Cyber Security | CySec

Zusammenfassung

1. CYBER SECURITY FOUNDATIONS

1.1. THE ASSETS OF AN ORGANISATION

An asset is ***anything*** within an environment that ***should be protected***. The loss or disclosure of an asset could result in:

- An overall security compromise
- Loss of productivity
- Reduction in profits
- Additional expenditures
- Discontinuation of the organization
- Numerous intangible consequences

1.1.1. Different Types of assets of an organization

- ***Information:*** includes all its its data
- ***Systems:*** includes any information technology (IT) systems that provide one or more services
- ***Devices:*** refers to any computing system
- ***Facilities:*** includes any physical location that it owns or rents
- ***Personnel:*** People working for an organization are also a valuable asset

1.1.2. Intellectual Property

Intangible (immateriell) assets like brand names.

1.1.2.1. How to protect intellectual property

- ***Copyright ©:*** Protection against unauthorized duplication of work. Computer software falls under literary works, but only the actual source code is protected. It is not necessary to officially register a copyright, you just need to be able to prove in court that you were the creator of a work. For example, send your work as a registered letter to yourself and keep it unopened. Protected until 70 years after the death of the last surviving author. Does not count for "works for hire".
- ***Trademarks:*** Words, slogans and logos. Not registered: ™, Registered: ®
- ***Patents:*** Protect intellectual property rights. They provide a period of 20 years during which the inventor is granted exclusive rights. After that, it is available for anyone to use. The invention must be new, useful and not obvious. Does not provide adequate protection for computer software.
- ***Trade Secrets:*** Because copyright and patents applications require you to publicly disclose the detail of your work, this removes the "secret" nature of your property. They also provide protection only for a limited period. Because of that, trade secret is one of the best way to protect computer software.

1.2. DATA CLASSIFICATION

Used to determine how much effort, money and resources are allocated to protect the data. It is inefficient to treat all data the same way. There are three data states:

Data at Rest *stored*, ***Data in Transit*** *transmitted over a network*, ***Data in Use*** *data in memory or storage buffer*

1.2.1. Sensitive data

- ***Personally Identifiable Information (PII):*** Any information that can identify an individual. Organizations need to notify individuals if a data breach results in a compromise of PII. *Examples: Name, social security number, biometric records*
- ***Protected Health Information (PHI):*** Any health-related information that can be related to a specific person. *Examples: Any information from the past, present or future that connects to the health of a specific person*

- ***Proprietary Data:*** Any data that helps an organization to maintain a competitive edge. *Examples: Software code, technical plans, internal processes, trade secrets.*

1.2.2. Government/military classification

		The unauthorized disclosure of top-secret data will have …
High	***Top secret***	***drastic effects*** and cause ***grave damage*** to national security.
	Secret	***significant effects*** and cause ***critical damage*** to national security
	Confidential	***noticeable effects*** and cause ***serious damage*** to national security.
	Sensitive but unclassified	Sensitive but unclassified is used for data that is for internal use.
Low	***Unclassified***	

1.2.3. Commercial business/private sector classification

High	***Confidential / Private***	***Confidential*** is used for data that is ***extremely sensitive*** and for internal use only. If proprietary data is disclosed, it can have ***drastic effects*** on the ***competitive edge*** of an organization. ***Private*** is used for data that is of a ***private or personal nature*** and intended for ***internal use only***.
	Sensitive	Sensitive is used for data that is ***more classified than public data***.
Low	***Public***	This is used for all data that ***does not fit in one of the higher classifications***.

1.2.4. Destroying sensitive data

- ***Erasing:*** Just delete the data (like removing an entry in a phone book). Easily retrievable.
- ***Clearing:*** Overwriting of data with unclassified data
- ***Purging:*** Repeats the clearing process multiple times and combines it with another method such as degaussing.
- ***Degaussing:*** Strong magnetic field that erases data on magnetic storage media.
- ***Destruction:*** In a way that the media cannot be reused or repaired like incineration, crushing, shredding, disintegration and dissolving.

1.2.5. Tracing or hiding sensitive data

- ***Steganography:*** Embedding a message within a file
- ***Watermarking:*** Embedding an image or pattern in paper that isn't readily perceivable.
- ***Digital Watermark:*** Secretly embedded marker in digital file. *Used by movie studios to find pirated copies*

1.3. VULNERABILITY

The ***weakness in an asset***. If a vulnerability is exploited, loss or damage to assets can occur.

- ***Common Vulnerabilities and Exposures (CVE):*** Standard identification number for vulnerabilities. Identifies a vulnerability. *Actual security problems found in software*
- ***Common Weakness Enumeration (CWE):*** List of software and hardware weakness types. *Types of exploits that software should not contain*
- ***Common Vulnerability Scoring System (CVSS):*** Uses the CIA triad principles within the metrics used to calculate the score. Rates how severe a vulnerability is.
- ***OWASP Application Security Verification Standards (ASVS):*** Provides a basis for testing web application technical security controls and provides a list of requirements for secure development.

1.3.1. Exploit

An exploit is a software that ***takes advantage of a vulnerability*** in order to ***cause harm*** to a system. An ***exploit kit*** is a ***compilation of exploits*** that are often designed to be served from webservers.

1.4. THREAT

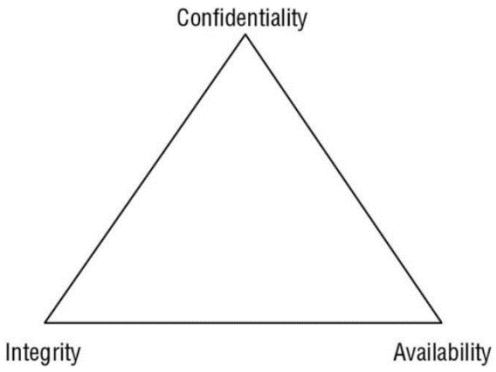
Any potential danger to an asset, intentional or accidental.

- ***Threat actor or agent:*** Intentionally exploits vulnerabilities. *Examples: Script kiddies, organized crime groups, state sponsors and governments, hacktivists, terror groups.*
- ***Threat intelligence:*** The knowledge about an existing or emerging threat to assets.
- ***Threat event:*** Accidental or intentional exploitation of vulnerabilities, natural or man-made. *Examples: Fire, earthquake, flood, system failure, human error, power outage.*

1.4.1. The STRIDE threat model

S	**Spoofing != Authenticity** An attack with the ***goal of gaining access*** to a target system using a ***falsified identity***.
T	**Tampering != Integrity** Any action resulting in ***unauthorized changes or manipulation of data***.
R	**Repudiation != Non-repudiation** The ability of a user or attacker to ***deny having performed*** an action or activity.
I	**Information disclosure != Confidentiality** The ***revelation or distribution*** of private, confidential or controlled information to external entities.
D	**Denial of service (DoS) != Availability** An attack that attempts ***to prevent authorized use*** of a resource.
E	**Elevation of privilege != Authorization** An attack where a limited user account is ***transformed into an account with greater privileges.***

1.5. CIA TRIAD



The primary goal of security is to protect the ***confidentiality***, ***integrity,*** and ***availability*** of assets.

