OCTOBER 2023

DEFENSIBLE SECURITY ARCHITECTURE

Barbara Krašovec

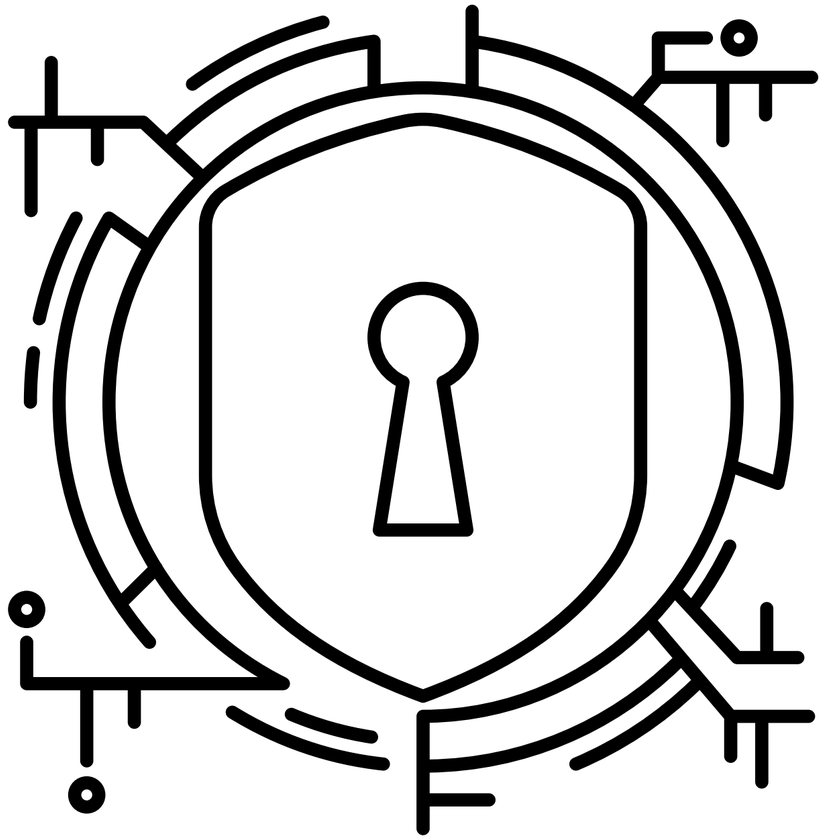


TABLE
of contents

Think like an attacker

”Think like a chef and see how well you do in the kitchen…”

- Adam Shostack

Defensible security architecture is a security model that designs, builds, operates and defends an infrastructure while continuously applying threat modelling and analysis during each of the process steps in a continuous lifecycle. It is based on the Zero trust principle.

Deficiencies of traditional approach

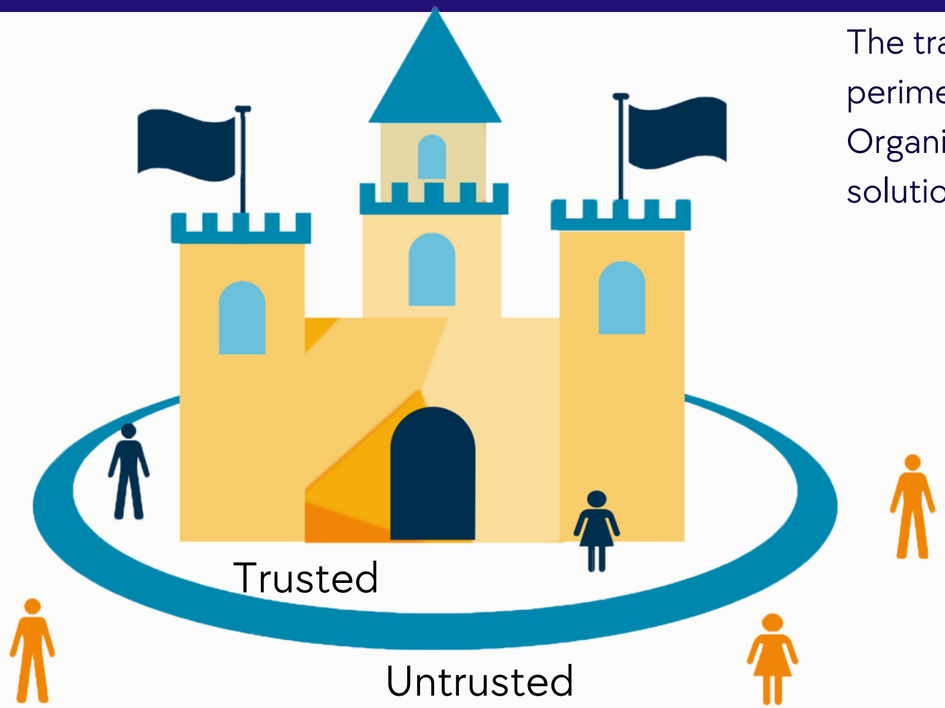


Image source: https://www.clouddirect.net/a-beginners-guide-to-zero-trust/t

"Castle-and-moat" is a network security model in which no one outside the network is able to access data on the inside, but everyone inside the network can. Imagine an organization's network as a castle and the network perimeter as a moat. Once the drawbridge is lowered and someone crosses it, they have free rein inside the castle grounds. Similarly, once a user connects to a network in this model, they are able to access all the applications and data within that network. (source Cloudflare)

The traditional approach focuses on defending the network perimeter. Everything inside the firewall is trusted.
Organisations deploy firewalls, IDS, IPS and other security solutions to prevent external attacks.

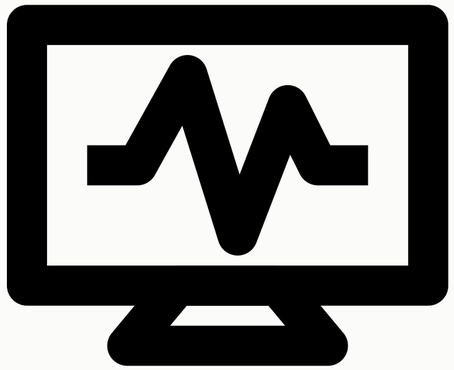
- How about insider threats?
- How about data breaches?
- What if services are running in the cloud?
- What if data is stored in the cloud?
- What if the equipment is used off-premises?
- What if employees work remotely?

Access control in this model is managed by virtual private networks (VPN). Remote users can connect via secure/encrypted tunnel to the organisation's services.
Problem:

- VPN becomes a single point of failure
- Traffic is encrypted, difficult to catch a malicious user.
- Encrypted traffic means slower performance
- Based on the server-client, when the server is overloaded, it has impact on usage (scalability problem).
- VPNs require maintenance, on server and client side.

