Cyber Security Foundations | CySec

Summary

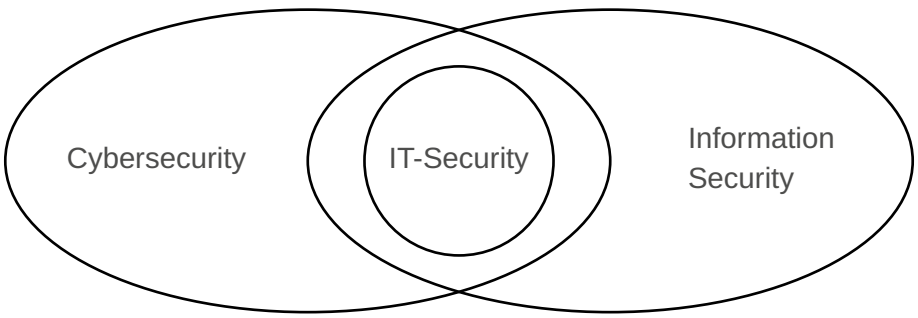
CONTENTS

1. PREFACE

Copypasted straight from the lectures. Best viewed as the anki deck, so that all of the hundreds of abbreviations and corpospeak can be burned into your brain for the duration of the exams and be forgotten right afterwards. Exam is the easiest there is, so do not fret.

2. INFORMATION SECURITY

Term	*Definition*
Information	An organization's data that has been processed, organized, or structured in a way that gives it meaning and value to an organization or individual.
Information security	Protection of the integrity, confidentiality and availability of information data whether in storage, transit or processing.
non-repudiation	prevents parties from denying actions they have performed.
accountability	ability to trace actions and decisions back to a specific person or system.
authentication	verifies the identity of a user or system.
authorization	determines what actions an authenticated entity is allowed to perform.
access control	restricts access to resources based on defined rules.
business continuity	ensures critical operations continue during disruptions.
security policy	a rule or expectation for protecting information.
compliance	adherence to laws, regulations, and (security-)standards.
asset	any item of value belonging to an organization – For example: information, systems, people and processes
attack	an act that intends to damage, steal or degrade an organizations assets
vulnerability	a flaw or weakness in a system that can be abused
exploit	a technique or method used to take advantage of a vulnerability
threat	an event or action with the potential to cause harm by exploiting a vulnerability
risk	the likelihood of a threat exploiting a vulnerability and the potential harm that could cause
control	a measure designed to reduce the potential risk of an attack – Can be achieved through training employees, enforcing policies or implementing technology



2.1. TYPES OF INFORMATION

- Personal information
- Business information
- Financial information
- Intellectual property
  - Copyright
  - Trademarks
  - Patents
  - Trade secrets
- System information

2.2. HOW CAN INFORMATION BE ATTACKED

- In storage
  - Data that is stored on a server or in a database short-term or long-term.
- In transit
  - Data that is currently being transported from one place to another.
- In use
  - Data that is currently being processed by a service or another entity.

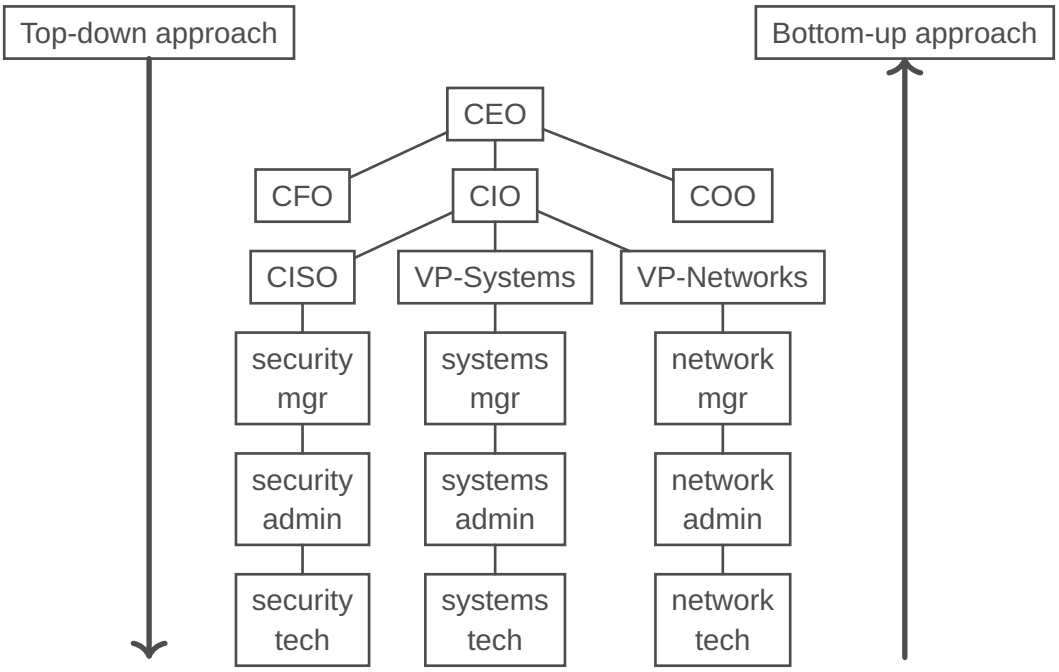
2.3. COMPONENTS OF AN INFORMATION SYSTEM (IS)

- Software
- Hardware
- Data
- People
- Procedures
- Networks

2.4. BALANCING SECURITY AND SYSTEM USABILITY

- Obtaining perfect information security is impossible.
- Security needs to protect the system without slowing people down.
- Too much security can lead to workarounds.
  - Example: If strong passwords are enforced, people might start writing them down on sticky notes.
- Too much convenience exposes the system to unnecessary risks.
- It’s all about finding that sweet spot between security and usability.
  - Example Solution: Employees must use multi-factor authentication. This way, they are free to use a less secure password without compromising the overall security.
- An even better, continuously review policies and involve users to find the best solution.

2.5. IMPLEMENTATION OF INFORMATION SECURITY



Bottom-Up	*Top-Down*
– Initiated by an organization's technical staff (system engineers, admins, etc.). – Implementations happen before policies are defined. – Often lacks support from management, budget and consistency. – Generally less effective and not scalable in large organizations.	– Initiated and supported by an organization's upper management. – Policies come first and provide guidance for implementations. – Ensures proper funding, authority and organization-wide enforcement. – Generally more effective and in-line with the business strategy

2.6. CIA TRIAD

The CIA triad is a foundational information-security model stating that systems should protect:
– **Confidentiality** - Keeping information secret
– **Integrity** - Keeping information correct and unaltered
– **Availability** - Ensuring information and systems remain accessible

	Confidentiality	*Integrity*	*Availability*
Goal	Prevent or minimize unauthorized access to information	Protecting the reliability and correctness of information.	Ensuring that subjects have timely and uninterrupted access to information.
Steps to ensure it	– Encryption – Access Control – Allow / enforce advanced authentication mechanisms	– Digital Signatures – Hashing and Checksums – Change Management	– Redundance and Backups – DDoS Protection – Incident Response

2.7. NON-REPUDIATION AND ACCOUNTABILITY

Example of security controls through which non-repudiation can be established: Digital certificates, session identifiers, transaction logs, etc.

2.7.1. Non-Repudiation

– Ensures that the subject of an activity or who caused an event cannot deny having performed an action or cannot deny that the event occurred.
– Non-Repudiation prevents a subject from claiming not to have sent a message, not to have performed an action, or not to have been the cause of an event.

2.7.2. Accountability

– Being responsible or obligated for actions and results.
– Non-Repudiation is an essential part of accountability. A suspect cannot be held accountable if they can repudiate the claim against them.

2.8. STRIDE MODEL

A structured model developed by Microsoft used in cybersecurity to identify and categorize threats to systems by looking at how they can be attacked.

Term	*Definition*
S(poofing)	Pretending to be someone else. (see Authenticity)
T(ampering)	Unauthorized data modification or altering. (see Integrity)
R(epudiation)	Denying actions without proof. (see Non-Repudiation)
I(nformation disclosure)	Exposing sensitive information. (see Confidentiality)
D(enial of service)	Making systems or services unavailable. (see Availability)
E(levation of privilege)	Gaining unauthorized rights or privileges.

2.9. MCCUMBER CUBE

Y-Axis: Security Goals (C.I.A. Triad)

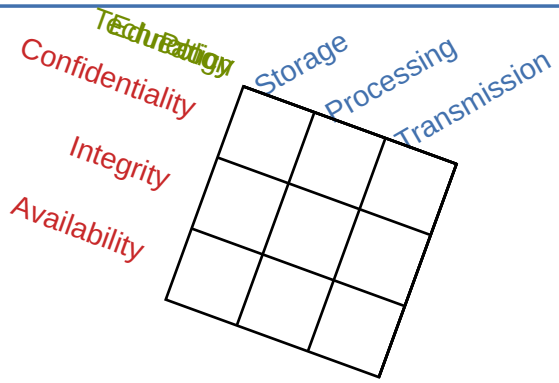
– Defines what needs to be protected.

X-Axis: Information States

– Describes where the information exists.

Z-Axis: Safeguards / Controls

– Defines how protection is implemented.



3. THREAT CATEGORIZATION

Term	*Definition*
Social Engineering	Manipulating people to reveal confidential information.
Software Attacks	Exploiting vulnerabilities in software to gain access to a system or steal data.
Denial of Service	Overloading one or multiple systems to make it unavailable.
Web Application Attacks	Exploiting vulnerabilities in websites or servers hosting websites.

Term	Definition
Password / Authentication Attacks	Attempting to bypass or compromise login systems to gain unauthorized access.
Physical Attacks	Bypassing technical controls by accessing physical infrastructure directly.

3.1. SOCIAL ENGINEERING

The psychological manipulation of individuals to trick them into revealing confidential information or performing actions that can compromise security.

Term	Definition
Phishing	Forged emails impersonating legitimate entities.
Spear Phishing	Targeted phishing against specific individuals.
Vishing	Voice-based phishing over phone or video calls.
Smishing	SMS / Text-based phishing.

3.2. SOFTWARE ATTACKS

Attacks involving malicious code or malware designed to damage systems, steal sensitive data, or gain unauthorized access to systems or services.

Term	Definition
Virus	Malware that attaches to programs and spreads.
Worms	Self-replicating malware that spreads over a network.
Trojan Horse	Malicious software disguised as legitimate applications.
Ransomware	Malware that encrypts victim’s data and demands payment to restore access.
Rootkits	Stealthy tools that hide malicious activity and maintain privileged access.

3.3. DENIAL OF SERVICE

Attacks aims at making a system or service unavailable by overwhelming it with excessive traffic or requests.

Term	Definition
DoS	Single source denial of service attacks.
DDoS	Denial-of-service attacks performed by multiple attackers or attacking devices.
Botnet	A network of compromised computers and other devices controlled by an attacker and used to together to flood a target with excessive traffic.
SYN-Flood Attack	Sending many connection requests without completing them.
Reflection Attack	Attacker sends requests to a service and spoofs the victim’s IP making the service send (many) replies to the victim instead of back to the attacker.

3.4. WEB APPLICATION ATTACKS

Exploits vulnerabilities in web applications to steal data, manipulate content, or gain unauthorized access.

Term	Definition
SQL Injection	An attacker inserts malicious SQL commands into an input to manipulate a database and access, modify, or delete data.
Cross-Site Scripting (XSS)	An attacker injects malicious scripts into a website that execute in other users' browsers to steal sensitive data.
Cross-Site Request Forgery (CSRF)	An attacker tricks a logged-in user's browser into sending unauthorized requests to a web application on their behalf.
Broken Authentication	Weak authentication mechanisms allow attackers to compromise passwords, sessions, or identities to gain unauthorized access.

3.5. PASSWORD / AUTHENTICATION ATTACKS

Attacks that attempt to bypass or compromise login systems to gain unauthorized access to a system or service.

Term	Definition
Rainbow Table Attacks	Attackers using precomputed hash lookup tables to reverse weakly hashed passwords back into plaintext.
Password Spraying	Attackers trying a few common password like "password" across many accounts to avoid lockouts or timeouts.
Credential Stuffing	Attackers using leaked usernames and passwords from previous breaches to attempt logins on other services.
Brute Force Attack	Attackers repeatedly try many username and password combinations until they successfully gain access to an account.

3.6. PHYSICAL THREATS

Threats or attacks that affect the physical infrastructure supporting information systems, usually bypassing technical controls overall.

Term	Definition
Theft of devices	Attackers physically steal hardware to gain direct access to stored data, credential, internal systems, or other sensitive data.
Hardware tampering	An attacker modifies or implants malicious components in physical equipment to intercept data, bypass security, or disrupt operations.
Power disruption	Attackers interrupt or manipulate power supply to shut down or destabilize critical systems or services impacting availability and business continuity.
Environmental damage	Natural or deliberate environmental events that damage infrastructure, causing data loss, downtime, or destruction of critical systems (e.g., earthquake, fire).

4. INFORMATION SECURITY MANAGEMENT

4.1. INFORMATION SECURITY GOVERNANCE

The system by which an organization directs and controls its information security strategy to ensure that it supports business objectives, manages risk appropriately, and complies with legal and other regulatory requirements.

Strategic Direction
– Defining security objectives aligned with business goals.

Leadership and Accountability
– Having clear roles and responsibilities for security decisions.

Risk Management
– Defining risks and ensuring they are identified and addressed appropriately.

Regulatory Compliance
– Ensuring adherence to laws and regulations (e.g. NIS2, HIPAA, CRA)

Responsibilities	*Functional Role Examples*
– Oversee overall corporate security posture (accountable to board) – Brief board, customers, public	– Chief Executive Officer
– Set security policy, procedures, program, training for company – Respond to security breaches (investigate, mitigate, litigate) – Responsible for independent annual audit coordination	– Chief Security Officer – Chief Information Officer – Chief Risk Officer – Department/Agency Head
– Implement/audit/enforce/assess compliance – Communicate policies, program (training)	– Mid-Level Manager
– Implement policy; report security vulnerabilities and breaches	– Enterprise Staff/Employees

4.2. INFORMATION SECURITY MANAGEMENT SYSTEM (ISMS)

A structured framework used to systematically manage and protect an organization's assets through various policies, processes and controls

Security governance defines **what** an organization wants to achieve. An ISMS defines **how** the organization wants to manage it.

Enterprise Information Security Policy (EISP)

– The information security policy that sets the strategic direction and scope for all an organization's security efforts.

Risk Management Process
– Definition of processes to identify assets, analyze threats and evaluate risk.

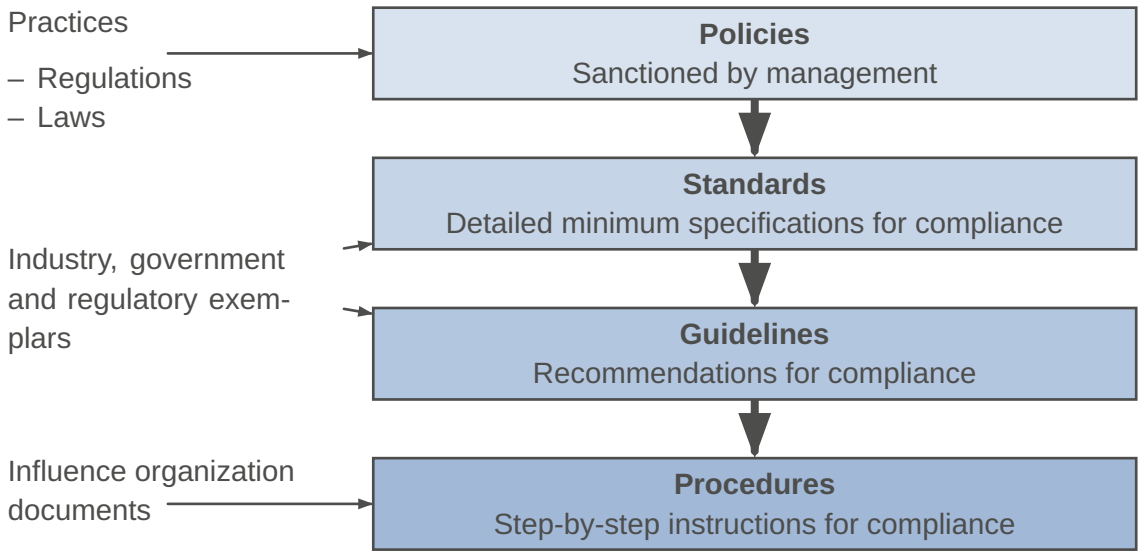
Security Awareness and Training
– Educational programs to ensure employees understand their security responsibilities.

Monitoring, Measurement and Audits
– Ongoing evaluation of control effectiveness and ISMS performance.

5. POLICY

A high-level, management-approved rule that defines mandatory organizational behavior and translates external laws and regulations into enforceable internal requirements.

Term	*Definition*
policy	instructions that dictate certain behavior within an organization.
guidelines	non-mandatory recommendations employees may use as a reference.
procedures	step-by-step instructions designed to assist employees in following policies.
practices	examples of actions that illustrate compliance with policies.
standard	a detailed statement of what must be done to comply with a policy.
de jure standard	a standard that has been formally evaluated and approved by a formal standards organization
de facto standard	a standard that is widely adopted or accepted by a public group.



What does a policy do? Establishes authority, accountability, and responsibilities for protecting information assets. Provides the foundation for standards, procedures and guidelines.

Who is responsible for policies? Policies are created and approved by senior management, ensuring organizational commitment. Management is responsible for enforcement while employees and users are responsible for compliance.

How is a policy enforced? By clearly communicating it to all relevant parties, integrating it into standards and procedures, monitoring compliance through audits and oversight, and applying defined disciplinary measures when violations occur.

Cyber Resilience Act (EU)
– Requires secure-by-design digital products and vulnerability management (starting December 2027).

Health Insurance Portability and Accountability Act (U.S.)
– Requires administrative, technical, and physical safeguards for protecting patient health data from disclosure.

NIS2 Directive (EU)
– Mandates cybersecurity risk management and incident reporting for critical and important entities.

Local Laws
– Many regions have their own data protection or breach notification laws in additional to national or EU regulations.

5.1. DESIGNING EFFECTIVE POLICIES

1) Development
   – Policies must align with organizational goals, business risks and legal requirements.
2) Distribution
   – Policies must be distributed to all affected entities in a timely manner.
3) Comprehension
   – Policies must be readable for, available to and read by all affected entities.
4) Compliance
   – Policies must be formally agreed to by act or affirmation.
5) Enforcement
   – Policies must be uniformly applied to all affected entities.
6) Review
   – Policies must be reviewed regularly in a changing environment.

5.2. ENTERPRISE INFORMATION SECURITY POLICY (EISP)

The high-level information security policy that sets the strategic direction, scope and tone for all an organization’s security efforts and policies.

– Guidance for the development, implementation and management of the security program.
– Sets the requirements that must be met by the information security blueprint.
– Defines the purpose, scope, constraints and applicability of the security program.
– Assigns responsibilities for the various areas of information security.
– Addresses legal compliance.

5.2.1. Elements of an EISP

Although the content of EISP documents varies among organizations, most EISP documents should include the following elements.

– Statement of Purpose
  – Statement of intent that defines the scope, objectives, and purpose of the enterprise information security policy and establishes its role as the foundation for all supporting security documents.
– Information Security Elements
  – Definition of information security that outlines the core principles and concepts, including confidentiality, integrity, and availability, guiding the organization’s security efforts.

– Need for Information Security
  – Definition of the importance of information security within an organization and its legal and ethical responsibility to protect information about customers, employees, and markets.
– Information Security Responsibilities and Roles
  – Description of the organizational structure that supports information security, including defined roles and responsibilities for management, employees, and users, as well as responsibility for maintaining the policy itself.

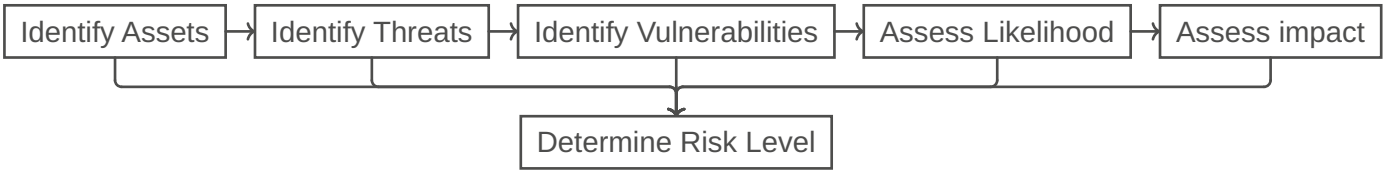
5.3. ISSUE-SPECIFIC SECURITY POLICY

An organizational policy that provides detailed, targeted guidance to instruct members of an organization in the use of a specific resource.

– Supports the EISP by translating it into an issue-specific guidance.
– Establishes rules for access, monitoring, and protection of the resource.
– Defines acceptable and unacceptable use of the specified technology or resource.
– Assigns responsibilities and accountability to users, administrators, and management.

6. RISK ANALYSIS

The process of identifying assets, threats, and vulnerabilities, and evaluating the likelihood and impact of potential adverse events to determine the level of risk.



6.1. IDENTIFYING ASSETS

Term	*Definition*
Asset	Any resource that has some kind of value to an organization and therefore requires protection.
Information Assets	Customer data, intellectual property, source code, etc.
Technical Assets	Services, applications, databases, networks, etc.
Physical Assets	Servers, devices, facilities, infrastructure, etc.
Human Assets	Employees, administrators, contractors, key personnel, etc.
Business Process Assets	Critical operational workflows

6.2. CLASSIFYING ASSETS

The process of assigning every asset to a class based on their value, sensitivity and impact if compromised

Term	*Definition*
Public	Information that can be shared without risk
Internal	Information for organization internal use only
Confidential	Sensitive information that could cause harm if disclosed

Term	Definition
Restricted	Highly sensitive, strictly limited and strongly protected information

6.3. IDENTIFYING THREATS

A potential event, actor, or action that could exploit a vulnerability and cause harm to an asset.