- ***Confidentiality:*** Prevent or minimize unauthorized access to data. Data must be protected while in storage, in process and in transit. *Example: encryption & access controls*
- ***Integrity:*** Protecting the reliability and correctness of data. Unauthorized Alterations should not occur while the data is in storage, in process and in transit. *Example: intrusion detection system and hash verifications.*
- ***Availability:*** Authorized subjects are granted timely and uninterrupted access to objects. *Example: redundancy and scalability, maintain backups*

1.5.1. Nonrepudiation and Accountability

Ensures that a subject of an activity is ***not able to deny having performed the activity***. Accountability means being responsible or obligated for actions and results. *Example: certificates, session identifiers, transaction logs*

1.6. RISK MANAGEMENT

The goal of risk management strategies is to ***reduce risk***. It is ***impossible*** to design a ***risk-free*** environment. Risk management is a detailed process of ***identifying*** risk factors, ***evaluating*** those factors and ***implementing*** cost-effective solutions for reducing risk.

Countermeasure: Any action or product that reduces risk through the elimination or lessening of a threat or a vulnerability within an organization.

1.6.1. Risk analysis

The process by which the goals of risk management are achieved

- ***Evaluation***, assessment and the ***assignment of value*** for all assets within the organization.
- ***Examining*** an environment for risk
- ***Evaluating*** each ***threat event*** to its likelihood and the cost of damage it would cause
- ***Assessing the cost*** of various countermeasures for each risk and creating a cost/benefit report

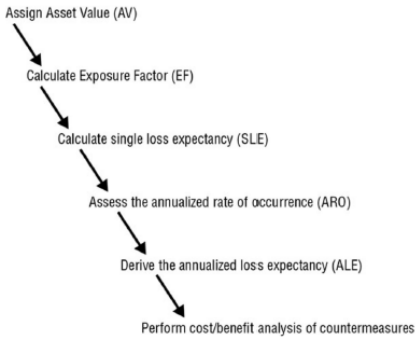
1.6.2. Key words

- ***Asset Valuation:*** Dollar value assigned to an asset based on actual cost and nonmonetary expenses.
- ***Exposure:*** Possibility for an asset loss because of a threat.
- ***Risk:*** Possibility that something could happen to damage, destroy or disclose data.
- ***Realized risk:*** A risk that happened. A threat actor or event has taken advantage of a vulnerability.
- ***Attack:*** Exploitation of a vulnerability by a threat agent.
- ***Breach:*** Occurrence of a security mechanism being bypassed or thwarted by a threat agent.

1.6.3. Risk assessment / analysis

How to decide which risks are acceptable and which are not? Once you develop a list of threats, you must individually ***evaluate each threat*** and its related risk.
There are two risk assessment methodologies: ***Quantitative risk analysis*** (assigns real dollar figures) and ***qualitative risk analysis*** (assigns subjective and intangible values). Both methods are necessary for a complete risk analysis in order to gain a balanced view of the security concerns.

Quantitative risk analysis



- ***Assign Asset value (AV):*** Assign a dollar value to the asset
- ***Calculate Exposure Factor (EF):*** Percentage of loss that an organization would experience if a specific asset is violated. *If the value of a building would be reduced from CHF 1'000'000 to CHF 250'000 by a fire, the exposure factor for the risk of fire would be 75%.*
- ***Calculate single loss expectancy (SLE):*** The cost associated with a single realized risk. $SLE = AV * EV$. *If an asset has a value (AV) of CHF 1'000'000 and an Exposure Factor of 75%, the single loss expectancy (SLE) would be* $\text{CHF } 1'000'000 * 0.75 = \text{CHF } 750'000$
- ***Assess the annualized rate of occurrence (ARO):*** The expected frequency with which a specific threat will occur in a single year. *The risk of fire in your building might have an ARO of 0.001.*
- ***Derive the annualized loss expectancy (ALE):*** The possible yearly cost of all instances of a specific realized threat against the asset. $ALE = SLE * ARO$. *If the Single loss expectancy is CHF 750'000 and the annualized rate of occurrence (ARO) is 0.001, the annualized loss expectancy (ALE) is* $\text{CHF } 750'000 * 0.001 = \text{CHF } 750$
- ***Annualized loss expectancy with a safeguard:*** Calculate the annualized loss expectancy (ALE) for the asset if the safeguard is implemented. This requires a new exposure factor (EF) and a new annualized rate of occurrence (ARO). The EF often stays the same. *If the expected frequency of a fire in your building in a year with a safeguard is 0.00001, The new ALE is* $\text{CHF } 750'000 * 0.00001 = \text{CHF } 7.50$

- ***Safeguard Costs:*** Compile a list of safeguards for each threat. Then you assign each safeguard a deployment value ***ACS*** (Annual cost of the safeguard). *If it costs CHF 500 to add a fire warning system to your building, the ACS of this safeguard is CHF 5000.*
- ***Calculating Safeguard Cost/Benefit:***
  ALE before safeguard – ALE with safeguard $- ACS$ = Value of Safeguard. If the result is negative, the safeguard is not a financially responsible choice. If the result is positive, then that value is the annual savings your organization may reap. The countermeasure with the greatest value from the cost/benefit formular makes the most economic sense. *If your ALE before safeguard is CHF 750, your ALE with safeguard is CHF 7.50 and the ACS is CHF 50'000 the cost/benefit is* $\text{CHF } 750 - \text{CHF } 7.50 - 500 = 242.50$ *which is a positive number. This safeguard would be a responsible choice.*

Security should be cost effective. Thus it is not prudent to spend more protecting an asset than it is worth to the organization. ***If the cost of the countermeasure is greater than the value of the asset or the cost of the risk, you should accept the risk***.

1.6.4. Risk handling

- ***Risk mitigation:*** The implementation of safeguards and countermeasures to eliminate vulnerabilities or block threats.
- ***Risk assignment:*** The placement of the cost of loss ono another entity, like purchasing insurance.
- ***Risk acceptance:*** If the cost/benefit analysis shows that countermeasure costs would outweigh the possible cost of loss, the risk can be accepted. Usually needs a sign-off letter.
- ***Risk deterrence:*** Process of implementing deterrents to would-be violators of security. *Examples: security cameras or guards, warning banners, motion detectors, strong authentication.*
- ***Risk avoidance:*** The process of selecting alternate options that have less associated risk. The risk is avoided by eliminating the cause: *Example: move to an inland location to avoid risks from tsunamis.*
- ***Risk rejection:*** A final but unacceptable possible response to risk is to ignore risk.

1.6.5. Residual risk

Once ***countermeasures are implemented***, the ***risk that remains*** is the residual risk. This is an accepted risk.

1.7. PRIVACY

Privacy is the ***right*** of an ***individual*** to ***control their personal data***. Data ***collection*** should be ***restricted***, data owners have a ***responsibility*** to ***respect*** and enforce privacy principles. Data remanence techniques should be used to ***permanently delete*** data.

1.7.1. USA Patriot Act of 2001

Allows authorities to monitor all communications to or from a person under a single warrant. Internet service providers may have to provide the government with a large range of information.

1.7.2. European Union General Data Protection Regulation (GDPR/DSGVO)

Companies have to inform authorities of serious data breaches within 24 hours. Individuals have access to their own data. The “right to be forgotten” allows people to require companies to delete their information. Violations can lead to heavy fines.

1.7.3. Pseudonymization

Makes it more difficult to identify individuals, can be reversed.