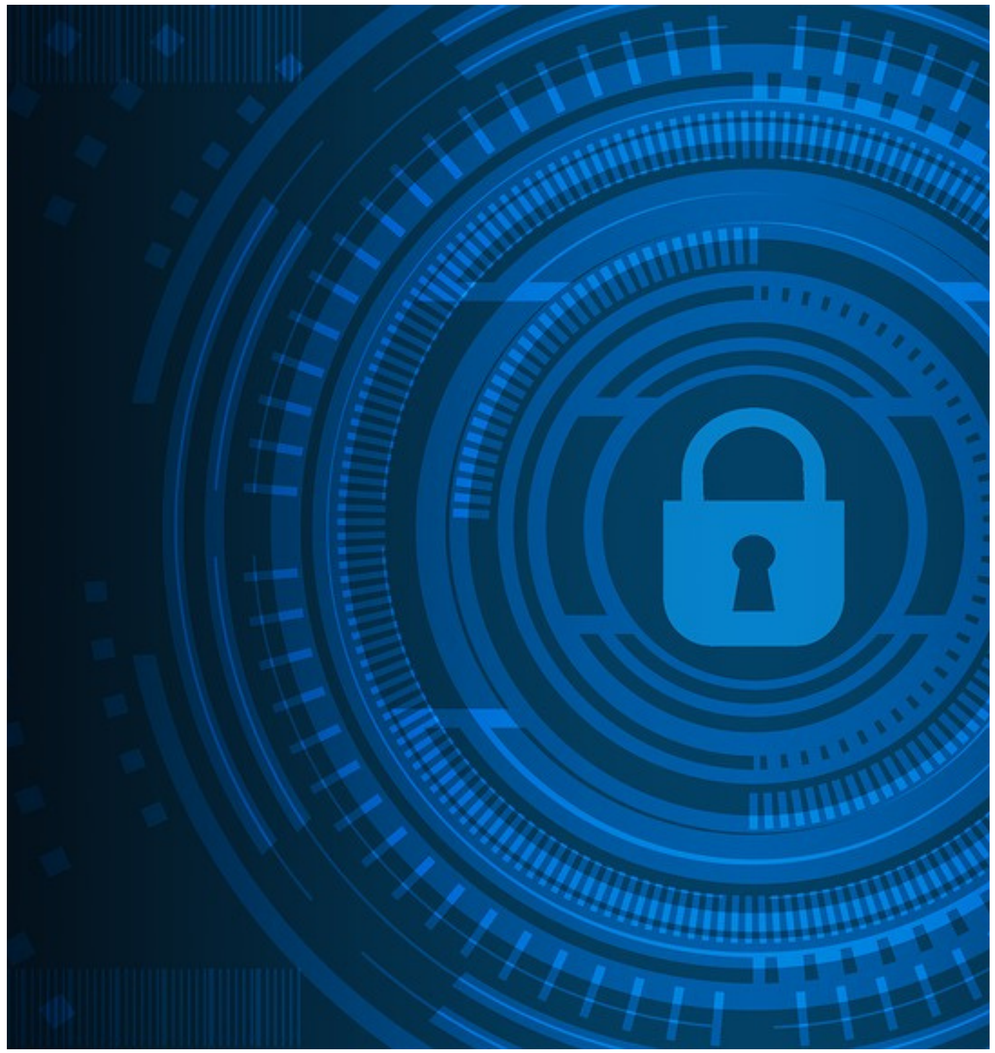
Design principles for a defensible architecture

- assign least privilege
- separate responsibilities
- apply zero trust principles
- simplest solution possible should be the goal
- define critical resources and audit sensitive events
- protect your data
- implement defence-in-depth
- choose verified security tools (no time for innovation here)
- provide enough training/education to develop sufficient skill set and for people to be familiar with the security procedures



WHAT TO MONITOR?

- privileged accounts activity
- brute force authentication failures
- authentication anomalies
- session anomalies (spikes on firewall)
- account anomalies
- excessive service failures
- unusual traffic
- signature matching to known vulnerabilities
- insider threat indications
- data exfiltration indicators
- IoC detection



Network security

Physical security

Network security controls and physical security controls are crucial to applying the defence-in-depth and zero-trust principle.

- protect areas,
- put an organisational policy in place that covers physical and network security,
- protect devices,
- patch regularly,
- separate network into multiple segments,
- encryption is important to secure data transfers etc.
- do not assign privileges and access on a “just in case” basis
- use firewalls, ACLs and other protection systems (IDS, IPS)
- monitoring and logging is crucial -> DETECT WHAT YOU CANNOT PREVENT

Data security

Ponemon Institute evaluates that the average cost of a data breach worldwide is 4.35M dollars.

Data security protects data from malicious threats: activity monitoring, network security, access control, encryption, and authentication.

WHICH DATA?

- **data in motion** (secure transfers + endpoint security of devices used in communication)
- **data at rest** (storage location)
- **data in use, data in memory** (host security)

To ensure privacy, we need security.

Privacy vs security

DATA PRIVACY

- Privacy relates to the appropriate use of data.
- Data privacy addresses proper handling, processing, and storage of data: security policies and permissions.
- Data privacy focuses on the control people have over their personal data and on how they can protect if from unwanted or harmful uses, such as sales to third parties.
- Standards: GDPR

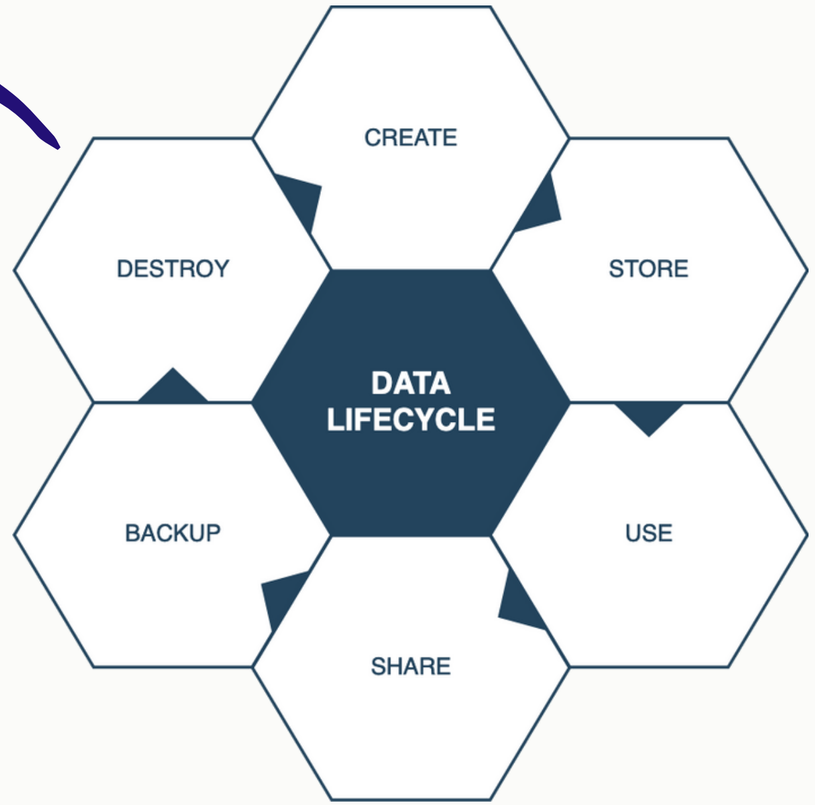
What kind of data is processed?
Where will it be stored?
For how long?

DATA SECURITY

- Security relates to the confidentiality, availability and integrity of data.
- Data security protects data from malicious threats: activity monitoring, network security, access control, encryption, and authentication.
- Data security focuses on the physical security of data on-premises and the logical security of digitalized data.
- Standards: ISO27k, PCI-CP..

How to protect a site, a server, where the data is stored?
How to protect equipment?
How to protect people ?

