system hardening - an introduction

general

- Hardening a system against:
  - (Security) Threats
- Threat: Everything that can possibly violate the security goals:
  - Confidentiality
  - Integrity
  - Availability

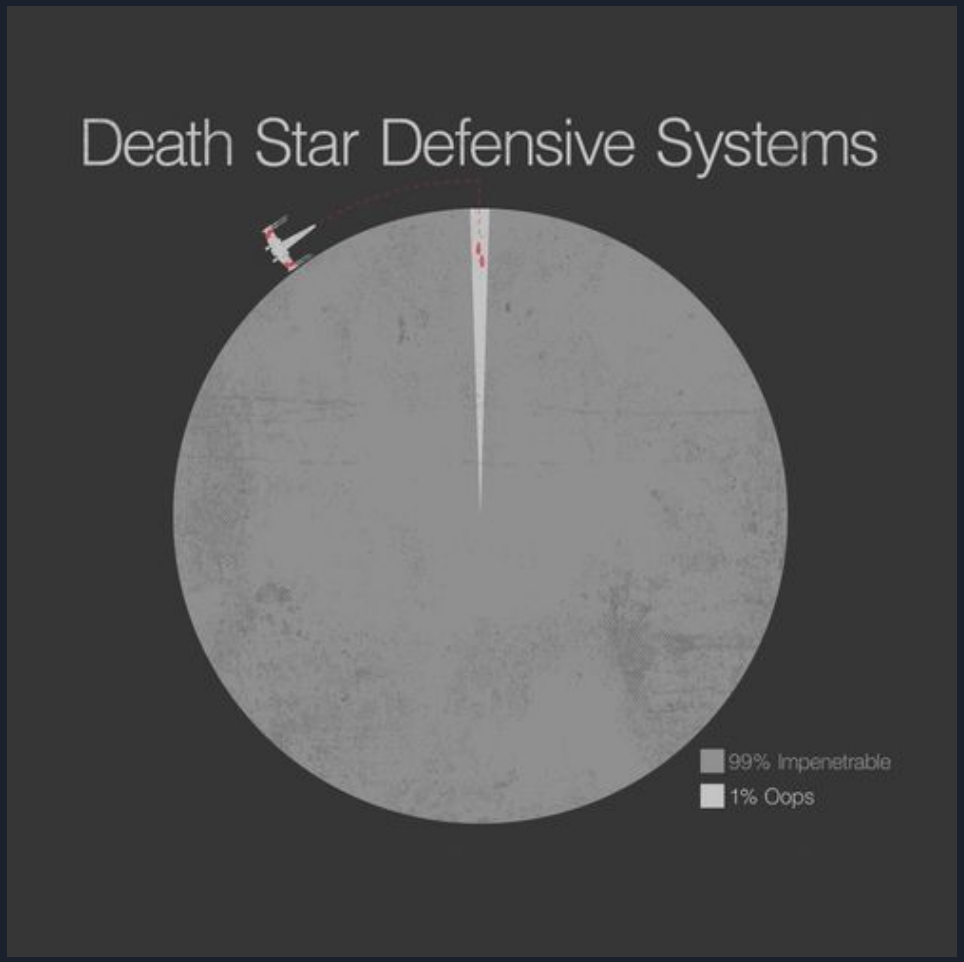
Why doing it and what's the impact?

Decrease the attack surface
Decrease available information after successful exploitation through an attacker
Increase the likelihood of detection of an event
Learn more about your system
    Installed software
    Configuration
    Inner workings

- Depends on the use case
- General impacts
  - Nice:
    - Performance
    - Stability
    - Manageability
    - System Know How
  - Bad:
    - Complexity
    - Usability