1.7.4. Anonymization

Removing all relevant data so it is impossible to identify the original subject.

2. IDENTITY AND ACCESS MANAGEMENT (IAM)

2.1. CONTROLLING ACCESS TO ASSETS

The goal is to provide access to authorized subjects and prevent unauthorized access attempts.

- ***Subject:*** active entity that accesses a passive object. *Example: Users, programs, computers*
- ***Object:*** passive entity that provides information to active subjects. *Example: files, databases, computers*

The management of the relationship between subjects and objects is known as access control.

2.1.1. Primary access control types

- ***Preventive:*** Attempts to thwart unauthorized activity from occurring.
  Examples: Fences, Locks, Alarm systems, data classification, penetration testing, training
- ***Detective:*** Attempts to discover unauthorized activity after it has occurred.
  Examples: Security guards, Motion detectors, Cameras, Honeypots, incident investigations
- ***Corrective:*** Attempt to correct any problems that occurred because of a security incident.
  Examples: Rebooting a system, antivirus solutions, backup and restore plans

2.1.2. Other access control types

- ***Deterrent:*** Discourage violation of security policies. Depends on choices of individuals.
  Examples: Policies, Training, Locks, Fences, Guards, Cameras, Security Badges
- ***Compensation:*** Provides various options to aid in enforcement and support of security policies.
  Examples: If data that should be encrypted isn't during transit, a compensation control can be added to protect the data.
- ***Directive:*** Directs, confines, or controls the action of subjects to force or encourage compliance.
  Examples: security policy requirements or criteria, posted notifications, escape route exit signs, monitoring
- ***Recovery:*** Extension of corrective control, but more advanced.
  Examples: Backup and restores, fault-tolerant drive systems, antivirus software, multisite solutions

2.1.3. Controlling access to assets

Access controls are also categorized by how they are implemented.

- ***Physical:*** prevent, monitor or detect contact with systems or areas within a facility. Touchable.
  Examples: Guard, Fences, locked doors, lights, laptop locks, cameras, alarms
- ***Technical or logical:*** Hardware or software used to manage access for resources. Uses Technology.
  Examples: Authentication methods, encryption, lists, protocols, firewalls, routers, intrusion detection systems
- ***Administrative / management:*** Policies defined by an organizations security policy.
  Examples: Procedures, hiring practices, background checks, security awareness and training efforts, reporting, testing

2.2. STEPS OF ACCESS CONTROL

- ***Identification:*** The process of a subject claiming (and proving) an identity. Like a username. The identity must be proven or verified before access is allowed.
- ***Authentication:*** The subject needs to provide additional information that corresponds to the identity they are claiming. Like a password. The process of verifying that the claimed identity is valid is authentication.
- ***Authorization:*** Checks if you're allowed to do what you want to do after authentication.
- ***Auditing:*** Logging of user activity. The purpose is to hold the subjects accountable for their actions.
- ***Accounting:*** Relies on the capability to prove a subject's identity through the other steps. To have viable accountability, you should be able to support your security decisions in a court of law.

2.2.1. Authentication

- ***Password / Basic authentication:*** static, weakest form of authentication. Should not be stored in plaintext. Weakness: prone to passive sniffing attacks.
- ***Password phrase:*** Easier to remember and encourages the use of a longer pw. *1P@ssedTheCySecEx@m*
- ***Cognitive password:*** Series of questions that only the subject should know.
- ***Smartcard:*** Credit card-sized ID or badge that has a circuit chip embedded. Contains information about the authorized user. Smartcards are tamper-resistant and provide an easy way to carry complex keys.
- ***Tokens:*** A carry-on password-generating device. An authentication server stores details of the token so the server always knows what number is displayed on the token. There are synchronous dynamic password tokens (use a clock) and asynchronous dynamic password tokens (use a counter).
- ***Onetime password generators:*** *Dynamic passwords* that change every time they are used. Generators are token devices that create those passwords. *Time-based one-time password (TOTP)* use a timestamp and remain valid for a certain timeframe. *HMAC-based One-time passwords (HOTP)* include a hash function to create passwords which remain valid until used.
- ***Challenge/Response:*** One party presents a question ("challenge"), and another party must provide a valid answer ("response") to be authenticated.
- ***Anonymous key exchange:*** Exchange credentials over unauthenticated secure channel.
- ***Server Certificates plus user authentication:*** Transmit password over authenticated secure channel.
- ***Mutual Public Key authentication:*** Bilateral use of public key signatures.
- ***Zero knowledge password proofs:*** a password-based authentication protocol that allows a claimant to authenticate to a verifier without revealing the password. *See example with different colored balls*

Summary: Vulnerability Matrix

Attack	A1 Basic Authentication	A2 One Time Passwords	A3 Challenge / Response	A4 Anonymous Key Exchange	A5 Zero Knowledge PW Proof	A6 Server Cert + User Auth	A7 Mutual Public Key Auth
Passive Password Sniffing	x						
Offline Brute Force Password Attack	x		x	x			
Active Man-in-the-Middle Attack (Phishing)	x	x	x	x			
Identity Theft on Server	x	x	x	x	x	x	
CA Compromise						x	x

2.2.1.1. Authentication Factors

- ***Type 1 (weakest):*** Something you know. *Example: Password, PIN, Cognitive passwords*
- ***Type 2:*** Something you have, like a physical device. *Example: Smartcard, USB drive, auth app*
- ***Type 3 (strongest):*** Something you are / you do. *Example: Fingerprint, keystroke pattern*

Multifactor authentication: Any authentication using two or more factors of different Types.

Secondary authentication factors

- ***Somewhere you are:*** Location based. *Example: IP Address, Caller ID, specific computer*
- ***Somewhere you aren't:*** to identify suspicious activity: *Example: User gets mail if log in at new location*

2.2.2. Authorization

Access control models

- ***Discretionary Access Control (DAC):*** Every object has an owner which can grant or deny access to other subjects. *Example: Windows File System.*
- ***Role Based Access Control (RBAC):*** Users are placed in roles which have been assigned privileges. *Example: Different groups for IT, HR and Marketing with different access rights.*
- ***Rule Based Access Control (RuBAC):*** Global rules that are applied to all subjects. Rules are referred to as restrictions or filters. *Example: Firewall blocks access to blick.ch for all users in the network.*
- ***Attribute Based Access Control (ABAC):*** Rules can include multiple attributes. More flexible than RuBAC. *Example: No one should be allowed to access digitec.ch but the users of the IT group.*
- ***Mandatory Access Control (MAC):*** Use of labels applied to both subjects and objects. *Example: if a user has a label of top-secret, he can be granted access to a top-secret document.*
- ***Implicit Deny:*** Most mechanisms use this. Access is denied unless it has been explicitly granted.
- ***Constrained Interface:*** Hide the capacity if the user does not have permission to use it. *Example: Hide advanced settings if user isn't in admin group.*
- ***Access Control Matrix (ACL):*** Table that includes subjects, objects and assigned privileges. System checks table to decide if user has the needed privileges for an action. Object focused and identify the subjects that can access the object. *Example: Read/write/execute permissions on files/folders per user/group*
- ***Capability Tables:*** Also used to identify privileges assigned to subjects. Subject focused and identify the objects that subjects can access.
- ***Content-Dependent Control:*** Restrict access to data based on the content within an object. *Example: A database view. A view retrieves specific columns, creating a virtual table.*
- ***Context-Dependent Control:*** Require specific activity before granting users access. *Example: a download page in an online shop is only visible if a user goes through the purchase process first.*

Principles

- ***Need to know:*** Subjects are granted access only to what they need to know. *«So viel wie nötig»*
- ***Least Privilege:*** Subjects are granted only the privileges they need to perform. *«So wenig wie möglich»*
- ***Separation of Duties and Responsibilities:*** No single person has total control over a system.