Data lifecycle



Data security is the process of safeguarding digital information **throughout its entire life cycle** to protect it from corruption, theft, or unauthorized access

GOVERNANCE -Who can create it, what is critical data, how to protect it?
Data security policy, classification, catalogue, resilience

DISCOVERY - where is the data, in DB, files, on the network?

PROTECTION - How to protect it?
Encryption, key management, access controls, backups, replicas etc.

COMPLIANCE
Compliance to the organisation's data and retention policy, regulations, GDPR etc.

DETECTION
Monitor who access data, UBA (User Behaviour analytics) and report if any unusual activity.

RESPONSE
Ensure fast and automatic recovery.

Data integrity

- ***FIM software***

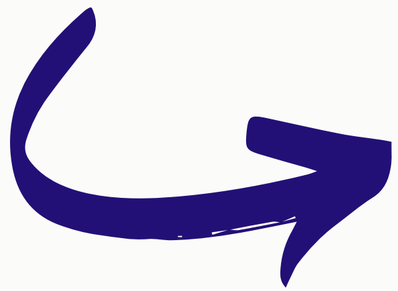
  is a software that monitors and detects file changes that could be indicative of a cyberattack and reports them.

  - FIM software: Tripwire, Samhain, OSSEC
  - you can set audit.rules on Linux, but only check sensible/critical folders

- ***HIDS software***

  stands for a host-based intrusion detection system and represents an application that monitors a computer or network for suspicious activities.

  - HIDS tools: OSSEC, Wazuh, AIDE



PROBLEMS:

- Poorly configured FIM software can lead to excessive alerts causing alert fatigue
- Not suitable for big data

Ausearch

```
~# ausearch --message USER_LOGIN --interpret | less
----
type=USER_LOGIN msg=audit(06/16/2022 07:44:45.104:1919347) : pid=3905900 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=157.245.245.11 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:44:55.132:1919370) : pid=3905953 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=root exe=/usr/sbin/sshd hostname=? addr=157.230.98.148 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:44:55.924:1919376) : pid=3905950 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=root exe=/usr/sbin/sshd hostname=? addr=159.65.171.230 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:01.267:1919383) : pid=3905957 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=103.246.240.28 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:11.482:1919395) : pid=3906040 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=198.12.227.59 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:33.156:1919407) : pid=3906069 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=180.76.106.84 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:44.932:1919429) : pid=3906145 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=188.128.39.113 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:45.698:1919435) : pid=3906146 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=(unknown) exe=/usr/sbin/sshd hostname=? addr=159.65.171.230 terminal=ssh res=failed'
----
type=USER_LOGIN msg=audit(06/16/2022 07:45:58.330:1919453) : pid=3906255 uid=root auid=unset ses=unset subj=system_u:system_r:sshd_t:s0-s0:c0.c
1023 msg='op=login acct=root exe=/usr/sbin/sshd hostname=? addr=82.118.225.196 terminal=ssh res=failed'
```

Data protection risk assessment

Risk assessment of the data should be conducted, to identify and evaluate the potential threats and vulnerabilities that may affect the confidentiality, integrity, and availability of data

Regulations: https://commission.europa.eu/law/law-topic/data-protection_en

SOME COMMON THREATS:

- unauthorised access
- data disclosure
- ransomware
- phishing
- insider threat
- data loss
- data corruption

- CLASSIFICATION OF DATA based on the severity of the risk and the likelihood of its compromise or exposure.

Data security solutions

Protect data from unauthorised or unlawful access, use, disclosure, modification, or destruction. Techniques:

- provide lifecycle management,
- IAM (strong password policy, centralised user and access management, least privilege)
- restrict access to data (ACLs, firewall, authN, authZ)
- data masking, obscure raw data and only display selected portions during operations
- monitoring activity of authorised users
- monitoring any unusual activity (large transfers, access from unusual location etc.)
- carefully chosen data storage rotation
- data policy for storage, deletion, access
- endpoint security
- auditing and penetration tests
- data transfers are restricted and allowed over secure channels,
- provide backup and replication,
- encryption and key management (on AWS, newly added resources will be terminated if encryption is not enabled
- apply SIEM, FIM.

Endpoint security

Endpoint security, or endpoint protection, refers to securing endpoints — such as desktops, laptops, and mobile devices — from cybersecurity threats. Endpoints can create entry points to organizational networks which cybercriminals can exploit. Endpoint security protects these entry points from malicious attacks. (Kaspersky)

From a hardware perspective, an endpoint is:

- server
- desktop
- IoT device
- network device
- laptop

From a software perspective, an endpoint is:

- Linux
- Unix
- Windows
- MacOS

GOAL OF ENDPOINT SECURITY:

- reduce attack surface
- reduce consequences of potential attack
- detect suspicious activity

HOW?

- patching
- encryption
- regular updates
- access control
- turning off unnecessary services

Prerequisites

- Asset inventory
- Configuration management
- Change control
- Security policies

Securing your desktop

- don‘t install software, that you don‘t need,
- don‘t install software from untrusted sources,
- install and configure a firewall, and an antivirus program
- keep all software up to date,
- make sure your passwords are well-chosen and protected,
- use YubiKeys, verification apps, and other authentication tokens to sign in.
- when practical, consider biometric authentication (e.g., fingerprint scanning).
- don't open suspicious attachments or click unusual links in messages,
- browse the web safely,
- don‘t plug in unknown external devices (such as a USB-key)
- encrypt your hard drive,
- make regular backups of your data

Securing IoT devices

Internet of Things (IoT) devices are Internet-connected objects, such as networked security cameras, smart refrigerators, fire alarms etc.

How to protect them?

IoT devices have firmware and software which is essentially an operating system that needs to be protected:

- Upgrade firmware and software regularly, especially when security patches are available.
- Use strong passwords to access the devices.
- Disable physical access to the devices.
- Use encryption to protect the communication of the device.
- Turn off features that you don't need.

Essentials for endpoint security

- Configuration management,
- applying security policies,
- change control,
- authentication and authorisation security measures,
- host hardening,
- logging and monitoring.

Configuration Management

Process of monitoring/deploying the hardware and software configuration in line with IT policies.

- Enables consistency and automation,
- enables traceability of configuration changes,
- reduces security breaches,
- reduced time to restore service,
- efficient change management,
- easier upgrade automation,
- higher quality of service,
- control over running processes and permissions over the files,
- configuration backup and documentation.


CFEngine
puppet labs
PalletOps
ANSIBLE
CHEF
SALTSTACK

Security policies

Password policies
Software policies
Hardware policies
Data protection policies
Access Policies
AUP

- How many characters are in the password? How often does it need to be changed?
- Which software is obligatory? SELinux installed or disabled? Which is forbidden? Server checklist required?
- Network devices from which vendor? A template for hardware configuration provided?
- When can data be deleted? Who can delete it? Where it will be stored, who will have access to it? From where?
- Which services can run on a server? Which services should be disabled?
- Who can access an endpoint, from where? Who has administrative privileges?
- Get consent from the user for monitoring, using and wiping his data. Also important for the user to know what is allowed and what is the normal operation.

