Content

1. Computer security principles - Základné princípy počítačovej bezpečnosti

Computer security is the protection of the items you value, called the assets of a computer or computer system. There are many types of assets, involving hardware, software, data, people, processes, or combinations of these. To determine what to protect, we must first identify what has value and to whom. [hardware, software, and data].

Value of the asstest: after identifying the assets to protect, we next determine their value. We make valuebased decisions frequently, even when we are not aware of them.

A **vulnerability** is a weakness in the system, for example, in procedures, design, or implementation, that might be exploited to cause loss or harm.

A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm.

A human who exploits a vulnerability perpetrates an **attack** on the system.

Controls prevent **threats** from exercising **vulnerabilities**.

Harm can also be characterized by four acts: **interception**, **interruption**, **modification**, and **fabrication**. **Confidentiality** can suffer if someone **intercepts** data, **availability** is lost if someone or something **interrupts** a flow of data or access to a computer, and **integrity** can fail if someone or something **modifies** data or **fabricates** false data.

Computer security seeks to prevent unauthorized viewing (confidentiality) or modification (integrity) of data while preserving access (availability).

Adversary (threat agent)
An entity that attacks, or is a threat to, a system.

Attack
An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

Countermeasure
An action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.

Risk
An expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular harmful result.

Security Policy
A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources.

System Resource (Asset)
Data contained in an information system; or a service provided by a system; or a system capability, such as processing power or communication bandwidth; or an item of system equipment (i.e., a system component--hardware, firmware, software, or documentation); or a facility that houses system operations and equipment.

Threat
A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Vulnerability
A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

1. Confidentiality – Utajenost

the ability of a system to ensure that an asset is viewed only by authorized parties

Data confidentiality : Assures that private or confidential information is not made available or disclosed to unauthorized individuals. **Privacy** : Assures that individuals control or influence what Information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Nastroje pre zabezpecenia utajenia:

- Sifrovanie
- Kontrola Pristupu
- Autentifikacia
- Autorizacia

- Fyzicka kontrola pristupu

A failure of data confidentiality:

- An unauthorized person accesses a data item.
- An unauthorized process or program accesses a data item.
- A person authorized to access certain data accesses other data not authorized (which is a specialized version of “an unauthorized person accesses a data item”).
- An unauthorized person accesses an approximate data value (for example, not knowing someone’s exact salary but knowing that the salary falls in a particular range or exceeds a particular amount).
- An unauthorized person learns the existence of a piece of data (for example, knowing that a company is developing a certain new product or that talks are underway about the merger of two companies).

2. Integrity - Integrita

the ability of a system to ensure that an asset is modified only by authorized parties

Data integrity : Assures that information and programs are changed only in a specified and authorized manner. **System integrity** : Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Nastroje pre zabezpecenia integrity

- Zalohovanie
- Kontrolne Sumy

we say that we have preserved the integrity of an item, we may mean that the item is:

- precise
- accurate
- unmodified
- modified only in acceptable ways
- modified only by authorized people
- modified only by authorized processes
- consistent
- internally consistent

- meaningful and usable

3. Availability - Dostupnost

the ability of a system to ensure that an asset can be used by any authorized parties

Nastroje:

- Redundancie
- Fyzicka ochrana

an object or service is thought to be available if the following are true:

- It is present in a usable form.
- It has enough capacity to meet the service's needs.
- It is making clear progress, and, if in wait mode, it has a bounded waiting time.
- The service is completed in an acceptable period of time.

We can construct an overall description of availability by combining these goals. Following are some criteria to define availability:

- There is a timely response to our request.
- Resources are allocated fairly so that some requesters are not favored over others.
- Concurrency is controlled; that is, simultaneous access, deadlock management, and exclusive access are supported as required.
- The service or system involved follows a philosophy of fault tolerance, whereby hardware or software faults lead to graceful cessation of service or to work-arounds rather than to crashes and abrupt loss of information. (Cessation does mean end; whether it is graceful or not, ultimately the system is unavailable. However, with fair warning of the system's stopping, the user may be able to move to another system and continue work.)
- The service or system can be used easily and in the way it was intended to be used. (This is a characteristic of usability, but an unusable system may also cause an availability failure.)