2.3. COMMON ACCESS CONTROL ATTACKS

- ***Access aggregation attacks (passive):*** Collecting multiple pieces of non sensitive information and aggregating them to learn sensitive information.
- ***Password attacks (brute force):*** Attack against online account or stealing database with pws.
- ***Dictionary attacks (brute force):*** Using every possible password in a predefined database of common passwords to try and find out the correct password. Often also scan for one-upped-constructed pws.
- ***Birthday attacks (brute force):*** Focuses on finding collisions in hash functions.
- ***Rainbow table attacks:*** large database of precomputed hashes which can be compared to the hashes in a stolen password database file.
- ***Sniffer attacks:*** Attacker uses a sniffer (software application) to capture information transmitted over a network.
- ***Spoofing / Masquerading Attacks:*** Pretending to be something or someone else. Attackers replace a valid IP address, email address or phone number with a false one and impersonate a trusted system.
- ***Social Engineering Attacks:*** Trick people into revealing information.
- ***Shoulder surfing:*** Read information on the computer screen of someone else or watch the keyboard as a user types.
- ***Phishing:*** Form of social engineering, attempts to trick users into giving up sensitive data or clicking a link.

- ***More sophisticated phishing:*** Link to a site that looks legitimate, mail with an infected file, link to a website that installs a drive-by download, use of social media.
- ***Spear phishing:*** Targeted to a specific group of users, such as employees within an organization.
- ***Whaling:*** Phishing that targets senior or high-level executives.
- ***Vishing:*** Phishing attack via instant messaging and VoIP.

2.4. PROTECTION MECHANISMS

- ***Layering (defense in depth):*** Use of multiple controls in a series. A failed control should not result in exposure of systems or data. Swiss cheese model.
- ***Abstraction:*** Simplifies security by enabling to assign security controls to a group of objects.
- ***Data Hiding:*** Preventing data theft by positioning the data that it is not accessible or seen.
- ***Security through obscurity:*** Not informing a subject about an object being present. Not really security.
- ***Encryption:*** Hiding the meaning of a communication from unintended recipients.

3. SYMMETRIC ENCRYPTION AND KEY EXCHANGE

3.1. CRYPTOGRAPHIC CONCEPTS

- ***Message or plaintext:*** Before a message is put into a coded form, it is known as plaintext *unencrypted*
- ***Ciphertext:*** The plaintext message gets encrypted to a ciphertext with a cryptographic algorithm.
- ***Cipher:*** The encryption algorithm.
- ***Cryptographic key:*** large number used for the cipher. The security of the key needs to be protected.
- ***One-Way functions:*** Mathematical operation that makes it impossible to retrieve the input values.
- ***Reversibility:*** We should be able to undo the operation of encryption (decryption)
- ***Nonce:*** Unique number that changes each time it is used. Is used to make sure that a key is not re-used twice. The nonce is public, whereas the key is private.
- ***XOR:*** Binary operator between two values that returns true if either input is true, but not both (encrypt). If applied twice, it reverses its effect (decrypt).
- ***Initialization vector (IV):*** A random bit string. Same length as the block size and is XORed with the message. IVs are used to create unique ciphertexts every time the same message is encrypted using the same key.
- ***Confusion:*** Occurs when the relationship between the plaintext and the key is so complicated that an attacker cannot find the key by analyzing said relationship. Substitution of bytes adds confusion.
  Example: Enigma machine, only confusion but no 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.
- ***Kerckhoff's principle:*** "The enemy knows the system". A cryptographic system should be secure even if everything about the system except the key is public knowledge.
- ***SP-Network:*** Algorithm that uses repeated substitution and permutation operations. Substitution: replacing bits with others. Permutation: Swapping bytes around.
- ***One Time pad:*** Cipher that uses XOR to encrypt and decrypt a message. Uses a key that's the same length as the message, XOR each message bit with each key bit. Impossible to break, but impractical because the key needs to be the same size as the message. And keys cannot be reused.

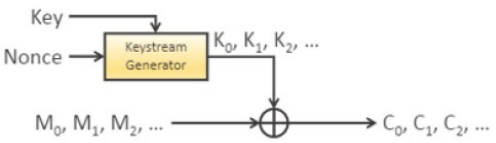
3.2. SYMMETRIC CRYPTOGRAPHY

Symmetric cryptography relies on a ***"shared secret" key*** that is distributed to all members who participate in the communications. The ***key*** is used by all parties to ***both encrypt and decrypt*** messages. It provides only the security service of ***confidentiality***.

Symmetric key cryptography does ***not implement nonrepudiation*** because each party uses the same key to encrypt and decrypt. It also does ***not implement message integrity***. But it is ***very fast***.

3.2.1. Stream cipher

We can approximate a one-time pad by generating an ***infinite*** pseudo-random ***keystream***. Stream ciphers work on messages of ***any length***.



Extremely ***fast*** and possible to encrypt long continuous streams but does not protect the ciphertext and one is not allowed to reuse keys.

3.2.2. Block cipher

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

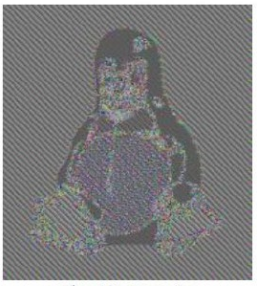
Advanced Encryption Standard (AES)

SP-Network that uses repeated substitution and permutation operations. The message gets split into matrices of 8 bits. Each round the AES does the following operations:

- ***SubBytes:*** Substitute each cell of 8 bits with another 8 bits which are chosen from a predefined loop up table.
- ***ShiftRows:*** Each row of the matrix is shifted to the left (each one step more than the row before)
- ***MixColumns:*** Performs matrix multiplication between the current matrix and a predefined given matrix. The sum operation is substituted by XOR and multiplication for AND.
- ***Key Addition:*** The matrix gets XORed with the key matrix.

Mode of operation for block ciphers

A mode of operation is the combination of multiple instances of block encryption into a usable protocol.


The ECB Penguin

- ***Electronic Code Book (ECB):*** Encrypts each block one after another. The same input creates the same output, so it is weak to redundant data divulging patterns. Not recommended.
- ***Cipher Block Chaining (CBC):*** XOR the output of each cipher block with the next input. Not parallelizable, not perfect.
- ***Counter Mode (CTR):*** Encrypting a counter (Nonce $+ n$) to produce a stream cipher. Can be parallelized. Does not encrypt the message, but a random number and uses that to XOR the message. Standard mode for all types of encryption cipher (AES).

3.2.3. Key distribution

Parties must have a secure method of exchanging the secret key.