Security policies templates

Document	Who should complete the template?	Audience	Description	Link
Top Level Infrastructure Policy	Infrastructure Management	All Infrastructure Participants (abides by)	This policy template defines the roles of actors in the Research Infrastructure and binds the policy set together	Google Doc
Incident Response Procedure	Infrastructure Management & Security Contact	Infrastructure Security Contact, Services (abides by)	This template procedure provides a step-by-step breakdown of actions to take following a security incident.	Google Doc
Membership Management Policy	Infrastructure Management	Research Community (abides by)	This policy template defines how Research Communities should manage their members, including registration and expiration.	Google Doc
Acceptable Authentication Assurance	Infrastructure Management	Research Community, Services (abide by)	This is a placeholder for the Infrastructure to determine rules for the acceptable assurance profiles of user credentials.	Google Doc
Risk Assessment	Infrastructure Management, Services & Security Contact	Infrastructure Management (completes)	This table can be used as a starting point for identifying whether a full Data Protection Impact Assessment is required.	Google Doc
Policy on the Processing of Personal Data	Infrastructure Management & Data Protection Contact	Research Community, Services (abide by)	This document defines the obligations on Infrastructure Participants when processing personal data.	Google Doc

AuthN and AuthZ

Authentication is the process of verifying who a user is, while authorization is the process of verifying what they have access to

- **single source of truth = IAM system**, used for provisioning and de-provisioning of accounts
  - Approval process (is your account still valid? For how long?)
- secure systems rely on **MFA** -> trend is passwordless authentication (SSO)
  - **something that you have** (access card, OTP generator on mobile device)
  - **something that you are** (fingerprint, face recognition)
  - **something that you know** (password)
- Audit your procedures and monitor user activity

Federated identity - when you connect to multiple service providers using the same identity provider. This is a big problem in case of stolen credentials or compromise.

Change control

FILE INTEGRITY CHECKS

- define which files should be monitored for change
- an attacker will always change some files on the system

File integrity checks use a mathematical function called a one-way hash. Hashes are stored and cannot be modified. At defined intervals, the hash against the original is checked. If the hashes do not match, an alert is issued. An alert is sent also when a new file appears in a folder. Poorly configured FIM and HIDS systems can lead to excessive alerts causing Alert Fatigue.

FIM = file integrity monitoring

- monitors and detects file changes that could be indicative of a cyberattack and reports them

HIDS = host intrusion detection system

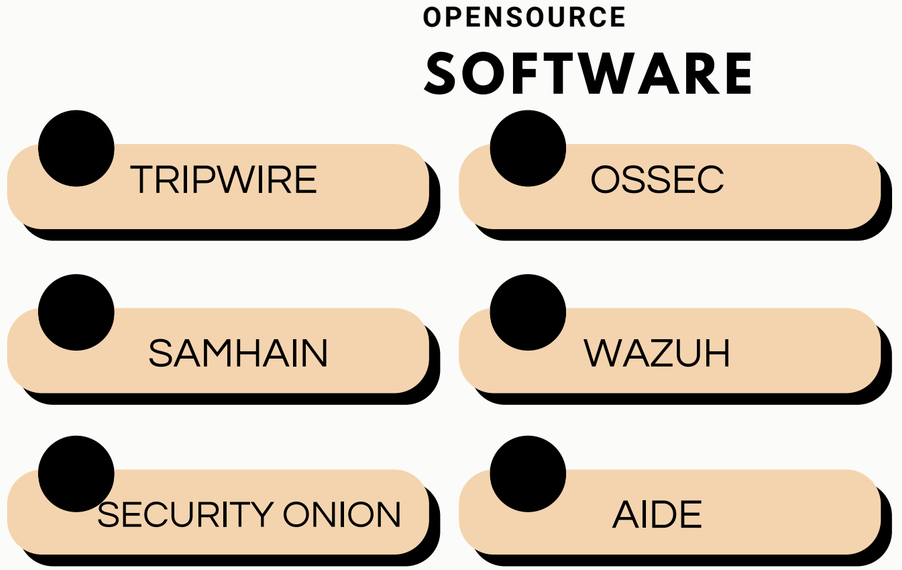
- monitors traffic on the host
- checks file changes, abnormal activity in logs or network traffic

```
Check trusted computing base:
#kernel modules
/lib/modules

#binaries
/bin, /sbin, /usr/bin, /usr/sbin,
/usr/local/bin, /usr/local/sbin

#system configuration
/etc

#critical files in:
/var/spool, /boot, /home
```



Linux and host hardening

Essentials of Linux operating system security

- modify kernel settings at runtime (sysctl), blacklist uneeded kernel modules,
- network: close unneeded ports, limit access and services (ACLs) also verify which ports are opened
- protect files - minimise access rights, FIM,
- automate what you can, use configuration management tools,
- access: use MFA or at least SSH keys to log in, use auditing, for passwords use strong change policy,
- logging and monitoring: use central logging.
- protect configuration files
- enable SElinux, auditing, and AppArmor to limit the capabilities of programs
- use VPN/Wireguard or similar to connect to other segments
- disable X11
- use also software firewall (nftables, firewalld, iptables etc)
- disable root cron jobs
- check for backdoors and rootkits

Host hardening - users and access

- It is poor practice if multiple admins share the same root account and/or use the same password on multiple servers - provide unique accounts to admins, and use session recording.
- use central IAM for user management,
- user credentials should never be transferred over an insecure channel,
- use MFA for SSH login. Disable password login to the servers, disable root login,
- enable auditing,
- ensure that private keys (SSH, X.509,...) are not stored unencrypted on publicly accessible servers (such as user interfaces or login nodes). Ideally, no private keys should be stored on publicly accessible servers,
- configure termination for inactive sessions,
- users should sign AUP,
- apply secure access rules,
- lock account after multiple failed attempts (faillog or fail2ban)
- use encryption (sftp, scp, ssh) for file transfer and accessing the machines
- make sure no other user than root has UID 0 (awk -F: '($3 == "0") {print}' /etc/passwd)
- lock all accounts with empty passwords

```
for user in `awk –F: '($2 == "") {print}'` /etc/shadow; do passed –l $user;done
```

Provide account provisioning and de-provisioning policies, and use an identity governance system (IAM = central service for account creation/deletion): the user accounts are created, modified and terminated on a single service which ensures consistency. By configuring user accounts for each service locally, it will require a lot of work to keep the validity of users (are they still employed, entitled to use the services?) in sync with their access to the servers.

Host hardening - packages and services

- automate OS deployment and configuration (using Foreman, Puppet, Ansible or others),
- minimise the number of installed packages (dnf remove package),
- disable/mask unused services (systemctl disable service),
- update regularly,
- apply patches and verify them with services like Nessus, Pakiti, OpenVas,
- subscribe to and follow security notifications for the software you rely on,
- ensure that all remote server management interfaces (e.g. BMC subsystems) are not exposed externally, access should be restricted to administrative users with ACLs in place. Disable default built-in administrative accounts that might be provided by default by the vendor. A unique administrative account configuration for your management systems is usually a good practice.
- regularly scan for vulnerabilities.