Concepts related to hardening

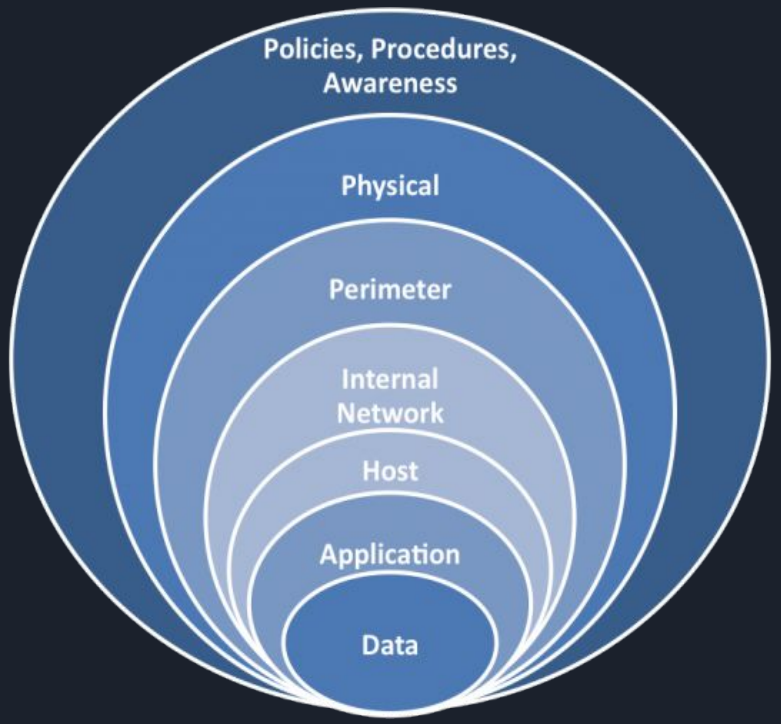
Reduction of attack surface



- Less attack surface -> less security risk
- But beware, only one risk is needed to make your death star explode
- Therefore think in layers and minimize the exposure on each layer
- Similar to
  - Minimum Exposure
  - Keeping it simple

Defense in depth

- Define measures for each layer
- If one line of defense gets breached, another layer can halt the attacker



Principle of least privilege

What is a privilege?

The ability to do something like accessing or modifying a resource

Restrict the privileges of parties within a system, so the party can only access the information needed.

Any subject (tool / user / service / …) should only run with the minimal privileges needed to complete a task

Other concepts

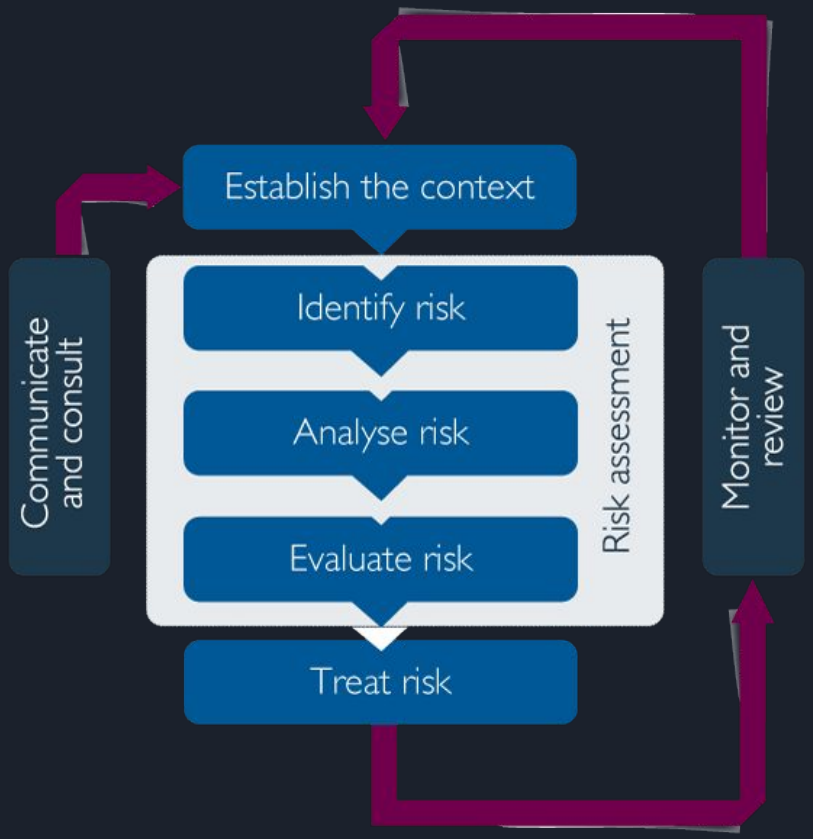
- Simplicity
- Open Design
- Compartmentalization
- Secure, Fail-Safe Defaults
- No Single Point of Failure
- Log security-relevant system events
- Usable security mechanisms

What measures to choose ?

What measures to choose ?

Risk management !