- ***Diffie-Hellmann (DH) key exchange:*** TLS is relying heavily on DH. Two parties can jointly agree a shared secret over an insecure channel. Not really a key exchange but a exchange of some parts of the mathematical key so each party can create the key by themselves. Uses prime numbers, modulo and logarithms. $(g^b)^c \ mod \ x = (g^c)^b mod \ x$
- ***Elliptic Curve cryptography:*** Replacement for the mathematics underpinning DH: Elliptic curve is a two-dimensional curve. The private key is a number, the public key is composed of two numbers. More difficult to solve than the logarithm of DH. Much stronger than other schemes for the same key length.

Ephemeral Mode (perfect forward secrecy): New key exchange for every new session.

4. ASYMMETRIC CRYPTOGRAPHY AND HASH FUNCTIONS

4.1. RSA

Public-key cryptosystem which is widely used. The keys are reversible, either can be used for encryption or decryption.

You should already know how to use RSA by this time, so I will skip this chapter.

4.1.1. Encryption using RSA

RSA is ***very weak for short messages***. It is not common to see encryption done using RSA. It is a lot slower than symmetric cryptosystems.

4.1.2. Signing using RSA

Signing is ***encrypted with the private key***. Mostly for verification of the ***integrity*** of the message. The message is hashed, and the hash is signed and sent with the message. The receiver hashes the message, decrypts the signature and checks if the hashes match.

4.2. HASH FUNCTIONS

Takes a message of any length and returns a pseudorandom hash of fixed length. Non-reversable. A good hashing algorithm should perform quickly but not too quickly because of brute force attacks.

- ***Functions:*** Hash functions iteratively jumble blocks of a message after another. The message is processed in blocks, every round there is a new hash. The last one is the final hash. Output must be indistinguishable from random noise. Bit changes must diffuse through the entire output.
- ***Password storage:*** PBKDF2 (Password-Based key derivation function 2) uses a hash like SHA-2 but runs it in a loop 5000 times so it is slower. Good for password storage.
- ***Hash collision:*** When two different inputs get the same hash, the function is broken. *like MD5*
- ***Current Standard:*** SHA-2 256 bits and 512 bits.
- ***HMAC (hash message authentication code):*** Splits a key in two and hashes twice. This way it is not vulnerable to length extension attacks.

5. HACKING & ETHICAL HACKING

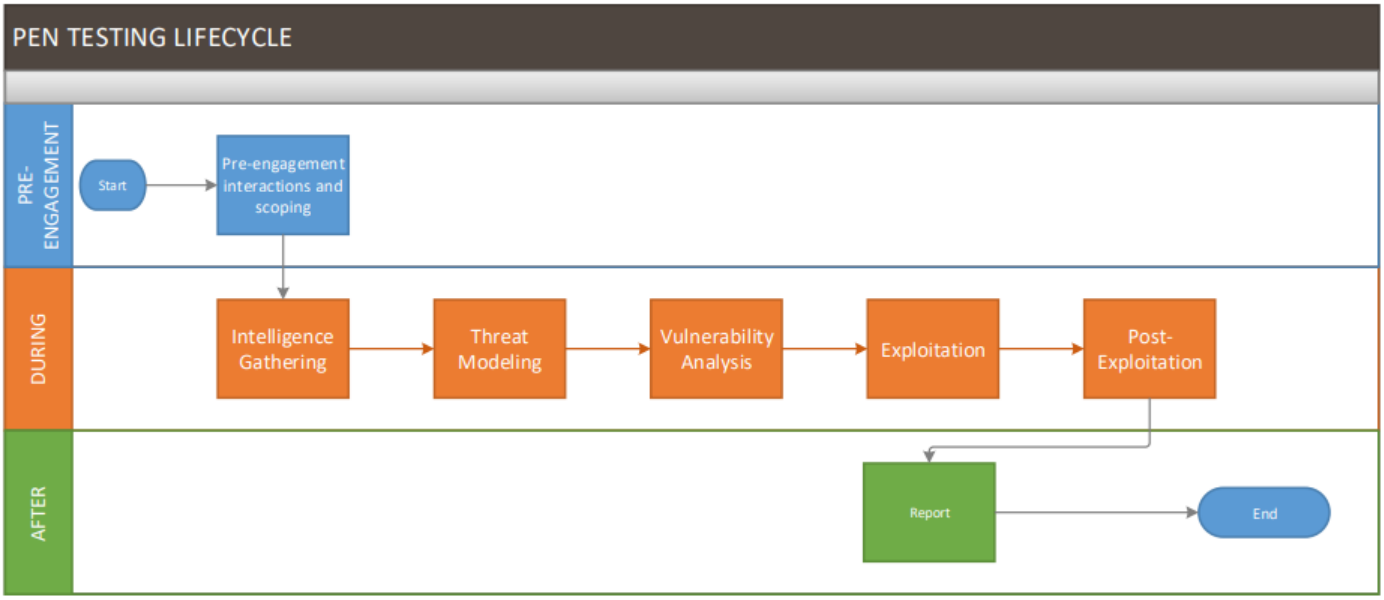
- ***Red Team:*** Offensive, focused on penetration testing.
- ***Blue Team:*** Assessment of network security and identification of possible vulnerabilities.
- ***Hacking:*** Exploiting vulnerabilities in systems and / or software to gain unauthorized access
- ***Ethical Hacking:*** Using tools and techniques to validate, audit and report on system / software vulnerabilities.

Hacker Types

- ***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.
- ***Script Kiddie:*** Individuals that use tools without understanding what they are doing.
- ***Cyber Terrorist:*** Skilled attacker whose purpose it is to further an ideology.
- ***State Sponsored:*** Hackers employed by the governement for both offensive and defensive activities.
- ***Hacktivists:*** A hacker whose activity is aimed at promoting a cause

5.1. PENETRATION TESTING

For pen testing, a contract, statement of work and nondisclosure agreement is drawn up.



6. ATTACK TYPES AND COMMON ATTACKS

6.1. VIRUSES, WORMS, TROJAN HORSES

Computer security threats that exploit various vulnerabilities to spread malicious payloads to computer systems. Most computer viruses and Trojan horses depend on irresponsible computer use by humans. Worms spread rapidly among vulnerable systems under their own power.

6.1.1. Zero Day Attack

Exploits a Zero-Day vulnerability which are security flaws discovered by hackers that have not been addressed by the security community.

6.1.2. Virus

A virus has two main functions: propagation and destruction. The propagation function defines how the virus will spread from system to system. The destructive power is delivered by the virus's payload.

Propagation techniques

- ***Master boot record (MBR) infection:*** Attack the portion of bootable media that the computer uses to load the operating system during the boot process.
- ***File infection:*** Infect executable files. Often easily detected.
- ***Macro infection:*** virus that is written in a macro language inside a software application / document.
- ***Service injection:*** Inject themselves into trusted runtime processes of the operation system.

Malware technologies

- ***Multipart Viruses:*** Use more than one propagation technique.
- ***Stealth Viruses:*** Hide themselves and fool antivirus to thing that everything is okay.
- ***Polymorphic Viruses:*** Modify their own code as they travel from system to system.
- ***Encrypted Viruses:*** Use cryptographic techniques to avoid detection.
- ***Logic Bombs:*** Lie dormant until they are triggered by one or more conditions like time or website login.
- ***Trojan Horses:*** Software program that appears "kind" but carries a malicious payload.
- ***Keystroke logging:*** Virus logs the keys struck on a keyboard.
- ***Ransomware:*** Uses encryption to encrypt files and then demands ransom for decrypting.
- ***Worms:*** Malicious code objects that propagate themselves without requiring any human intervention.
- ***Spyware:*** Monitors action and transmits important details to a remote system that spies on you.
- ***Adware:*** Displays advertisements on infected computers.