Examples: Power outage, insider threat, vishing attack

6.4. SECURITY CONTROLS

Measures to reduce risk by detecting, preventing, responding to, or mitigating threats to organizational assets.

6.4.1. Types

Term	Definition
Administrative / Management Controls	Policies, procedures, security training, security governance, etc.
Technical / Logical Controls	Firewalls, encryption, access control systems, system hardening, etc.
Physical Controls	Physical locks, surveillance cameras, secure access badges, turnstiles, etc.

6.4.2. By Function

Term	Definition
Preventive Controls	Stop incidents before they occur. e.g., Firewalls, access control, encryption, etc.
Detective Controls	Identify incidents when they occur. e.g., Intrusion detection, log monitoring, SIEM, CCTV, etc.
Corrective Controls	Limit damage and restore systems after an incident. e.g., Backups, system restore, incident response, etc.
Deterrent Controls	Discourage malicious behavior. e.g., Warning banners, monitoring notices, disciplinary policies, etc.
Compensating Controls	Reduce risk when a primary control cannot be implemented. e.g., Network isolation, layered security, alternative safeguards, etc.

6.5. BUSINESS CONTINUITY MANAGEMENT

Ensures that critical business functions can continue during and after incidents or disruptions such as cyberattacks, system failures, or physical incidents.

Even with strong security controls in place, incidents can and will still occur at some point. BCM prepares the organization to operate and recover during these times.

6.5.1. Key Objectives

– Maintain critical operations during incidents.
  e.g., backups, redundant services, manual processing, etc.
– Minimize downtime and financial impact.
  e.g., fast system restore, emergency support contracts, incident response team, etc.

– Protect people, assets and reputation
  e.g., evacuation plans, fire suppression systems, customer notification processes, etc.
– Enable fast and structured recovery
  e.g., disaster recovery playbooks, tested backup restoration, post-incident review processes, etc.

6.6. SECURITY AND AWARENESS TRAINING

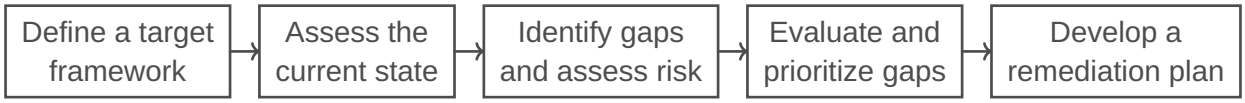
A coordinated program designed to ensure that all members of an organization understand their security responsibilities and have the knowledge and skills to protect information assets.

	Awareness (Level 1)	***Training (Level 2)***	***Education (Level 3)***
Objective	Seeks to teach members of an organization **what** security is and what to do in certain situations	Seeks to train members of an organization **how** they should react and respond to certain situations	Seeks to educate members of an organization as to **why** the organization reacts the way it does
Complexity Level	Offers **basic information** about threats and responses	Offers more **detailed knowledge** about detecting threats and teaches skills needed for effective reaction	Offers the background and **depth of knowledge** to gain insight into how processes are developed and enables ongoing improvement
Teaching Method	– Videos – Newsletters – Posters – Informal Training	– Informal Training – Workshops – Hands-on Practice	– Theoretical Instruction – Discussions / Seminars – Background Reading
Impact timeframe	Short-term	Intermediate	Long-term

6.7. GAP ANALYSIS

The process of comparing an organization's current security posture with a required or desired target to identify missing or insufficient controls.

– Risk Analysis: What could go wrong?
– Gap Analysis: Where are we non-compliant or under-protected?



6.8. SECURITY FRAMEWORK

A structured set of principles, processes, and controls that organizations use to manage risks and protect their information systems, assets, and operations.

Framework	***Definition***
ISO/IEC 27000	Global standard for information security management systems (ISMS).
NIST Cybersecurity Framework	Practical framework for managing cyber risk
CIS Controls	Defines 18 highly practical technical security controls

Framework	*Definition*
ISACA CORBIT	An IT governance and risk management framework.

6.9. ISO/IEC 2700

A set of standards for ISMS, helping organizations systematically protect information assets using a risk-based approach.

Standard	*Definition*
ISO/IEC 27000	Introduction, terminology, and key concepts (e.g., risk, asset, control, etc.)
ISO/IEC 27001	Defines requirements to establish, implement, maintain, and improve an ISMS.
ISO/IEC 27002	Practical guidance for implementing controls.
ISO/IEC 27005	Focuses on risk management methodology.
ISO/IEC 27017	Additional guidance for cloud services.
ISO/IEC 27018	Focuses on privacy and personal data protection in cloud environments.

6.10. NIST CYBERSECURITY FRAMEWORK

A risk-based guideline that helps organizations to structure, manage, and improve their cybersecurity activities across the full lifecycle of prevention, detection, and response.

– It's organized into five core functions; Identify, Protect, Detect, Respond, Recover.
– Provides categories and subcategories of cybersecurity outcomes and controls.
– Includes implementation tiers to assess cybersecurity maturity.
– Is very flexible and adaptable to any organization or business sector.
– Not certifiable, primarily used as guidance and best practice.

6.11. RISK, VULNERABILITIES AND THREATS

6.11.1. Vulnerabilities

A potential weakness in an asset or its defensive control system. Can be **known** or **unknown**.

Examples:
– Software vulnerabilities (Bugs, design flaws, …).
– Human vulnerabilities (sharing passwords, …).

6.11.2. Threat

A potential malicious action, or event that aims to damage, or steal unauthorized access to assets. A threat exploits system vulnerabilities.

– Threat: What?
– Threat Actor: Who?
– Threat Vector: How?

6.11.3. Threat Actors

Motivations	*Actor Types*

– Service disruptions
– Data exfiltration
– Disinformation
– Chaotic / Vandalism
– Financial
  – Blackmailing
  – Fraud
– Political
	– Hackers
	– Unskilled Attackers & Script Kiddies
	– Hacker Teams & Hacktivists
	– Nation-State Actors
	– Organized Crime
	– Internal Threat Actors

6.11.4. Threat Vector

The path, method, or delivery mechanism that a threat uses to reach an asset and exploit a vulnerability.

Types:
– Software Vectors (Bugs, Virus, …)
– Network Vectors (Bluetooth, Open Ports, Remote Network, …)
– Lure-Based Vectors (Drop Attacks with USB Sticks, Trojans, …)
– Message-Based Vectors (SMS, Email, IM, Web and Social Media, …)
– Supply Chain Vectors (Updates, Libraries, …)

6.11.5. Attack surface

The sum of vulnerabilities, pathways, or methods (Threat vectors) that hackers can use to gain unauthorized access to the network or sensitive data, or to carry out a cyberattack.

6.11.6. Risk

The probability of an unwanted occurrence, such as an undesirable event or loss.

– The definition of risk implies threats and vulnerabilities: A risk is only here if we have an existing vulnerability, threat, and threat vector!
– Risk = Vulnerability (Value & Exposure) + Threat (Threat Actor & Threat vector)

At what cost are we willing to accept what risk? The answer to that question gives us risk management.

6.12. RISK MANAGEMENT

The process of identifying, assessing, prioritizing and mitigating threats to an asset from an organisation.

Risk management framework

Structure of the strategic planning and design of the entirety of the risk management efforts (planning).

Risk management process

Implementation, analysis, evaluation of the risk management framework (doing).

6.12.1. RM Framework

1) ***Executive Governance & Support***: Support from management and users.
2) ***Framework Design***: Defining the methods and risk appetite strategy.
3) ***Framework Implementation***: Rollout of the plan (through →RM process).
4) ***Monitoring & Review***: How effective is the entire system?
5) ***Continuous Improvement***: Continuous adaption to new , or existing threats.

6.12.1.1. Framework Design

Defining the methods and risk appetite strategy

Term	*Definition*
Risk appetite (strategic)	The quantity of risk that organizations are willing to accept, to achieve their goals.
Risk tolerance (specific)	The acceptable risk organizations are willing to accept for a specific asset.
Residual risk	The risk that still remains after all controls have been applied.

6.12.1.2. Framework Implementation

Framework Implementation starts after:

- The RM framework and process is finished designing.
- The structure of the RM process & framework is defined.

The methodologies are dependent on the risk appetite:

- Direct rollout
- Pilot-test
- Phased approach

6.12.1.3. Monitoring & Review

- How successful was the framework in the last cycle?
  - Designing
  - Implementing
- What issues require adjustments to the plan?

6.12.2. Risk Management Process

Risk assessment: The identification, analysis, and evaluation of risk as initial parts of risk management.

Risk treatment & Risk Owner: The application of safeguards or controls to reduce the risks to an organization's information assets to an acceptable level.

1) **Risk identification**: Where and what is the risk?
2) **Risk analysis**: How severe is the current level of risk?
3) **Risk evaluation**: Is the current level of risk acceptable?
4) **Risk treatment**: What do I need to do to bring the risk to an acceptable level?

6.12.2.1. Risk Identification

The recognition, enumeration, and documentation of risks to an organization's information assets.

Where and what is the risk?

- What are the assets of the organisation? (Internal Asset Register, Weighted Asset Table)
  - Data, Software, Hardware, Networks, Employees, procedures, …
- What are the threats of the organisation? (ATT&CK is a globally-accessible knowledge base).
  - Human error, Attacks from hackers, forces of nature, day zero attacks …
- What are the vulnerabilities? (CVE and CVSS helps with that question).
  - Lack of training, known bugs in the system, day zero exploit
- Precision is key: If THIS then THAT, because OF …

6.12.3. CVE & CVSS

- A Common Vulnerabilities and Exposures (CVE) is an industry-wide standard identification number for vulnerabilities.

– The Common Vulnerability Scoring System (CVSS) uses the CIA triad principles within the metrics used to calculate the CVVS base score and assigns severity scores to a vulnerability

6.12.4. Risk Analysis

A determination of the extent to which an organization's information assets are exposed to risk.

Identify the severity of every identified threat and vulnerability.

– What is the probability of an attack?
– What would be the impact of an attack?
  – Quantitative risk analysis assigns real dollar figures to the loss of an asset.
  – Qualitative risk analysis assigns subjective and intangible values to the loss of an asset.
– Existing Security Controls shall be considered

6.12.5. Quantitative Risk Analysis

1) Assign Asset Value (AV)
2) Calculate Exposure Factor (EF)
3) Calculate single loss expectancy (SLE)
4) Assess the annualized rate of occurrence (ARO)
5) Derive the annualized loss expectancy (ALE)
6) Perform cost/benfit analysis of countermeasures

6.12.5.1. AV

1) Identify the organization's information assets.
2) Classify them.
3) Categorize them into useful groups.
4) Prioritize them by overall importance.

Example 1: **Weighted Asset Table**

		Impact on Revenue	*Impact on Profitability*	*Impact on Reputation*		
#		**0.3**	**0.4**	**0.3**	**TOTAL (1.0)**	**Importance**
1	Customer order via SSL	5	5	5	5	Critically Important
2	Customer service request via e-mail	3	3	5	3.6	Important
⋮	⋮	⋮	⋮	⋮	⋮	⋮

6.12.5.2. EF

Exposure factor (EF): Represents the percentage of loss that an organization would experience if a specific asset is violated by a realized risk.

– In most cases, a realized risk does not result in the total loss of an asset. The EF simply indicates the expect.

6.12.5.3. SLE

Single loss expectancy (SLE): The cost associated with a single realized risk against a specific asset. It indicates the exact amount of loss an organization would experience if an asset were harmed by a specific threat occurring.

– SLE = asset value (AV) × exposure factor (EF)
– Example: if an asset is valued at $200,000 and it has an EF of 45 % for a specific threat, then the SLE of the threat for that asset is $90,000.

6.12.5.4. ARO

Annualized rate of occurrence (ARO): The expected frequency with which a specific threat or risk will occur within a single year.
– Example: The ARO of an earthquake in Paris may be .00001, whereas the ARO of an earthquake in San Francisco may be .03 (for a 6.7+ magnitude).

6.12.5.5. ALE

Annualized loss expectancy (ALE): The possible yearly cost of all instances of a specific realized threat against a specific asset.
– ALE = single loss expectancy (SLE) × annualized rate of occurrence (ARO)
– If the SLE of an asset is $90,000 and the ARO for a specific threat (such as total power loss) is .5, then the ALE is $45,000.
On the other hand, if the ARO for a specific threat (such as compromised user account) is 15, then the ALE would be $1,350,00

6.12.5.6. ALE with Safeguards

– You must calculate the ALE for the asset if the safeguard is implemented.
  – This requires a new EF and ARO specific to the safeguard.
  – The whole point of a safeguard is to reduce the ARO and/or reduce the SLE. The best of all possible safeguards would reduce the ARO to zero.
  – In most cases, the EF to an asset remains the same even with an applied safeguard because if the safeguard fails, the loss on the asset is usually the same as when there is no safeguard.
– Safeguard Costs
  – You must first compile a list of safeguards for each threat. Then you assign each safeguard a deployment value = ACS (Annual cost of the safeguard).

6.12.5.7. Value of a Safeguard

Net Value or Cost/Benefit of a safeguard:

– Negative value: not a responsible choice.
– Positive value: Then the value represents the yearly savings in cost that you CAN have (because the rate of occurrence is just an expected value).

Safety needs to be cost effective. Do not use more resources or money for the protection of an asset as the value of the asset itself!

6.12.6. Risk Evaluation

The process of comparing an information asset's risk rating to the numerical representation of the organization's risk appetite or risk threshold to determine if risk treatment is required.

Risk Evaluation: Compare the risk with the risk appetite of the organization.

– Can the company live with the analysed level of risk (From the CVSS, the quantitative risk analysis, qualitative risk analysis)?
– Levels: Expansionary, Conservative or Neutral

The Risk appetite from the RM Framework must be translated into a value so it can be compared to each analysed risk.

– For the quantitative risk analysis, the risk appetite can be translated into a numerical value!

Goal: The risk must be smaller or equal as the risk appetite.

- Important Indicators for Business Impact:
  - Maximum Tolerable Downtime (MTD)
  - Recovery Point Objective (RPO)
  - Recovery Time Objective (RTO) & Work Recovery Time (WRT)

6.12.7. Risk Treatment

Mitigation risk treatment strategy: The risk treatment strategy that attempts to eliminate or reduce any remaining uncontrolled risk through the application of additional controls and safeguards in an effort to change the likelihood of a successful attack on an information asset; also known as the defense strategy.

The company now has a list of information assets with unacceptable levels of risk.

- The appropriate strategy must be selected and applied.

Four basic strategies to treat risk:

1) Mitigation: Apply safeguards that eliminate or reduce the remaining uncontrolled risk.
   - Example: Firewall, Training, …
2) Transfer: Shift risks to other areas or outside entities.
   - Example: Outsourcing
3) Acceptance: Understand the consequences of choosing to leave an information assets vulnerability facing the current risk level (after formal evaluation).
4) Termination: Remove or discontinue the asset from the organization's operating environment.

6.12.7.1. Mitigation

- Fix vulnerabilities
- Applying controls (tools, processes, rules to mitigate risk)
  - Endpoint Hardening (preventive Control): Secure a "endpoint" (device: laptop, server, …) by reducing its vulnerabilities and shut down potential threat vectors!
- Reduce final impact (If zero-day attacks, unknown vulnerabilities, or a taken risk happen)
  - EDR (Endpoint Detection and Response): Software that watches for suspicious behaviour and responds with certain measures.
  - XDR (Extended Detection and Response): Watching everywhere (not just on endpoints) and respond with certain measures (shut down infected laptop, …)

6.12.8. Other RM Frameworks

- OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) by the Carnegie Mellon University.
- FAIR (Factor Analysis of Information Risk) by Jack A. Jones.
- ISO Standards: ISO 27005 and ISO 31000: (explanation: https://en.wikipedia.org/wiki/ISO/IEC_27005).
- NIST Risk Management Framework (RMF): https://csrc.nist.gov/publications/sp

7. IDENTITY & ACCESS MANAGEMENT (IAM)

IAM deals with provisioning and protecting digital identities and user access permissions. Or in other words: The right people can access the right resources for the right reasons at the right time. To ensure this we need Access Controls.

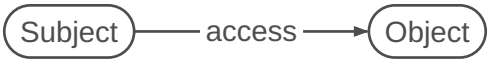
7.1. ACCESS CONTROL

Any hardware, software, or administrative policy or procedure that controls access to resources. The selective method by which systems specify who may use a particular resource and how they may use it.

The goal is to:

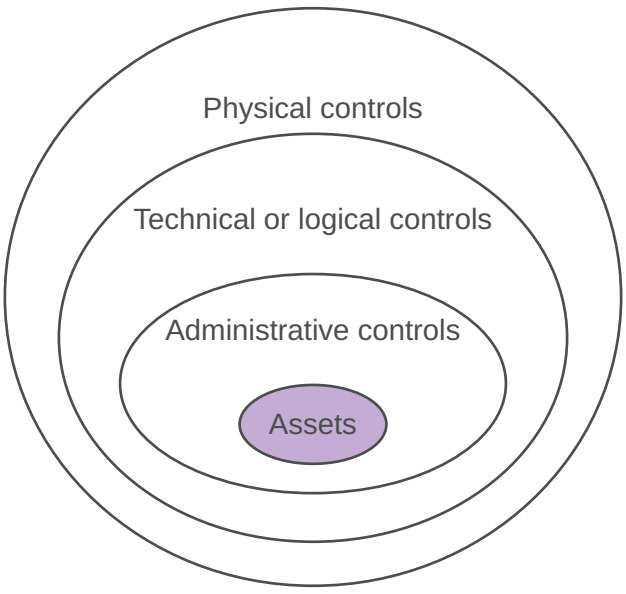
– PROVIDE access to authorized subjects
– PREVENT access to unauthorized access attempts and unauthorized subjects

Term	*Definition*
Subject	Active entity that accesses a passive object. – Anything that can access a resource can be a subject: users, programs, processes, services, computers,…
Object	Passive entity that provides information to subjects – Anything that can provide resources: files, databases, computers, programs, processes, services, printers, …

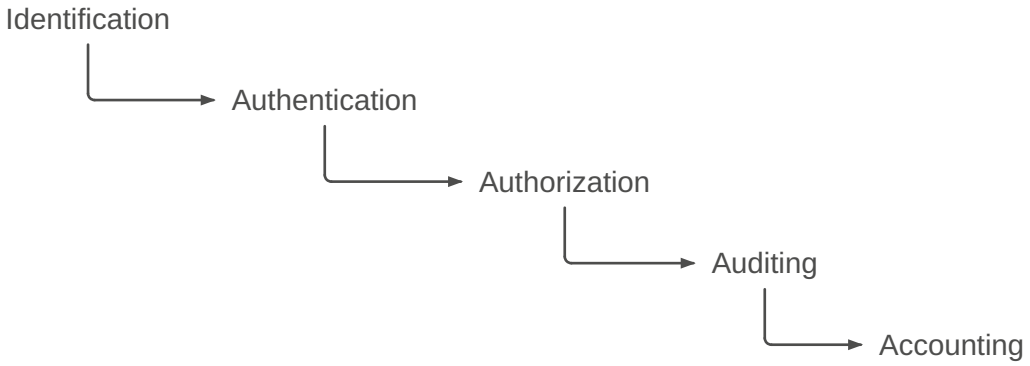


7.1.1. Control methods

Term	*Definition*
Physical controls	Items that you can physically touch. Included are physical mechanisms deployed to prevent, monitor, or detect direct contact with systems or areas within a facility – Examples: guards, fences, motion detectors, locked doors, sealed windows, lights, cable protection, laptop locks, badges, swipe cards, guard dogs, video cameras, mantraps, and alarms
Technical or logical controls	Hardware or software mechanisms used to manage access and to provide protection for resources and systems – Examples: authentication methods (username, passwords, biometrics,…), encryption, access control lists, protocols,…
Administrative controls	Policies and procedures defined by an organization's security policy or other regulations or requirements – Examples: policies, procedures, hiring practices, background checks, data classifications and labeling, security training,…



7.2. MECHANISMS



7.2.1. Identification

The subject is claiming an identity.

