TOPIC 3. OPERATING SYSTEM SECURITY

Different areas to secure any operating system

José Manuel Redondo López *"S-81 Isaac Peral"* Project v3.0

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Logo: @creative_vanesa (*Instagram*)

Topic goals

José Manuel
Redondo López

- **Understand how to manually harden an operating system**
  - What are the basic operating system areas to cover for that
- **Learn the areas that require control and special care**
- **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 team effort

Source: zona-militar.com

Security as a "chain"

José Manuel
Redondo López

- **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!)

Security defects in software

José Manuel
Redondo López

- **Malware takes advantage of defects (security bugs or vulnerabilities) in software**
  - Targeting bugs or loopholes **to exploit them**
  - **Outdated** applications (e.g., Internet Explorer versions supported by 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**
  - So the security problem persists!
- **Security advisories from application providers 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

José Manuel
Redondo López

- **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 management and authentication complexity…
  - Increasing short-term costs for training and maintenance
- **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 the disabled nodes
  - Introduce recovery options for malware or ransomware
- **OS installation must be done with the most secure settings (next topic)**
  - ASR covers both Linux-based and Windows OS installations

INSECURE DESIGN OR USER ERRORS

José Manuel
Redondo López

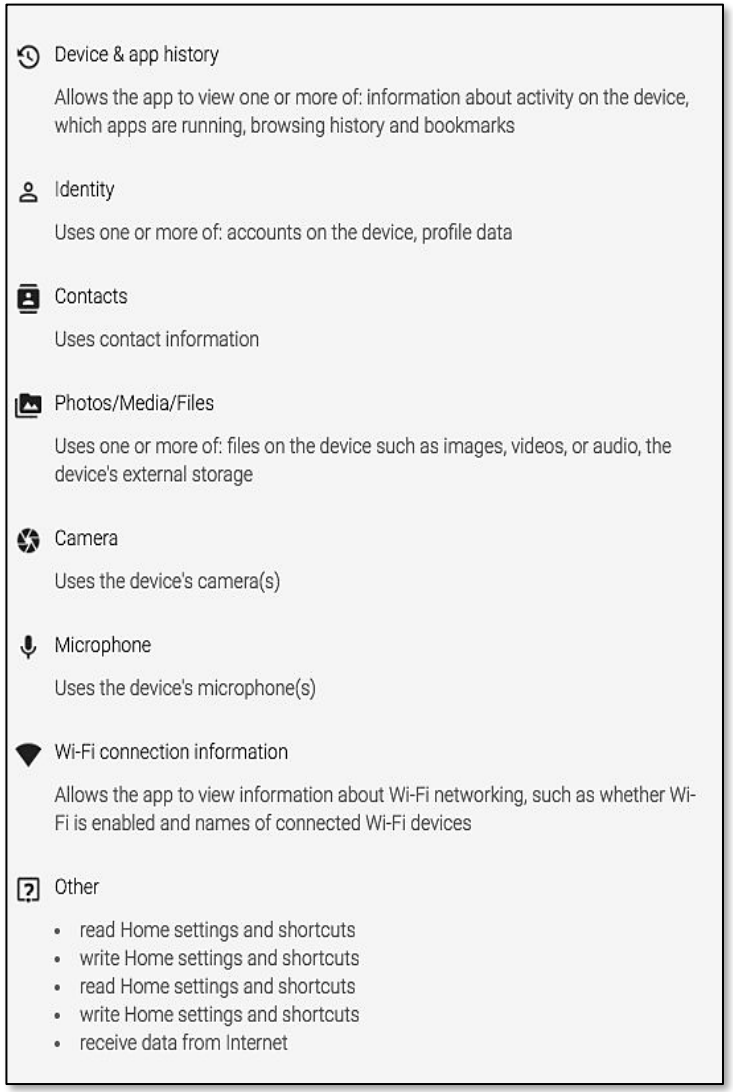
- **Sometimes common operations were designed with no security in mind**
  - Attackers frequently trick users to perform these operations so these are exploited. E. g.:
    - Microsoft vishing scam and RATs (Remote Administration Tools)
    - The Windows embedded Remote Assistance support ("Microsoft call" scam)
- **A popular example is the external boot / run problem**
  - Computers usually boot from an HDD/SSD
    - But external (USB) booting devices can be configured
  - This also happens when **running software automatically** upon external hardware insertion (USB)
  - This mechanism was designed as a convenient feature, but initially 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

José Manuel
Redondo López

- **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 their internal structures
    - 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 these privileges to attack or damage the system
    - Almost all currently popular OS and many applications give too many privileges to parts of their code

Source: Android App permissions



USERS AS POTENTIAL RISK SOURCES

José Manuel
Redondo López

- **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 untrusted 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, others may be "inside jobs"**
  - **Studies determine that half of the attacks to an organization come from actual employees!**

Users as liabilities: Countermeasures

José Manuel
Redondo López

- **Training**
  - Every user must receive **training** about the company resources and how they can or cannot be used
  - **A set of rules to create a security policy must be created**
  - **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) (next topic)**
  - 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

< Go to Index

Source: zona-militar.com

Boot >

Logging >

General OS Hardening

First steps in securing our OS

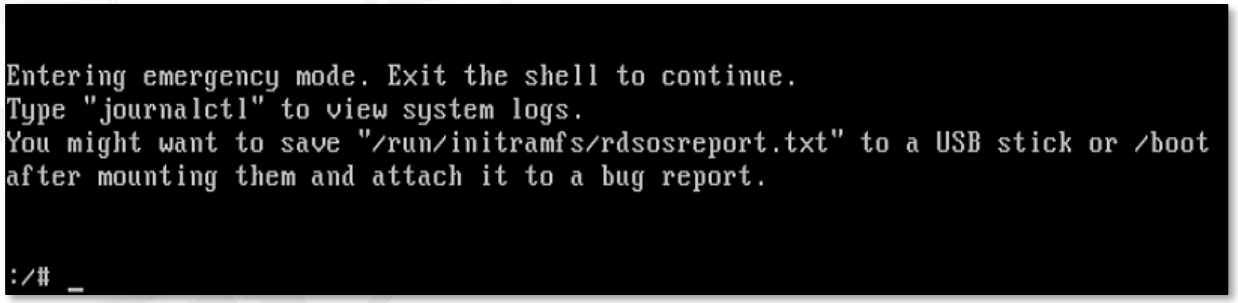
Security on the boot process

We don’t (want to) start the fire

Establish a boot / bootloader password

José Manuel
Redondo López

- **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 **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