1. First get to know your system
2. Identify your threats
3. Rate them, order, prioritize
4. Act based on risks (e.g. do nothing, find requirements based on threats)



- Scale the depth of the process to meet your security goals.
- A more formal process would be the SQUARE process

Sources of measures

BSI Grundschutz Katalog

- Provides a baseline of protections
- Helps to identify components, threats and measures

Common Vulnerabilities and Exposures (CVE)

- Database of public information of exploits, vulnerability and exposures
- Helps to identify threats and measures

More sources

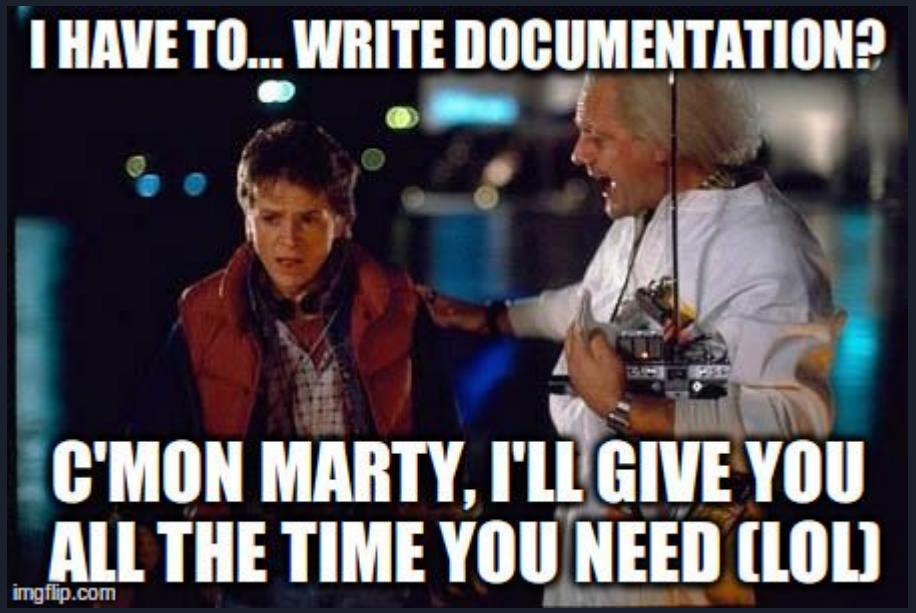
  - Center of internet security (very detailed harding for applications)
  - Cisofy (harding for linux and unix systems)
  - Security Technical Implementation Guides (STIGs)
  - NIST SP 800-123: Guide to General Server Security
  - NSA Guides
  - Google + RTFM (the manual sometimes is indeed useful)

hardening steps

1. Nice to have: isolate the system
2. Take a known good system state as start point
3. Create an list of components installed
4. Find measures
5. Apply measures
6. Create a baseline defined on the hardened state
7. Return the system into production
8. Use the system as template

`system documentation`

- Step 1: Do it
  - Level 0: baseline + changes
  - Level 1: dependencies, configurations, maintenance access, responsibilities
- Can be easily combined with checklists !
- Some tools:
  - Markdown based tools
    - Notebook
    - Confluence
  - MS stuff:
    - OneNote
    - Word
  - zim (OneNote Alternative)
  - echo “$EDITOR”