4. Further concepts

Authentication: the ability of a system to confirm the identity of a sender

Auditability: the ability of a system to trace all actions related to a given asset.

Anonymity: It is a property that certain records or transactions do not belong to any individual.

The tools for ensuring anonymity are: **Aggregation** - a combination of data from multiple users. **Mixing** - aggregating information from multiple pages and combining it into folders that cannot be broken down. **Proxy** - trusted agents that replace the real identity of the user. **Pseudonym** - a fictitious identity of a user who pretends to be an identity.

Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source. Proves, that data, process and rights of the system are true. Under Integrity. Main tool is Digital Signature.

Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and afteraction recovery and legal action. Because truly secure systems aren't yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

2. Cyptography

Cryptography conceals data against unauthorized access.

An important element in computer security is the use of cryptographic algorithms.

Cryptography - is a scientific discipline that deals mainly with the creation of ciphers, the aim of which is to hide sensitive data from unauthorized persons. Its goal is information systems security with a focus on:

- confidentiality - during data transfer, storage on media
- integrity (data integrity) - correctness of the content of the transmitted message
- authentication - confirmation of the sender's identity

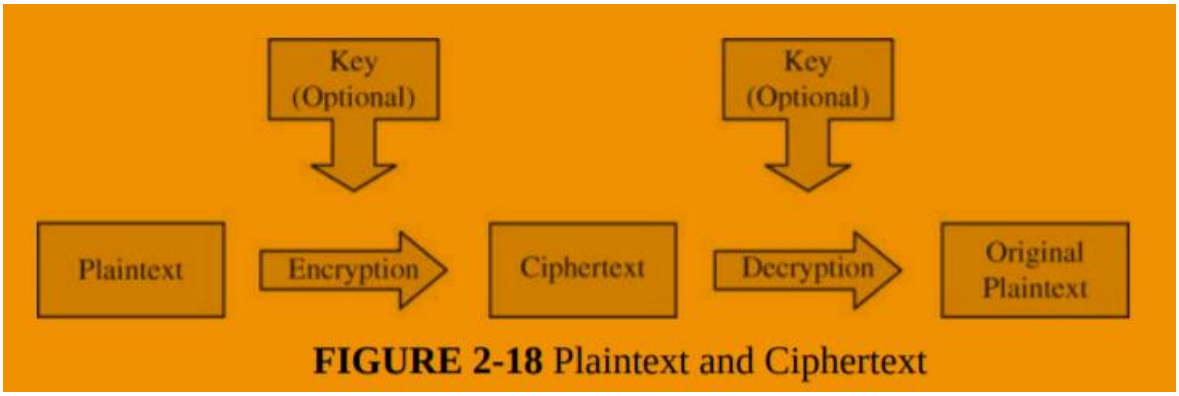


FIGURE 2-18 Plaintext and Ciphertext

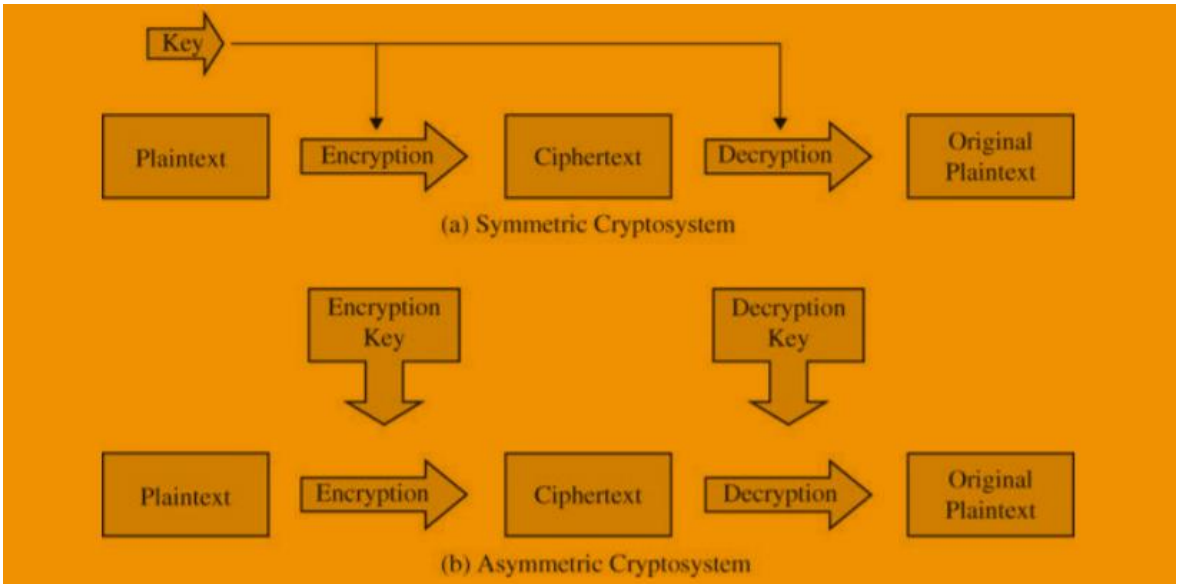


FIGURE 2-19 Symmetric and Asymmetric Encryption

Work factor: amount of effort needed to break an encryption (or mount a successful attack)

Applications:

- SSL and TLS (https://)

- VPN
- Hash on a downloaded file
- Digital signatures

Cipher – a system used to create an encoded or secret message (not key)

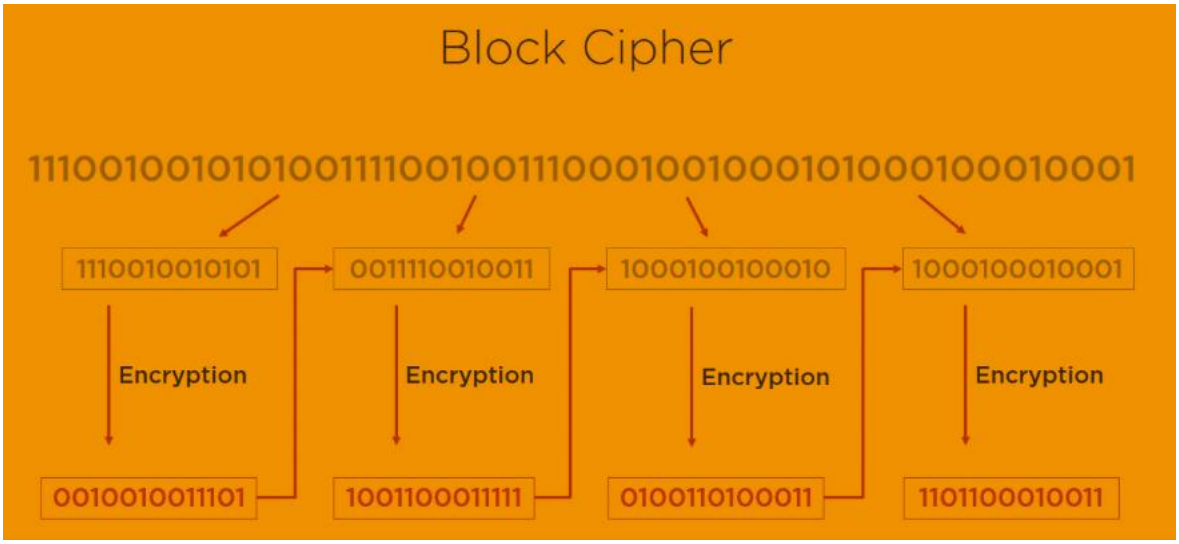
Cryptography and CIA:

- Confidentiality: file, message and link encryption
- Integrity: digital signatures, hashes

1. Stream cipher vs block cipher

2.1.1. Block cipher

A block cipher is an encryption algorithm that encrypts a fixed size of n-bits of data - known as a block - at one time. The usual sizes of each block are 64 bits, 128 bits, and 256 bits. So for example, a 64-bit block cipher will take in 64 bits of plaintext and encrypt it into 64 bits of ciphertext. In cases where bits of plaintext is shorter than the block size, padding schemes are called into play. Majority of the symmetric ciphers used today are actually block ciphers. DES, Triple DES, AES, IDEA, and Blowfish are some of the commonly used encryption algorithms that fall under this group.