José Manuel
Redondo López

- **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 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`
    - **Log files**, what happens while the OS runs
  - `/var/log/audit`
  - 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

José Manuel
Redondo López

- **Some OS allow to flag partitions with security properties. Ex:. 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 in the last theory topic)
  - `noexec`: useful for partitions without binaries (like `/var`)
    - Or containing binaries that should not be executed…

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

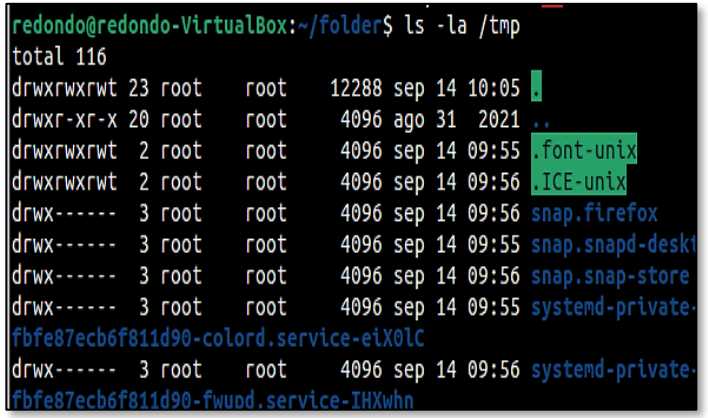
- **Example**
  - The boot directory of a Linux system contains important kernel-related files
  - It should be restricted to read-only permissions and flagged with the three previous properties

```
  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

José Manuel
Redondo López

- **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, etc.
    - …



- **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

José Manuel
Redondo López

- **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 (OS course)
  - 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, some kernel modules should not be loaded if not used (E. g.: USB, Bluetooth…)
    - To protect the system against the exploitation of any implementation flaws
  - And this again could be done installing **only** the software / drivers you **really** need

- **General disk management, quotas, etc. are explained in ASR**
  - Both in Linux and Windows

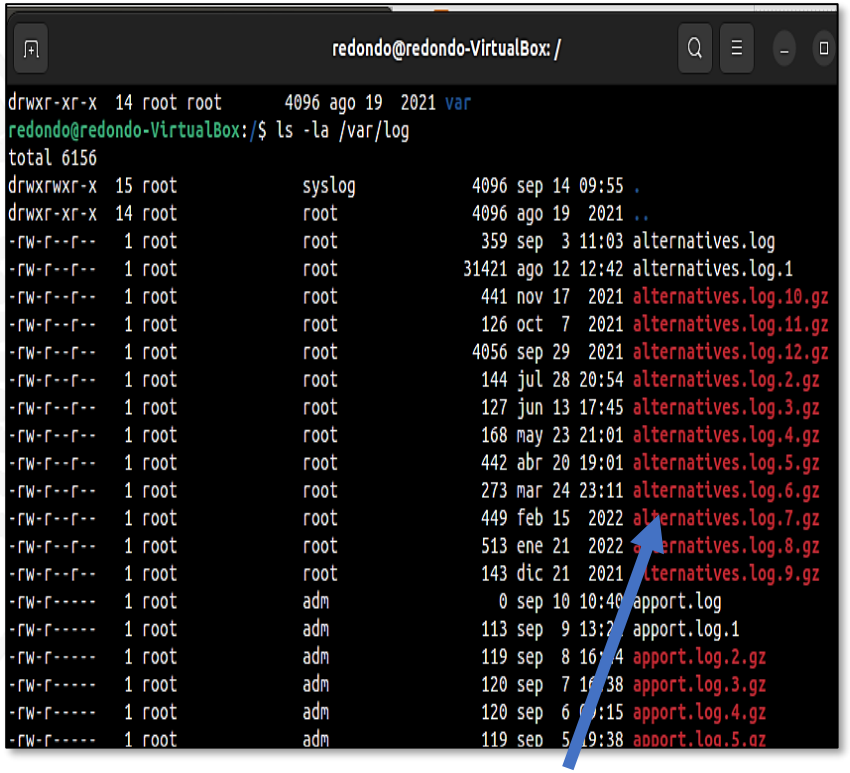
Logging and Auditing

Registering what happened

IMPLEMENTING A PROPER AUDIT CONFIGURATION

José Manuel
Redondo López

- **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
    - These include
      - 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)

Implementing a proper audit configuration

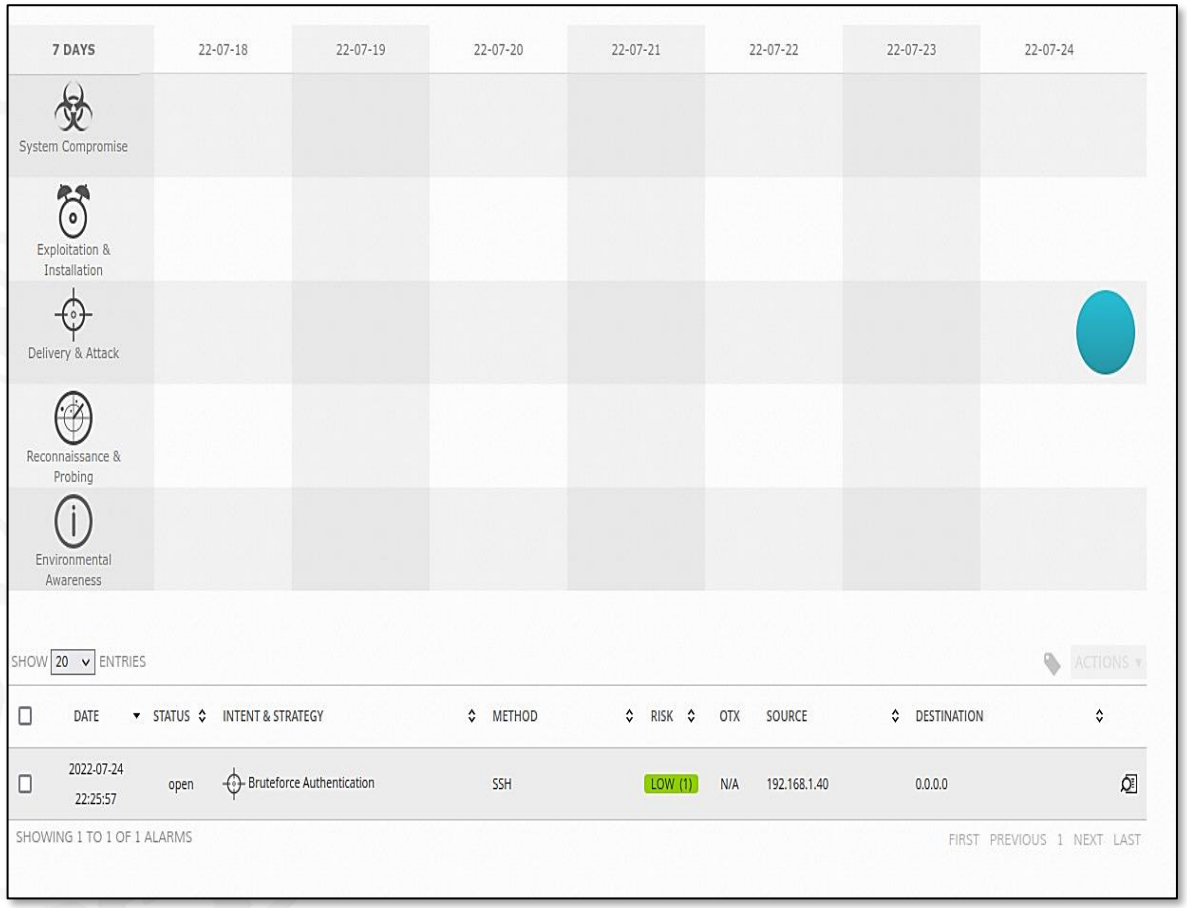
José Manuel
Redondo López

- **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**
    - Modifying sensitive parts of the OS by an administrator is allowed
    - But may also indicate compromise if unwanted changes are detected!
- **Log locations per OS will be explained in ASR**
  - Also, the forensics course teach you how to analyze and interpret log contents
  - Additionally, ML/IA techniques (SI course) may be also used to analyze log contents and identify threats

IMPLEMENTING A PROPER AUDIT CONFIGURATION

José Manuel
Redondo López

- **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)

REMOTE STORAGE OF SYSTEM LOGS

José Manuel
Redondo López

- **Most computers in highly secure infrastructures do not keep local logs, but 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
  - 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**

LOGWATCH

https://sourceforge.net/projects/logwatch/

José Manuel
Redondo López

- **Properly dealing with log files can be a complex and time-consuming task**
- **We should at least automate local log checking for anomalies periodically**
- **LogWatch enables us to do that**
  - Customizable log analysis system
  - It parses through the system's logs and creates a report analyzing areas that you specify
  - It is easy to use and will work without needing further configuration in most systems
  - **Tutorial:** https://www.techrepublic.com/article/how-to-install-and-use-logwatch-on-linux/
- **For thorough analysis techniques, you need to enroll the forensics course ☺**
  - For quick manual review tips, see the image

SANS CRITICAL LOG REVIEW CHECKLIST FOR SECURITY INCIDENTS

This cheat sheet presents a checklist for reviewing critical logs when responding to a security incident. It can also be used for routine log review.

GENERAL APPROACH

1. Identify which log sources and automated tools you can use during the analysis.
2. Copy log records to a single location where you will be able to review them.
3. Minimize "noise" by removing routine, repetitive log entries from view after confirming that they are benign.
4. Determine whether you can rely on logs' time stamps; consider time zone differences.
5. Focus on recent changes, failures, errors, status changes, access and administration events, and other events unusual for your environment.
6. Go backwards in time from now to reconstruct actions after and before the incident.
7. Correlate activities across different logs to get a comprehensive picture.
8. Develop theories about what occurred; explore logs to confirm or disprove them.

POTENTIAL SECURITY LOG SOURCES

- Server and workstation operating system logs
- Application logs (e.g., web server, database server)
- Security tool logs (e.g., anti-virus, change detection, intrusion detection/prevention system)
- Outbound proxy logs and end-user application logs
- Remember to consider other, non-log sources for security events.

TYPICAL LOG LOCATIONS

- Linux OS and core applications: /var/log
- Windows OS and core applications: Windows Event Log (Security, System, Application)
- Network devices: usually logged via Syslog; some use proprietary locations and formats

WHAT TO LOOK FOR ON LINUX

Successful user login	"Accepted password", "Accepted publickey", "session opened"