Short Example

Approach: Layer by Layer

Physical Access

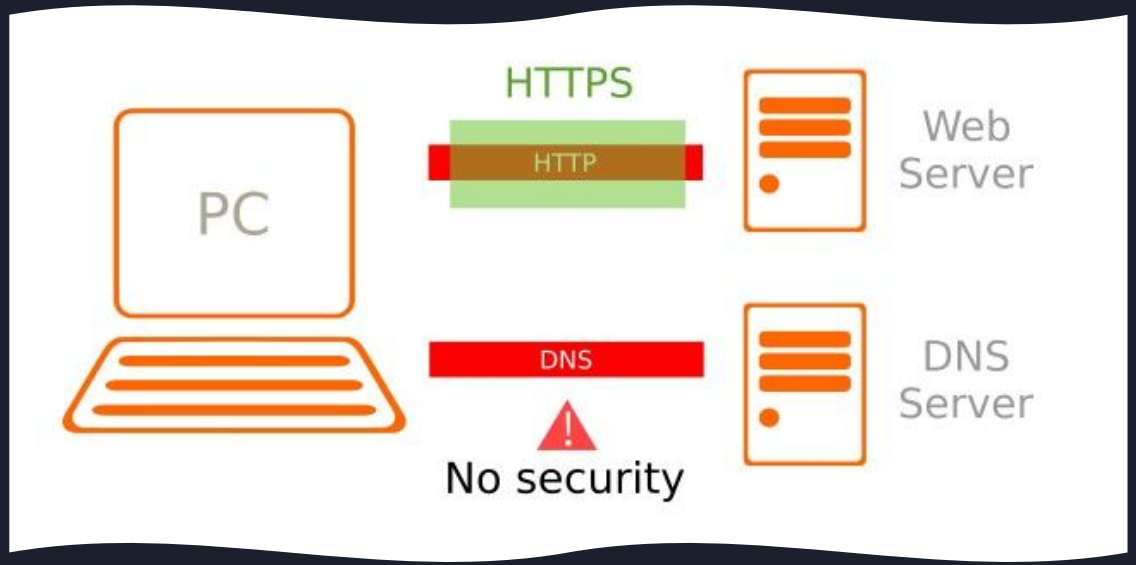
- Threat: Attacker gains physical access to the machine and does evil stuff.
- Measures:
  - Set a boot password (hdd, boot and master password)
  - Lock the PC if booted
  - Use the full disk encryption (LUKS vs HDD built in encryption)
- Threat: Attacker modifies the boot code.
- Measures:
  - Use Secure Boot with our own keys
  - Sign the boot code and verify the integrity of the boot code
- Threat: Attacker modifies the hardware
- Measures:
  - Use the built in intrusion sensor
  - Or lots of duct tape

Network Layer

- Threat: Attacker intercepts our network traffic
- Measure:
  - Encryption
    - TLS, Secure NTP, DNS,
- Threat: Attacker tracks us per MAC Address or SSID we search
- Measure
  - Random MAC for searching SSIDs
  - Random MAC for connection to a network
- Threat: Attacker attacks the protocol stack
- Measure:
  - Use a hardened kernel
  - Harden the kernel settings for networking
  - Use applications to detect the attacker (Intrusion detection systems (IDS))

Network Layer - In detail

- Bind services to local host
- Use a firewall
- Harden the kernel and applications for network usage
- Disable IPv6 if not needed
- Hardening of TLS (e.g. RFC 7525)
- VPN, Encrypted Tunnels (e.g. Proxies)
- Encryption of DNS is a rising issue



Host layer

- Asset: a arch linux install
- Threat: Attacker attacks on of our applications in user space
- Measure:
  - Isolate the application (systemd (per cgroups), chroot, container, docker)
- Threat: Malware infects our system
- Measure:
  - Install antivirus software
  - Check for changes in the system (e.g. aide, tripwire)
- Threat: Sensitive key material gets stolen
- Measures:
  - Move keys to a secure storage (e.g. a hardware dongle)
  - Use a decent key management to minimize impact of compromisation

Encryption

- Keep your data confidential
- Per disk:
  - home, root, var, ...
  - don‘t forget swap
- Per folder:
  - per fuse-fs in the sync folder (e.g encfs)
  - encrypt the home drives of the user
- Per application
- A word of warning on LUKS disk encryption:
  - Do not forget to backup the HEADERS or it will byte you if the drives failed in the first block

Application Layer - e.g. Browser

- Asset: a browser
- Threat: Attacker exploits our system via the browser
- Measure:
  - Isolate the browser
  - Harden the browser
- Threat: Attacker tracks us across different
- Measure:
  - Enforce HTTPs usage
  - Use a hardened browser
  - Use plugins against tracking

Isolation of applications under linux

- fully virtualized environment
  - qemu + kvm, virtualbox, vmware, ...
- namespaces (container)
  - docker, lxc, nspawn, ...
  - firejail !!!
- In a jail / chroot
- Limited per cgroups
  - Can be done per systemd unit files (e.g. limit ressources, block access to files, ...)
- As a low privileged user
  - never a user who is root or is intended to become root
- Don‘t even think about it

`systemd-nspawn`

- Container without docker or lxc
- Nice:
  - Comes pre installed with most distros
  - Only depends on systemd (and the kernel)
  - Can be easily integrated (per systemd services)
  - Can use predefined images
  - Many more features
- Bad:
  - ... ???
- Examples: Wordpress, skype, steam, ...