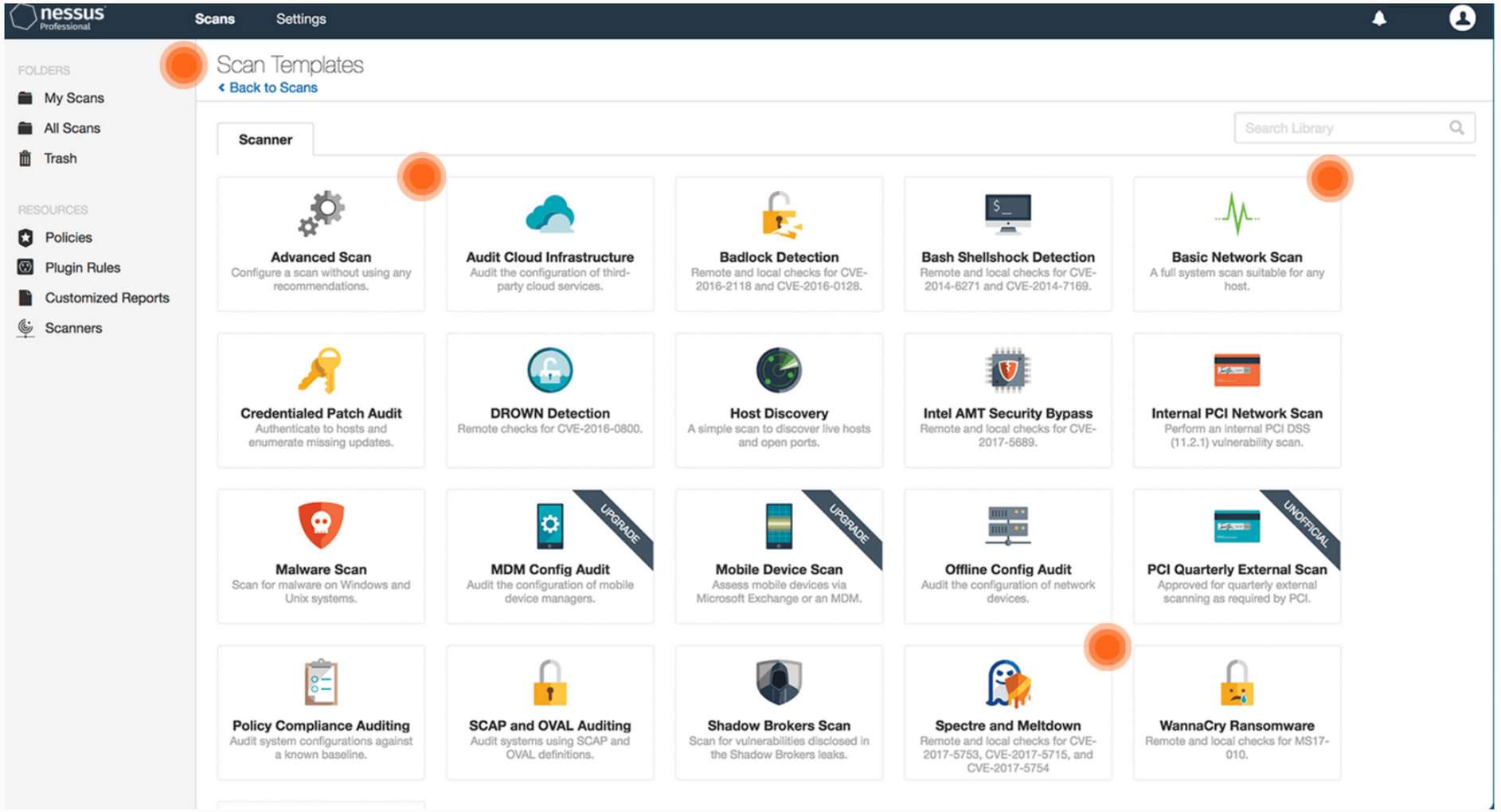
Linux hardening tools

Advanced task for a sysadmin. Checklists available, but demand knowledge. Some Linux hardening tools available:

- **Nessus**: security vulnerability scanning tool (checks services and alerts about misconfigurations)
- **Zeus**: configuration audit, security assessment, self-assessment, system hardening for AWS
- **OpenSCAP**: vulnerability scanning and security audit tool
- **Lynis**: scan system for expired SSLs, outdated software, no password user accounts, files etc.

Nessus

https://www.tenable.com/products/nessus/demo



OpenSCAP

Security Content Automation Protocol (SCAP) is a framework for security standards,it provides tools for assessment, measurement and enforcement of security baselines - how to harden your system and detect misconfigurations.

- Guidelines for Linux,
- validated by NIST (National Institute of Standards and Technology),
- CIS control included,
- command-line tool oscap, GUI is scap-workbench,
- note that the tool has a limited span of checks and guidelines.