2.1.2. Stream Cipher

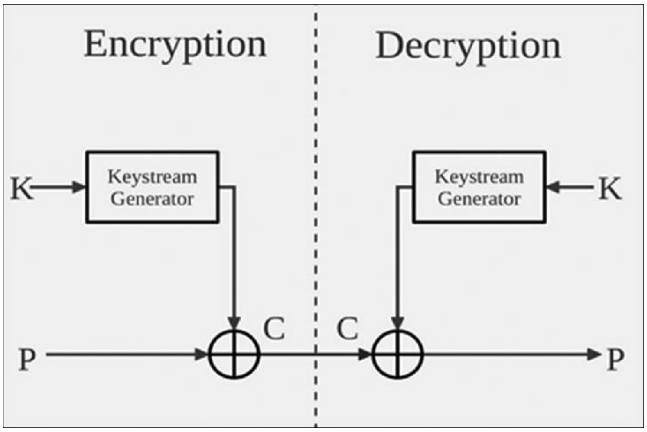
A stream cipher is an encryption algorithm that encrypts 1 bit or byte of plaintext at a time. It uses an infinite stream of pseudorandom bits as the key. For a stream cipher implementation to remain secure, its pseudorandom generator should be unpredictable and the key should never be reused. The pseudorandom keystream is typically generated serially from a random seed value

using digital shift registers. The seed value serves as the cryptographic key for decrypting the ciphertext stream. Stream ciphers represent a different approach to symmetric encryption from block ciphers.

The One-Time Pad, which is supposed to employ a purely random key, can potentially achieve "perfect secrecy". That is, it's supposed to be fully immune to brute force attacks. The problem with the one-time pad is that, in order to create such a cipher, its key should be as long or even longer than the plaintext. In other words, if you have 500 MegaByte video file that you would like to encrypt, you would need a key that's at least 4 Gigabits long.

RC4 - RC4, which stands for Rivest Cipher 4, is the most widely used of all stream ciphers, particularly in software. It's also known as ARCFOUR or ARC4. RC4 steam chiphers have been used in various protocols like WEP and WPA (both security protocols for wireless networks) as well as in TLS. Unfortunately, recent studies have revealed vulnerabilities in RC4, prompting Mozilla and Microsoft to recommend that it be disabled where possible. In fact, RFC 7465 prohibits the use of RC4 in all versions of TLS.

These recent findings will surely allow other stream ciphers (e.g. SALSA, SOSEMANUK, PANAMA, and many others, which already exist but never gained the same popularity as RC4) to emerge and possibly take its place.



2. Symmetric Key Cryptography

This encryption system is called symmetric because the encryption key can be derived from the decryption key and vice versa. Most symmetric algorithms have the same encryption and decryption key. These algorithms, known as secret key algorithms, require both the recipient and the sender to agree in advance on the key they will use. Security is based on the secrecy of the key. Two basic conditions:

1. the need for a strong encryption algorithm

2. both the sender and the recipient must own a copy of the secret key and must keep it secret, otherwise this method does not make sense

This method has 5 basic parts: text, encryption, algorithm, secret key, encrypted text and decoding algorithm

- File encryption for transmision – AES256
- Encryption of files in storage
- Financial transactions – triple DES
- VPC encryption – AES256

Weaknesses:

- Method for secure transfer is needed
- Does not provide non-repudiation
- Key management can be difficult
- Not recommended: digital signatures, key transfer, web security, message security

2.2.1. DES

A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round. Also makes the decryption easy. We can imagine this as a framework for encryption algorithms. Balanced and unbalanced feistel netwoks exist.

The Data Encryption Standard (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure. DES uses a 64-bit block and a 56-bit key. If the block is smaller padding comes.

Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these two types of attack.