Failed user login	"authentication failure", "failed password"
User log-off	"session closed"
User account change or deletion	"password changed", "new user", "delete user"
Sudo actions	"sudo: ... COMMAND=..." "FAILED su"
Service failure	"failed" or "failure"

WHAT TO LOOK FOR ON WINDOWS

- Event IDs are listed below for Windows 2000/XP. For Vista/7 security event ID, add 4096 to the event ID.
- Most of the events below are in the Security log; many are only logged on the domain controller.

User logon/logoff events	Successful logon 528, 540; failed logon 529-537, 539; logoff 538, 551, etc
User account changes	Created 624; enabled 626; changed 642; disabled 629; deleted 630
Password changes	To self: 628; to others: 627
Service started or stopped	7035, 7036, etc.
Object access denied (if auditing enabled)	560, 567, etc

WHAT TO LOOK FOR ON NETWORK DEVICES

- Look at both inbound and outbound activities.
- Examples below show log excerpts from Cisco ASA logs; other devices have similar functionality.

Traffic allowed on firewall	"Built ... connection", "access-list ... permitted"
Traffic blocked on firewall	"access-list ... denied", "deny inbound"; "Deny ... by"
Bytes transferred (large files?)	"Teardown TCP connection ... duration ... bytes ..."
Bandwidth and protocol usage	"limit ... exceeded", "CPU utilization"
Detected attack activity	"attack from"
User account changes	"user added", "user deleted", "User priv level changed"
Administrator access	"AAA user ...", "User ... locked out", "login failed"

WHAT TO LOOK FOR ON WEB SERVERS

- Excessive access attempts to non-existent files
- Code (SQL, HTML) seen as part of the URL
- Access to extensions you have not implemented
- Web service stopped/started/failed messages
- Access to "risky" pages that accept user input
- Look at logs on all servers in the load balancer pool
- Error code 200 on files that are not yours

Failed user authentication	Error code 401, 403
Invalid request	Error code 400
Internal server error	Error code 500

OTHER RESOURCES

- **Windows event ID lookup:** www.eventid.net
- **A listing of many Windows Security Log events:** ultimatewindowssecurity.com/.../Default.aspx
- **Log analysis references:** www.loganalysis.org
- **A list of open-source log analysis tools:** securitywarriorconsulting.com/logtools
- **Anton Chuvakin's log management blog:** securitywarriorconsulting.com/logmanagementblog
- **Other security incident response-related cheat sheets:** zeltser.com/cheat-sheets

Authored by Anton Chuvakin (chuvakin.org) and Lenny Zeltser (zeltser.com).

Reviewed by Anand Sastry.

Cheat sheet version 1.0.

DIGITAL EVIDENCE COLLECTION | ADMINISTRATIVE | WORK PLAN | CASE SETUP | IMAGE INTEGRITY | ANALYSIS PREP | ANALYSIS | INTERPRETATION- REVIEW | REPORTING

1. ADMINISTRATIVE STUFF

- Review Policies and Laws
- Chain of Custody form
- Digital Evidence Collection Form
- Consent form (if needed)
- ✓ Evidence Tracked and Stored

2. WORK PLAN (docs)

- Review Policies and Laws (if needed)
- Gain understanding of:
  - -Background
  - -If applicable, previous work
  - - Requirements/Goal of analysis
  - - Deliverable
- Create Analysis Work Plan
- Create Investigative Plan

3. SETUP CASE FOLDER (example)

- EV1 (Evidence Files)
- WC (Working Copy of Files)
  - Case # - Project Name
    - Custodian Name
      - Media Type – EV #
        - Case File
        - Index
        - Reg Files
        - Internet Hist
        - Case Processor
        - Logs

  (Analysis Folders)
      - Media Type – EV #
      - Media Type – EV #

*Organize output neatly!

4. CONFIRM IMAGE INTEGRITY

- ✓ Compare Acquisition and Verification Hash values (MD5, SHA)
- Save Verification Reports

GENERAL FORENSIC ANALYSIS CHECKLIST V.1.1

THE PURPOSE OF THIS REFERENCE GUIDE IS TO PROVIDE AN OVERVIEW AND OUTLINE OF COMMON PROCESSES, SOFTWARE, AND BEST PRACTICES FOLLOWED BY PROFESSIONALS CONDUCTING COMPUTER FORENSIC ANALYSIS

BY DAVID NIDES (12/16/2011)
TWITTER: @DAVNADS
BLOG: DAVNADS.BLOGSPOT.COM
EMAIL: DNIDES@KPMG.COM

CREDITS TO: ED GOINGS, ROB LEE & SANS QUESTIONS/FEEDBACK–CONTACT US!

SANS COMPUTER FORENSICS and INCIDENT RESPONSE

5. BEFORE YOU GET STARTED..

- ✓ Check physical size of drive and compare to physical label accounting for all drive space (Check for DCA/ HPA).
- ✓ Identify & compare logical partition size(s) to physical drive size to identify any deleted partitions or unused disk space
- ✓ Retrieve time zone settings for each disk and apply correct time zone, if applicable
- ✓ Rename hard disk volumes as necessary to "Recovery", "C", etc.

GATHER SYSTEM INFORMATION

- ✓ Determine OS, service pack, OS install date, application list, owner, machine name, and other basic information.
- ✓ Retrieve user profile information (names, SIDs, create and last logon dates)

PRE-PROCESSING ANALYTICS

- ✓ Conduct hash analysis, indentify "known" and/or "notable" files.
- ✓ Conduct file signature analysis, review renamed files.
- ✓ Identify encrypted files (entropy)
- ✓ Mount ALL compound files (VHD, VMDK, ZIP,RAR, Email containers, Reg Files, etc
- ✓ Index Case (DT Search, WDS, Encase, AD..)
- ✓ Generate metadata (and extended) listings/ reports

RP / VSC

- ✓ Identify if services turned on/used