– Example: Typing a username, swiping a smartcard, waving a token device, speaking a phrase, or positioning your face, hand, or finger in front of a camera or in proximity to a scanning device

Important: All subjects must have unique identities

– IT systems track activity by identities, not by the subjects themselves
– A subject's identity is typically labeled as, or considered to be, public information

A subject must provide an identity to a system to start the other processes (authentication, authorization, and accountability)

7.2.2. Authentication

The process of verifying that the claimed identity (from identification) is valid

– Example: password
– Identification and authentication are often used together as a single two-step process

Authentication information used to verify identity is private information and needs to be protected

To authenticate the claimed identity it is common to use multiple factors These factors are often categorized in three different categories:

1) Type 1
   – **Something you know**. Passwords, PINs, …
2) Type 2
   – **Something you have**. Physical devices that a user possesses can help them provide authentication
3) Type 3
   – **Something you are or something you do**. It is a physical characteristic of a person identified with different types of biometrics

7.2.2.1. Authentication Schemes

1) Basic Authentication: Classical username / password pair transmitted in the clear
2) One Time Passwords: Transmitted in the clear but used only once
3) Challenge / Response: Response is a function of password and one-time challenge
4) Anonymous Key Exchange: Exchange credentials over unauthenticated secure channel
5) Zero-Knowledge Password Proofs: Does not permit offline-based password attacks
6) Server Certificates plus User Authentication: Transmit user password over unilaterally authenticated secure channel
7) Mutual Public Key Authentication: Bilateral use of public key signatures

Attack vulnerability Matrix

Attack	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Passive Password Sniffing	✗						
Offline Brute Force Password Attack	✗		✗	✗			
Active Man-in-the-Middle Attack	✗	✗	✗	✗			
Identity Theft on Server	✗	✗	✗	✗	✗	✗	
CA Compromise						✗	✗

7.2.2.2. Type 1 Factor: Passwords

Passwords are typically static. They are the weakest form of authentication

- Users often choose passwords that are easy to remember and therefore easy to guess or crack
- Randomly generated passwords are hard to remember, and many users write them down
- Users often share their passwords, or forget them
- Passwords are rarely stored in plaintext.
  - A system will create a hash of a password using a hashing algorithm
- Best practices and policies
  - Enforce a minimum length
  - Complexity rules (uppercase/lowercase, non-alphanumeric, etc…)
  - Ageing and expiration
  - Reuse and history
- Password managers mitigate the risk of poor credential management

7.2.2.3. Type 2 Factor: Tokens

A token device, or hardware token, is a device that users can carry with them

- An authentication server stores the details of the token, so at any moment, the server knows what number is displayed on the user's token

Hard Authentication Tokens
- No transmission of the token itself e.g. Smartcards, Hardware OTP Token

Soft Authentication Tokens
- Software token transmitted to the user e.g. via Authenticator App, SMS, Email or phone

Dynamic Password Tokens
- Synchronous dynamic passwords are time-based and synchronized with an authentication server (TOTP)
- Asynchronous dynamic password is based on a Challenge-Response principle. Passwords are generated based on an algorithm and an incrementing counter, which remains valid until used (HOTP)

7.2.2.4. Type 2 Factor: Smartcard

A smartcard is a credit card–sized ID or badge and has an integrated circuit chip embedded in it

- Smartcards store information about the authorized user that is used for identification and/or authentication purposes
- Implements certificate-based authentication (private key and sometimes a PIN to activate the card)
- Most current smartcards include a microprocessor and one or more certificates. The certificates are used for asymmetric cryptography such as encrypting data or digitally signing email
- Smartcards are tamper-resistant and provide users with an easy way to carry and use complex encryption keys

7.2.2.5. Type 2 Factor: One-Time Passwords

Onetime passwords are dynamic passwords that change every time they are used

- Onetime password generators are token devices that create passwords
- The PIN can be provided via a software application running on the user's device (e.g., smartphone)

TOTP (Time-based One-Time Password)
- Uses a timestamp and remains valid for a certain timeframe, such as 30 seconds
- This is similar to the synchronous dynamic passwords used by tokens

HOTP (HMAC-based One-Time Password)
- Includes a hash function to create onetime passwords. It creates HOTP values of six to eight numbers
- This is similar to the asynchronous dynamic passwords created by tokens. The HOTP value remains valid until used

7.2.2.6. Type 3 Factor: Biometrics

Biometric authentication uses physiological characteristics to provide authentication for a provided identification.

Biometrics make measurements and compare them with unique points of reference. This may lead to these errors:

- False reject rate (FRR) (Type 1 Error): percentage of authorized users who are denied access
- False accept rate (FAR) (Type 2 Error): percentage of unauthorized users who are granted access
- Crossover error rate (CER): The point at which the rate of false rejections equals the rate of false acceptances

7.2.2.7. Multifactor Authentication

Multifactor authentication is any authentication using two or more factors

- For a positive authentication, elements from at least two, and preferably three factors should be verified
  - When two authentication methods of the same factor are used together, the strength of the authentication is no greater than it would be if just one method were used
  - Using more types or factors results in more secure authentication

7.2.2.8. Authentication factors

Numbered from weak to strong

1) Type 1: Something you know
2) Type 2: Something you have
3) Type 3: Something you are/ you do
4) Multi-Factor: 2 types
5) Multi-Factor: 3 types

7.2.2.9. Secondary Authentication Factors

In addition to the three primary authentication factors, there are some others

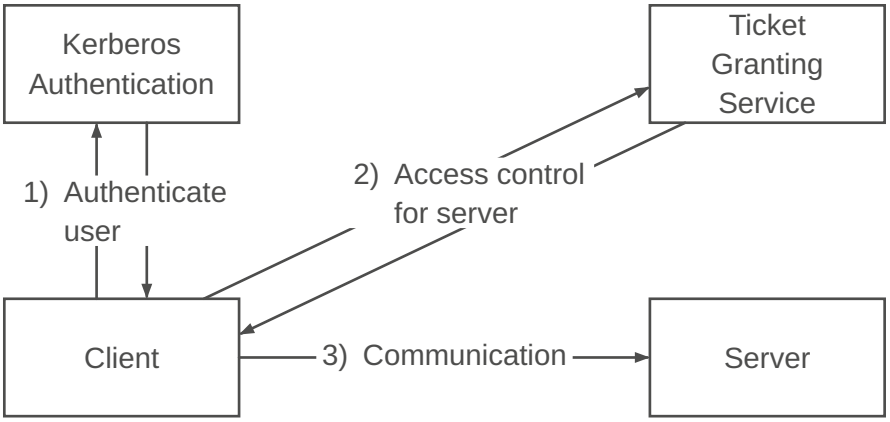
- Somewhere You Are
  - The somewhere-you-are factor identifies a subject's location based on a specific computer, a geographic location identified by an Internet Protocol (IP) address, or a phone number identified by caller ID
- Somewhere You Aren't
  - Many IAM systems use geolocation technologies to identify suspicious activity

– For example, imagine that a user typically logs on with an IP address in Switzerland. If a user is trying to log on from a location in India, it can block the access even if the user has the correct username and password

7.2.2.10. Authentication Frameworks

– Kerberos: Create Authentication through a trusted third party.
– RADIUS: Provide centralized authentication, authorization, and accounting (AAA) for network access.

7.2.2.10.1. Kerberos



An authentication system that uses symmetric key encryption to validate an individual user's access to various network resources by keeping a database containing the private keys of clients and servers that are in the authentication domain it supervises.

– Authentication in UNIX-based TCP/IP networks
– Use of symmetrical cryptography (DES)
– Relies on the mediation services of a trusted referee or notary
– Based on the work by Needham and Schroeder on trusted third-party protocols as well as Denning and Sacco's modifications of these
– Current release is Kerberos v5 (RFC 1510 , September 1993)
– V5 supports additional encryption ciphers besides DES

Term	*Definition*
Principal	A Kerberos participant
Principal's Master Key (MKey_p)	A long-term secret shared between the principal (user, service, or host) and the Key Distribution Center (KDC). This key is typically derived from the principal's password and is used to encrypt and decrypt authentication tickets.
Kerberos Ticket	Temporary credential that allows a user to access specific network services
Authentication Server (AS)	Verifies who the client is, gives TGT
Ticket Granting Server (TGS)	Grants access to specific services, gives ST
Ticket Granting Ticket (TGT)	Given by the AS
Service Ticket (ST)	Given by the TGS
Key Distribution Center (KDC)	A server that verifies and manages authentication credentials and distributes session keys to users and services within a network

Kerberos Step-By-Step
1) The user wants to get authenticated at a Service.

2) The user sends a request to the Authentication Server (KDC) asking for a Ticket Granting Ticket (TGT). This request is encrypted with the hash of the user's password.
3) The Authentication Server looks up the user, authenticates him using the hashed password and sends back the TGT. (notice, the password itself never travels across the network)
4) The user wants to access a specific service. He sends the TGT to the Ticket Granting Server (TGS)
5) The TGS verifies the TGT and issues a Service Ticket to the client
6) The client presents this Service Ticket directly to the Service he wants to use.
7) The Service decrypts the ticket, verifies the client and grants access. He can also send a message back to the client to prove its own identity.

7.2.2.10.2. Remote Authentication Dial-In User Service (RADIUS)

A networking protocol that provides centralized Authentication, Authorization and Accounting (AAA) management for users who use a network service.

Used to secure network nodes: Enterprice Wi-Fi (802.1x), VPNs, Switches

AAA
– Authentication: Verifying the user's identity
– Authorization: Granting specific network privileges (assigning specific IP, …)
– Accounting: Tracking network resource for auditing, billing, …

RADIUS Architecture
1) User requests network access from the Network Access Server (NAS)
2) NAS prompts the RADIUS server for credentials (username / password, or certificate)
3) RADIUS server evaluates the request and returns one of three responses:
   – Access-Accept: User is authenticated, NAS grants network access
   – Access-Reject: Invalid credentials, NAS denies access
   – Access-Challenge: Server requires more information (MFA, or Token)
4) When connected, NAS sends Accounting-Request to log the session.

7.2.2.10.3. RADIUS vs Kerberos Vulnerability Matrix

Attack	*Kerberos*	*RADIUS*
Passive Password Sniffing		
Offline Brute Force Password Attack	✗	✗
Active Man-in-the-Middle Attack		✗
Identity Theft on Server		
CA Compromise		✗

7.2.3. Authorization

The process of authorization ensures that the requested activity or access to an object is possible given the rights and privileges assigned to the authenticated identity

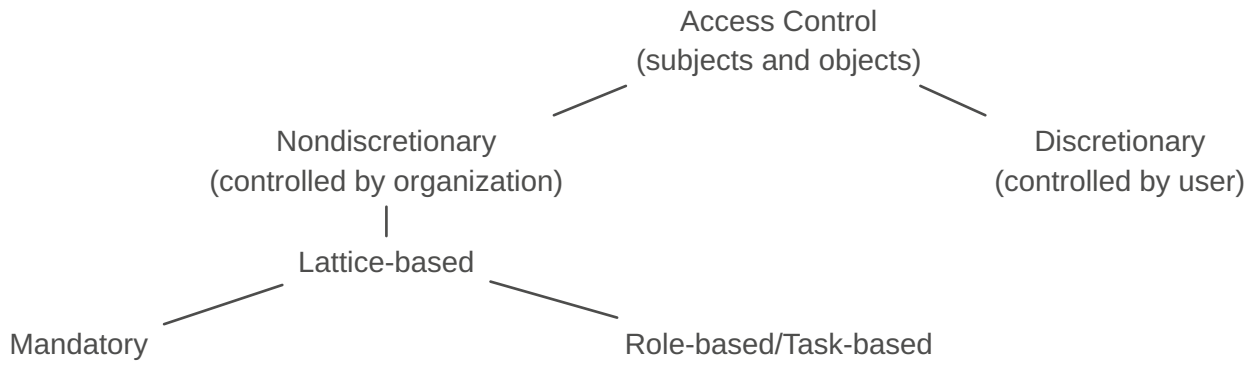
Or in other words: Once a subject is authenticated, access must be authorized

– Just because a subject has been identified and authenticated does not mean they have been authorized to perform any function or access all resources within the controlled environment

Identification and authentication are all-or-nothing aspects of access control. This is NOT the case with authorization:

– Authorization has a wide range of variations between all or nothing for each object within the environment

7.2.3.1. DAC and NDAC



7.2.3.1.1. Discretionary access control (DAC)

Access controls that are implemented at the judgment or option of the data owner. Every object has an owner, and the owner can grant or deny access to any other subjects → The owner (or user) chooses who has access!

- Most flexible and widely used e.g. file system security
- Data owner can modify access control list (ACL)
- Example: User has a hard drive and wants to share it with coworkers. He decides who he shares it with.

7.2.3.1.2. Nondiscretionary access control (NDAC)

Access controls that are implemented by a central authority.

- Example: US-Hospital where access is based on rules and regulations like HIPAA (DSG covers that in Switzerland)

7.2.3.1.2.1. Lattice-based access control (LBAC)

A variation on mandatory access controls that assigns users a matrix of authorizations for particular areas of access, incorporating the information assets of subjects such as users and objects.

- Mandatory access control (MAC):

  Use of labels applied to both subjects and objects. This means each collection of information is rated, and all users are rated to specify the level of access.
  - Example: Information are labelled as top secret → only users that are labelled top secret are granted access to this information!
- Role-based (RBAC) / Task-based (TBAC) access control:

  privileges are tied to a role or a job (role-based) or to a task or assignment (task-based).
  - Example: Project manager has access to corresponding information about his project. (role-based)
  - Example: A technician is only allowed into a server room in his planned maintenance timeslot (task-based)

7.2.3.2. Least privilege design principle

Access rights should be limited in scope, time, and function

- “Just enough access” is usually better than broad permanent access

Users and systems should only get the permissions they actually need

- Reduces attack surface and limits damage after account compromise
- Helps prevent misuse of admin accounts and service accounts
- Supports separation of duties and stronger compliance

Requires regular access reviews and removal of unused permissions

7.2.4. Auditing

A subject's actions are tracked and recorded

Purpose: Hold the subjects accountable for their actions while authenticated on a system

7.2.5. Accounting

The consumption of resources by a subject is measured, metered, and collected.

Purpose: Provide a record of resource usage for billing, capacity planning, and trend analysis.

7.3. ESTABLISHING ACCOUNTABILITY AND NON-REPUDIATION

Accountability means actions can be traced to a specific identity

– Proving to regulators that your data is secure
– Link a human to the activities of an identity
– Requires unique user identities, no shared accounts and trong authentication
– Support your security decisions and their implementation
– Supports incident investigation, compliance, and trust in transactions

Non-repudiation means a user cannot credibly deny a performed action

– Logging and audit trails must be complete, accurate, and protected
– Digital signatures are a key mechanism for non-repudiation

7.4. COMMON ACCESS CONTROL ATTACKS

7.4.1. Access Aggregation Attacks (passive attack)

– Access aggregation refers to collecting multiple pieces of nonsensitive information and aggregating them to learn sensitive information.
– Reconnaissance attacks are access aggregation attacks that combine multiple tools to identify multiple elements of a system, such as Internet Protocol (IP) addresses, open ports, running services, operating systems.

7.4.2. Password Attacks (brute-force attack)

– Online: Attacks against online accounts
– Offline: to steal an account database and then crack the passwords.

7.4.3. Dictionary Attack (brute-force attack)

An attempt to discover passwords by using every possible password in a predefined database or list of common or expected passwords also called a password-cracking dictionaries

– Dictionary attack databases also include character combinations commonly used as passwords, but not found in dictionaries
– Dictionary attacks often scan for one-upped-constructed passwords. A one-upped-constructed password is a previously used password, but with one character different.
– For example, password1 is one-upped from password, as are Password, 1password, and passXword

7.4.4. Birthday Attack (brute-force attack)

A birthday attack focuses on finding collisions. Its name comes from a statistical phenomenon known as the birthday paradox

– The birthday paradox states that if there are 23 people in a room, there is a 50 percent chance that any two of them will have the same birthday. (This is not the same year, but instead the same month and day, such as March 30)

– With February 29 in a leap year, there are only 366 possible days in a year. With 367 people in a room, you have a 99.99 percent chance of getting at least two people with the same birthdays. Reduce this to only 23 people in the room, and you still have a 50 percent chance that any two have the same birthday

You can reduce the success of birthday attacks by using hashing algorithms with enough bits to make collisions computationally infeasible, and by using salts.
– MD5 is not collision free
– SHA-3 (short for Secure Hash Algorithm version 3) can use as many as 512 bits and is considered safe against birthday attacks and collisions – at least for now

7.4.5. Rainbow Table Attacks

A rainbow table reduces the time by using large databases of precomputed hashes

– It takes a long time to find a password by guessing it, hashing it, and then comparing it with a valid password hash

A password cracker can then compare every hash in the rainbow table against the hash in a stolen password database file

– When using the rainbow table, the password cracker doesn't spend any time guessing and calculating hashes. It simply compares the hashes until it finds a match
– This can significantly reduce the time it takes to crack a password

Salting
– adds a unique random value to each password before hashing
– prevents identical passwords from producing identical hash values

7.4.6. Sniffer Attacks

A sniffer (also called a packet analyzer or protocol analyzer) is a software application that captures traffic traveling over the network

– A sniffer attack (also called eavesdropping attack) occurs when an attacker uses a sniffer to capture information transmitted over a network

The following techniques can prevent successful sniffing attacks:
– Encrypt all sensitive data (including passwords) sent over a network. Attackers cannot easily read encrypted data with a sniffer
– Use onetime passwords (OTP) when encryption is not possible or feasible. OTPs prevent the success of sniffing attacks, because they are used only once, also see next chapter Kerberos
– Protect network devices with physical security. Controlling physical access to routers and switches prevents attackers from installing sniffers on these devices

8. CRYPTOGRAPHY

Term	*Definition*
Plaintext/Cleartext	Before a message is put into a coded form, it is known as a plaintext or cleartext
Ciphertext/Cryptogram	The sender of a message uses a cryptographic algorithm to encrypt the plaintext and produce a ciphertext or cryptogram
Cipher	Encryption algorithm An algorithm is a set of rules, usually mathematical, that dictates how enciphering and deciphering processes are to take place
Key/Cryptovariable	A key is nothing more than a number (usually a very large binary number)

Term	Definition
Key space	– Every algorithm has a specific key space. A key space is defined by its bit size – The key space is the range of numbers from 0 to 2^n, where n is the bit size of the key – A 128-bit key can have a value from 0 to 2^{128}
One-Way Functions	A one-way function is a mathematical operation that easily produces output values for each possible combination of inputs but makes it impossible to retrieve the input values – In practice, it's never been proven that any specific known function is truly one way – Cryptographers rely on functions that they believe are one way – It's always possible that they might be broken by future cryptanalysts
Reversability	Being able to undo the operation of encryption
Nonce	The nonce must be a unique number each time it is used – Used to make sure that a key is not re-used twice – The main feature of a nonce is that it is a number that is only used once (nonce) – It can be a counter (for example) – The nonce is public, whereas the (shared) key is private
Initialization vector (IV)	An IV is a random bit string – It is the same length as the block size and is XORed ($\oplus$) with the message – IVs are used to create unique ciphertext every time the same message is encrypted using the same key
Steganography	Steganography is the art of using cryptographic techniques to embed secret messages within other content. Some steganographic algorithms work by making alterations to the least significant bits of the many bits that make up image files.

8.1. OBJECTIVES

- Confidentiality (Privacy)
  - Only authorized persons should read a message, get to know sener/receiver, know about the existence of a message
- Integrity
  - Data should be demonstrably unaltered from sender to recipient
- Authentication
  - The identity of an author of a message should be clearly verifiable
- Non-repudiation
  - The sender of a message should not be able to deny authorship or having performed an action.
- Procedures do not necessarily have to fulfill every objective

8.2. KERCKHOFF'S PRINCIPLES

Term	Definition
Security through obscurity	The security of a system or process depends on the confidentiality of its secrecy of its functioning

Term	Definition
Kerkhoff's Principle (Auguste Kerkhoffs, 1883)	The security of an encryption method is based on the secrecy of the key and not on the secrecy of the encryption algorithm

A cryptographic system should be secure even if everything about the system, except the key, is public knowledge
- This principle makes algorithms known and public, allowing anyone to examine and test them
- The principle can be summed up as "The enemy knows the system"
- Public exposure may expose weaknesses more quickly, leading to the abandonment of insufficiently strong algorithms and quicker adoption of suitable ones
- A large number of cryptographers adhere to this principle, but not all agree
- Some believe that better overall security can be maintained by keeping both the algorithm and the key private

8.3. SHANNON'S PRINCIPLES

Term	Definition
Confusion	Confusion occurs when the relationship between the plaintext and the key is so complicated that an attacker can't merely continue altering the plaintext and analyzing the resulting ciphertext to determine the key – The mapping between input and output is very confusing – Substitution of bytes adds confusion – Example: S-Box
Diffusion	Diffusion occurs when a change in the plaintext results in multiple changes spread throughout the ciphertext – A small change in the input leads to a big change on the output – Permutation of bytes adds diffusion; also known as a transposition – Example: P-Box

8.4. CAESAR CIPHER

One of the earliest known cipher systems was used by Julius Caesar to communicate with Cicero in Rome while he was conquering Europe

- To encrypt a message, you simply shift each letter of the alphabet three places to the right
- The Caesar cipher became known as the ROT3 (or Rotate 3) cipher
- The Caesar cipher is a substitution cipher that is mono-alphabetic

8.5. SP-NETWORK

An SP-Network (Substitution-Permutation Network) is an algorithm that uses repeated substitution and permutation operations

- Substitution: Replacing bytes with others
- Permutation: Swapping bytes around
- The substitutions and permutations are combined into a round.
- Rounds are then repeated many times

8.6. XOR

XOR is a binary operator between two values that returns true if either input or the other is true but not both
- Extremely useful in cryptography
- x is deciding whether y will change
- Applying XOR twice reverses its effect
  - Think of x as encrypting y, and then decrypting it again

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

8.7. ONE-TIME PAD (OTP)

We can design a cipher that uses XOR to encrypt and decrypt a message
- Use a key that's the same length as the message
- XOR each message bit with each key bit

```
M 01011010 00110101
          ^
K 01001011 10111001
          =
C 00010001 10001100
```

If you take away the key, there is no way to find the message because there is no statical mapping between the input and the output

But
- The OTP is not practical
- A 1 GB file would need a 1 GB key!
- If you ever reuse a key, the entire cipher is broken

8.8. HASHING

- A hash function maps data of any size to a fixed-size output in a deterministic and hard-to-reverse way
- A 128 bit hash function will return a 128 bit string, regardless of how much bits have been hashed in
- Hash functions are used everywhere. Message authentication, integrity, passwords etc.
  - For example, SHA-256 can be used to verify data integrity
- A good hash algorithm should perform quickly but it shouldn't be too quick because if it's too quick, it is easy to break