Data Layer

- Asset: top secret documents
- Threat: Attacker reads documents
- Measure:
  - Use a sophisticated access control mechanism
  - Encryption
  - Hide them (e.g. per steganographie)
- Threat: Attacker deletes documents
- Measure:
  - Make backups and keep logs

Hardening tips for applications

Specific Applications - Web Server

- Permissions
  - Run the web server as low privileged user
- Use TLS
  - For encryption of the traffic
  - For authentication of the user
  - Use secure and modern TLS settings
  - Maybe use techniques like DANE or HSTS
- Limit Information send by the server (e.g. Header, error messages)
- Disable error traces
- Use web application firewall (WAF; e.g. modsecurity)
- Isolate the web server if possible
- Use a reverse proxy for better auditing and more

Firewalls

Typical Firewalls

- iptables
- nftables
- arptables
- ebtables

“Classic” ACLs:

- TCP Wrapper
- SELinux

Specific Applications - SSH

- Authentication
  - Two-factor Authentication with SSH (per pam)
  - Disable root login
  - Disable passwords (use keys)
  - Limit authentication tries
  - Hardware token for more secure key management
- Disable old protocols version (version 1)
- Use DNS hostname checking
- Watch for changes of the signature of the server
- User and groups configuration;
  - Least privilege for users (e.g. user which is only allowed to read certain files)
  - Whitelist users
- Change the port
- Use modern crypto

Specific Applications - sudo; git

sudo

- Enable auditing
- Use the fine-grained access control mechanisms
- Get mails on sudo access

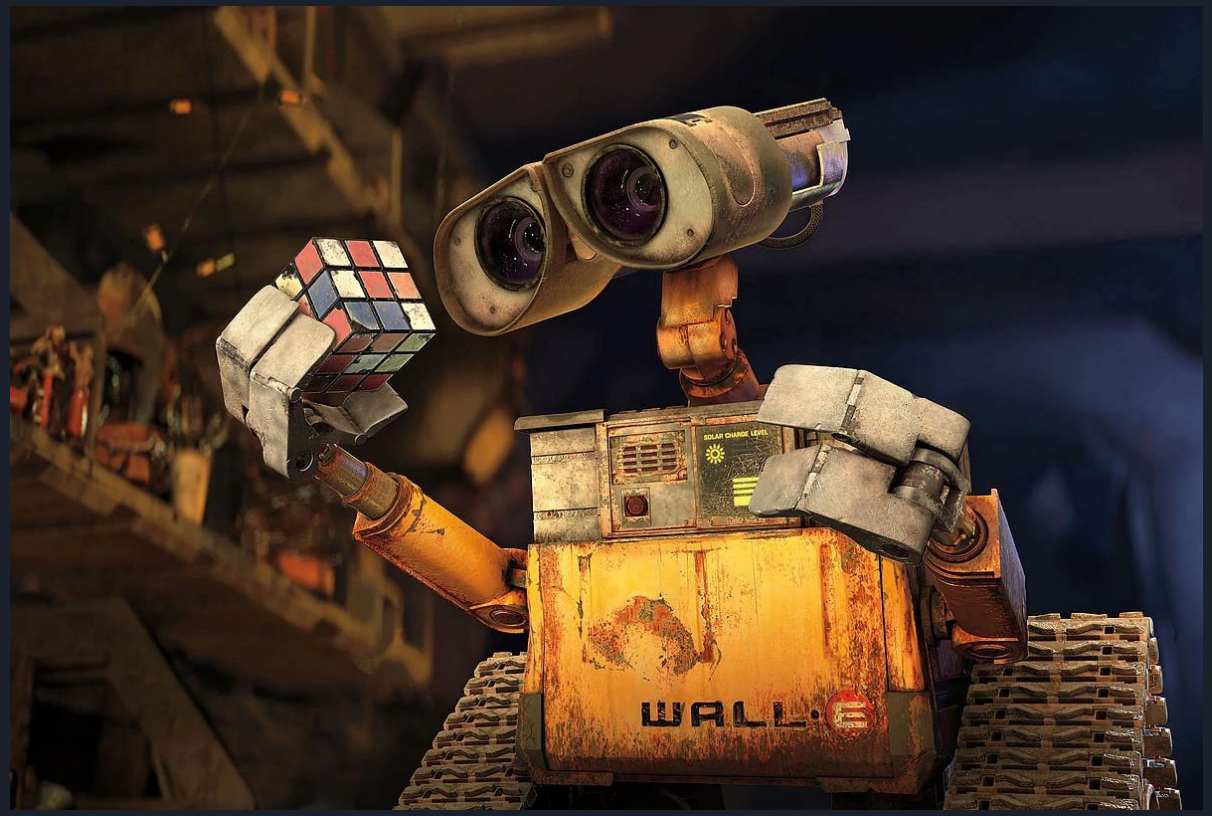
git

- use a proper identity
- sign your commits
- secure access to the repo
- follow the git best practices