- ✓ Extract or make available accordingly for analysis

MOUNTING / VIRTUAL EMULATE

- ✓ Mount - Malware/Virus Scan (Don't forget about MBR)
- ✓ Mount - Stego Scan
- ✓ Virtually Emulate – conduct behavior and live analysis

KEYWORD SEARCHING

- ✓ Create keyword list & QC syntax formatting/code page usage (may be iterative process)
- ✓ Perform targeted or full disk search including unallocated and slack areas.
- ✓ Create hit report/stats

FILTERING

- ✓ Filter data based on meta data and exteneded meta data such as Date and Time values, File Extension, and etc.

EXPORT

Export files from case for independent analysis with specialty tools. For example:

Memory
Memorize
Redline
Volatility (SIFT)
Passwords
AD PRTK
Passware
Ophcrack
Shellbags
Shellbags.py (SIFT)
Internet History
WebHistorian
LNK Files
Tzworks
Lslnk (SIFT)
Event Logs (.evt & .evtx)
Tzworks
GrokEVT
MFT
AnalyzeMFT
Ntfswalk (SIFT)
index/$I30
INDXParse.py (SIFT)
SIFT Workstation is a great resource for tons of tools!

Email
NUIX
Clearwell
Recover My Email
Bulk extractor (SIFT)
Image Mounting
FTK Imager
ImDisk
Live View
OSFMount
Virtual Box
Stego
Outguess
Hashing
Md5deep (SIFT)
Sha256deep (SIFT)
Hashdeep (SIFT)
Registry
Reg Ripper (SIFT)
Registry Decoder
AD Registry Viewer
Reglookup (SIFT)
YARU (SIFT)
Windows Journal Parser
Tzworks

6. EQUATION FOR SUCCESSFUL ANALYSIS: (TIMELINES + MANUAL ANALYSIS) x (PASSION + TIME + RESEARCH + RESOURCES) = "WINNING"

GENERAL AREAS

Analyze general folder locations for artifacts or data of interest. For example:
>Desktop
>User folders
>Documents
>Desktop
>Network Shortcuts
>Recently Opened File folders
>System Temp folders
>Browser Temp Folders
>System Folders (malware)
>Autorun

TIE IT ALL TOGETHER USING TIMELINE ANALYSIS – FLIP SIDE! → 5

REGISTRY ANALYSIS (Win only)

The System registry hives (NTUSER.DAT, SYSTEM, SOFTWARE, SECURITY, SAM..) contain a vast of amount of information that can be imperative to your analysis. For Example:
-OS Version (SOFTWARE)
-Last logged on user (SAM)
-Last Failed Logon (SAM)
-Username & SID (SAM)
-Shutdown (SYSTEM)
-TimeZone (SYSTEM)
-Drives Mounted by User (NTUSER.DAT)
-File Ext Associations (NTUSER.DAT)
-Installed application list (SOFTWARE)
-Search History (SOFTWARE)
-Removable Storage Devices (SYSTEM)
Using Registry analysis tools, such as Reg Ripper, can aid your analysis..

SYSTEM ARTIFACTS

There are many system related artificacts that may contain potentially relevant information including:
-Backups (RP, VSC, others)
-Event Logs (.evt, evtx)
-Shell bags (Registry)
-Jump Lists
-Misc. Logs (firewall, AV, Apps, etc)
-Removable Media Connections
-LNK files
-Prefetch
-PageFile

SOFTWARE RESIDUE

Identify software (i.e Wiping tools, P2P, Sticky Notes, hacker tools, etc) and perform analysis on associated files (binary-malware analysis), logs, settings, registry and etc.

MEMORY RESIDUE

If applicable, perform memory analysis to gather volatile information including:
-Handles; files, directories, processes, registry keys, Semaphores, events and sections
-Open network ports
-Hooks: Driver IRP, SSDT and IDT driver tree

E-MAIL/IM/SOCIAL ARTIFACTS

Identify email clients or web access on system and perform analysis on associated data stores or application residue/settings:
Client based (file examples):
Windows - .OST, .PST, .MSG, Temp folder for Outlook attachments, Lotus Notes - .NSF, Mac - .EML, .EMLX, .MBOX
Web based (OWA, Facebook, Twitter, etc.):
-Logs -Internet History reconstruction -Cache

INTERNET

Identify installed browsers and perform analysis on artifact such as:
-Parse Internet history files (index.dat, sqlite, etc)
-Check temp folders
-Parse cookies
-Cached pages
-Form History/Auto complete files
-Favorites/bookmarks
-Toolbars
-WebSlices
-Browser plug ins
-Registry analysis
-Carve unallocated for deleted history artifacts

...BUCKETING ANALYSIS "TO DO" ITEMS LIKE THIS CAN HELP.

7. INTERPRETATION/REVIEW OF ARTIFACTS (examples) ..

File Download	Open/Save MRU	Open/Run MRU	Email	Skype History	Index.dat / places.sqlite	downloads.sqlite	P2P Logs	
File Opening/ Creation	MACB	MRU by ext	MS Office MRU	LNK Files	Index.dat file://	JumpLists	Shellbags	Prefetch
File Knowledge	XP Search ACMRU	Win7 Search WordWheelQuery	Thumbs.db	Win7 Thumbnails	Recycle Bin	Browser Artifacts	Last visited MRU	
USB Usage	Identification	First & Last Insert Times	User Identification	Volume Name	Drive Letter	LNK Files	Event Logs	Mount Point
Program Execution	User Assist	LastVisited MRU	Run MRU (Start->Run)	Prefetch	MUI Cache	Jump Lists	Event Logs	
Physical Location	Time Zone	Wireless SSID	Vista/W7 Network History	Cookies	Search Terms	IP	Image exif	
Account Usage	Last Password Change	Group Membership	Success/Fail Logons	Logon Types	RDP Usage	Account Logon/ Authentication	SID Attributes	
Browser Usage	History	Search Box Terms	Cache	Session Restore	Flash & Super Cookies	Suggested Sites	Memory Fragments of Private Browsing	AJAX Transactions	Typed URLS

Have more? Let me know!

→ 8

8.REPORTNG

- ✓ Document findings comprehensively
- ✓ Fact based Interpretation
- ✓ Remember who the audience is
- ✓ Remember requirements/expectations

LOG PARSING CHEAT SHEET

GREP

GREP allows you to search patterns in files. ZGREP for GZIP files.