8.8.1. How it works

- Usually hash functions iteratively jumble blocks of a message after another
  - This is a one-way function
  - There is no way to revert back and restore the initial message
- A hash function is kind of like a washing machine for bits
  - The initial hash is usually defined in the spec
  - A new current hash is created every round
  - We loop for every block of the message
- When we run out of message, we use the current hash as the final hash

8.8.2. Strong hash functions

The output must be indistinguishable from random noise
- It should look like you have just generated random numbers
- It shouldn't look like the output is based on the input
- With SHA-256, the output is pretty much undistinguishable from random noise

Bit changes must diffuse through the entire output
- This is called the avalanche effect

– With SHA-256, a small change in the message makes a big change in the hash

8.8.3. Important properties

It shall to be quick but not too quick

It shall introduce diffusion

– Most hash functions, even MD5, adhere to that

Given a hash, we can’t reverse it

– Most hash functions, even MD5, adhere to that

Given a message and its hash, we can’t find another message that hashes to the same thing

– That’s a collision
– That is problematic because we use hashes to verify that things hasn’t been changed. So finding collision undermines the whole idea
– MD5 is broken in that sense
– shattered.io: two different PDFs that have the exact same SHA-1 hash

8.8.4. Current standards in cryptographic hash functions

The current standard is the SHA-2 family with 256-bit and 512-bit variants

SHA-3 is not better or worse than SHA-2

– SHA-3 is a completely different function (Keccak algorithm)
– SHA-3 was designed in case something happens to SHA-2

8.8.4.1. KMAC 128/256

KMAC 128/256 is a new SHA-3 based KECCAK MAC
– standardized in Dec 2016, NIST SP 800-185
– alternative method to SHA-2
– permutation-based hash algorithm (sponge construction)
– very strong resistance to the pre-image

Optimized for parallel processing; efficient on various platforms

8.8.5. Hash functions for Password Storage

Hash functions such as SHA-256 are not good to store passwords because there are too fast

– They are designed to be quick to summarize data
– Vulnerable to brute-force attacks. The attackers hash the passwords and compare with the hashes to see if there are any matches

The hashes are iterated to slow them down on purpose

– Argon2 is memory-hard and designed to resist brute-force attacks
– Configurable parameters for memory usage, iterations, and parallelism.
– Slows down attackers – makes massive guessing attempts significantly harder.
– This is exclusively good for login and passwords and not not suitable for general-purpose hashing
– It is totally useless for any kind of other hash function usage

8.8.6. Comparison of hash families

Type/ Family	*Output Length*	*Rounds*	*Security*	*Use / Application*	*Examples / Libraries*
MD5	128-bit	4	128-bit, fast, insecure	Legacy systems, checksums	OpenSSL, hashlib (Python)
SHA-1	160	80	160-bit, insecure	Legacy signatures, integrity checks	OpenSSL, hashlib (Python)
SHA-2	224, 256, 384, 512	64, 80	Secure, widely used	Digital signatures, certificates	OpenSSL, hashlib (Python)
SHA-3 (Keccak)	224, 256, 384, 512	24	Resistant to certain attacks, flexible	Modern crypto applications	hashlib (Python ≥3.6)
bcrypt	184-bit		Adaptive, salted, GPU-resistant	Password storage	bcrypt (Python)
Argon2	256		Winner of Password Hashing Competition, highly secure, configurable	Password storage, key derivation function	argon2-cffi (Python)
PBKDF2	Varies		Iterative, widely supported, configurable iterations	Password storage	hashlib.pbkdf2_hmac (Python)

8.8.7. Where are hashes used?

Hashing lets us ensure that a message hasn't been altered

1) Digital signatures
2) Message Authentication Codes (MAC)
   – A short piece of information used for authenticating and integrity-checking a message.

8.8.8. HMAC

– MAC approaches may have issues due to the structure of common hash functions like SHA-256
  – MAC with SHA-1 and SHA-2: possibility of length extension attack
– Hash based MAC (HMAC) is a standardized form and the most common approach, it splits a key in two and hashes twice
  – We hash two times to be safer.
  – We split the key into two and we hash twice with each key
  – Then not vulnerable to length extension attack

8.9. SYMMETRIC CRYPTOGRAPHY

Symmetric key algorithms rely on a shared secret key that is distributed to all members who participate in the communications.

– This key is used by all parties to both encrypt and decrypt messages
– The sender encrypts with the shared secret key and the receiver decrypts with it.
– When large-sized keys are used, symmetric encryption is very difficult to break.

It provides for the security service of confidentiality

8.9.1. Stream ciphers

We can approximate a one-time pad by generating an infinite pseudo-random keystream

– Stream ciphers work on messages of any length
– The nonce guarantees that each keystream is unique, even if the same key is reused

8.9.1.1. Pros and cons

Pro	*Contra*
Encryption of long continuous streams, possibly of unknown length	The keystream must appear statistically random
Extremely fast with a low memory footprint, ideal for low-power devices	You must never reuse a key + nonce
If designed well, it can seek to any location in the stream	Stream ciphers do not protect the ciphertext (= no guaranteed integrity)

8.9.2. Block ciphers

Block ciphers take an input of a fixed size and return an output of the same size

– Block ciphers attempt to hide the transformation from message to ciphertext through confusion and diffusion
– Most block ciphers are SP-Networks

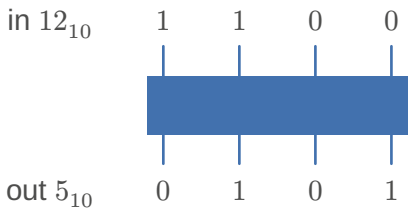
The Advanced Encryption Standard (AES) is an SP-Network

– Almost everything uses AES
– There are others (e.g. Feistel Ciphers)

8.9.2.1. S-Box

Provides Confusion

Visualization

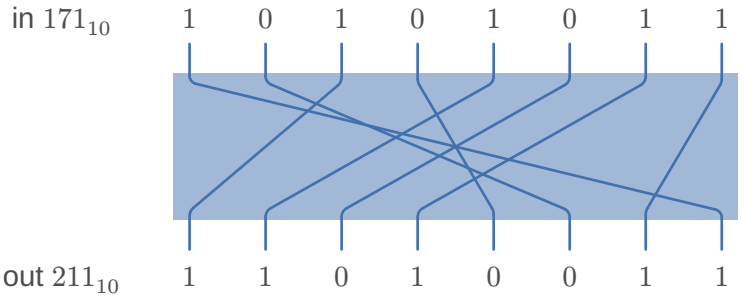


Lookup Table

in	0	1	...	12	...	15
out	6	13	...	5	...	4

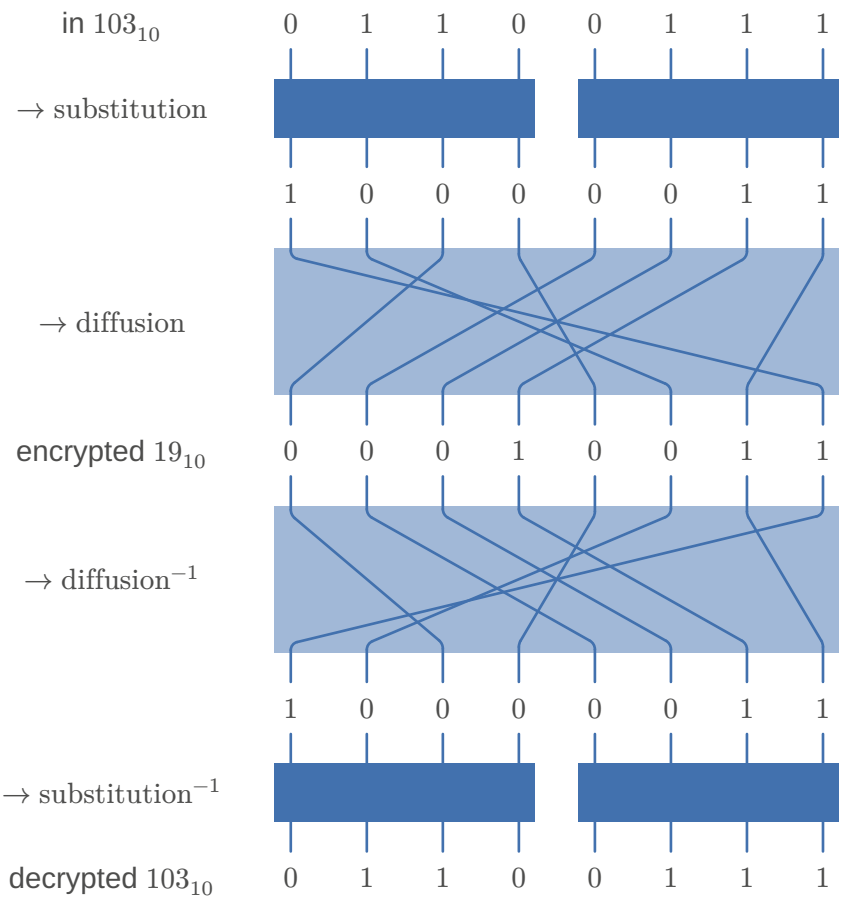
8.9.2.2. P-Box

Provides Diffusion



8.9.2.3. Encryption and decryption in a basic SP-Network

Combines S-Box and P-Box

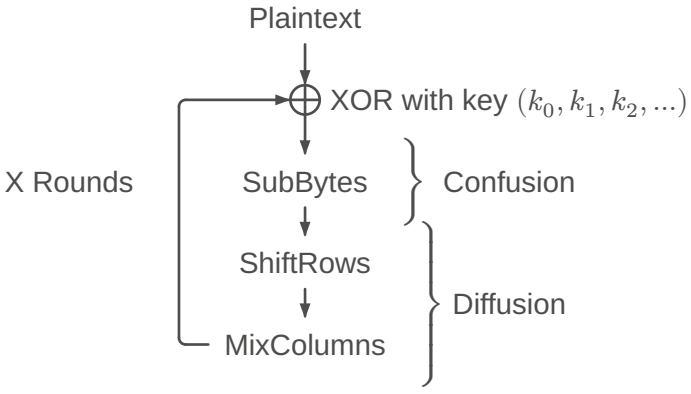


8.9.2.4. Advanced Encryption Standard (AES)

- A standard built around the Rijndael algorithm
  - Superseded DES as a standard in 2002
- SP-Network with a 128-bit block size
  - Key length of 128, 192 or 256-bits
  - 10, 12 or 14 rounds
  - Each Round:
    - SubBytes
    - ShiftRows
    - MixColumns
    - Key Addition

Round Structure

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$



Key addition to a Block / XOR

128 bits block after XOR with the extended key

b_{00}	b_{01}	b_{02}	b_{03}	b_{04}	b_{05}	b_{06}	b_{07}	b_{08}	b_{09}	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}

b_{00}	b_{04}	b_{08}	b_{12}
b_{01}	b_{05}	b_{09}	b_{13}
b_{02}	b_{06}	b_{10}	b_{14}
b_{03}	b_{07}	b_{11}	b_{15}

SubBytes

It is a lookup table, there is no fixed point (byte 15 doesn't end up byte 15)

There is no opposite bit flap (10101010 didn't become 01010101)

$S(b_{00})$	$S(b_{01})$	$S(b_{02})$	$S(b_{03})$	$S(b_{04})$	$S(b_{05})$	$S(b_{06})$	$S(b_{07})$	$S(b_{08})$	$S(b_{09})$	$S(b_{10})$	$S(b_{11})$	$S(b_{12})$	$S(b_{13})$	$S(b_{14})$	$S(b_{15})$

$S(b_{00})$	$S(b_{04})$	$S(b_{08})$	$S(b_{12})$
$S(b_{01})$	$S(b_{05})$	$S(b_{09})$	$S(b_{13})$
$S(b_{02})$	$S(b_{06})$	$S(b_{10})$	$S(b_{14})$
$S(b_{03})$	$S(b_{07})$	$S(b_{11})$	$S(b_{15})$

ShiftRows

$S(b_{00})$	$S(b_{05})$	$S(b_{10})$	$S(b_{15})$	$S(b_{04})$	$S(b_{09})$	$S(b_{14})$	$S(b_{03})$	$S(b_{08})$	$S(b_{13})$	$S(b_{02})$	$S(b_{07})$	$S(b_{12})$	$S(b_{01})$	$S(b_{06})$	$S(b_{11})$

$S(b_{00})$ $S(b_{04})$ $S(b_{08})$ $S(b_{12})$	No changes
$S(b_{05})$ $S(b_{09})$ $S(b_{13})$ $S(b_{01})$	1 to the left
$S(b_{10})$ $S(b_{14})$ $S(b_{02})$ $S(b_{06})$	2 to the left
$S(b_{15})$ $S(b_{03})$ $S(b_{07})$ $S(b_{11})$	3 to the left

MixColumns

MixColumns is done using a matrix multiplication

– Add operation is an XOR

– Multiplication operation is a multiplication within that finite field (modular polynomial)

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} b_{00} \\ b_{05} \\ b_{10} \\ b_{15} \end{pmatrix} = \begin{pmatrix} 2b_{00} + 3b_{05} + b_{10} + b_{15} \\ b_{00} + 2b_{05} + 3b_{10} + b_{15} \\ b_{00} + b_{05} + 2b_{10} + 3b_{15} \\ 3b_{00} + b_{05} + b_{10} + 2b_{15} \end{pmatrix}$$

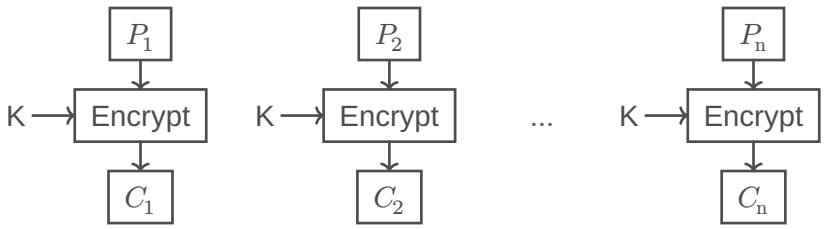
8.9.2.5. Modes of operation for block ciphers

– Realistically, messages of exactly 128-bits are pretty unlikely
  – We need some mechanism to encrypt messages that are longer or shorter
– A mode of operation is the combination of multiple instances of block encryption into a usable protocol
– There are several modes of operations, in this lecture we only cover the following:
  – Electronic Code Book (ECB)
  – Cipher Block Chaining (CBC)
  – Counter Mode (CTR)

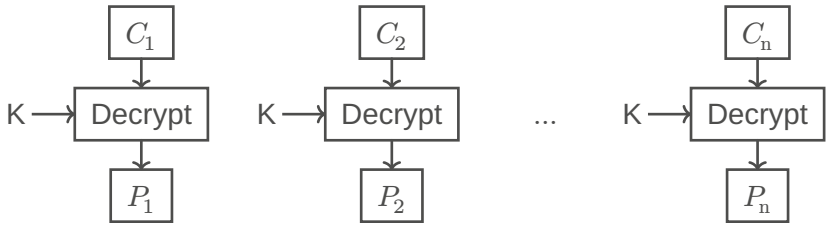
8.9.2.6. Electronic Code Block (ECB)

– Just encrypt each block one after another with same key
– Weak to redundant data divulging patterns
– Electronic codebook is not recommended!

Encryption



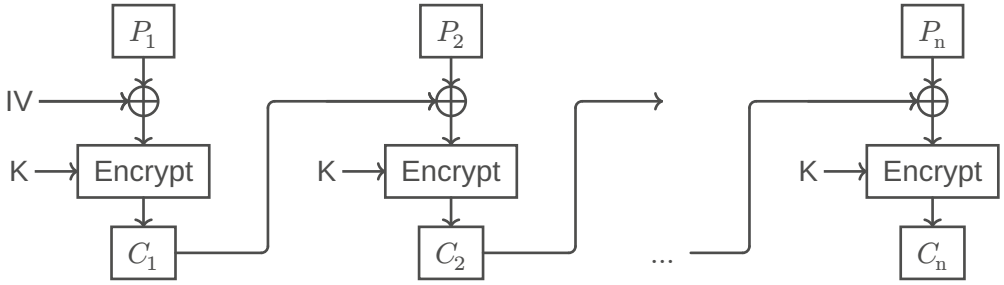
Decryption



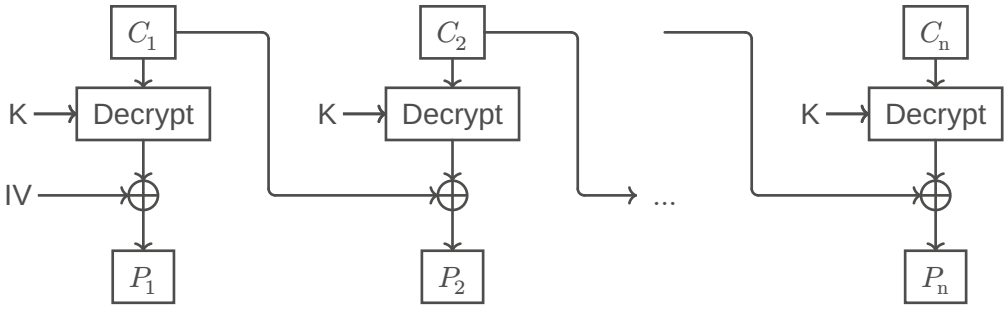
8.9.2.7. Cipher Block Chaining (CBC)

– XOR the IV with the first input, then XOR the output of each cipher block with the next input
  – Not parallelizable
  – It is better than ECB but not perfect

Encryption



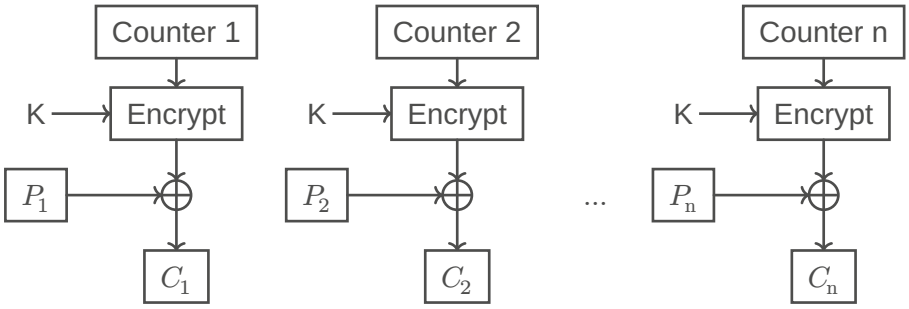
Decryption



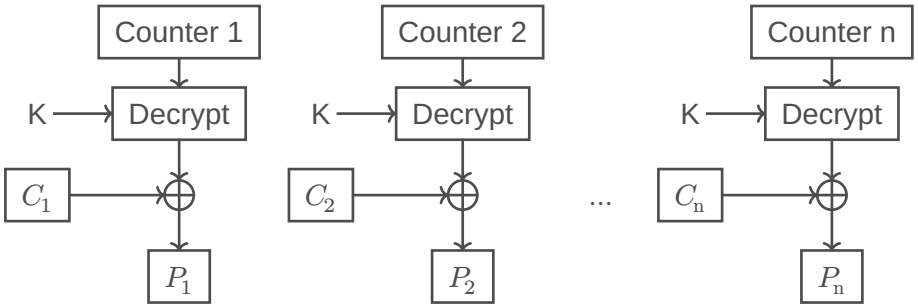
8.9.2.8. Counter Mode (CTR)

– Encrypting a counter to produce a stream cipher
  – Pretty good - can also be parallelized!
  – Convert a block cipher into a stream
– We don't encrypt the message
  – We encrypt a number and use the random number that comes out to XOR the message
– Standard mode for all type of encryption cipher (AES)

Encryption



Decryption



8.9.3. Remarks

Key distribution

– Parties must have a secure method of exchanging the secret key before establishing communications with a symmetric key protocol

Symmetric key cryptography does not implement non-repudiation

– Because any communicating party can encrypt and decrypt messages with the shared secret key, there is no way to prove where a given message originated

Symmetric key cryptography does not implement message integrity

The major strength of symmetric key cryptography is the great speed at which it can operate

– Symmetric key encryption is very fast, often 1′000 to 10′000 times faster than asymmetric algorithms
– Lots of the processor have an AES instruction set
– Alternative to AES: the Chacha20 cipher (for devices that don't have a CPU with an AEC instruction set)

8.10. ASYMMETRIC CRYPTOGRAPHY

– Asymmetric encryption is a cryptographic method that relies on mathematically linked key pairs.
– While public-key encryption uses the public key (e, n) to encrypt and the private key (d) to decrypt, digital signing reverses this process – the private key (d) signs the message and the public key (e, n) verifies the signature, ensuring authenticity and integrity.
– RSA (Rivest–Shamir–Adleman) algorithm was the first public-key encryption algorithm developed/published for commercial use.

8.10.1. Rivest - Shamir - Adleman (RSA) Algorithm

RSA is It is widely used for secure data transmission. There are two very useful use cases for RSA:

1) Encryption that only the owner of the public key can read
2) Signing that must have been performed by the owner of the private key

It provides both encryption and/or authentication.

Term	*Definition*
Discrete logarithm	$f(g,a,p) = g^a \bmod p$ (Diffie-Hellman, Elliptic Curves, DSA)
Factoring	$f(p,q) = p \cdot q$ (RSA)

8.10.2. Euler's Theorem

Let $n \in \mathbb{N} \setminus \{0\}$ and $z \in \mathbb{Z}$ with $\gcd(z,n) = 1$. Then $z^{\varphi(n)} \equiv 1 \bmod n$.