Specific Applications - Key Storage

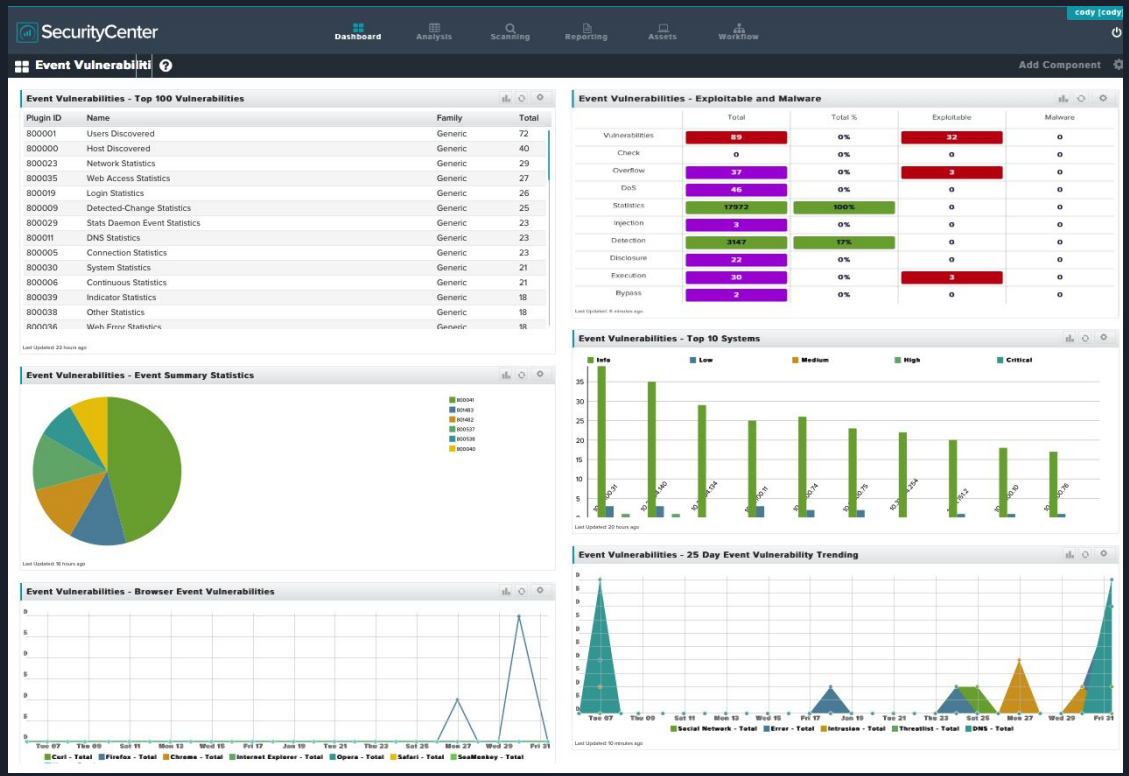
- A difficult topic, but here's a simple example
- Store the asymmetric keys in a hardware dongle and keep a backup offline
  - Only subkeys are on the token, master key is offline
  - Key can not be extracted from the token
  - Cryptographic operations are done on the token
- What can be done with them:
  - Create a set of gpg keys for multiple usages:
    - For SSH (e.g. for authentication)
    - For GPG (e.g. for sign and encryption; mails, git, passwords)
    - For TLS (e.g. your own CA or client cert)
- Some tokens
  - Yubikey 4
  - GPG Card

Tools for automated auditing and hardening



Nessus Scanner

- Commercial Tool with a community version
- Used to assess vulnerabilities in systems



`lynis`

Audit tool from cisofy

- Can audit linux system
  - Kernel Settings, Distro Settings, Applications Settings and more
- Can be used for penetration tests
- Can audit dockerfiles