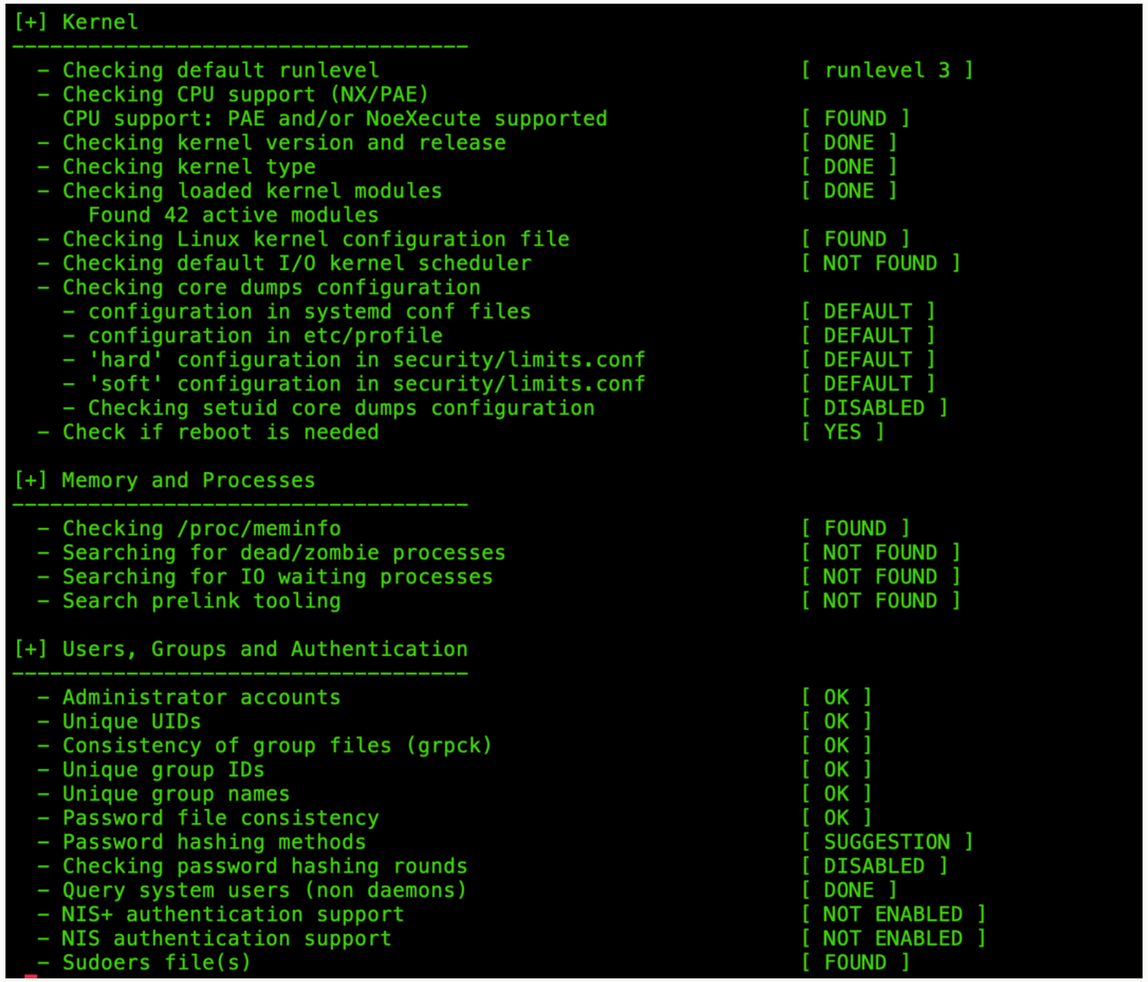
OpenSCAP report

Title	Severity	Result
Guide to the Secure Configuration of Red Hat Enterprise Linux 8 1x fail 1x notchecked		
System Settings 1x fail 1x notchecked		
Installing and Maintaining Software 1x notchecked		
System and Software Integrity		
GNOME Desktop Environment		
Updating Software 1x notchecked		
Ensure gpgcheck Enabled In Main yum Configuration	high	notapplicable
Ensure gpgcheck Enabled for All yum Package Repositories	high	pass
Ensure Red Hat GPG Key Installed	high	pass
Ensure Software Patches Installed	high	notchecked
Account and Access Control		
Protect Accounts by Configuring PAM		
Set Lockouts for Failed Password Attempts		
Set Password Quality Requirements		
Set Password Hashing Algorithm		
Ensure PAM Displays Last Logon/Access Notification	low	notapplicable
Protect Physical Console Access		
Protect Accounts by Restricting Password-Based Login		

Lynis

Security auditing tool for systems running Linux or Unix-based operating system

- Security scan,
- file permissions checks,
- tips for additional OS hardening: kernel parameters (sysctl), SSH configuration, PAM configuration etc.,
- vendor guides included,
- supports multiple standards, such as NIST and also CIS benchmarks.



Rootkit detectors

Rootkit is a type of malware that allows an attacker to gain access and control a target server.

- most common is to install them through phishing or other social engineering attacks or by exploiting a vulnerability
- can be a hidden process
- different kinds: hardware, application, kernel rootkit
- examples: Diamorphine, Flame, Stuxnet etc.

What to log?

Logging

- problem are different formats, timestamps, timezones
- use centralised log management, then analyse
- normalise logs (same format for all)
- provide log rotation
- specify log rotation policy (diskspace, regulatory requirements)
- visualise vital logs
- software: NXlogs, ELK, Graylog, Loki, rsyslog, syslog-ng

Logging checklist

https://www.sans.org/brochure/course/log-management-in-depth/6

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.

Application security

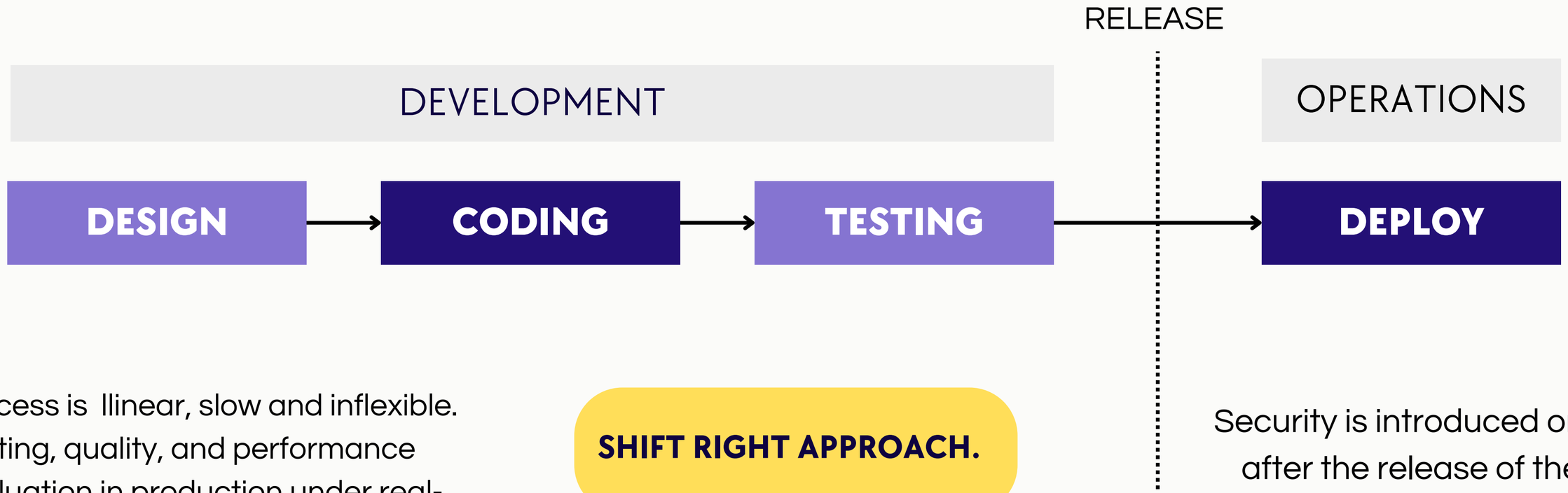
FACTS

- All software has bugs and vulnerabilities.
- Little software is written from scratch (taken from other sources and modified).
- We are often not aware of the dependencies/libraries used.
- Security must be implemented by design.
- If you find a vulnerability in the development phase, it is less costly as when the software is already released

DEFINITION

Application security (short AppSec) i**ncludes all tasks that introduce a secure software development life cycle** to development teams. Its final goal is to improve security practices and, through that, to find, fix and preferably prevent security issues within applications. It encompasses the whole application life cycle from requirements analysis, design, implementation, verification as well as maintenance (source Wikipedia)