6.2. SQL INJECTION

Allow a malicious individual to directly perform ***SQL transactions*** against the underlying database. Databases will process multiple SQL statements at the same time if you end each one with a semicolon.

Example: A bank customer enters an account number to gain access to a dynamic web application that retrieves current account details. The web application uses a SQL query to obtain that information: `SELECT * FROM transactions WHERE account_number = "<number>"`. If the user's account number is `145249`, it is possible to enter the following into the <number> field: `"145249"; DELETE * FROM transactions WHERE "a" = "a"`. This is a valid SQL transaction containing two statements, the second of which deletes all the records stored in the database. Occurs when an application sends untrusted data to an interpreter.

6.2.1. Protection against SQL Injection Attacks

- ***Use prepared statements:*** Application may pass parameters to it but cannot alter the statement.
- ***Limit Account privileges:*** The database account should have the smallest set of privileges possible.
- ***Perform input validation:*** Remove HTML Chars from allowed inputs. Safest form: whitelist validation, only specified which input is allowed, everything else is not allowed.

6.3. MAN IN THE MIDDLE (MITM)

The attacker inserts itself in-***between*** the ***client*** and the ***server***. The traffic is forced through the attacker machine. The attacker can now ***view*** and ***control*** all ***network traffic***.

6.4. MAN IN THE BROWSER (MITB)

Trojan horse program installed as a ***browser plugin***. Can capture form data, such as usernames and passwords. Can inject JavaScript into webpages or hijack authentication sessions.

6.5. BUFFER OVERFLOW

Buffer overflow vulnerabilities exist when a developer does not properly validate user input to ensure that it is of an appropriate size. Input that is too large can "overflow" a data structure to affect other data stored in the computers memory.

6.6. CROSS-SITE SCRIPTING (XXS)

Scripts can be ***embedded*** in web pages by using the HTML tags `<script></script>`. A successful attack can allow the attacker to execute JavaScript in the victim's browser.
It can be found in any page that displays user supplied data.

Example: Consider a web application that contains a text box asking the user to enter his name. When the user clicks submit, the web application loads a new page that says «Hello, Name». If a script is entered, like `<script>alert("hello")</script>`, the web application reflects the input on the web page and the browser processes it and executes the script.

6.7. CROSS-SITE REQUEST FORGERY (XSRF)

Attackers embed code in one website that ***sends a command to a second website*** if the user clicks on a link. ***Protection against XSRF:*** using secure tokens and checking the referring URL in requests.

Example: An attacker wants to steal funds from user accounts of an online banking site. The attacker goes to an online forum and posts a message containing a link. The link is actually a link directly into the money transfer site that issues a command to transfer funds to the attacker's account. If the user who clicks on the link happens to be logged into the banking site, the transfer succeeds.

6.8. DOS SYN FLOOD

Resource consumption attack that has the ***goal of preventing legitimate activity***. The attacker sends TCP SYN segments to open ports with a spoofed source IP address. The server replies with SYN/ACK to spoofed source. Leads to a lot of half-open connections until the server cannot accept new connections.

6.9. DDOS AND BOTNETS

A botnet is a connection of internet-connected devices whose security has been breached. Each compromised device added to a botnet is known as ***bot*** or ***zombie***. Botnets are increasingly ***rented out*** by cyber criminals. The controller of a botnet can direct the activities of these compromised computers. If ***each zombie conducts a DoS attack*** against the victim, it is called a ***distributed DoS*** (DDoS). The victim may be able to discover zombie systems that are causing the DoS attack but won't be able to track down the actual attacker.

6.10. ADVCANCED PERSISTENT THREATS

Advanced Persistent Threats (APTs) are ***sophisticated attacks*** in which a specific target is attacked over a ***longer period of time*** (years). They are used by ***skilled attackers*** to penetrate networks undetected, steal data or conduct ***espionage***. APTs are ***difficult to detect*** and use ***advanced techniques*** to circumvent security measures. Comprehensive protection and proactive security measures are required to defend against APTs. Often associated with ***zero-day exploits***.

6.11. XML EXTERNAL ENTITY ATTACKS (XXE)

Attack range: DoS, Inclusion of local files, port scanning, overloading of XML-Schema, NTLM authentication material theft by accessing files on a network share.

Example: A new variable named `include` is created with the `!ENTITY` command. With the `SYSTEM` parameter, one can read external files and store them as a value, like here with the file `/etc/passwd`. which is subsequently printed on the website in the <description> tag.

Countermeasures: Xerces Hardening, hardening of the XML parser.

7. TRANSPORT LAYER SECURITY

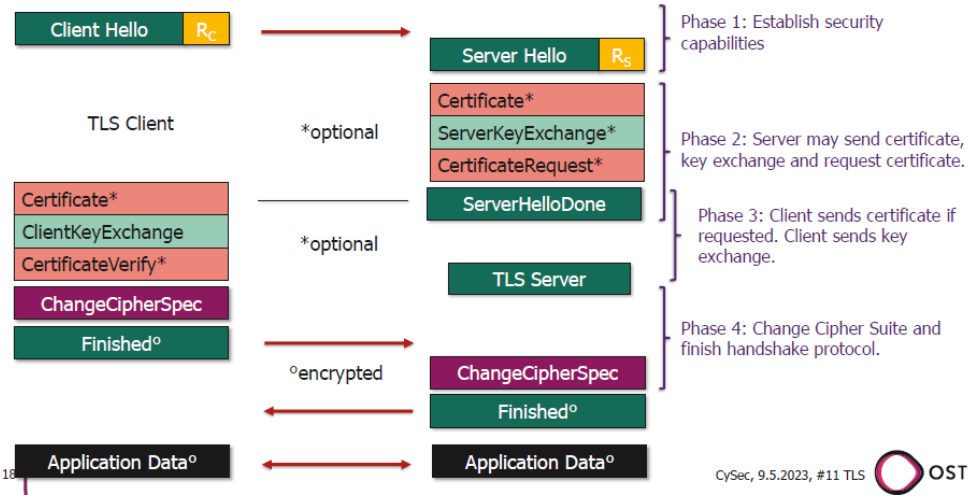
SSL (secure socket layer) and TLS (transport layer security) are ***network security protocols*** for setting up ***authenticated*** and ***encrypted connections*** and exchanging data. TLS is the primary mechanism for encryption for HTTP communication, that is ***HTTPS***.

7.1. TLS ARCHITECTURE

- ***Connection:*** Every connection is associated with one session.
- ***Session:*** Association between a client and a server created by the Handshake protocol. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

7.2. TLS HANDSHAKE

Is used before any data are transmitted. Allows server and client to authenticate each other, negotiate encryption, keys, and MAC (Message authentication code) algorithms. The exchange has four phases. The keys are shared via RSA or DH.