8.10.2.1. Euler's φ-Function (Totient)

Let $n \in \mathbb{N} \setminus \{0\}$ and $\mathbb{Z}_n^* = \{x \in \mathbb{Z}_n \mid x \text{ has a multiplicative inverse in } \mathbb{Z}_n\}$.
Then $\varphi(n)$ defines:

$$\begin{aligned}\varphi(n) &= \text{Nr. of elements in } \mathbb{Z}_n \text{ with mult. inv.} \\ &= \text{Amount of Numbers } 1 \leq q \leq n \text{ with } \gcd(q,n) = 1 \\ &= |\mathbb{Z}_n^*|\end{aligned}$$

We also call those numbers ***relatively prime with*** n.

Rules

1) Let $n \in \mathbb{N}$ a prime number, then $\varphi(n) = n - 1$
2) Let $n \in \mathbb{N}$ a prime number and $p \in \mathbb{N} \setminus \{0\}$, then $\varphi(n^p) = n^{p-1} \cdot (n-1)$
3) Let $m, n \in \mathbb{N} \setminus \{0\}$ and $\gcd(m,n) = 1$, then $\varphi(n \cdot m) = \varphi(n) \cdot \varphi(m)$

8.10.2.2. Computation

1) Choose two prime numbers p, q
2) Calculate $n = p \cdot q$
3) Calculate $\varphi(n) = (p-1)(q-1)$
4) Choose a, b, so that $a \cdot b \equiv 1 \bmod \varphi(n)$
5) Forget $p, q, \varphi(p \cdot q)$

Public key is now n, b, private key is n, a

– Encrypt: $z = c^a \bmod n$
– Decrypt: $c = z^b \bmod n = c^{a^b} \bmod n$

8.10.2.3. Weaknesses

– RSA is very weak if you encrypt short messages

  - OAEP is added in short messages
- It is not common to see encryption done using RSA
  - Encryption with RSA was used by TLS to send shared keys around but is not anymore. (DH instead)
  - Signing is being done by using RSA
- RSA is 1000x slower than symmetric crypto systems

8.10.2.4. Optimal Asymmetric Encryption Padding (OAEP)

Standard padding for encryption using RSA. It is a pseudo random padding that introduces an IV to the process and then hashes it. The server that receives the ciphertext will decrypt it and will have to do the exact same padding to make sure that the messages match up.

- Adds random bits (salt) and pads short messages to proper length.
- Optional hash over additional data (ad) for authenticity.
- Uses Mask Generating Functions (MGF1/MGF2, e.g., SHA3).
- Structure: 0x00 | maskedSalt | maskedDB
  - encrypted with RSA trapdoor permutation.

8.10.2.5. Message integriy verification

1) Signer's side
   - The message is first hashed
   - Hashing generates a unique digest for message integrity verification
   - The hash is signed and sent with the message
2) Verifier's side
   - The receiver computes the hash of the received message, applies the sender's public key to the signature, and verifies that both hashes match.

8.10.2.6. Signing

- A message is sent as a challenge to prove the server's identity.
- The client sends to the server a message that it has to sign.
  - The server is going to use PSS padding and sign it with its private key
  - The client verifies the signature using the public key.
  - If the signature is valid for the original message, the server has successfully proved itself.
- A similar challenge-response mechanism is used in TLS to authenticate the server during the handshake, where RSA-PSS is used as the signature scheme.

8.10.2.7. Probabilistic Signature Scheme (PSS)

- Message is hashed before applying the RSA signature function.
- A randomly chosen salt is added to the hash value.
- Padding is applied to match the required length based on the key size.
- Uses a hash function H and random salt for security and unpredictability
- Provides strong resistance against attacks compared to deterministic signatures.

8.10.2.8. From RSA to DSA and Elliptic Curves

Within a few years, RSA is going to become too slow because always bigger keys will have to be used. The main alternative to RSA for signature is the Digital Signature Algorithm (DSA)

- ECDSA (or DSS) is much faster than RSA and it will become a standard very soon
  - Elliptic Curve Digital Signature Algorithm (ECDSA) or Digital Signature Standard (DSS)
- Ed448 and Ed25519 are various types of DSA with different curves

DSA can't be used for encryption. It only works one way, for the signing part. It acts a lot like RSA, but uses mathematics similar to Diffie Hellman. It can also make use of Elliptic Curves.

8.10.3. Discrete Logarithm Problem

When operating $\mod n$, we call the operation a discrete logarithm:
– $a^b \mod n \equiv c$
– $b \equiv \log_a(c) \mod n$
– Discrete logarithms are much harder to compute

8.10.4. Primitive roots of prime numbers

g is said to be a ***primitive root*** of the prime number p, if $g \mod p, g^2 \mod p, g^3 \mod p, ..., g^{p-1} \mod p$ are distinct

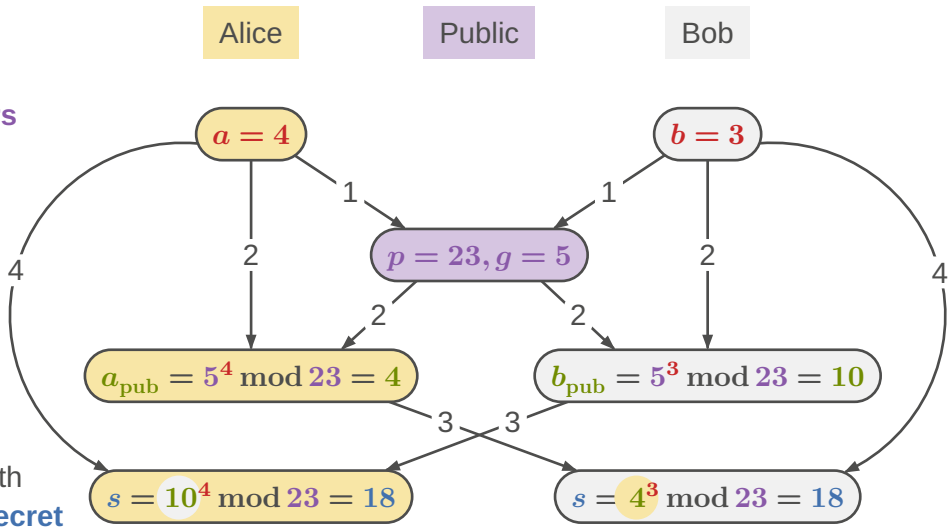
If the multiplicative order of a number $g \mod p$ is equal to Euler Totient Function $\varphi(p)$, then it is a primitive root

8.10.5. Diffie-Hellman (DH)

– With Diffie-Hellman, two parties can jointly agree a shared secret over an insecure channel
– Every communication handshake on the internet is powered by DH
– We are exchanging some parts of the mathematical key and then we secretly create the key ourselves

Process:
– p is usually 4096 or 6144 bits
– g is a primitive root of p
– **private keys** are values between 1 and p
– The **shared secret** (often called the pre-master secret) serves as the foundation for deriving all subsequent session keys.
  – The raw shared secret is not used directly for encryption, as it is typically a very large integer (e.g., 4096 bits in RSA or DH) and may not have uniform entropy.
  – We derive a master secret using a hashed-key derivation function (HKDF), for example the SHA-256 hash function
– The only way to find **a** or **b** is to solve the Discrete Logarithm Problem.

1) Agree on **public parameters** (prime and generator)
2) Combine **private key** with **the parameters**
3) Send resulting **public keys** to each other
4) Combine other's **pubkey** with **private key** to get shared **secret**



Elliptic-Curve Diffie Hellman (ECDH) is becoming the standard nowadays due to shorter keys.

8.10.6. Elliptic Curve Cryptography (ECC)

– Elliptic curves are a drop-in replacement for the mathematics underpinning regular Diffie-Hellman
– ECDHE = Elliptic Curve Diffie-Hellmann Ephemeral
– Elliptic curve is a two-dimension curve
  – $y^2 = x^3 + ax + b$
  – The private key is a number
  – The public key is composed of two numbers (x, y)

– The elliptic curve discrete logarithm problem (ECDLP) is slightly more difficult to solve than the discrete logarithm problem
– Elliptic curves are much stronger than traditional public-key schemes for the same key length in bits

8.10.7. Perfect Forward Secrecy in Ephemeral DH KEX

– In most protocols, running Diffie-Hellman in ephemeral mode forces a new key exchange for every new session
  – This is called PERFECT FORWARD SECRECY
– DH in ephemeral mode means a new DH every time, not necessarily every message
  – Refreshing a webpage would generate a new secret every time
  – If anyone is sniffing our traffic and is breaking our keys, they don't get any of the previous messages.

Ephemeral session keys should be used for communication and long-term public keys for authentication.

9. TRANSPORT LAYER SECURITY (TLS)

TLS is a transport layer protocol and supports:
– Confidentiality (AES)
– Integrity (MAC / HMAC)
– Authentication (Server, optional Client, RSA Certificate)

TLS can encrypt communication over the internet with any protocol. Secure Socket Layer (SSL) was the original name, which after version 3.0 became Transport Layer Security (TLS), which currently exists as version 1.3

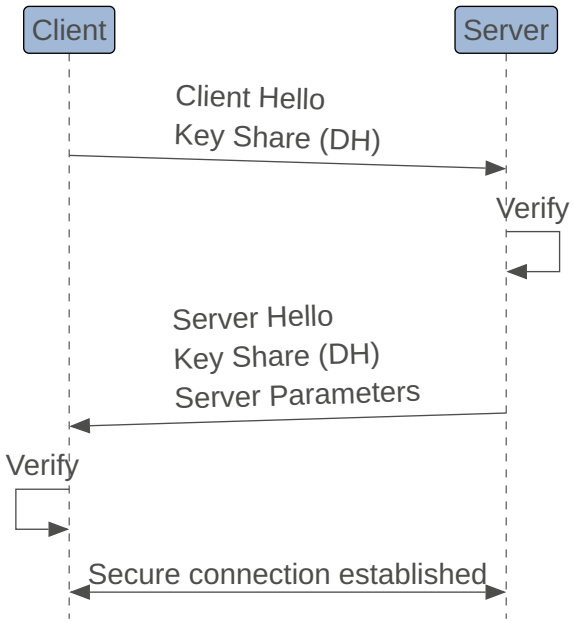
– TLS < 1.2 is insecure
– TLS 1.2 is secure if configured correctly
  – It supports cipher suites that are considered insecure

9.1. HANDSHAKE

Allows server and client to:

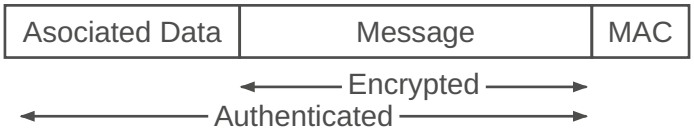
1) Authenticate each other
2) Negotiate encryption and MAC algorithms
3) Negotiate cryptographic keys to be used

In the diagram the 0RTT Handshake is shown.



9.2. AUTHENTICATED ENCRYPTION WIITH ASSOCIATED DATA (AEAD)

AEAD is a technique used in cryptography which provides both the encryption and authentication and also additional Associate Data (AD) along with the encrypted message. It allows the recipient to check integrity of both encrypted and unencrypted information and provides confidentiality.



9.2.1. AES Galois Counter Mode (GCM)

AES is a block cipher in CTR mode and GCM is the Message Authentication Code (MAC). GCM computes a Galois Message Authentication Code (GMAC) over the ciphertext and the additional data.

10. PUBLIC KEY INFRASTRUCTURE (PKI)

A set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

A PKI is used to bind a public key to an identity of a person or an organization. The binding is performed with a registration process by a Registration authority (RA) (based on Certificate Policies (CP) and Certification Practice Statements (CPS)) and an issuance of a certificate by a Certificate Authority (CA). The CA itself can be validated to be able to perform this service by an independent Validation Authority (VA).

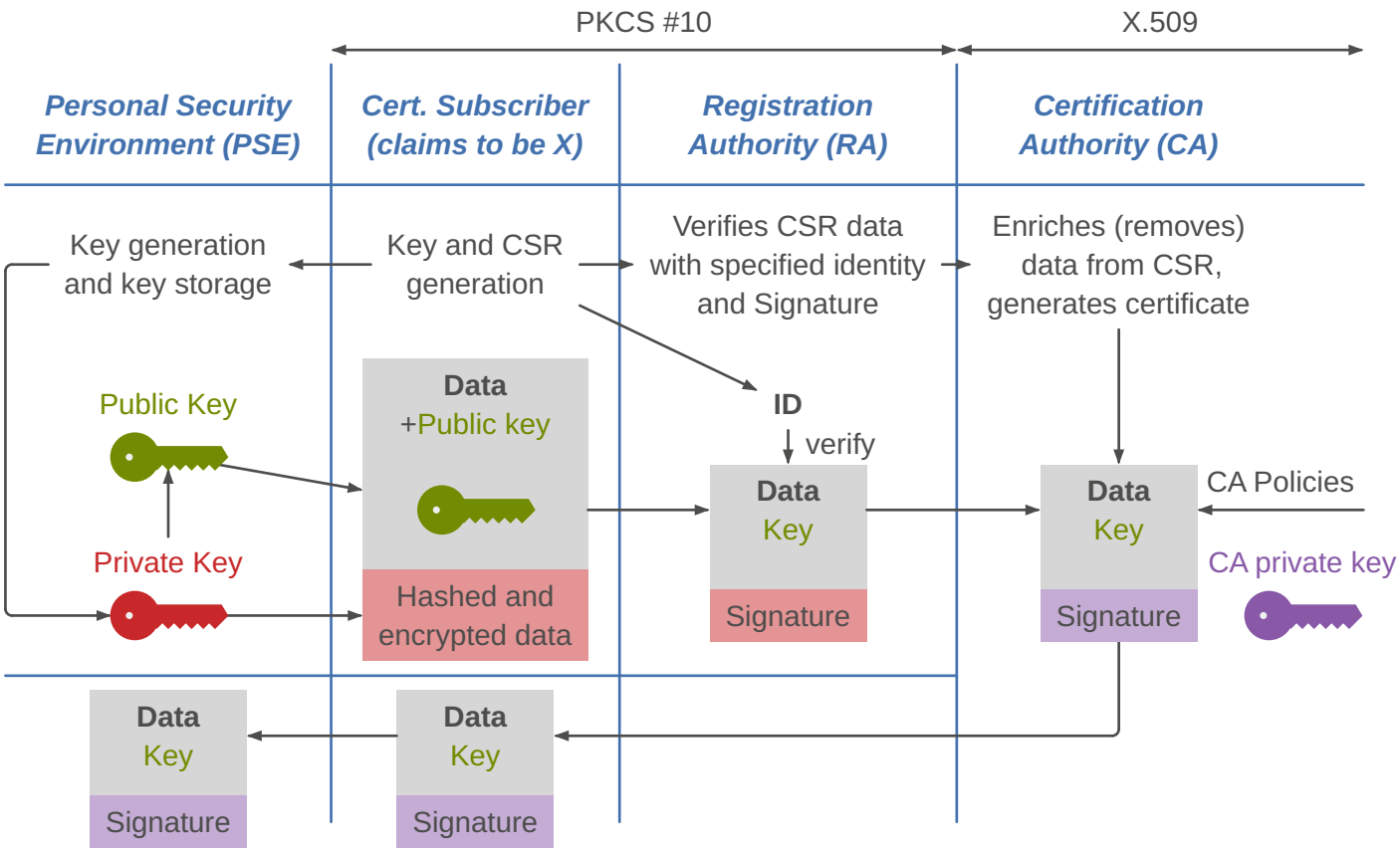
Term	*Definition*
TSA	Time Stamp Authority

– Certificate Lifecycle

10.1. DIGITAL CERTIFICATE

A digital certificate enables i.a. identification, document signing, non-repudiation and data integrity through encryption and authentication.

Process of certificate creation:



Hybrid cryptosystem: Combining different methods (hashing, encrypting) for optimal use-cases.

– Hashing: Digital fingerprint
– Symmetric cryptography: Bulk encryption
– Asymmetric cryptography: Signing or exchanging symmetric keys

10.1.1. Certificate pinning (HPKP)

Certificate pinning explicitly trusts only one certificate, all other root anchors are ignored. There are 3 different pinning models: root pinning, intermediate CA pinning and end entity pinning. Pinning poses a big risk if the HPKP is hacked, because certificates are no longer fully validated. Legacy since 2017.

10.1.2. X.509

10.1.2.1. Encoding

ASN.1 (Abstract Syntax Notation 1) Object Representation
– Distinguished Encoding Rules
  – DER-encoded binary X.509 (.der)
  – Base64-encoded X.509 (.cer,crt)

10.1.2.2. Formats

– .csr PKCS#10 - Certificate Signing Request
– *.p7b PKCS#7 - Format for exchanging certificate chains*
– .cms Cryptographic Message Syntax (deprecated PKCS#7)
– .crl Certificate Revocation List

Container formats

– .pfx, .p12, .pkcs12
  – identical formats
  – include private key and certificate

– .pem
  – base64 encoded DER data
  – private key or certificate

10.2. CIPHER SUITES

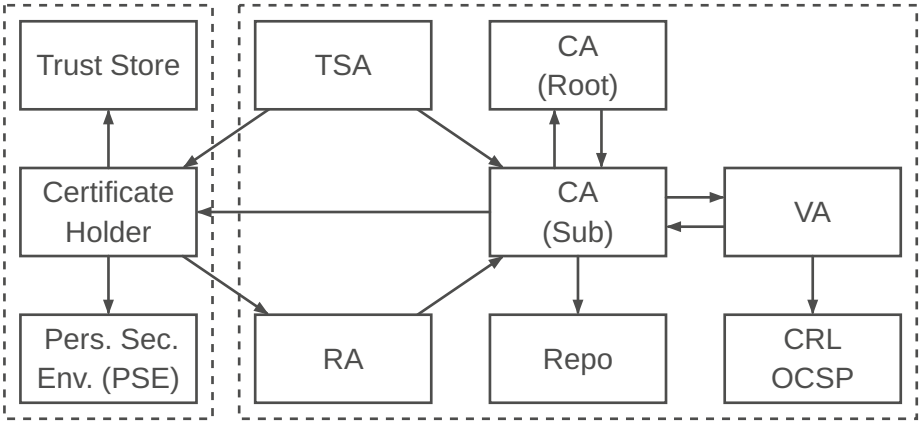
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

	Key		Bulk	Message
Encryption	exchange	Signature	encryption	Authentication
protocol	_algorithm	_algorithm	_WITH_algorithm	_Code (MAC)

10.2.1. Recommended HTTPS Cipher suites (TLS 1.2)

– Should support PFS (Perfect Forward Secrecy).
– Diffie–Hellman key exchange-based PFSs
  – DHE-DSA, DHE-RSA
– Elliptic Curve Diffie–Hellman-based PFSs
  – ECDHE-ECDSA, ECDHE-RSA
– GCM > CBC
– ECDHE_ECDSA > ECDHE_RSA > DHE_RSA (over RSA)
– P521 > P384 > P256
– 256-bit > 128-bit
– SHA512 > SHA384 > SHA256 (SHA-1 should no longer be used)

10.3. PKI COMPONENTS



10.3.1. Certificate Authority (CA)

– Trusted issuer for digital certificates
– Creates digital certificates from Certificate Signing Requests (CSRs)
– Guarantees the content of the certificates
– Signs Certificate Revocation Lists (CRLs)

10.3.2. Subscriber

– End-Entity / Certificate Holder / Client
– Creates keys, CSR and sends it to RA for authenticity verification

10.3.3. Trust- and keystore

10.3.3.1. Truststore

Central storage for certificates from trusted entities.

- Implemented in OSes, applications or middleware and includes
  - Certificates for trusted entities
  - Trust-Chains and public keys
- Ideally only limited amount of certificates
- Should be verified after each update
- Content usually decided by Business/IT

10.3.3.2. Keystore

Central storage for private keys and certificates.

Neither Truststores nor Keystores should leak to the public.

10.3.4. Registration Authority (RA)

- Intermediary between Subscriber and CA
- Validates the CSRs and ensures that subscriber is authorized to receive that certificate
- Forwards validated CSRs to CAs
- Revokes certificates when they expire

10.3.5. Validation Authority (VA)

- Provides the means to verify the integrity of digital certificates
- Enables remote real-time certificate validation
- Makes sure that information regarding revoked certificates is available and up-to-date
- CA sends updates to VA regarding the state of a certificate, which in turn delivers then to clients
- VA can be a separate entity or included in the CA

RFC 5280

10.3.6. Certificate Revocation List (CRL)

Blocklist of revoked X.509 certificates. Includes serial numbers and timestamps for certificates and is signed by the CA. RFC 5280 RFC 6818

Revoked: Irreversibely revoked

Hold: Temporarily revoked

10.3.7. Certificate Status Protocol (OCSP)

Alternative to CRL for validating the state of certificates. RFC 6960

Pros:

- OCSP-Responder can deliver near real-time up-to-date information
- Non-revoked certificates can be differentiated from faked certificates?

10.3.8. OCSP Stapling

Alternative to CSP (the alternative to CRL). I love my field of research. RFC 6961

11. E-MAIL

11.1. MIME

Multipurpose Internet Mail Extensions

Contains various header fields and is split into multiple body parts

***Example** 2:*

```
From: trinity@matrix.org
To: neo@matrix.org
MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary=boundary1
--boundary1
Content-Type: text/plain; charset=us-ascii
Dear Neo, please study the attached Word document.
--boundary1
Content-Type: application/msword; name="Matrix.doc"
Content-Transfer-Encoding: base641
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfH
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbTrfv=
--boundary1--
```

RFC 1521 RFC 1522

11.2. S/MIME

Secure Multipurpose Internet Mail Extensions provides two security mechanisms: digital signatures and encryption.