Traditional approach to software development

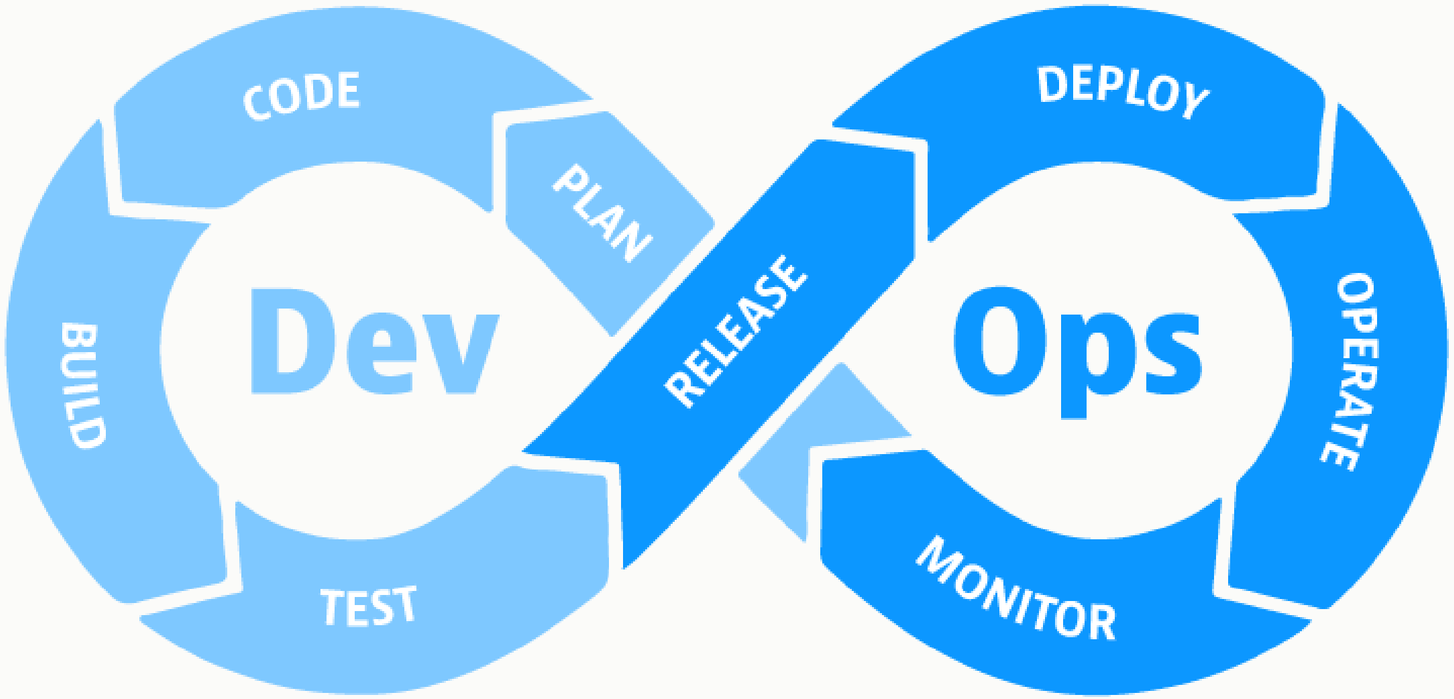


- Process is llinear, slow and inflexible.
- Testing, quality, and performance evaluation in production under real-world conditions

SHIFT RIGHT APPROACH.

Security is introduced only after the release of the software.

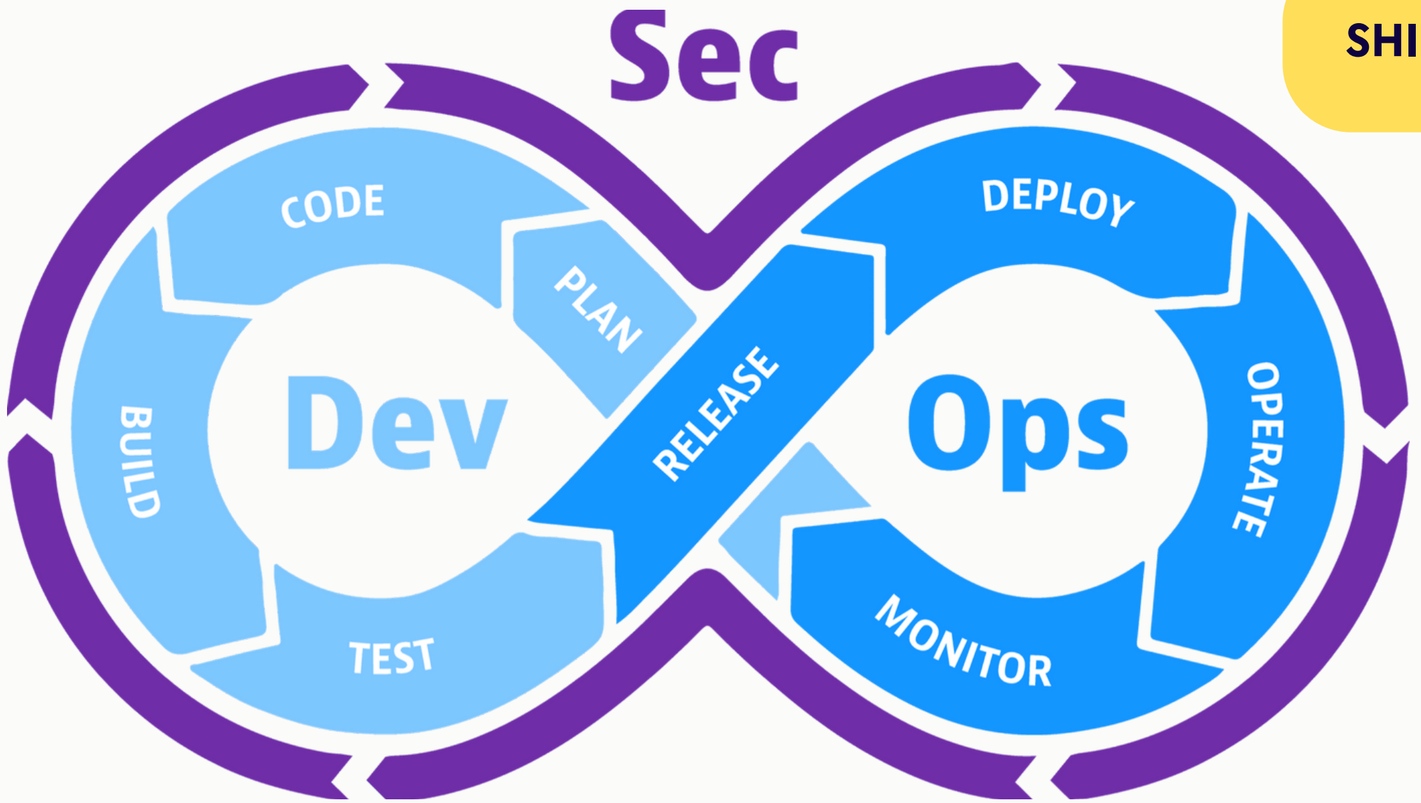
Dev-ops approach to software development



Five pillars of DevOps:

- culture,
- automation,
- measurement,
- sharing, and
- learning

Dev-ops approach to software development



SHIFT LEFT APPROACH.

The term "shift left" refers to the efforts of a DevOps team to guarantee application security at the earliest stages in the development lifecycle, as part of an organizational pattern known as DevSecOps.

- Dev-ops cycle is agile, fast, includes continuous integration
- DEV-SEC-OPS: security in all phases of the cycle
- Requires collaboration between developers, security engineers and operations.