DES – 64bit key, 16 rounds, cracked in 1998

Tripe DES (quickfix) – 168bit key, 48 rounds. The changes made encryption time 3 times longer as well.

Blowfish and Twofish – free algorithms

2.2.2. TLS / SSL

Further in Web Security. P.S. worth to learn because you can use in 2 topics! ☺

2.2.3. AES

128, 192, 256 bit keys, 10;12;14 rounds. 128 bit data. With fewer round and longer keys better perofrmance and higher security is achieved.

Permutation and diffusion needs to be introduced into the algorithm. This way the cyper cannot be analysed. We dont want 1 byte in one byte out,that will be easy to analyze. Grid (4x4) apply the cyptio funsion – introduce some permutation, add round key and that one round. Repeat it and done. SP network – Substitution and permutation network.

3. Asymmetric Key Encryption

Pros:

- Secure key exchange
- Authentication + Encryption
- Digital Signature
- Key exchange; Message encryption; Part of web security; Message authenication

Contra

- Slower than symmetric key cryptography
- Not a competition, use both types together

2.3.1. Diffie – Helman

Uses a public and private key to generate symmetric key. The original algorithm was found vulnerable against man in the midle attack. If the attacker intercepts the public key he can send the client its own and read the messages. Can be solved by digital sigantures and an another level of authentification.

2.3.2. RSA

Other: ECC (mbe even better than RSE, less vulnerable against quantum computing, google uses it in its certificate) , el gamal (usually the slowest, extendion of Diffie-Helman)

RSA (Rivest Shamir Adleman) - very widespread and used for asymmetric encryption

The public key is used for encryption and the private key is used for decryption

the encryption principle is that there is no inverse function to the encryption and decryption function that would allow you to decrypt the message with the same key that was used to encrypt the message

the public and secret key is generated by the addressee of the message to be transmitted in encrypted form, the public key is provided by the addressee to the sender of the message

- two large prime numbers p and q are generated
- let n = p.q - the number n is called the module
- let m = lcm [(p-1); (Q-1)] - lowest common multiple
- we find such a number e that the greatest common divisor of the numbers m and e is the number 1
- find another number d such that d.e mod m = 1
- the number e is called the public exponent and the number d is called the private exponent
- the public key is a pair (n, e)
- private key is a pair (d, e)
- at present it is very difficult to obtain the number d from the pair (n, e)

- if someone were able to factorize the number n to the primes p and q, they could calculate the number d and get to the private key
- RSA encryption security is therefore based on the assumption that factoring large numbers is extremely complex
- The discovery of a simpler factorization method could compromise the security of RSA encryption
- encryption security is therefore based on the great computational complexity of factoring large numbers
- For encryption to be secure, the p and q numbers must have at least 100 decimal digits

The base idea: there are mathematical operations which are easier to do in one way than into another. Simplified scenario:

907 and 773 – 2 primes

n = 701,111 = 907 x 773

It is so much easier to determine the results of 907 x 773, than to determine which 2 primes to multiplicate to get 701,111.

4. Use of cryptography to provide integrity

2.4.1. Hashing algorithms

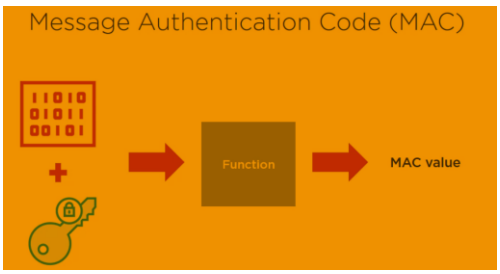
They take data and produce a unique hash value.

MD5 – vulnerability (different input provides the same hash value), not used anymore in SSL.