***Example** 3:*

```
Content-Type: multipart/signed;
protocol="application/pkcs7-signature";
micalg=sha256; boundary=boundary1
--boundary1
Content-Type: multipart/mixed; boundary=boundary2
... multipart message with various MIME-types ...
--boundary1
```

```
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfH
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbTrfv=
--boundary1--
```

RFC 2046 RFC 8551

11.2.1. Signed S/MIME with Multiple Signatures

– Theoretically, each signer may use a different digest algorithm.
– In practice, SHA-256 is used throughout.

11.2.2. Encapsulated S/MIME Signatures

MIME content carried within a CMS ***SignedData*** object. RFC 5652

***Example** 4:*

```
Content-Type: application/pkcs7-mime;
  smime-type=signed-data;
  name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfH
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbTrfv=
```

Pro	***Contra***
The MIME content is protected against modifications caused by transfer encoding changes enforced by intermediate mail transfer agents.	The message cannot be read unless the recipient's mail client supports S/MIME.

11.2.3. Encrypted S/MIME Messages

MIME content carried within a CMS ***EnvelopedData*** object. RFC 5652

***Example** 5:*

```
Content-Type: application/pkcs7-mime;
  smime-type=enveloped-data;
  name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfH
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbTrfv=
```

Pro	*Contra*
Protected against transfer encoding changes by intermediate mail transfer agents (MTAs)	Encrypted content is opaque; requires S/MIME support to read

11.2.4. Encrypted S/MIME Messages with Multiple Recipients

The MIME entity is encrypted once using a symmetric content-encryption key, which is then individually encrypted for each recipient using their respective public keys.

11.2.5. Storage of Signature and Encryption Keys

Signature Key

– The private signature key represents the signer's digital identity and must not be copied.
– Best stored on a tamper-proof smart card. Mandated by digital signature legislation.

Encryption Key

– Backup copy protects against information loss. If the private encryption key gets lost or damaged, all encrypted communication received in the past cannot be accessed and read any more.
– In a corporate environment, a deputy should be able to access the encrypted email of an employee who is absent due to vacation, accident, sickness or even death. **the corporate overlords must be mightier than any privacy concerns**

11.3. PGP

– Describes how existing algorithms can be used to exchange data securely on the Internet
– Uses
  – Asymmetric encryption
  – Symmetric encryption
  – Hash functions
  – Compression
– Key pairs (public & private) are used
– GnuPG is an **open-source** (goated) implementation of this standard www.gnupg.org

11.3.1. Web of Trust in PGP

– There is no central authority that manages all the keys, but each participant decides for themselves who they trust (Decentralized trust model)
– Direct trust
  – Keys of persons who have been personally checked
– Indirect Trust
  – Keys from people who have been checked by people who in turn have been personally checked
– This creates chains of trust that become less trustworthy depending on the number of trustworthy ???

11.4. CRYPTOGRAPHY IN DETAIL

11.4.1. Sign-then-Encrypt

– Default in S/MIME
– Advantages:
  – Signer knows the message
  – Signature is hidden
– Disadvantages (Naïve Implementation):
  – Recipient could re-encrypt to someone else

- Disadvantages (Less Naïve Implementation)
  - Ciphertext integrity not protected → Oracle attacks possible

11.4.2. Encrypt-then-Sign

- Default in PGP
- Similar to Encrypt-Then-MAC
- Advantages:
  - Signature ensures integrity of the encrypted message
- Disadvantages (Naïve Implementation):
  - Signer might not know message
  - Recipient could re-sign ciphertext

11.5. SENDER AUTHENTICATION

- Mechanism to verify the legitimacy of the sending domain.
- Prevents email spoofing (falsified sender address) and impersonation attacks (posing as a trusted person).
- Core building block against phishing and spam.
- SMTP does not authenticate the sender by default.
- Anyone can falsify the “From” address without protection.
- The Core mechanisms of e-mail sender authentication include:

11.5.1. Sender Policy Framework (SPF)

RFC 7208

- SPF specifies which servers may send email for a domain.
- A mail server or spam filter that supports SPF checks the sender address in the SMTP MAIL FROM command via a DNS TXT query.
- Example: `strongsec.net TXT "v=spf1 a mx a:lists.strongswan.org -all"`

11.5.2. DomainKeys Identified Mail (DKIM)

RFC 6376

- Global public key infrastructure (PKI) based on DNS
- For verifying sender domain & message integrity
  - Sending server signs selected headers & body and added as DKIM-Signature
  - Receiving server retrieves public key from DNS TXT record

11.5.3. Domain-based Message Authentication, Reporting, and Conformance (DMARC)

RFC 7489

Combines SPF and DKIM, defining actions for failed checks.

- Ensures that emails align with the domain’s SPF & DKIM policies.
- Domain owner publishes DMARC policy in DNS.
- Receiving server verifies email against SPF & DKIM.
- Policy determines handling of failed emails like none, quarantine, or reject.
- It provides aggregate and forensic reports to help monitor compliance and detect abuse.

12. WEB

12.1. COMMON WEAKNESS ENUMERATIONS (CWE)

– Classification of vulnerabilities
– Standardized list of software weaknesses
– Helps developers identify and communicate risks
– Example: CWE-79: Cross-Site Scripting (XSS), CWE-89: SQL Injection

12.2. OPEN WEB APPLICATION SECURITY PROJECT (OWASP)

– Community-driven project for web application security
– Known for OWASP Top 10 – the most critical web security risks
– Examples: Injection, Broken Authentication, Security Misconfiguration
– Practical security guidance for web apps

12.3. APPLICATION SECURITY RISKS

12.3.1. Broken Access Control

Example: Accessing another user's invoice by changing a URL ID.

– Avoidance Insecure Direct Object References (IDOR)
  – Prevent users from accessing other objects by manipulating IDs
  – Use random UUIDs or hashes instead of numeric IDs
  – Implement server-side checks to verify whether the user is authorized to access the
– Centralized Access Control Logic
  – Implement access control centrally in the backend – not in the frontend
  – Use frameworks or middleware to enforce permissions consistently

12.3.2. Injection

– SQL Injections
– Cross-site Scripting (XSS)
  – Reflected: The malicious code is sent to the server via a request and is immediately "reflected" back to the user's browser.
  – Stored or Persistent: The malicious code is permanently stored on the target server.
  – DOM-based: The vulnerability exists entirely within the client-side JavaScript, which processes data from an untrusted source (e.g., the URL fragment #) and writes it to the Document Object Model (DOM) in an unsafe way.
– Directory and Query Injection
  – LDAP, XPath statements
  – XML and SOAP
– Operating System commands (Shell Injection)

12.3.2.1. Mitigation SQL Injection

1) Secure Programming
   – Prepared Statements
   – Stored Procedures
2) Infrastructure Security
   – DB Least Privileges
   – Web Application Firewall (WAF)

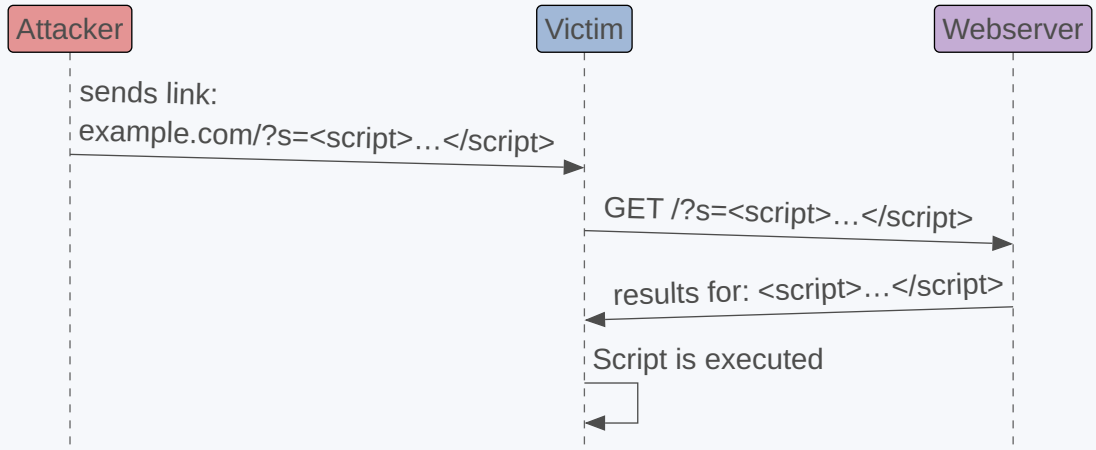
3) Secure Programming
   – Do not disclose SQL errors to the user
   – Anonymous error messages instead

12.3.2.2. XSS

12.3.2.2.1. Reflected XSS

Data provided by a web client is used immediately by server-side code to generate a page of results for that user

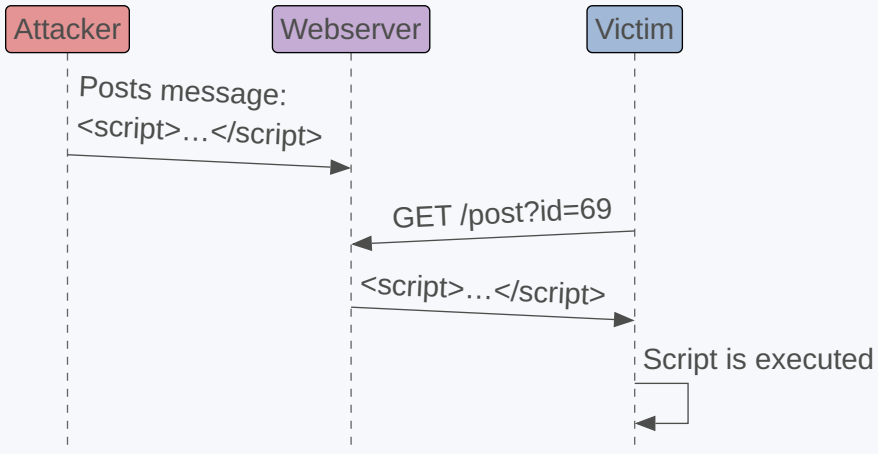
***Example** 6:*



12.3.2.2.2. Stored XSS

Data provided by a web client is stored in a database. This data is then presented to the user unencoded

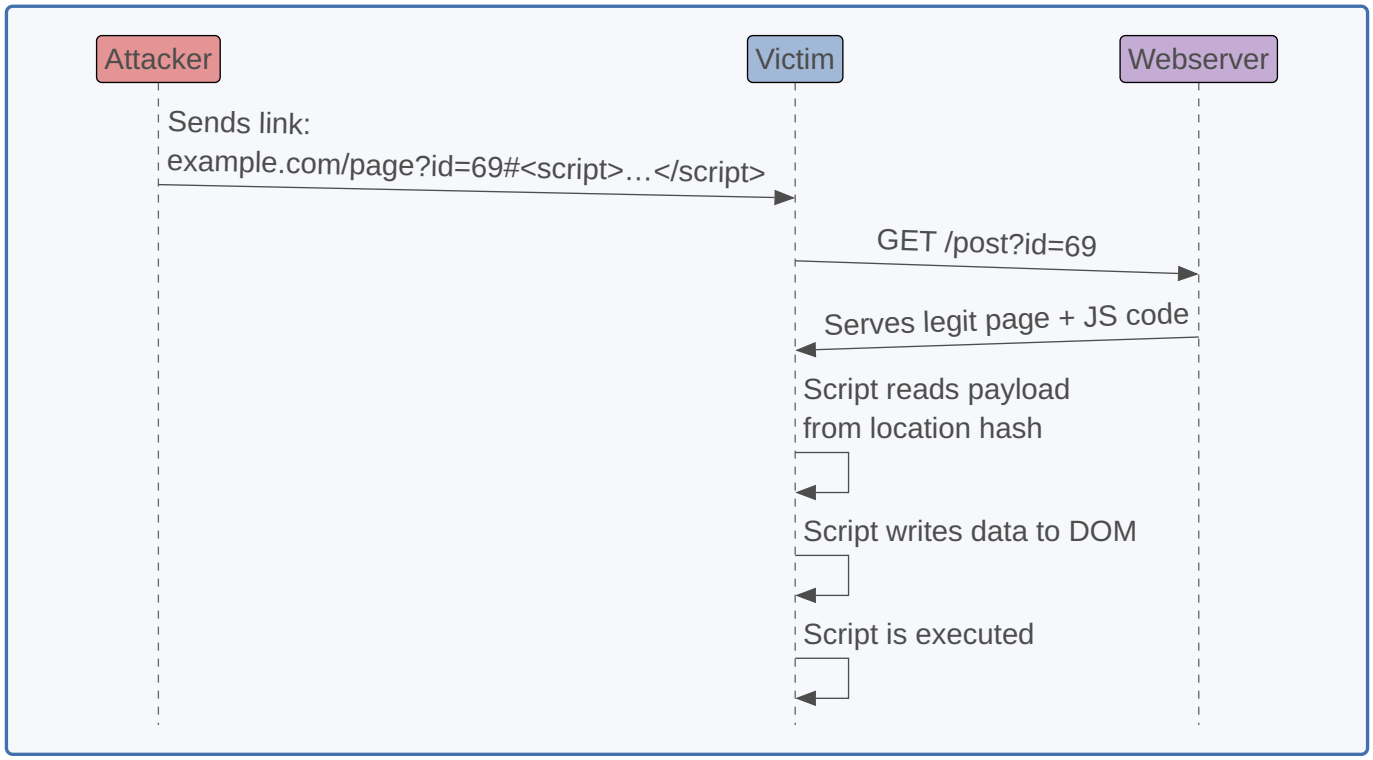
***Example** 7:*



12.3.2.2.3. DOM-based XSS

Attack payload is embedded in the DOM object in the victim's browser used by the original client side script, so that the client side code runs in an "unexpected" manner

***Example** 8:*



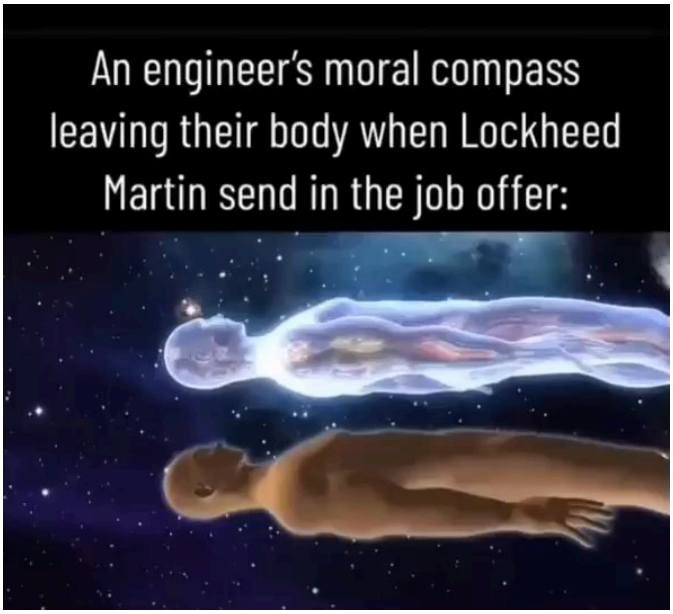
12.3.2.2.4. Mitigation

- Prio 1: Secure Programming
  - input sanitization
  - when storing user input (input encoding)
  - when loading data from db (ouput encoding)
- Prio 2: Security Mechanisms
  - CSP (Content Security Policy)
  - X-XSS-Protection Header
  - Cookie with HttpOnly, Secure, SameSite=Strict
  - Web Application Firewall

13. CYBER KILL CHAIN

A model developed by Lockheed Martin in 2011, adapted from a military concept.

1) Reconnaissance: Gather information on the target.
2) Weaponization: Construct a custom weapon to attack the target.
3) Delivery: Transmit the weapon to the target.
4) Exploitation: The vulnerability is triggered or the target tricked.
5) Installation: Establish persistence, install backdoors or services.
6) Command & Control: The compromised host connects to the attacker.
7) Actions on Objectives: Execute required tasks. Exfiltration, encryption, sabotage, etc.



13.1. WHY START WITH THE ATTACKER?

Defenders who understand how an attack unfolds are the ones who spot it early. Thinking like an attacker is the starting point for every detection strategy.

You can only detect what you understand.

Shared language: A model of attacker behaviour gives the whole team a common vocabulary for alerts, playbooks and post-mortems.

Events rarely look dangerous alone: A login at 2 a.m., a PowerShell script, an outbound connection. Individually noise, together a pattern.

Every control has a counter: Attackers adapt. Defenders who know the phases know where to push back when one control fails.

The leverage point: Breaking any single step in the attack sequence causes the whole intrusion to fail.

13.2. PHASE 1: RECONNAISSANCE

The attacker gathers information about the target. No systems are touched in a way that looks malicious yet.

Possible Information Sources

– Public profiles such as LinkedIn, GitHub or other social media.
– DNS lookup, WHOIS, IP scanning or Shodan research.
– Find potentially leaked credentials of target or people related to the target.
– Legal information through Handelsregister / Zefix.
– Physical information from Google Maps, Street view or other geo software.

13.3. PHASE 2: WEAPONIZATION

The attacker constructs a bespoke weapon to attack the target. Typically combining an exploit with a payload.

Exploit: Office Macro Execution, Unpatched Software, Malicious PDF File

Payload: Remote Access Tool, Ransomware, Spyware / Data Stealer

Weapon: Targeted Phishing Email, Crafted Network Request, Login Attempt on Service

13.4. PHASES 3 & 4: DELIVERY AND EXPLOITATION

The first moment the attacker touches the defender's environment. This is also the defender's best realistic chance to stop the attack.

Delivery: How the weapon reaches the victim: phishing email (still the #1 vector), drive-by downloads, USB drops, exploitation of exposed services, supply-chain compromises.

Exploitation: The payload runs. Either a vulnerability is triggered (browser, Office, VPN appliance) or the user executes the code voluntarily, tricked by social engineering.

Controls that work here: Email filtering, web gateways, EDR, patching, application allow-listing, user awareness training. This is where most security budget gets spent.

Why does SOC care?: Alerts at this stage are common and actionable. Blocking a phishing wave is cheap, cleaning up ransomware is expensive.

13.5. PHASES 5 & 6: INSTALLATION AND C2

The attacker has code running and now wants to keep it that way and talk to it remotely.

Installation: Establish persistence so that the attacker survives reboots and user log-offs. Typical techniques: scheduled tasks, services, registry run keys, WMI subscriptions, startup folders.

Command & Control (C2): A communication channel back to the attacker. Usually HTTPS to a legitimate-looking domain, sometimes DNS tunnelling, increasingly cloud services (Dropbox, Azure blobs, Telegram).

Beaconing: Most C2 traffic beacons: small periodic check-ins, often jittered, that are tiny in volume but constant in rhythm.

What defenders look for: Unusual outbound connections, new persistence mechanisms, EDR process-tree anomalies, DNS to newly registered domains, certificate oddities.

13.6. PHASE 7: ACTIONS ON OBJECTIVES

The attacker does what they came to do. By this point the defender has lost all initiative and is in damage-control mode.

Exfiltration: Steal sensitive data (e.g., customer data, source code, intellectual property)

Destruction or Encryption: Ransomware, deliberate sabotage, deletion of files or database records.

Lateral Movement: Moving to other hosts from inside the compromised network.

Economic Damages: Cost and reputational damage by leaking sensitive data or disrupting availability.

13.7. BREAKING THE CHAIN

Defenders do not need to be perfect. They just need to be earlier than the attacker's next step.

Defense-in-depth mapping: Each control (email filter, EDR, network IDS, identity protection, backup) maps to specific phases. Gaps are visible as phases no control covers.

Dwell time matters: Mandiant M-Trends 2024 reports a global median dwell time of 10 days. That is 10 days for an attacker to move between phases, undetected.

Metric you can measure: Time from phase-1 artefact entering your environment to phase-7 impact. Any additional day you take away from the attacker reduces damage.

Detections, not tools: Buying another appliance without knowing which phase it improves coverage on is expensive theatre.

13.8. KILL CHAIN VS. MITRE ATT&CK

MITRE ATT&CK is a knowledge base of real-world adversary tactics, techniques, and procedures (TTPs), maintained by the MITRE Corporation since 2013. The Kill Chain explains the story. ATT&CK provides the dictionary.

Kill Chain: Seven linear phases, strategic view, great for explaining intrusions and for mapping controls to gaps.

ATT&CK: Matrix of 14 tactics and hundreds of techniques. Every technique is linked to real-world threat groups and malware.

Tactics: Initial Access, Execution, Persistence, Privilege Escalation, Defence Evasion, Lateral Movement, Exfiltration, Impact, and more.

14. INCIDENT RESPONSE

14.1. EVENT VS INCIDENT

Term	*Definition*
event	any observable occurrence in a system or network. A user logs in, a file is created, a packet arrives. Most events are routine.
alert	an automated notification from a security tool that an event matched a detection rule and may warrant investigation.
false positive	an alert or suspected adverse event that turns out to be benign.
adverse event	an event with possibly negative consequences, worth investigating.
incident	a confirmed adverse event that threatens the confidentiality, integrity, or availability of information assets.
data breach	an incident resulting in unauthorised access to or disclosure of protected data.

14.2. ESCALATION PATH

Raw Events (Millions per day): Every log line, packet, and API call. Collected and stored, very rarely looked at individually.