```
$grep <pattern> file.log
```

-n: Number of lines that matches
-i: Case insensitive
-v: Invert matches
-E: Extended regex
-c: Count number of matches
-l: Find filenames that matches the pattern

NGREP

NGREP is used for analyzing network packets.

```
$ngrep -I file.pcap
```

-d: Specify network interface
-i: Case insensitive
-x: Print in alternate hexdump
-t: Print timestamp
-I: Read pcap file

CUT

The CUT command is used to parse fields from delimited logs.

```
$cut -d ":" -f 2 file.log
```

-d: Use the field delimiter
-f: The field numbers
-c: Specifies characters position

SED

SED (Stream Editor) is used to replace strings in a file.

```
$sed s/regex/replace/g
```

s: Search
g: Replace
d: Delete
w: Append to file
-e: Execute command
-n: Suppress output

SORT

SORT is used to sort a file.

```
$sort foo.txt
```

-o: Output to file
-r: Reverse order
-n: Numerical sort
-k: Sort by column
-c: Check if ordered
-u: Sort and remove
-f: Ignore case
-h: Human sort

UNIQ

UNIQ is used to extract uniq occurrences.

```
$uniq foo.txt
```

-c: Count the number of duplicates
-d: Print duplicates
-i: Case insensitive

DIFF

DIFF is used to display differences in files by comparing line by line.

```
$diff foo.log bar.log
```

How to read output?
a: Add
c: Change
d: Delete
#: Line numbers
<: File 1
>: File 2

AWK

AWK is a programming language use to manipulate data.