SHA-1 and SHA-256 (best) haval, tiger

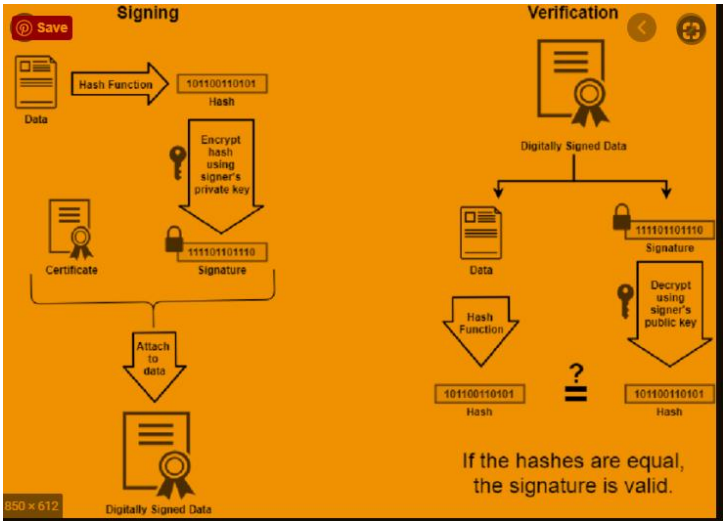
2.4.2. Message Authentication Code (MAC)

The principle is to verify that the key / message sent is actually from the sender. (or authentic). It focuses on whether the content of the message has not been grounded, or whether the message is sent really from the source from which it should have been. It also checks the time and correct order of messages. It uses agreed encryption, so only the sender and recipient know the key.



Hash MAC (HMAC)	CBC-MAC and CMAC
Symmetric key is added to the message	Message encrypted with symmetric block cipher in cipher-block-chaining (CBC) mode
Key and message are hashed	Result is the MAC value
Resulting value is the MAC	Message sent unencrypted with MAC value added
MAC is transmitted with message	Receiver uses symmetric key to validate MAC and sender
Other end uses symmetric key to authenticate MAC value	CMAC is a more secure method of the same process

2.4.3. Digital signatures



Email: S/MIME – Secure Multipurpose Internet Mail Extention, standard for encrypting and digitally signing email. RDS + SHA (example).

PGP – Pretty Good Privacy, RSA + MD5. Examples of different types types of cryptography methods combined.

2.4.4. PKI – Public Key Infrastructure

Trusted third party = **Certificate Authority** (internal or external) they provide **Certificate** using Registration Authority.

Similar to: You go to a certificate to verify that you say the truth when you say you know this and this.

5. Comparison of symmetric and assymetric keys

The problem with symmetric encryption is in key transmission. The K key must be transmitted through a medium. This has been one of the biggest priorities for international espionage in the past. It was no longer possible to transfer the key via an electronic channel, which is very easy to

listen to. Physical transmission, on the other hand, is very slow. Asymmetric encryption solves this problem very effectively. Asymmetric encryption is a series of procedures in which we unambiguously convert the text T1 to the text T2 using the key Kn (n = 1,2). It consists of two parts. The first part (encryption) converts the text M to the text T using the key K1 (usually referred to as the public key). The second part (decryption) converts the text T to the text M, using the key K2 (usually referred to as the private key). In principle, no mathematical procedure can be used to obtain K2 from K1. The K2 private key is a key owned only by the person to whom the message is addressed. K1 is a public key that can be owned by anyone (so that person can provide it for download on the Internet). The text M encrypted with the key K1 can therefore only be decrypted with the key K2, which is only available to the person to whom the message is addressed (it follows that the text T to the text M cannot be decrypted even by the person who encrypted it because he does not have the private key K2 , required for this operation).

3. Identification and autentification

- Identification is the act of asserting who a person is.
- Authentication is the act of proving that asserted identity: that the person is who she says she is.

Identities are typically public or well known. Authentication should be private.

Authorization is the allocation of permissions for specific types of access to restricted information. In the real world, **authorization** is conferred on real human beings; in contrast, information technology normally confers authorization on user **identifiers** (IDs). Computer systems need to link specific Ids to particular authorized users of those IDs. Even inanimate components, such as network interface cards, firewalls, and printers, need IDs. Identification is the process of ascribing an ID to a human being or to another computer or network component. Authentication is the process of binding an ID to a specific entity. For example, authentication of a user's identity generally involves narrowing the range of possible entities claiming to have authorized use of a specific ID down to a single person.