Alerts (Thousands per day): Subset of events that matched a rule, anomaly model, or signature. Mostly noise, misconfiguration or suspicious looking benign activity.

Adverse Events (Dozens per day): Alerts that survived triage and deserve individual investigation. An analyst looks at the context and decides what to do next.

Incidents (A few per year): Classified as actual incidents. IR plan activates, roles are assigned, and the clock starts on containment and recovery.

Breaches (One every few years): Large scale incidents involving data disclosure. IR plan activates and notification obligations apply.

14.3. CLASSIFYING INCIDENTS

14.3.1. Pipkin's Indicators

Donald Pipkin's three-level framework for deciding whether an adverse event is an actual incident. Escalating too early burns the SOC and its analysts out. Escalating too late lets the attacker finish the chain. The framework forces a deliberate, evidence-based call.

Possible Indicators: Weak signals, worth logging but not acting on alone.
Unfamiliar files in user directories, unusual CPU or disk consumption, unexplained system crashes.

Probable Indicators: Strong signals, typically trigger investigation.
Activity at unexpected hours from unusual locations, new accounts, sudden privilege changes, repeated phishing-reaching-inbox reports.

Definite Indicators: Confirmed incidents, activate the IR plan.
Documented malware presence, data exfiltration observed, ransomware encryption in progress, attacker logged in and moving laterally.

14.3.2. Classifier's Dilemma

Every detection rule is a trade-off between missing real attacks (false negatives) and burying the team in noise (false positives).

True Positive: Real malicious activity, alert is raised, and incident is investigated.
Incident contained early, damage minimized.

True Negative: Benign activity correctly ignored. Not measurable in practice.
Nothing happens.

False Positive: Benign activity mis-flagged as malicious.
Analyst wasted time investigating issue.

False Negative: Real attack missed. Often only visible after a post-mortem.
Incident is discovered too late, potentially irreversible damage has been done.

Confusion matrix:

True Positive	False Positive
False Negative	True Negative

14.4. DETECT AND RESPOND

14.4.1. From "Prevent Everything" to "Assume Breach"

The philosophical shift that created modern Detect & Response. Prevention is still essential, but no longer sufficient on its own.

"Prevent Everything" Mindset (until ~2010): Perimeter firewall, AV on endpoints, patching. Assumption: if the perimeter holds, we are safe.

Why It Broke: Laptops leave the office. Attackers live inside trusted networks. Insiders exist. Supply chains are compromised. The perimeter is everywhere and nowhere.

"Assume Breach" Mindset (today): Assume the attacker is already inside some part of the environment. Design for fast detection and fast containment, not just for prevention.

Consequence: Equal investment across Identify, Protect, Detect, Respond and Recover. Resilience is the goal, not impenetrability.

14.4.2. The Metrics That Matter

Detect & Respond is one of the few security domains with good, measurable metrics.

"If you cannot measure it, you cannot improve it."

MTTD (Mean Time to Detect): From initial compromise to first alert. Today's global median is about 10 days A decade ago; it was over 200. (M-Trends Report 2026)

MTTR (Mean Time to Respond): From first alert to incident contained. Good SOCs are in the hours-to-days range, not weeks or days.

Dwell Time (MTTD + MTTR): Total time an attacker was active in the environment. This represents the defender's and attacker's running clock.

14.4.3. Who Actually Does What? SOC, CSIRT, CERT

SOC (Security Operations Centre): The always-on monitoring function. Runs the SIEM, triages alerts, drives day-to- day detection. Typically tiered (T1 triage, T2 investigation, T3 engineering).

CSIRT (Computer Security Incident Response Team): Handles confirmed incidents end-to-end. Often virtual, pulled together when needed, includes IT, legal, communications, management.

CERT (Computer Emergency Response Team): Originally the name of the Carnegie Mellon team. Today used for national or sector-level coordination bodies (e.g., GovCERT.ch, NCSC-CH, US-CERT, ENISA).

In-House vs. Managed: Many organisations outsource SOC functions to an MSSP (Managed Security Service Provider). CSIRT responsibilities usually stay in-house.

14.4.4. The NIST IR Process

1) Preparation
   Policy, plan, team, tools, training. Everything that must be ready before the first alarm fires.
2) Detection & Analysis
   Spot signals, classify them, decide whether to trigger the IR plan.
3) Containment, Eradication & Recovery
   Stop the bleeding, remove the attacker, restore services.
4) Post-Incident Activity
   After-action review, lessons learned, feed back into Preparation.

14.4.4.1. Phase 1: Preparation

All the work that must happen before the first real incident. Unglamorous, easily neglected, always the single biggest factor in how well an IR goes.

People: Named CSIRT roles and on-call rotation. Executive sponsor. Legal and communications liaisons. Clear chain of command.

Process: IR policy approved by management. IR plan with playbooks. Incident categories and severity scoring. Alert roster and escalation paths.

Technology: Logging enabled and centralized. IDS / EDR / SIEM deployed and tuned. Forensic tooling, evidence kits.

Practice: Tabletop exercises at least annually. Technical purple-team exercises. Test the backups (actually restore them).

14.4.5. Phase 2: Detection & Analysis

The phase where Detect & Respond happens. The SOC sees alerts, classifies them, and decides what to do and whether the IR plan fires.

Detection Sources: IDS and IPS and often forgotten direct alerts from users and the helpdesk. Anti-virus on endpoint devices. SIEM correlation rules. Threat intelligence feeds. External reports from peers or regulators.

Triage: For each alert: enrich (asset owner, user context), classify (event vs. incident candidate), score severity, decide to escalate or dismiss.

Key questions during triage: What exactly happened? Which hosts, users, time range? Is this a false positive? Does this match any known indicator of an active campaign?

Handover point: Once an alert is classified as a confirmed incident, it moves from SOC operations to full CSIRT activation.

14.4.6. Phase 3a: Containment

Short-Term Containment

– Isolate affected hosts
– Block malicious IPs through firewall rules
– Disable compromised accounts
– Apply temporary filtering rules

Long-Term Containment

– Network segmentation of affected zone
– Rebuild bastion hosts
– Rotate credentials / keys at scale
– Deploy additional monitoring

14.4.7. Phase 3b & 3c: Eradication and Recovery

Remove the attacker's presence and get the business back online. Carefully, because rushing either step is how organizations get re-compromised.

"If rebuilding is cheaper than investigating, rebuild."

Eradication: Identify Root Cause: How did the attacker get in? Which vulnerability, credential, or misconfiguration was used?

Eradication: Remove Persistence: Possible services, scheduled tasks, registry run keys, malicious OAuth apps, rogue forwarding rules.

Eradication: Rebuild and Patch: Rebuild from stable images / snapshots, patch the root-cause vulnerability, rotate all credentials that could have been exposed, including API keys and certificates.

Recovery: Bring affected systems back in stages. Validate integrity. Monitor closely for reinfection for at least 30 days.

14.4.8. Phase 4: Post-Incident Activity

The step most often skipped, and the most valuable one. An incident you do not learn from is one that will happen again.

After-Action Review (AAR): Everyone involved sits down and walks through what happened. No blame, no finger-pointing. What worked, what did not, what should change. The AAR is written up and shared.

Outputs feed back into Preparation: Updated IR plan and playbooks. New detection rules and signatures. Hardening changes and patches. Training material.

Indicators of Compromise (IOCs): Technical artefacts observed during the incident (hashes, domains, IPs) are documented and often shared with CERTs and peer organizations.

If it is not written down, it did not happen: Documentation is also a legal protection for the organization and its staff.

14.4.9. Common IR Mistakes

Most of these are process and communication problems, not technical ones.

Unclear Chain of Command: Nobody is sure who has the authority to isolate a production system at 3 a.m. The attacker uses the confusion.

No Central operations center: Incident runs out of a chat thread; half the decisions are undocumented. Two weeks later nobody remembers.

Containing too slowly: Fear of "making it worse" means the attacker keeps moving. Damage compounds.

Wiping evidence before Forensics: Rebuilding the affected server before the forensic image is taken. Root cause is now unknowable.

Skipping the after-action Review: Same incident next year. Same surprise. Same mistakes.

No tested Backups: Everyone assumed the backups worked. Ransomware just proved they did not.

14.5. INTRUSION DETECTION SYSTEMS

14.5.1. Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) monitors systems or network traffic, flags suspicious activity, and raises alerts. A burglar alarm, not a lock.

Firewall

– Enforces access-control policies within a network.
– Decides which traffic is allowed to pass between network zones and which isn't.
– Operates on basic allow / deny rules.
– Usually operated by network teams or admins.

Intrusion Detection System

– Detective / monitoring device that inspects traffic or host activity.
– Raises alerts when unusual or potentially malicious behavior is detected.
– It does not block anything; human action is required when an alert is raised.
– Usually operated by the SOC or security analysts.

14.5.2. Intrusion Prevention System (IPS)

When an IDS is placed inline with the traffic and is given the authority to actively block or modify malicious activity it becomes an Intrusion Prevention System (IPS).

– Sits inline in the data path in the network.
– Can automatically interrupt or block malicious traffic.
– Acts faster / instantly but a false positive now means downtime for real users.
– Usually operated by the SOC and networking teams together.

14.5.3. Signature-Based Detection

Match traffic or behavior against a known pattern. The classical approach, still the backbone of most commercial IDS products.

How it works: Each known attack has a fingerprint; a byte sequence, a URL pattern, a command string. The IDS compares every packet or event against a predefined rule set and alerts on matches.

Pro	*Contra*
– Low false-positive rate. – Alerts are precise and actionable. – Rule feeds are easy to share and update (Emerging Threats, ET Pro, vendor feeds).	– Blind to zero-days and unknown new threat variants. – Easily bypassed by small mutations (obfuscation, packing, custom protocols). – Rule sets grow large, performance and maintenance cost rises.

14.5.4. Anomaly-Based Detection

Learn what "normal" looks like, then alert on deviations. The only technique that has any chance of catching truly novel attacks.

How it works: Build a baseline of normal behavior; typical volumes, login times, process trees, DNS query patterns. Statistical or ML models flag observations that fall outside the baseline.

Pro	*Contra*
– Can detect novel attacks and zero-days. – Catches insider threats and credential abuse that signatures miss. – Adapts to the specific environment.	– High false-positive rate during training and after legitimate changes (new software, new users, seasonal patterns, etc.). – Hard to explain alerts. “The model says so” is rarely convincing at 03:00 in the morning.

14.5.5. Intrusion Detection in Practice

Real-world IDS products usually blend approaches. Nobody ships a pure-signature or pure- anomaly product anymore.

Stateful Protocol Analysis: An IDS understands networking protocols (SMB, HTTP, DNS, TLS). It alerts when a session violates the specification, even without a specific signature.

Heuristic and Behavioral Rules: Rules that describe suspicious sequences (e.g., MS Word spawning PowerShell that writes to AppData folder).

AI-Assisted Classification: Supervised models trained on labelled samples (phishing URLs, beacon profiles), often bolted onto signature engines.

Hybrid in Practice: Most real-world products blend all the above. Suricata does signatures and protocol analysis. An EDR does behavior as well as AI and telemetry.

14.6. SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM)

14.6.1. Why Do We Need a SIEM?

Every device, service and app produces logs. Someone must read them.

The Volume Problem: A mid-size company easily produces 20+ GB of logs per day from 40+ distinct sources. No human reads all of that.

The Correlation Problem: Interesting attacks touch several systems in sequence. Looking at any single system misses the pattern.

The Retention Problem: Forensics often needs logs from 90+ days ago. Most systems rotate logs out in days.

The Compliance Problem: PCI-DSS, ISO 27001, FINMA, NIS2 and friends all demand that security logs exist, are reviewed, and are retained.

14.6.2. SIEM Architecture

1) Sources: Endpoints, firewalls, servers, identity providers, cloud platforms, SaaS applications, IDS / IPS, EDR.
2) Collectors: Agents, syslog receivers, API-pull connectors, log forwarders. Responsible for getting data off the source and into the SIEM reliably.
3) Parse & Normalize: Map every log into a common schema (CIM for Splunk, ECS for Elastic). A Windows login and a Cisco login become comparable.
4) Correlate & Store: Detection rules, ML models, search index, long-term data lake. The core of the product.
5) Consume: Alerts to the SOC, dashboards for leadership, search for analysts, reports for auditors.

14.6.3. Correlation: Turning Noise Into a Story

Correlation rules express the attacker's behavior across sources and time. One event is a log line. Three in the right order might be an incident.

Example scenario: Possible account compromise: 10 failed VPN logins for user X, followed by a successful VPN login, followed by a login from a foreign IP, followed by a large SharePoint download within 15 minutes.

The rule encodes the sequence: Nothing here would fire a useful alert alone. Every event is common. Chained in that order, within that window, the pattern is suspicious.

Written in the SIEM's query language: Splunk SPL, Kusto (Microsoft Sentinel), Elasticsearch DSL, Sigma (open, vendor-agnostic).

Detection engineering is its own discipline: Writing, tuning, and retiring correlation rules is a full-time job. Good detection engineers are the scarce resource, not the tool.

14.6.4. What SOCs use a SIEM for

Detection is the headline, but the other uses are what usually pays for the tool.

Threat Detection: Real-time alerts on known and learned attack patterns. The reason SIEMs exist.

Incident Investigation: When an alert fires, the SIEM is where analysts pivot across logs to reconstruct what happened.

Threat Hunting: Hypothesis-driven search through historical data. "Have we seen this IOC anywhere in the last 90 days?"

Compliance Reporting: Proof that logs exist, are reviewed, and are retained. Auditors love canned SIEM reports.

UEBA (User & Entity Behavior Analytics): Baselines "normal" per user and flags outliers. Often a paid add-on module.

Fraud & Insider Risk: Finance, HR and legal teams are sometimes internal customers of the SIEM too.

14.6.5. SIEM vs. SOAR vs. XDR

SIEM: Collect, normalize, correlate, alert. Broad scope across all log sources. Detection-heavy. Human response.

SOAR (Security Orchestration, Automation and Response): Sits on top of the SIEM. Runs playbooks: isolate host, disable account, open a ticket, enrich indicators, notify legal.

XDR (Extended Detection and Response): Vendor-bundled detection across endpoint, network, identity, email and cloud. Often sold as a lighter alternative to SIEM+SOAR, at the cost of vendor lock-in.

In real enterprises: SIEM + SOAR is the classic stack. XDR is popular in smaller organizations and in vendor-aligned ones that have standardized on one provider.

14.6.6. Where SIEM Projects Fail

Alert Fatigue: Too many low-quality alerts. Analysts stop reading. The one that mattered was in there, somewhere.

Garbage In / Garbage Out: If logs are incomplete, mis-parsed, or missing timestamps, no correlation rule can save you. Log quality is the foundation.

Ingestion Cost: Most commercial SIEMs price per GB per day. "Log everything" quickly becomes unaffordable, so you have to be deliberate about what to collect.

Tuning Debt: Rules that fit last year's environment do not fit this year's. Someone has to own them and retire what is no longer useful.

Skills Gap: Running a SIEM well needs detection engineers and threat hunters, not just "log admins". That is a scarce role.

15. OPEN-SOURCE INTELLIGENCE (OSINT)

The process of collecting, analyzing, and making decisions based on information accessible to the general public. It's the structured processing of raw data into actionable intelligence.
– Information are from media, social networks, forums, government data, and commercial datasets.

15.1. TOOLS AND FRAMEWORKS

– Centralized platform for tools categorized by data type (IP, Email, Social Media etc.)
– Shodan /Censys - Searching for exposed infrastructure.
– Maltego - Link analysis and data visualization.
– SpiderFoot - Data Collection Tool for Public Information

15.2. CYBER THREAT INTELLIGENCE (CTI)

– Systematic collection, processing, and analysis of information about threats.
– The goal is to enable a proactive rather than reactive response to attacks.
– It transforms raw data (e.g., from OSINT) into actionable intelligence.
– It supports decision-making by identifying relevant threats, actors, and their methods (TTPs1).
– Typical sources include Incident Response, Threat Intelligence Reports, Malware Analysis, and standardized frameworks like MITRE ATT&CK

15.2.1. Sources for Threat Intelligence

External Sources:
– Regular feed of threat data from a threat intelligence subscription service
– Cyberintelligence vendors whose services can be employed
– Many of the sources of vulnerability information, such as National CERTs
– Information sharing and analysis centers

Internal Sources:
– Event logs from technical infrastructure, such as operating system logs
– Alerts from security systems such as firewalls, malware protection, DLP, network-based intrusion detection systems (NIDSs), gateway proxy servers, and physical security systems
– Direct feeds from security event management utilities, such as those produced by security event logging software or a security information and event management (SIEM) system
– Dedicated teams that perform information security–related activities

15.2.2. The 3 Levels

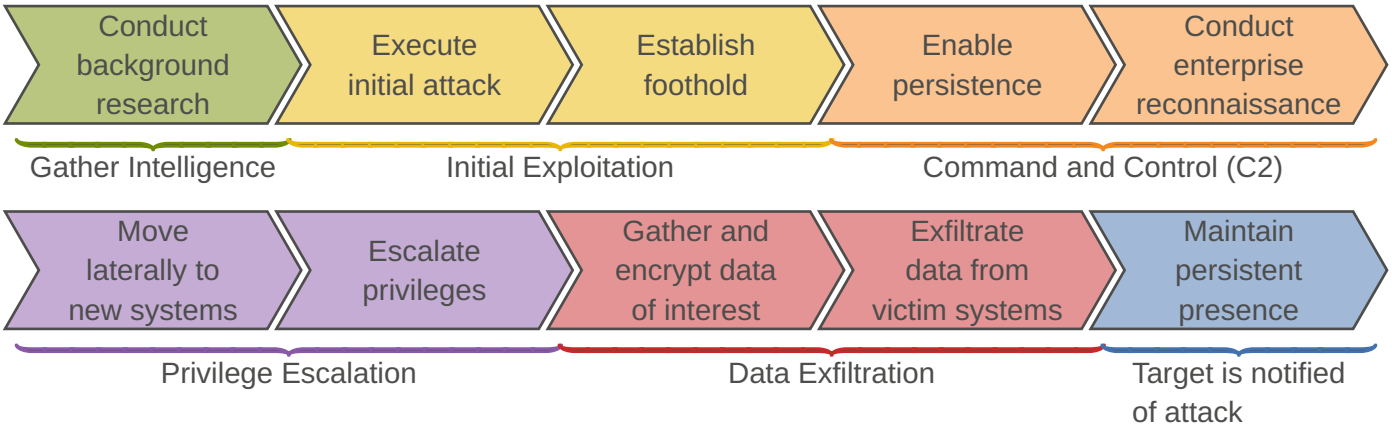
1) Strategic Level for Executives & Management
   – Who is attacking and why?
   – Example: Situation reports on state-sponsored APT 1 groups targeting a specific industry
2) Operational Level for SOC Teams & Analysts
   – How does a specific attack unfold?
   – Example: TTPs of a known APT mapped to MITRE ATT&CK
3) Tactical Level for SIEM systems & Firewalls
   – Which concrete indicators do I need to block?

– Example: Malicious IP, Domains, File hashes, Malicious signatures

15.2.3. Advanced Persistent Threat (APT) Attack

– A network attack in which an unauthorized person gains access to a network and stays there, undetected, for a long period of time.
– The intention of an APT attack is to steal data rather than to cause damage to the network or organization.
– APT attacks target organizations in sectors with high-value information, such as national defense, manufacturing, and the financial industry. APTs differ from other types of attacks in their careful target selection and persistent, often stealthy, intrusion efforts over extended periods

15.2.3.1. APT Attack Lifecycle



16. ETHICAL HACKING

Term	*Definition*
Hacking	– Exploiting vulnerabilities in systems and/or software to gain unauthorized access – Security control compromise – Produce behaviours outside of system/software's original intent
Ethical Hacking	– Using tools and techniques to validate, audit and report on system/software vulnerabilities – Vulnerability existence reporting

16.1. TAXONOMY BY ETHICAL INTENT

Term	*Definition*
Black Hat	Malicious, destructive hacker that usually remains anonymous
Grey Hat	Those possessing Black hat skills who focus on both offense and defense
White Hat	Those possessing black hat skills who primarily focus on defense

16.2. TAXONOMY BY SKILLS, MOTIVATION, ORGANIZATIONAL AFFILIATION

Term	*Definition*
Script Kiddie	Individuals that use tools without understanding what they are doing
Cyber Terrorist	Skilled attacker whose purpose it to further an ideology
State Sponsored	Hackers employed by the government for both offensive and defensive activities

Term	*Definition*
Hacktivist	A person who breaks into a computer system in order to pursue a political or social aim

16.3. INDEPENDENT ETHICAL HACKING

– HackerOne is the leading global crowdsourced security platform.
– Bugcrowd is a major crowdsourced security platform combining global security researchers.
– BugBounty is Switzerland's leading AI-powered security testing platform

16.4. PENETRATION TESTING VS VULERABILITY SCANNING

Term	*Definition*

Penetration testing	– Manual process
	– Cybersecurity professional tries to uncover weaknesses
	  – find a way to break into your system
	  – In-depth analysis

Vulnerability scanning	– Automates process
	– Periodic scans
	– First step performed by penetration testers
	  – determine the overall state of your systems
	  – trigger in-depth manual reviews

16.4.1. Contractual Framework for pentesting

– Pen Testing Contracts
– Statement of Work (SoW)
  – Activities to be performed
  – Pen testing timeline
  – Scope
  – Location of the work
  – SoW can be a standalone document or part of a Master service agreement (MSA)
– Non Disclosure agreement (NDA)
  – Types of NDAs include Unilateral, Bilateral, and Multilateral agreements

16.4.2. Frameworks and Methodologies

16.4.2.1. Penetration Testing Execution Standard (PTES)

– Community-driven industry standard
– End-to-End Penetration Testing Methodology
– https://pentest-standard.readthedocs.io/_/downloads/en/latest/pdf/

16.4.2.2. NIST 800-115

– Technical Guide to Information Security Testing and Assessment
– Government-oriented testing guidance
– Technical guide to information security testing and assessment
– https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

16.4.2.3. EC-Council

– The EC-Council is a global organization providing cybersecurity certifications such as the Certified Ethical Hacker
– https://www.eccouncil.org/

16.4.3. Pentesting approaches based on knowledge level

Term	*Definition*
Black-box Testing	– No internal knowledge – External attacker perspective – Simulates external threat actors
Gray-box Testing	– Limited knowledge – Realistic authentication – Insider-based threat modeling
White-box Testing	– Full internal knowledge – Deep analysis across all system components – Complete documentation access

17. MALICIOUS CODE

Malicious code refers to software-based security threats that exploit weaknesses in networks, operating systems, or applications to deliver harmful payloads to targeted systems.