Secure Coding Practice Quick-reference Guide

Secure coding

- **OWASP standard**

OWASP Developer Guide: https://owasp.org/www-project-developer-guide/
OWASP Secure Coding Practices Checklist: https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist.html

- **Recommendations**

- Take code from trusted sources and use trusted libraries (however log4j)
- keep software bill of materials (you want to document all dependencies and where they came from)
- Apply security measures before the release of softare

Top 10 vulnerabilities by OWASP

1. Injection
2. Cryptographic failures
3. Insecure Design
4. Broken Access Controls
5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery (SSRF)

- **How to prevent these?**
  - use security scanning tools
  - use encryption
  - use central logging
  - check if authorisation rules apply: verify that a requested action or service is approved for a specific entity
  - ensure proper authentication, login pages must use TLS, users validity should be properly managed - the system needs both defined provisioning and de-provisioning, recommended to use a single source of truth (use SSO when possible)
  - with vulnerability testing (SAST, DAST)

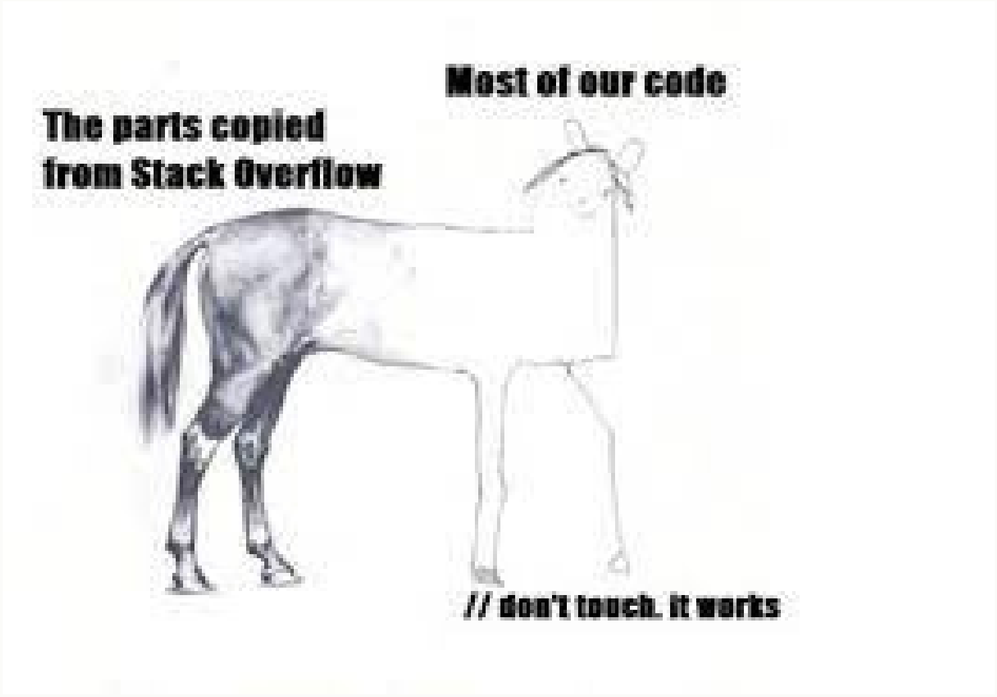
AI Chatbots in Software Development

- Code is generated quickly
- Useful to debug code

BUT:

- easy to inject vulnerability (misconfiguration, bad code, malicious code
- you expose your IP
- you can release your proprietary code by mistake

When you copy code from stackoverflow and GitHub



Cybersecurity threats

- ***In 2023***
  - data breaches (social engineering), phishing,
  - email is still the leading infection vector
  - cloud security
  - IAM (MFA) - stolen credentials, brute-force password attacks
  - zero-day software vulnerabilities
  - malware: trojans, ransomware, rootkits
  - remote work exploits (using VPN channels to get to the office network)
  - DoS
- ***Future?***
  - same as in 2023
  - compromises of IoT
  - AI (deep fake + AI to develop attacks)
  - Quantum (breaking encryption - problem of insecure communication channels)
  - lack of skills (not enough security experts)

HPC Security

- ***Challenges***

- Hard to detect anomalies and inappropriate activities.
- What to monitor?
- No shared threat intelligence and best practices.
- Ensuring a balance between security and performance.
- It is easy "to hide" in the system.
- File integrity - too many files to scan them.
- Cannot really separate development and production environment (too large scale).
- Rapidly changing environment.
- Scalability of consequences of an incident.

- ***Main risks***

- misuse of cycles (e.g. bitcoin mining)
- loss of file integrity,
- intelectual property confidentiality,
- insider threats,
- supply chain attacks.

HPC is traditionally a closed system with little focus on security. With international collaboration, large datasets and access to external storage systems, HPC needs to open some components to the Internet, and security gains importance.

Vendor security assessment

Vendor security assessment

- Make security part of the procurement process and put security requirements in the contract with the vendor.

Assess vendor's security maturity:

- vendor's transparency
- vendor's support in security incidents, vulnerability handling
- vendor's approach to product support (regular maintenance, defined product lifecycle, development process)

What to require from the vendor:

- defined security policies and procedures,
- redundancy for critical services,
- automated processes for building and deployment,
- detailed software documentation,
- controls for software and updates.

Example: How to prepare for a ransomware

Most often attack vector is phishing or a similar social engineering attack.

1. Backup
2. IR plan
3. Education, raising awareness
4. Technological security controls

Backup

- maintain an offline, encrypted backup
- regularly test disaster recovery
- maintain “golden images” of critical systems
- for cloud services, consider a multi-cloud solution to avoid vendor lock-in

COMMON MISTAKES

- no backup
- restoring from backup doesn’t work
- backup is very old
- backup is on the same storage as a compromised server

IR exercise

- prepare for PII breaches or disclosure of sensitive information (e.g. health records) - who to inform?
- prepare contact lists in advance
- assign roles: who is the coordinator, who lead investigator, who will contact the media
- prepare VM image for investigation
- prepare a sandbox for analysis
- define communication tools

COMMON MISTAKES

- no IR response plan
- contact list contains contacts that are not updated
- the organisation has no tools and no servers where they could run analysis

Fundamental security measures

- implement zero trust
- conduct regular vulnerability scanning
- update OS/packages regularly
- harden devices and host
- limit the use of RDP and other desktop services (most often entry point)
- for Samba, disable SMBv1 and SMBv2, only use SMBv3 (often used to propagate malware in lateral movement)
- limit SMB traffic, disable external access to SMB/NFS

COMMON MISTAKES

- only a single control is used to protect a service (firewall)
- access to SMB not restricted from internal networks
- no OS hardening in place
- storage systems not in a separate network segment
- also same storage used for multiple services (in DMZ, internal network etc.)

User-related security measures

- educate people on phishing
- apply MFA
- use central IAM (easy to block user's access to other machines in case of compromise
- disable root login
- change default admin credentials
- enforce account lock-out after multiple failed attempts
- store passwords in secure DB with strong hashing algorithms
- don't save passwords in browsers or in plain text files
- implement a strong password policy (password at least 15 characters)

COMMON MISTAKES

- all mentioned not applied

Email security

- implement filters on the email gateway
- use SPAM filters and apply additional rules for recognised phishing emails
- ensure macro scripts are disabled on MS Office files transmitted via email (recent versions of MS Office have this set by default -und Application/Settings/Security)

COMMON MISTAKES

- users' ignorance (click on malicious link or they open a malicious attachment)

QUIZ

1. What are the deficiencies of the traditional approach to security architecture?

2. Are the following statements true or false?
   - The biggest problem of file integrity monitoring is a badly configured system that causes alert fatigue and big data.
   - Ransomware is one of the common threats to data security.
   - Putting a disk in a case to prevent damage is a matter of data security.
   - Endpoint security encompasses OS security and host hardening.
   - Central IAM service is not recommended as it represents a single point of trust.
   - Shift left approach is a synonym for penetration testing when an application is released in production.
   - OWASP standard for secure coding includes also error handling and logging.

3. How would you recommend an admin to log in to Linux servers? What are the common mistakes to avoid?

4. Why is it beneficial to use configuration management?

5. Name at least 3 security controls that could be applied to prevent ransomware.