```
$awk {print $2} foo.log
```

Print first column with separator ":"
```
$awk -F: '{print $1}' /etc/passwd
```

Extract uniq value from two files:
```
awk 'FNR==NR {a[$0]++; next} !($0 in a)' f1.txt f2.txt
```


@FRØGGER_
THOMAS ROCCIA

< Go to Index

Source: zona-militar.com

Hardening services / software

Security applied to programs

Maintaining and updating software

José Manuel
Redondo López

- **Software maintenance is extremely important to have a secure system**
  - Patch software as soon as revisions 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
- **Activating package signatures ensures the software has not been tampered with and that it has been provided by a trusted vendor**
  - This is normally provided by default in Windows, but should be enabled in Linux
- **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

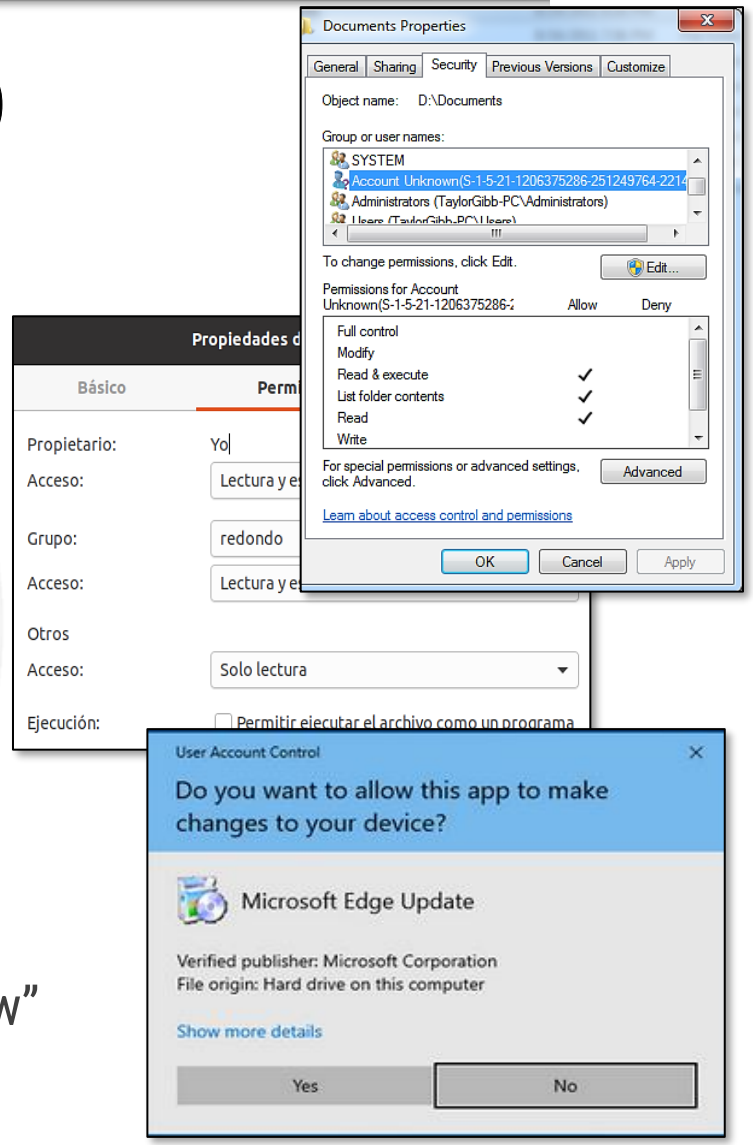
José Manuel
Redondo López

- **Enable task scheduling**
  - Enabling task scheduling is essential: 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
  - Time synchronization services are usually enabled by default
  - Try to not to change the date manually
- **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, but we should increase the difficulty

TYPES OF ACCESS CONTROL

José Manuel
Redondo López

- **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**, elevated 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

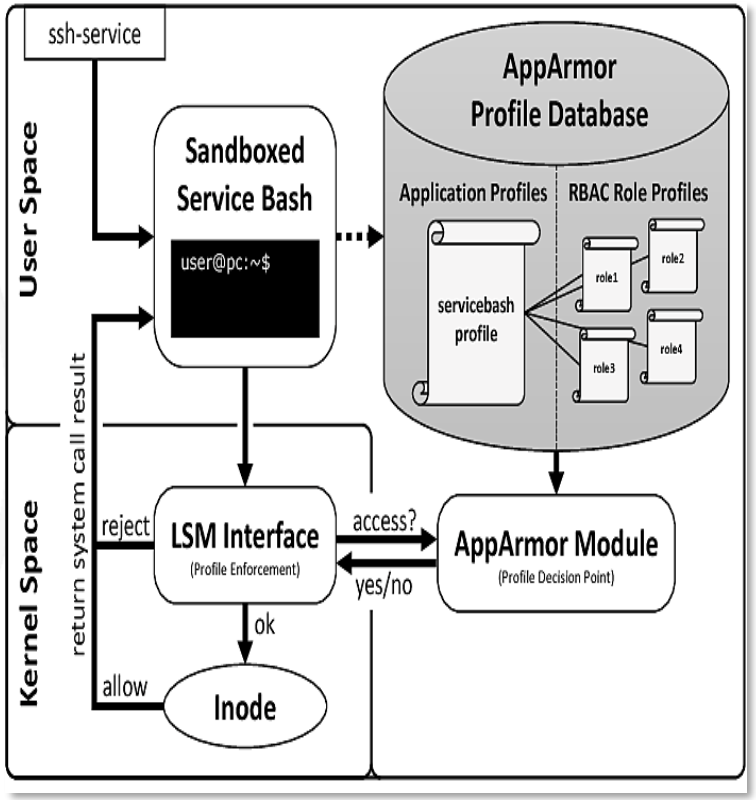


Source: https://helpdesk.athens.edu/hc/en-us/articles/4404584871575-Microsoft-Edge

Mandatory Access Controls (MACs)

José Manuel
Redondo López

- **While Windows have Mandatory Integrity Control (MIC), the Linux equivalent 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!
- **A series of measures ensure proper operation**
  - Ensure that it is enabled and "**enforcing**"
    - It can confine potentially compromised processes to a security policy
    - It is 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



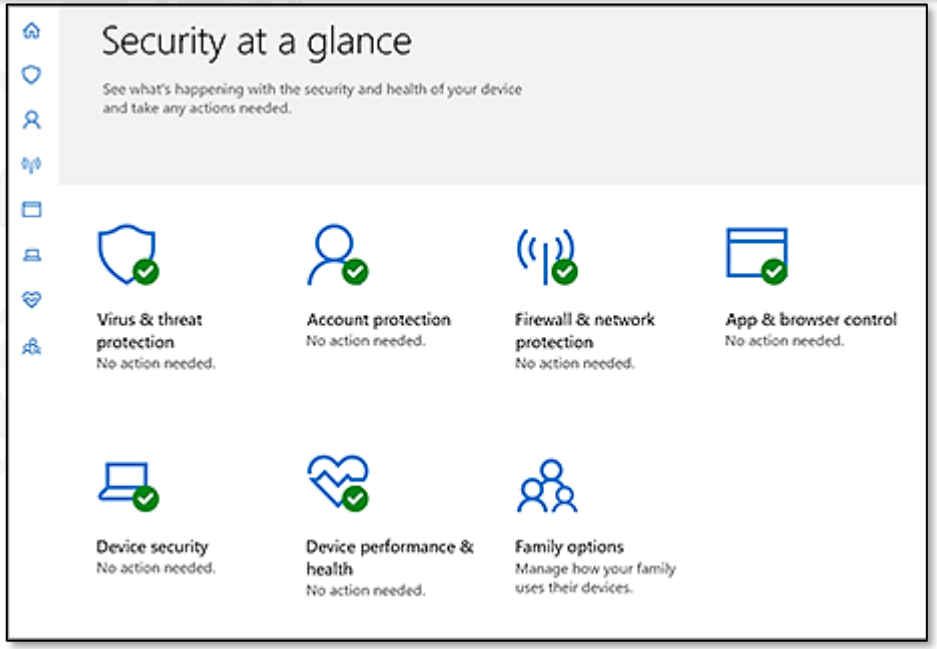
Source:
https://www.semanticscholar.org/paper/Using-RBAC-to-Enforce-the-Principle-of-Least-in-Kern-Anderl/c3438119a1d466f69568b48d6ae3f1664d1e3830

Programs: Malware

José Manuel
Redondo López

- **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)
```