– User-Dependent Malware: Most computer viruses and Trojan horses rely on user interaction or unsafe behavior to propagate from one system to another.
– Self-Replication Threats: Worms, by contrast, are self-replicating and spread autonomously across vulnerable systems without requiring human involvement.

17.1. BASIC FUNCTIONS OF COMPUTER VIRUSES

– As with biological viruses, computer viruses have two main functions, propagation and payload execution
– According to the AV-Test Institut, one of the major antivirus software vendors, >450,000 new malware variants appear on the internet every day!
– The propagation function defines how the virus will spread from system to system.
  – Viruses use new and innovative methods to escape detection and bypass increasingly sophisticated antivirus technology.
– The malicious impact is delivered by the virus's payload, which executes the attacker's intended malicious activity.
  – This could be anything that negatively impacts the confidentiality, integrity, or availability of systems or data
– Anyone with a minimal level of technical expertise can create a virus and unleash it upon the internet. GO CRAZY!!1!1 (be sure to target winbloat only)

17.2. DRIVE-BY DOWNLOADS: PASSIVE MALWARE INFECTION

– A drive-by download refers to the unintentional download of malicious code to your computer or mobile device that leaves you open to a cyberattack
  – You don't have to click the download button or open a malicious email attachment to become infected
  – A drive-by download can take advantage of an app, operating system, or web browser that contains security flaws due to unsuccessful updates or lack of updates.
– Unlike many other types of cyberattack, a drive-by doesn't rely on the user to do anything to actively enable the attack
  – Drive-by downloads may happen when visiting a website or clicking a link, or clicking on a pop-up window
– Example: Lumma Stealer

  - It is Malware-as-a-Service (MaaS) that specializes in exfiltrating sensitive data such as login credentials, credit card information, and cryptocurrency wallet details from infected Windows systems

17.3. ZERO-DAY EXPLOITS AND UNKNOWN VULNERABILITIES

- A Zero-day attack exploits a Zero-day vulnerabilities
  - Zero-day vulnerabilities are security flaws discovered by hackers that have not been thoroughly addressed by the security community
- A Zero-day exploit is not known to the software vendors
  - The delay between the discovery of a new type of malicious code and the issuance of patches and antivirus updates is known as the window of vulnerability
- Lots of system are long vulnerable to Zero-day attacks because of the slowness in applying updates on the part of system administrators

17.3.1. APT the Use of Zero-Day Exploits

- APTs are sophisticated adversaries with advanced technical skills and significant financial resources
  - These attackers are often military units, intelligence agencies, or shadowy groups that are likely affiliated with government agencies
- One of the key differences between APT attackers and other malware authors is that these malware developers often have access to zero-day exploits that are not known to software vendors
  - Because the vendor is not aware of the vulnerability, there is no patch, and the exploit is highly effective
  - Malware built by APTs is highly targeted, designed to impact only a small number of adversary systems and difficult to defeat.
  - Example: APT Case RUAG, Stuxnet

17.4. MECHANISMS OF VIRAL PROPAGATION

By definition, a virus must contain technology that enables it to spread from system to system. Once the virus has “touched” a new system, they use one of several propagation techniques to infect the new victim and expand their reach

- Traditional Propagation Techniques
  - Master Boot Record (MBR) infection
  - File infection
- Modern Propagation Techniques
  - Macro infection
  - Script-based infection
  - Process / Services injection
  - Fileless techniques

17.4.1. Master Boot Record Infection

- The Master Boot Record (MBR) viruses attack the portion of bootable media that the computer uses to load the operating system during the boot process (hard disk, USB)
- The MBR doesn’t contain all the code required to implement the virus’s propagation and destructive functions
  - The MBR is extremely small (usually 512 bytes)
- MBR viruses store the majority of their code on another portion of the storage media
  - The system reads the infected MBR
  - The virus instructs it to read and execute the code stored in this alternate location
  - The system loads the entire virus into memory

17.4.2. File Infection

- The file infector viruses infect executable files
- These viruses are often self-contained executable files that escape detection by using a filename similar to a legitimate operating system file (also called companion virus)
- Standard file infector viruses are often easily detected
  - by comparing file characteristics such as size and modification date before and after infection
  - by comparing hash values

17.4.3. Service Injection

- The Service Injection viruses inject themselves into trusted runtime processes of the operating system, such as svchost.exe, winlogin.exe, and explorer.exe
- The malicious code is able to bypass detection by any antivirus software running on the host because those processes are trusted.

17.4.4. Macro Infection

- A macro virus is a virus that is written in a macro language, a programming language which is embedded inside a software application word processors and spreadsheet applications
  - Macro programs are embedded in documents and the macros are run automatically when the document is opened
  - Macro viruses proliferate because of the ease of writing code in the scripting languages (such as VBA)
- Macro viruses first appeared on the scene in the mid-1990s to infect documents created in the popular Microsoft Word environment
  - In 1999, the Melissa virus spread through the use of a Word document that exploited a security vulnerability in Microsoft Outlook to replicate
  - The I Love You virus exploited similar vulnerabilities in early 2000
- Software developers made important changes to the macro development environment, restricting the ability of untrusted macros to run without explicit user permission
  - This resulted in a drastic reduction in the prevalence of macro viruses

17.4.5. Fileless Techniques

- Fileless Execution : Operates entirely within system memory, leaving no malicious code on disk. This makes it inherently resistant to traditional signature-based detection.
- Memory-Resident Execution: The malicious payload resides and operates exclusively in RAM.
- Living-off-the-Land (LotL): Abuse of built-in Windows tools like PowerShell or WMI, without using external binaries. Since these tools are trusted by the OS, their abuse often remains undetected.
- Registry-Based Persistence: Encoded shellcode or scripts are concealed within Windows Registry keys. Upon system startup, a native loader (e.g. regsvr32.exe) silently retrieves and executes the payload directly in memory.

17.5. MALWARE TECHNOLOGIES

17.5.1. Multipartite Viruses

- Multipartite viruses use more than one propagation technique in an attempt to penetrate systems that defend against only one method or the other.
- Example: The Marzia virus
  - It infects the command.com system file qualifying it as a file infector virus
  - 2 hours after the first infection, it writes malicious code to the system’s master boot record qualifying it as a boot sector virus.

17.5.2. Stealth Viruses

- Stealth viruses hide themselves and tool antivirus packages into thinking that everything is functioning normally.
- Example: A stealth boot sector virus might overwrite the system's MBR with malicious code and modify the operating system's file access functionality.
  - When the antivirus package requests a copy of the MBR, the modified operating system code provides it with a clean version of the MBR free of any virus signatures. When the system boots, it reads the infected MBR and loads the virus into memory.

17.5.3. Polymorphic Viruses

- Polymorphic viruses modify their own code as they travel from system to system.
  - The propagation and destruction techniques of the virus remain the same
  - The signature of the virus is somewhat different each time it infects a new system
- This constantly changing signature will render signature-based antivirus packages useless
  - Antivirus vendors have "cracked the code" of many polymorphism techniques and are able to detect known polymorphic viruses
  - It takes vendors longer to generate the necessary signature files to stop a polymorphic virus

17.5.4. Encrypted Viruses

- Encrypted viruses use cryptographic techniques to avoid detection
- Encrypted viruses use a very short segment of code known as the virus decryption routine
  - It contains the cryptographic information necessary to load and decrypt the main virus code stored elsewhere on the disk
- Each infection utilizes a different cryptographic key, causing the main code to appear completely different on each system (polymorph)
  - The virus decryption routines often contain signatures that render them vulnerable to updated antivirus software packages

17.5.5. Logic Bombs

- Logic bombs are malicious code objects that infect a system and lie dormant until they are triggered by the occurrence of one or more conditions such as time, program launch, website logon
- Example: The Michelangelo virus
  - It infected a system's MBR
  - It hid itself until March 6 – the birthday of the famous Italian sculptor Michelangelo
  - On that date, it reformatted the hard drives of infected systems and destroying all the data they contained
- Example: a logic bomb in South Korea in March 2013.
  - This malware infiltrated systems belonging to South Korean media companies and financial institutions and caused both system outages and the loss of data

17.5.6. Trojan Horses

- A Trojan horse is a software program that appears "kind" but carries a malicious, behind-the-scenes payload
- Example: a Rogue antivirus software
  - This software tricks the user into installing it by claiming to be an antivirus package (using a pop-up ad that mimics the look and feel of a security warning)
  - Once the user installs the software, it either steals personal information or prompts the user for payment to "update" the rogue antivirus
  - The "update" simply enables the Trojan!

17.5.7. Keystroke logging

- It is the action of recording (logging) the keys struck on a keyboard
  - The person using the keyboard is unaware that their actions are being monitored
- Data can then be retrieved by the person operating the logging program
  - A keylogger can be either software or hardware
  - Keyloggers are most often used for stealing passwords and other confidential information

17.5.8. Ransomware

- Ransomware infects a target machine and then uses encryption technology to encrypt files stored on the system with a key known only to the malware creator.
  - The user is unable to access their files and receives a pop-up message warning that the files will be permanently deleted unless a ransom is paid within a short period of time.
  - The user then often pays this ransom to regain access to their files.
- Examples: Cryptolocker, WannaCry, Petya, Nyetya, BABUK (2021)

17.5.9. Worms

- Worms are malicious code objects that propagate themselves without requiring any human intervention
- Example: Code Red Worm. Code Red performed three malicious actions on the systems it penetrated:
  - It seeks many new targets by randomly selected hundreds of Internet Protocol (IP) addresses and then probed those addresses to see whether they were used by hosts running a vulnerable version of IIS
  - It defaced HTML pages on the local web server
  - It planted a logic bomb that would initiate a denial-of-service attack against the IP address `198.137.240.91`, which at that time belonged to the web server hosting the White House's home page
- Example: Stuxnet
  - Stuxnet was searching for systems using a controller manufactured by Siemens and used in the production of material for nuclear weapons
    - When it found such a system, it executed a series of actions designed to destroy centrifuges attached to the Siemens controller
  - Stuxnet appeared to begin its spread in the Middle East, specifically on systems located in Iran in 2010.
    - It is alleged to have been designed by Western nations with the intent of disrupting an Iranian nuclear weapons program.
  - Stuxnet marks two major evolutions in the world of malicious code
    - the use of a worm to cause major physical damage to a facility
    - the use of malicious code in warfare between nations.
  - can you hear the sound of freedom eagles

17.5.10. Spyware & Adware

- Spyware monitors your actions and transmits important details to a remote system that spies on your activity
- Adware uses a variety of techniques to display advertisements on infected computers
  - The simplest forms of adware display pop-up ads on your screen while you surf the web
  - More advanced versions may monitor your shopping behavior and redirect you to competitor websites
- Adware often take advantage of third-party plug-ins to web browsers, to spread their malicious content
  - The original plug-in code is supplemented with malicious code that spreads malware, steals information, or performs other unwanted activity
- Example: Windows

17.6. ANTIVIRUS & ENDPOINT SECURITY

- An Antivirus software is a computer program used to prevent, detect, and remove malware

  - If possible, the antivirus package eradicates the virus, disinfects the affected files and restores the machine to a safe condition
  - If the software doesn't know how to remove the virus, the files can be quarantined. A common strategy is to send the suspicious files to a sandbox where they are executed in an isolated but monitored environment
  - If the software doesn't know how to remove the virus and cannot be quarantined, the infected files can be deleted in an attempt to preserve system integrity
- Antivirus solutions are not only protecting systems from viruses
  - These tools are often able to provide protection against worms, Trojan horses, logic bombs, rootkits, spyware,…

17.6.1. Antivirus Detection: Signature-based

- The vast majority of the antivirus packages utilize a method known as signature-based detection
  - An antivirus package maintains an extremely large database that contains the characteristics of all known viruses
  - The antivirus scans storage media periodically
- The signature-based antivirus package is only as effective as the virus definition file upon which it's based
  - Your antivirus software will not be able to detect newly created viruses
  - An outdated definition file will quickly render your defenses ineffective

17.6.2. Antivirus Detection: Heuristic-based

- The antivirus analyze the behavior of software, looking for the signs of virus activity.
  - Such as attempts to elevate privilege level, coverage of electronic tracks, and alteration unrelated or operating system files
- If the software behaves suspiciously in that environment, it is added to blacklists throughout the organization, rapidly updating antivirus signatures

17.6.3. Antivirus Detection: Data integrity

- Data integrity antivirus functionality is designed to alert administrators to unauthorized file modifications
  - Unless a new software, application of an operating system patch has been installed, sudden changes in executable files may be a sign of malware infection
- These systems work by maintaining a database of hash values for all files stored on the system
  - These archived hash values are then compared to current computed values to detect any files that were modified between the two periods

17.6.4. Endpoint Security

- Deep Visibility & Continuous Monitoring
  - EDR tools capture all system events, including process executions, network connections, and registry changes.
- Behavioral Analysis
  - Detection of fileless threats and complex attack chains that bypass traditional signature-based antivirus software.
- Automated Response and Remediation
  - When anomalous behavior is detected, the infected machine is automatically isolated from the network to prevent lateral spread.
- Example: Microsoft Defender for Endpoint, CrowdStrike Falcon, SentinelOne

18. OT AND IT/OT

OT: Operational Technology, software and hardware for monitoring …

Comparison	*IT*	*OT*
Lifecycle	3-5 yrs	15-20yrs
Security	Data and cyber security	Operational and physical security
Update-frequency	Frequent software-patches	Rarely
Availability	High but with maintenance-times	Permanent

19. THE FUTURE OF CYBERSECURITY

19.1. TRENDS

19.1.1. Social Engineering

The most attacks are still without any code. New shift: Vishing (rep: Voice + Phishing) seems to be used more often.

– Attacks on IT-Helpdesks: Pretending to be a user and trying to reset Password + MFA
– Attackers pretend to be IT-Helptesk (often with multiple calls!)

19.1.2. Identity Theft

The Theft Resource Center states three trends in Identity Theft:

1) Artificial Intelligence (AI) technology makes it easier for thieves to coerce unsuspecting victims into giving away their identity credentials
2) Identity thieves are increasingly able to access a variety of existing accounts.
3) Individuals are becoming more curious about protecting their identity.

19.1.3. Geopolitical Tensions / Conflict

Due to geopolitical tensions, threat actors remained persistent and adaptive in 2025 and evolve throughout 2026.

19.1.4. Expansion of Supply Chain Attacks

Supply Chain Attacks have doubled in 2025.

– ENSIA Foresight 2030: Supply Chain-Attacks are classified as the most critical threat
– EU NIS-2: Commits the operators of critical infrastructure to protect the whole supply chain

19.2. POST-QUANTUM CRYPTOGRAPHY

19.2.1. Quantum Computer

– Using qubits instead of classical bits
  – Superposition of states
– Can solve certain problems exponentially faster than classical computer
  – No generic speedup
  – Good in breaking Factoring and Discrete Logarithm, still bad in breaking Hashing
– Not practical (yet)
  – Current systems have a few qubits
  – E.g., Google Willow has 105 qubits
  – Challenge: Error correction

19.2.2. Affected Components

Vulnerable: Asymmetric Cryptography

– Signatures: RSA, ECDSA, EdDSA
– Key Exchange: Diffie-Hellman

Less Impacted: Symmetric Cryptography
– Block/Stream Cipher
– Hash functions (HMAC)

→ Double keys

19.2.3. New NIST Standards

– NIST defines cryptographic standards
– PQC NIST competition
  – 2017: 90 candidates submitted
  – 2024: 3 schemes standardized
– New Standards
  – Signatures
    – ML-DSA (CRYSTALS-Dilithium)
    – SLH-DSA (Sphincs+)
  – Key Encapsulation
    – ML-KEM (CRYSTALS-Kyber)

19.2.4. Transition to PQC

Until 2028:
– Define necessary policies/goals (regulations)
– Create a cryptographic inventory
– Build an initial plan for migration

Until 2031:
– High-priority PQC transitions complete
– Refine plan for remaining migration

Until 2035:
– Complete migration to PQC complete

19.3. IOT SECURITY

19.3.1. Elements of IoT

– Connectivity
  – Communications protocols, protocol stacks.
– Things += Microcontroller + SW + Sensors + Actuators
– Zero-Touch Provisioning
– Security
  – CIA, Authentication & Authorization

19.3.2. Growing Attack Surface

19.3.2.1. Explosive Device Growth

– Over 21 billion IoT devices worldwide (2025)
– Trend rising sharply
– Often insufficiently protected

19.3.2.2. Typical Vulnerabilities

– Default passwords

– Missing or inadequate encryption
– Outdated firmware
– Lack of update mechanisms

19.3.2.3. Attack Vectors

– Botnets
– Mirai (2016) is arguably the most famous IoT botnet
– Man-in-the-Middle
– Supply Chain Attacks

19.3.2.4. Cyber Resilience Act (CRA)

– CRA is being implemented in stages (until 2027)
– EU-wide regulation that establishes binding cybersecurity requirements for hardware and software products
– Outdated firmware
– Lack of update mechanisms

19.3.3. IoT security risks and design flaws

– IoT devices often lack security-by-design due to cost and time constraints.
– Weak authentication, insecure updates, and exposed interfaces are common.
– Complex dependencies between devices, networks, and cloud services increase risk.
– Design flaws can compromise entire IoT ecosystems, not just individual devices.

19.3.4. Threat examples

19.3.4.1. Service Disruption

– Description: IoT services or devices are rendered unavailable or unreliable.
– How it works:
  – Denial-of-Service (DoS/DDoS) against devices or gateways
  – Resource exhaustion on constrained devices (CPU, memory, battery)
  – Malicious firmware updates or protocol abuse
– Domains:
  – Smart household appliances
  – Industrial IoT systems
  – Safety-critical components (vehicle braking subsystems, medical devices)
– Impact: Loss of availability and reliability, economic damage

19.3.4.2. Unauthorized Control

– Description: Attackers gain active control over IoT devices or actuators and can issue malicious commands.
– How it works:
  – Exploitation of weak authentication (default credentials, missing mutual authentication)
  – Software/firmware vulnerabilities (buffer overflows, insecure update mechanisms)
  – Compromised communication channels (unencrypted or unauthenticated protocols)
– Domains:
  – Smart homes (locks, lighting, HVAC)
  – Connected vehicles (infotainment, driver assistance subsystems)
  – Medical devices (insulin pumps, implanted or external controllers)
– Impact: Loss of safety and physical integrity

19.3.5. Comparing IT, IoT, and OT Environments

Feature	*IT Security (Office/IT)*	*IoT Security (Connected)*	*OT Security (Industrial)*
CIA Priority	C > I > A	I > C > A	A > I > C
Primary Goal	Confidentiality (Data Protection)	Data Integrity & Device Safety	Availability (Process Protection)
Asset Focus	Servers, Laptops, Databases	Smart Devices, Trackers, Cameras	PLC, Motors, Robots, Sensors
OS Type	Windows, Linux, MacOS (Standard)	Embedded / Lightweight Linux	Real-time (RTOS), Proprietary
Lifecycle	3–5 Years (Short-lived)	2–7 Years (Fast-paced)	15–30 Years (Legacy Systems)
Patching	Regular (e.g. “Patch Tuesday”)	OTA (Over-the-Air) targeted	Rare (only during downtime)
Failure Impact	Data loss, Reputation damage	Mass manipulation, Botnets	Physical damage, Risk to life

19.3.6. The Future of IoT Security: Strategies & Trends

- Security-by-Design
  - Integrate security starting from the development phase, rather than adding it as an afterthought.
- Zero Trust Architectures
  - No device is automatically trusted →continuous authentication is required.
- Automated Patch Management & Standardization
  - OTA (Over-the-Air) updates as a standard to close security vulnerabilities.
  - Use of open standards, such as Matter.
- Compliance with standards:
  - ETSI EN 303 645: The leading European standard for the cybersecurity of consumer IoT devices.
  - IEC 62443: A series of standards for the cybersecurity of industrial automation and control systems (IIoT).

19.4. CYBERSECURITY AND AI

AI Tools enable attackers to generate Phishing Mails, Malware Code, Deepfakes, etc. Also, companies integrate AI-Systems, which come with new vulnerabilities.

19.4.1. Claude Mythos

- Announced in April 2026 by Anthropic, the developers of Claude AI.
- A model that is strikingly good at cyber security tasks such as finding zero-day exploits.
- Project Glasswing aims to strengthen existing cybersecurity systems across 40 organizations, including Amazon, Apple, Google and Microsoft, with the help of Mythos.
- Can be used by defenders to strengthen their systems but also by attackers to more effortlessly and effectively write malware and abuse zero-day exploits.