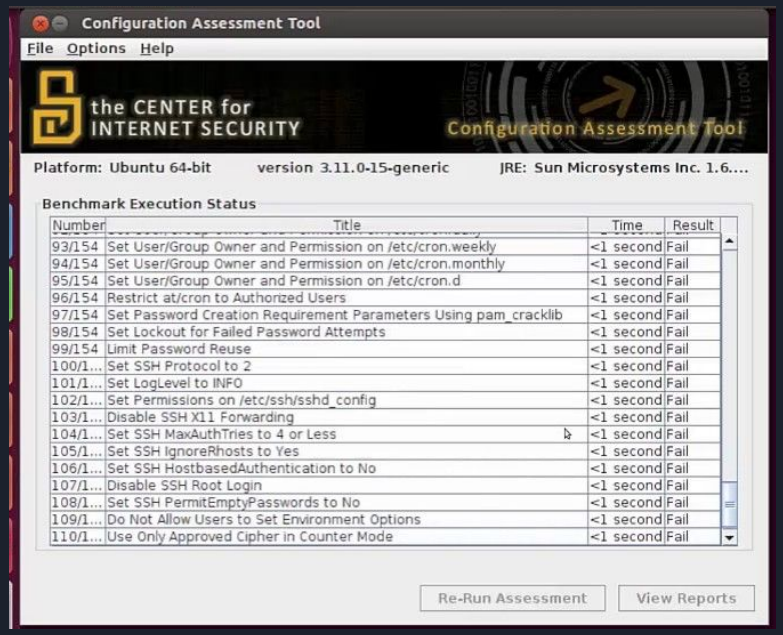
```
Lynis security scan details:

Hardening index : 64 [############        ]
Tests performed : 198
Plugins enabled : 1

Components:
- Firewall               [V]
- Malware scanner        [X]
```