7.3. FORWARD SECRECY (FS)

Gives assurances that ***session keys will not be compromised*** even if ***long-term secrets*** used in the session key exchange are ***compromised***. For HTTPS this is the ***private signing key*** of the server. By generating a ***unique session key for every session*** a user initiates, the compromise of a single session key ***will not affect any other data*** than that exchanged in the specific session.

7.4. HEARTBEAT PROTOCOL

Periodic signal to indicate ***normal*** operation or to ***synchronize*** other parts of a system. This assures the sender that the recipient is ***still alive*** and ***generates activity*** across the connection during idle periods.

7.5. TLS ATTACKS

There are four categories of attacks in TLS: Attacks on the ***handshake*** protocol, attacks on the ***record*** and application data protocols, attacks on the ***PKI*** and other attacks.

8. DETECTION & RESPONSE

8.1. CYBER DEFENSE FRAMEWORKS

- ***Cyber Kill Chain:*** Has eight phases:
  - Reconnaissance *Attackers assess the situation from outside, to identify targets and tactics.*
  - Intrusion *Attackers use identified tactics and leverage malware or security vulnerabilities to get in your systems.*
  - Exploitation *Exploiting vulnerabilities and delivering malicious code to get better foothold.*
  - Privilege *Escalation Attackers often need more privileges on a system, for this they need to escalate their privileges.*
  - Lateral movement *Once in the system, attackers can move to other systems and accounts to gain leverage.*
  - Obfuscation *To stay hidden, attackers need to cover their tracks and lay false paths, compromise data and clear logs.*
  - Denial of Service *Disruption of normal access to stop the attack from being monitored or blocked.*
  - Exfiltration *The extraction stage: getting data out of the compromised system.*
- ***Diamond model:*** Consists of four basic components: adversary (Name, alias, origin), infrastructure (IP addresses, malware used), victim (location, goal, person), capability (Attack methods, targets)
- ***STIX/TAXII:*** Structured Threat Information eXpression, JSON based document / Trusted Automated eXchange of Intelligence Information, standardized language.
- ***MISP:*** threat intelligence platform that facilitates the exchange of threat intelligence, Indicators of compromise, targeted malware and attacks, financial fraud, ..
- ***MITRE ATT&CK:*** Adversarial Tactics, Techniques & Common Knowledge. Framework to document common tactics, techniques and procedures that APT use against Windows enterprise networks.

8.2. SIEM (SECURITY INFORMATION AND EVENT MANAGEMENT)

Responsible for collecting log and event data from various sources and aggregating, identifying, categorizing, and analyzing it in real time. With a SIEM solution, security problems should be detected automatically as well as the ability to send alerts.

8.3. SOAR (SECURITY ORCHESTRATION, AUTOMATION AND RESPONSE SOLUTIONS)

Also collects data from various sources, but also supports the incident responder in managing the crisis and rolling out security countermeasures. It enables automated intervention.

8.4. WAF (WEB APPLICATION FIREWALL)

A specific form of application firewall that filters, monitors, and blocks HTTP traffic to and from a web service. By inspecting HTTP traffic, it can prevent attacks exploiting a web application's known vulnerabilities, such as SQL injection, cross-site scripting (XSS), file inclusion, and improper system configuration.

8.5. ANTIVIRUS

Computer program used to ***prevent, detect, and remove malware***. There are different ways to detect a virus: ***Signature-based*** (database with characteristics of known viruses), ***Heuristic-based*** (looks at behaviour of software) and ***Data integrity*** (checks for unauthorized file modifications).

The building blocks of cryptography need to be carefully assembled into protocols that keep our systems secure. There are numerous well-established protocols out there.

MAC = message authentication code

8.6. PROTOCOLS

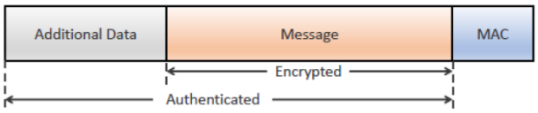
8.6.1. Authenticated Encryption (AE)

Authenticated Encryption (AE) is an encryption scheme which simultaneously assures the data ***confidentiality*** and ***authenticity***.

- ***Encrypt-then-MAC (EtM):*** Encrypt plaintext, produce MAC from ciphertext, send both together.
- ***Encrypt-and-MAC (E&M):*** Encrypt plaintext, produce MAC from plaintext, send both together.
- ***MAC-then-Encrypt (MtE):*** Produce MAC from plaintext, encrypt plaintext and MAC, send ciphertext.

8.6.2. Authenticated Encryption with Associated Data (AEAD)

Variant of AE. Allows recipient to check integrity of both encrypted and unencrypted information. A MAC is often attached to the end of the ciphertext.



Examples

- ***AES Galois Counter Mode:*** AES is a block cipher in CTR mode and GCM is the MAC. It computes a Galois MAC (GMAC) over the ciphertext and the additional data. Provides confidentiality and integrity.
- ***ChaCha20_Poly1305:*** Chacha20 is a stream cipher and Poly1305 is a MAC. Used on mobile phones.

8.7. SIGNATURES AND CERTIFICATES

Digital Signatures: Hashes are used with RSA and DAS to create digital signatures. They prove the authenticity of the sender. The server only proves that it has the private key, it does not prov that the server can be trusted yet.

Digital Certificates: Are a mechanism through which we can verify the ownership of a public key. This verification uses a trusted third party. Usually managed through a Public Key infrastructure.

8.7.1. Certificate Issuance

The server generates a private and public key. It creates a ***Certificate Signing Request (CSR)*** which needs to be verified by a ***Certification Authority (CA)***. The CA does some identification checks and then creates and signs the certificate with its private key. The server how has a ***certificate signed by a CA*** that it can use to prove its identity.

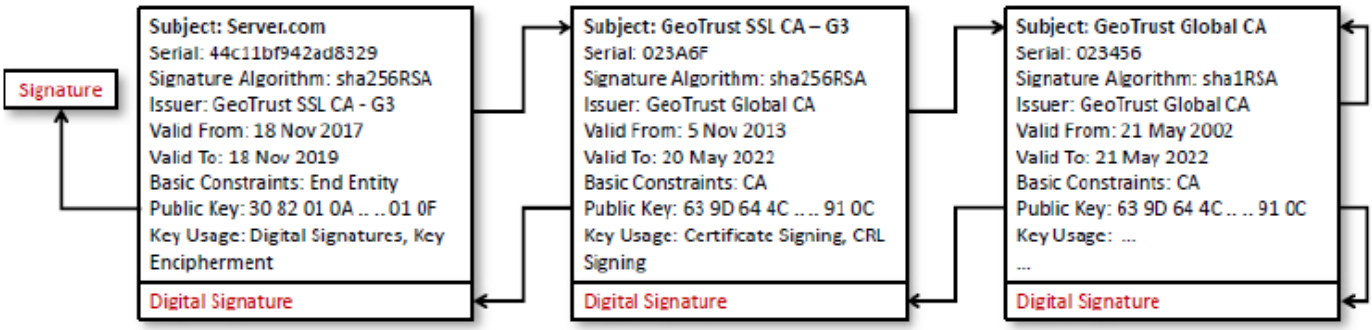
8.7.2. Certificate Use

During a ***TLS handshake***, the server supplies to the client a signature backed up by a certificate. In order to ***trust*** the public key in the certificate, the server has to ***check that the digital signature*** on the certificate is valid. The only way to check the validity is to use the ***public key of the issuer of the certificate to decrypt the digital signature***. This is called ***chain of trust***.