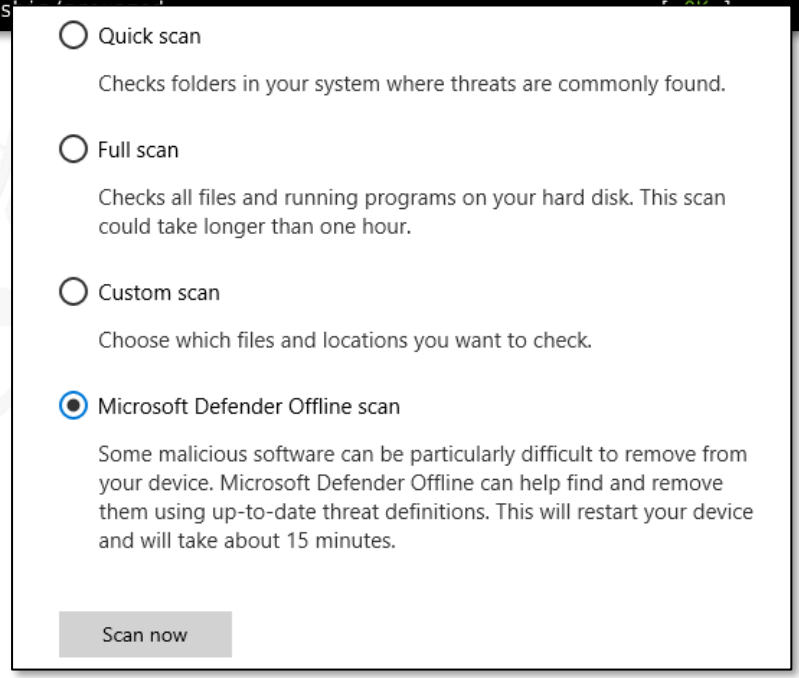


Programs: Rootkits

José Manuel
Redondo López

- **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

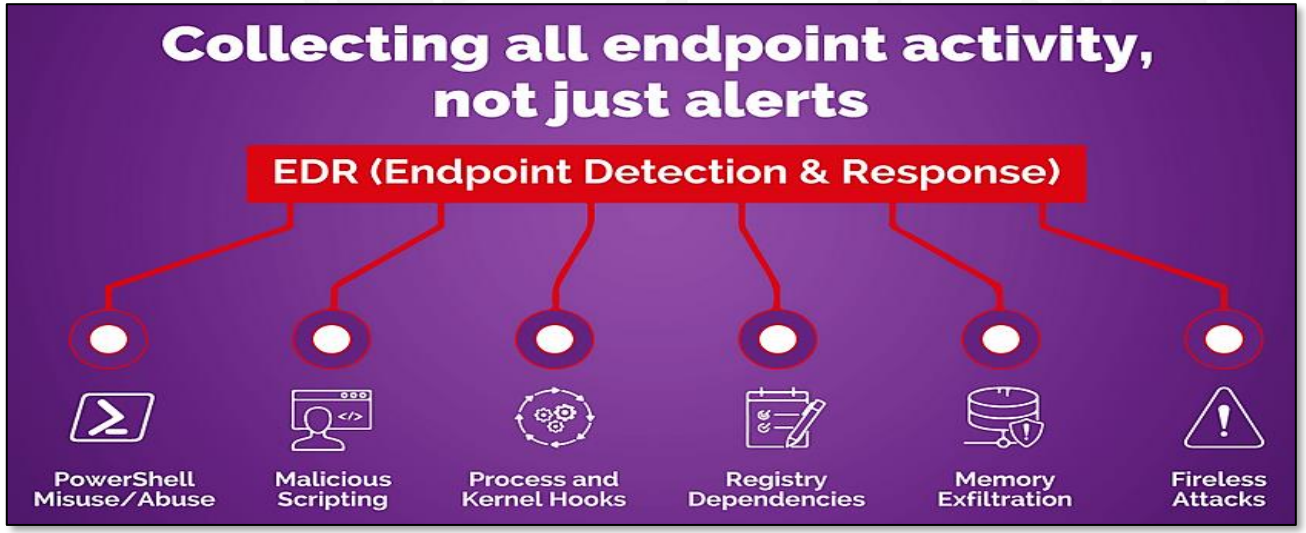
```
redondo@miw:~$ sudo rkhunter --check
[ Rootkit Hunter version 1.4.6 ]

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) technology

José Manuel
Redondo López

- **Continually monitors and responds to mitigate cyberthreats**
- **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 allow and block lists, and other advanced response capabilities
  - Some use the MITRE ATT&CK classification and framework for threats (topic 5 and 7)



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

< Go to Index

Passwords >

Source: zona-militar.com

Users

Hardening user management

General rules

José Manuel
Redondo López

- **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 individually

Accounts and Access

José Manuel
Redondo López

- **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 critical access services**
  - Like SSH: https://medium.com/mi-rincon/doble-factor-de-autenticacion-2fa-c2c8d728b978
  - We will talk more about MFA on Topic 6
- **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 by pressing specific key to edit the boot option

Session configuration files

José Manuel
Redondo López

- **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 e 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
  - This can be achieved by including in `/etc/profile` and in `/etc/bash.bashrc`: `umask 027`

LOGIN BANNERS

José Manuel
Redondo López

- **Login banners warn any possible intruders that may want to access your 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 IT 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

José Manuel
Redondo López

- **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**
  - **Protect direct `root` logins:** in Linux systems, disabling direct `root` logins ensures proper accountability
    - Use multifactor authentication on privileged accounts
  - **Protect system account logins (service accounts):** not giving interactive login privileges to system accounts upon login
    - Makes using them more difficult to attackers

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

Password policy

José Manuel
Redondo López

- **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 the cryptography topic)
- **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?*
  - *How quickly can a user change their password after a previous change? (users may bypass password history requirements by changing their password multiple times until they put the initial password again)*
  - *When must a user change their password? After 90 days? After lockout due to excessive log on attempts?*
  - *How different must the next password be from the last password?*
  - *Is the user prevented from using his username or other account information in the password?*

Common password protection measures

José Manuel
Redondo López

- **Set a password expiration policy: requires setting a password warning, minimum and maximum age**
  - **Password warning age** enables users to make the change at a practical time 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
- **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

José Manuel
Redondo López

- **Password hashing algorithm**
  - **The most used algorithm is SHA-512**
  - Default in most modern RedHat and Debian distros (see cryptography topic)
- **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 requires 5 passwords

Password strength policy: Password quality requirements

José Manuel
Redondo López

- **Using a complex / strong password helps to increase the time and resources required to compromise it**
- **Operating systems usually allows us to enforce it 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
  - **Usage of digits, lowercase, and uppercase letters** in a password (separately)
  - **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 same consecutive characters**

Password policy implementation example: PAM module

José Manuel
Redondo López

- **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, and sometimes the system defaults must be restored if manual changes prevent working 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

< Go to Index

Source: zona-militar.com

FILES

Hardening the access to our data

CRITICAL FILES PERMISSIONS AND LIMITS

José Manuel
Redondo López

- **OS security relies on file and directory permissions to prevent unauthorized users from reading or modifying files to which they should not have access**
  - Permissions for many files must be set restrictively to ensure sensitive information is properly protected
  - The default restrictive permissions for files which act as important security databases must be verified and maintained
    - `passwd`, `shadow`, `group`, and `gshadow` (**Linux**)