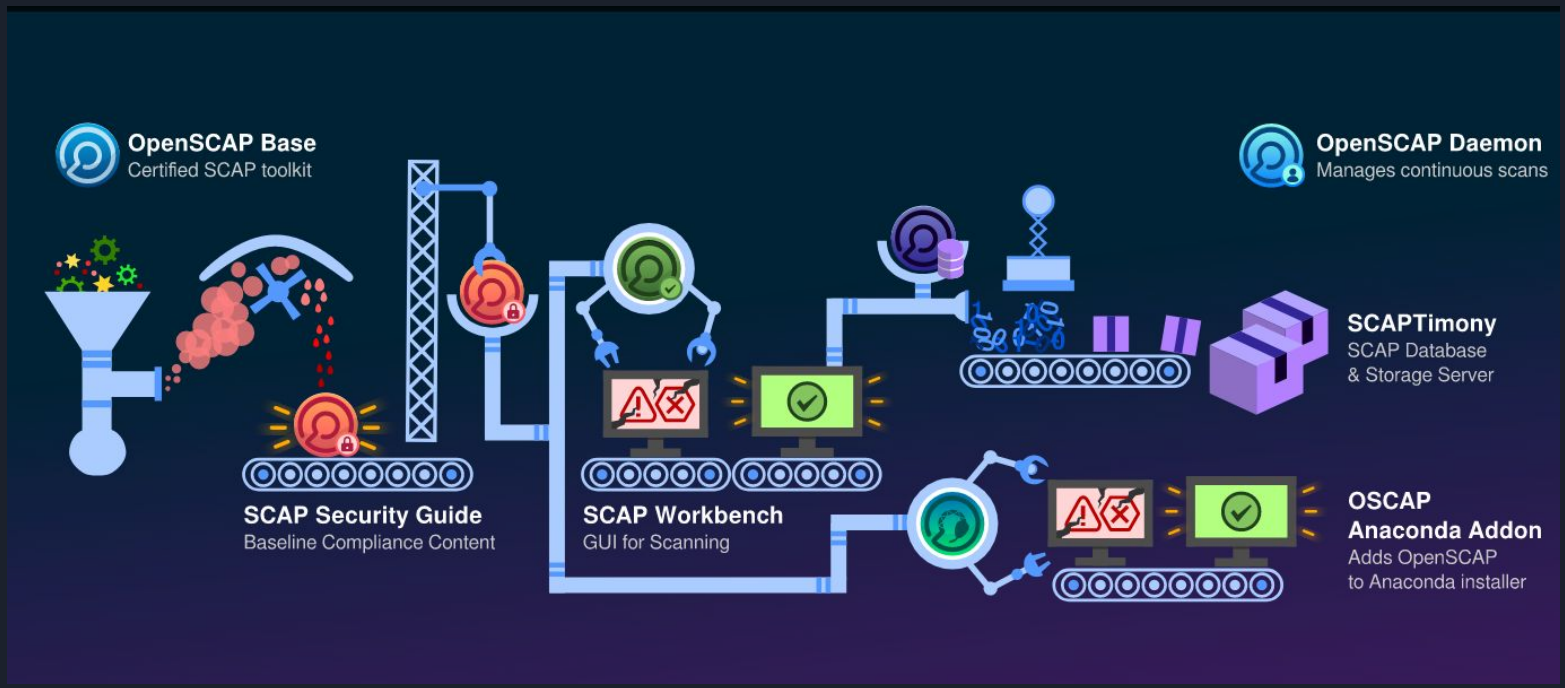
cis-cat

- From the center for internet security
- Can run with their benchmarks
- Is free and has a pro version for large scale usage



OpenSCAP

- Based on the Security Content Automation Protocol (SCAP)
- Currently a hot standard for auditing / hardening
- Has a complete pipeline built for hardening systems
- SCAP can be used with multiple tools (most of the time)



`do it yourself (diy)`

- If you got a specific set of requirements
  - most task are easy to automate (e.g. ansible, bash, python, ...)
  - many good examples are out there
- Take a look at http://dev-sec.io
  - Open source and automated (Big Thx to Simon for adding this)

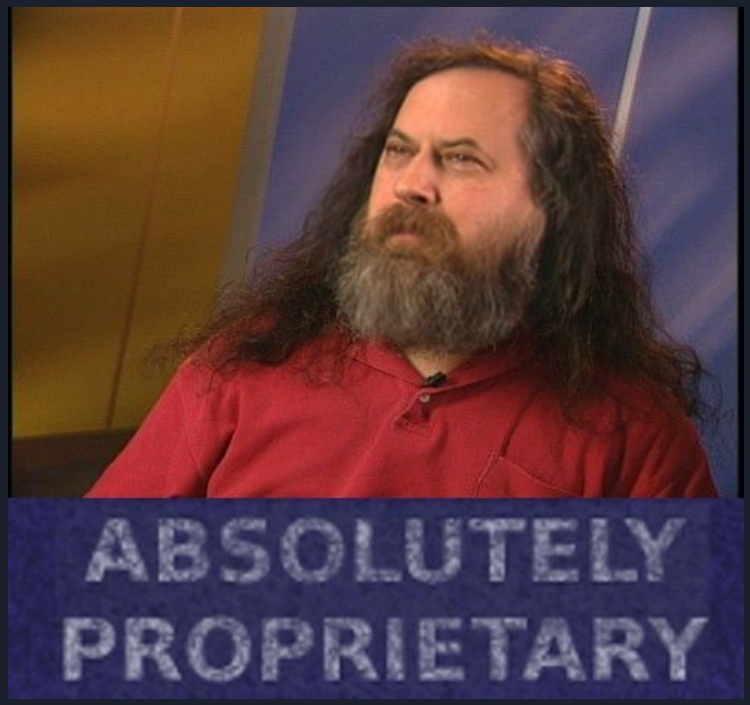
ANSIBLE CHEF™

the unspoken

- Nexpose (Rapid7)
- Secutor Compliance Automation Toolkit (S-CAT) (ThreatGuard)
- SCAP Compliance Checker (SPAWAR)
- BigFix Compliance (IBM)
- System Center Configuration Manager (Microsoft)
- Security Center (tenable networks)

the unspoken - extended edition

- Secutor Prime 5 (ThreatGuard)
- Qualys (Qualys)
- SAINT Security Suite 8 (Saint)
- BMC Server Automation (bmc)
- IBM Endpoint Manager (IBM)
- Policy Auditor 6.2 (intel)
- Tripwire Enterprise 8 (Tripwire)



More useful links

- Sources of measures in the beginning
- Web
  - Archlinux Wiki - Security
  - Other distro wikis (e.g. red hat security guide)
- Books
  - Linux Hardening in Hostile Networks. Server Security from TLS to Tor
  - Network Hardening: An Automated Approach to Improving Network Security
- Other
  - Google and RTFM

thats it ! thx