8.7.3. Chain of Trust

Normally, the chain involves ***multiple certificates***. Chains always end in a ***root certificate*** that is located on the client's machine. The root certificate is ***self-signed*** and is ***trusted*** because it is ***built into*** the operating system.

There are two ***assumptions*** that have to be true for the public infrastructure to be secure: The ***private key*** of a server is ***kept secret***, and the ***trust store*** or the ***root certificate*** on the client ***hasn't*** been ***manipulated***.



8.8. 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) 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)***.

8.8.1. Public Key Certificate (X.509)

Is an electronic document proving the ownership of a public key. Includes information about the key, the identity of its owner and of the entity that has verified the certificate.

8.8.2. Quality of a certificate

There exist four categories depending on the type checks done during registration and use:

- ***Domain Validated (DV) Certificate:*** Is issued after the proof of right of usage is established.
- ***Organization Validated (OV) Certificate:*** Needs validation of company name and domain name.
- ***Extended Validation (EV) Certificate:*** Are issued once an entity passes a strict procedure.
- ***Qualified Website Authentication Certificate (QWAC):*** Certificate under eIDAS Regulation

Where to get ***validity information*** of the certificate: ***Certificate Revocation List (CRL)*** or ***Online Certificate Status Protocol (OCSP).***

8.8.3. Trust service providers (TSP) – Certificate Authorities

The Registration Authority (RA) and the Certificate Authority (CA). The provide trusted identity information, support secure authentication, support integrity-assured and encrypted communication.

9. EMAIL SECURITY

9.1. SENDER POLICY FRAMEWORK (SPF)

v=spf1 mx ip4:193.135.215.47/32 ip4:193.135.215.55/32 -all

Allows senders to define ***which IP addresses are allowed*** to ***send mail*** for a particular ***domain***.

9.2. DOMAINKEYS IDENTIFIED MAIL (DKIM)

Provides an ***encryption key and digital signature*** that ***verifies*** that an email message was ***not faked***.
It works by adding a ***digital signature*** to the headers of an email message. That signature can be ***validated*** against a public key in the organizations ***DNS records***.

9.3. DOMAIN-BASED MESSAGE AUTHENTICATION, REPORTING AND CONFORMANCE (DMARC)

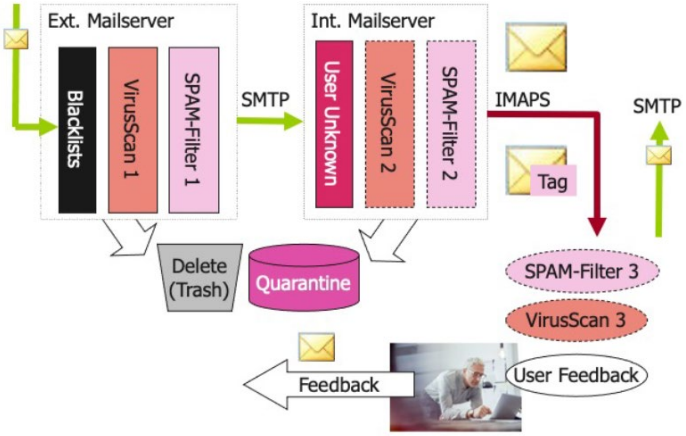
Unifies the SPF and DKIM authentication mechanisms into a common framework and ***allows domain owners to declare how they would like emails from that domain to be handled if it fails an authorization test***. DMARC policies are published in the DNS records.

9.4. SECURE / MULTIPURPOSE INTERNET MAIL EXTENSIONS (S/MIME)

MIME: Expansion of a mail message with extensions for e.g. security or to ***support*** text in character sets other than ASCII as well as ***attachments*** of audio, video, images and application programs.

S/MIME is a standard for ***public key encryption*** and ***signing of MIME*** data.

9.5. SPAM



10. KERBEROS (KEY DISTRIBUTION SYSTEM)

Kerberos is a ***key distribution center*** (KDC). It is a ***network authentication protocol*** that works on the basis of ***tickets*** (security tokens) ***to allow nodes communicating*** over a non-secure network to prove their identity to one another in a secure manner. Each Kerberos participant – called a principal – shares a common secret with the KDC, the ***principals master key***.

The maser keys of all principals are stored in the KDC database, encrypted using the KDC master key. The server which is hosting the KDC service must be physically secured by locking it away in a burglar-proof room.

10.1. SINGLE SIGN ON AUTHENTICATION WITH KERBEROS

- The ***client*** sends the principal name and a ***request*** for a ***TGT*** (Ticket granting ticket) to the KDC.
- The KDC ***generates a session key*** and a ***TGT***.
- It ***sends*** the session key and the TGT ***to the client***.
- When the client wants to use a service, it sends a ***service ticket request*** to the KDC.
- The KDC ***validates the client***. It then generates a ***service session key*** and sends it to the client.
- The ***client sends it to the service provider*** together with the service session key, the principal name and a timestamp.
- The ***service provider enables the service*** for the client.

11. FEDERATION

Federated identity is way to ***use an account from one website to create an account and log in to a different site***. ***Protocols:*** SAML (Security Assertion Mark-up Language), OAuth 2.0, OIDC (OpenID Connect)

11.1. SAML

XML-Based standard for exchanging authentication and authorization data. Uses a XML document called Assertion that the Identity Provider sends to the service provider.

11.2. OAUTH 2

Provides secure access delegation mechanism for website. It defines a process for end-users to grant controlled access for third-party website to their private resources stored on a resource server.

There are two client types: Confidential (Applications with a backend) and Public (Single page applications).

Credentials used to access protected resources are called ***access tokens***. They grant access to a specific protected resource defined by the scope parameter. They are issued by the authorization server.

Grant Types: Defines the messages that are exchanged between the parties to issue an access token to the client. Authorization Code + PKCE is recommended.

11.3. AUTHORIZATION CODE FLOW

- The **client** sends an authorization request to the **resource owner**.
- The **resource owner** gives consent to the **auth server**.
- The **auth server** sends an authorization grant to the **resource owner**.
- The **resource owner** sends the authorization grant to the **client**.
- The **client** sends the authorization grant and his client ID / Secret to the **auth server**.
- The **auth server** sends an access token to the **client**.
- The **client** sends the access token to the **resource server**.
- The **resource server** sends the protected resource to the **client**.

11.4. IMPLICIT FLOW

- The **client** sends an authorization request to the **resource owner**.
- The **resource owner** gives consent to the **auth server**.
- The **auth server** sends an access token to the **resource owner**.
- The **resource owner** sends the access token to the **client**.
- The **client** sends the access token to the **resource server**.
- The **resource server** sends the protected resource to the **client**.

11.5. AUTHORIZATION CODE FLOW WITH PROOF KEY FOR CODE EXCHANGE (PKCE)

Originally designed for ***mobile/native clients*** as they ***cannot store a static client secret securely***, which may ***allow malicious apps*** which pose as the original app to ***steal codes/tokens***.

With PKCE, the ***client generates a dynamic secret*** which later can be ***verified by the auth server***. This ***guarantees*** the auth server is ***always talking to the actual client*** who initiated the OAuth flow.