    - The SAM file (**Windows**)
- **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

José Manuel
Redondo López

- **Ensure no world-writable (or "All users" group) files exist**
  - 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
    - But not world-writable!
- **In Linux, verify that all world-writable directories have sticky bits set**
  - Not setting the sticky bit 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 (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 forensics 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

José Manuel
Redondo López

- **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 drives…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**
  - This also may reveal malware presence before it is too late
  - Tripwire, Aide and OSSEC (previously mentioned) are three of the most famous ones
- **Details about filesystems and their contents are explained in ASR**

< Go to Index

How to evaluate and achieve security

Hardening in an efficient way

Source: zona-militar.com

IMPLEMENTING AND EVALUATING SECURITY

José Manuel
Redondo López

- **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 over each sensitive part of the OS**
  - In this presentation we reviewed the most sensitive parts
  - Each of these security-enforcing operations is called a **security control**
  - Therefore, hardening any OS **means implementing a (potentially 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 applies to potentially any software product!**
  - Hardening software means obtaining its best list of security controls
  - And implement each one of them (except if that breaks legitimate functionality!)

Implementing and evaluating security

José Manuel
Redondo López

- **Unfortunately, it is frequent that installed software / OS have not been submitted to hardening processes**
  - They have their default post-install configuration
  - These configurations are most of the times inadequate security-wise
- **Software / OS could be even not updated ever after installation!**
- **But, even if hardening measures are taken, these can be inadequate**
  - **You cannot perform manual hardening** operations trusting your own skill
    - A documented list of security controls is required to ensure completeness, documentation and traceability
  - **You cannot trust any list security control list you find somewhere**
    - Is it complete? Is it properly documented? Is it maintained? Is it **CORRECT**? Can you trust the author?...
    - **Please avoid forums, StackOverflow, blog articles, etc.!**

Implementing and evaluating security

José Manuel
Redondo López

- **You need to put strong requisites to the security control lists you use**
  - **Validated** by trustable third parties: organizations, companies, governments…
  - **Complete**: it must secure every "sensitive" software part (like the OS parts we reviewed)
  - **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
- **But you did not give us security control lists sources!**
  - **That's because we will give you some of them in the next topic!** ☺
  - For Linux, Windows, software…everything you need to make them (properly) secure

Implementing and evaluating security

José Manuel
Redondo López

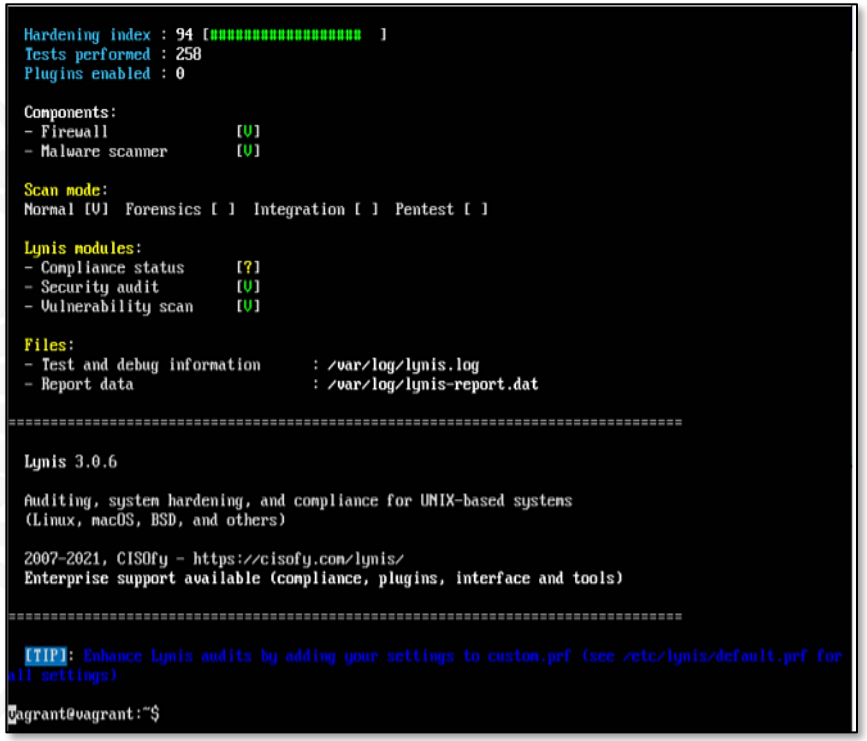
- **So, having an extensive list of properly validated security controls is the only way of ensuring a proper security level**
  - But…this list **does not have to be applied manually**!
  - In fact, many security controls can be checked and applied **automatically**!
- **Automatically applying security 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?**
  - We can get an automated security evaluation with tools like Lynis (for Linux systems)

Lynis

https://github.com/CISOfy/lynis

José Manuel
Redondo López

- **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 the next topic



- **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

SELF-EVALUATION ACHIEVEMENT LIST: OS SECURITY

José Manuel
Redondo López

Rank	Concept
	Know the different parts that are involved into the security of a system
	Know the security aspects of the boot process and methods to automate log analysis
	Being able to determine if permissions of files are adequate from a security point of view
	Understand the importance of security control sources and automation
	Know the difference between MAC and DAC access control policies
	Being aware of secure software update, management, and installation policy
	Know products able to detect different types of malware
	Know how to securely manage user accounts and access
	Know how to use Lynis to evaluate a Linux OS security level
	Being able to design an effective password policy
	Know the security aspects of the log system of an OS
	**This is OSparta!: Be able to identify the critical security areas of any OS**

TOPIC 3. OPERATING SYSTEM SECURITY

Source: zona-militar.com