1. Autentification

Authentication is based on something you know, are, have or do.
Authentication mechanisms use any of three qualities to confirm a user's identity:

- What only you know (passwords and passphrases)
- What only you have (tokens: physical keys, smart cards)
- What only you are (static biometrics: fingerprint, face, retina, and iris recognition)
- What only you do (dynamic biometrics: voice, handwriting, and typing recognition).
- Where you are (terminals with strategic plases, GPS signals)

Every password can be guessed; password strength is determined by how many guesses are required.

Problems with passwords:

- Forgotten passwords - saving in sealed envelopes in the vault, sending the password to e-mail ...)
- Password guessing - it is necessary to choose a strong password
- Intentional and unintentional disclosure of a password
- They can be stolen without knowledge of the user

How to store passwords: An encrypted system file is a good way to prevent unauthorized password browsing. It is best to use a one-way cipher to store the password. When a user enters their password, it is encrypted in the same way and then compared. It is necessary that the selected cipher does not provide the same result for different input words. If two users choose the same password, they will easily find out after looking at the password file, as their encrypted passwords will match. Therefore, in Unix + systems, the so-called "Salt" to expand the password. Salt is a 12-bit number composed of system time and process identifier. Salt is unique to each user, and can be stored unencrypted. The encrypted password then consists of the password itself in conjunction with the "salt". Encrypted in this way, even if the same passwords will certainly be different.

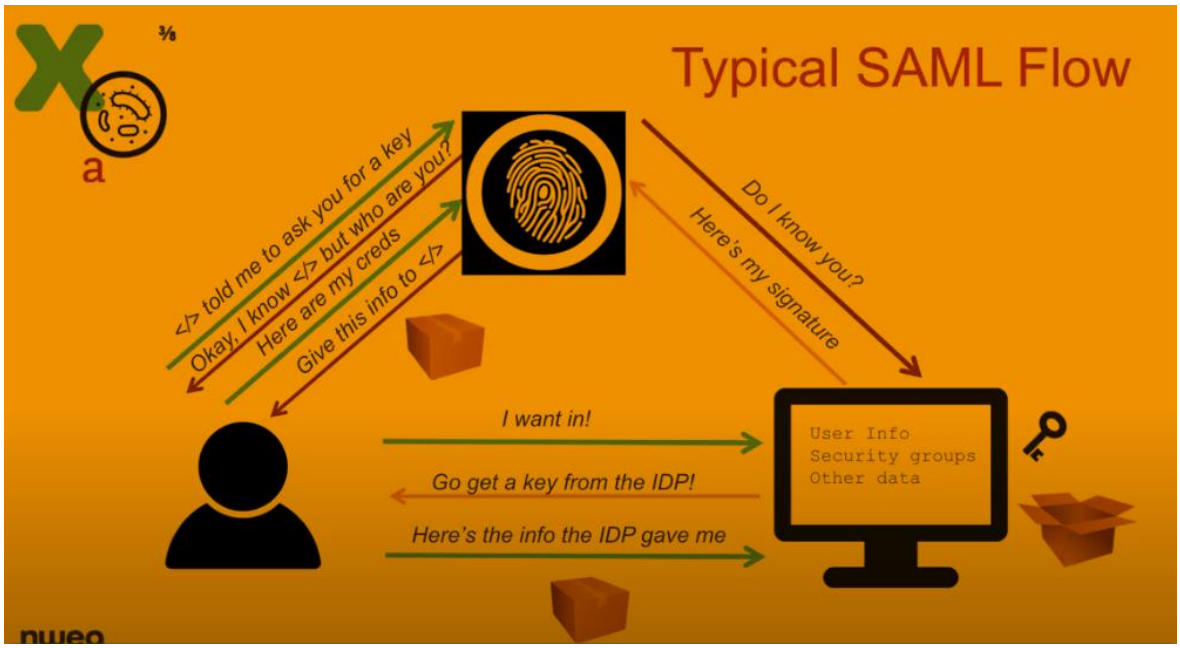
Rainbow Table (database of most frequent passwords) or Brute Force attacks (trying all the possbile combination) are examples of how authentication can be attacked.

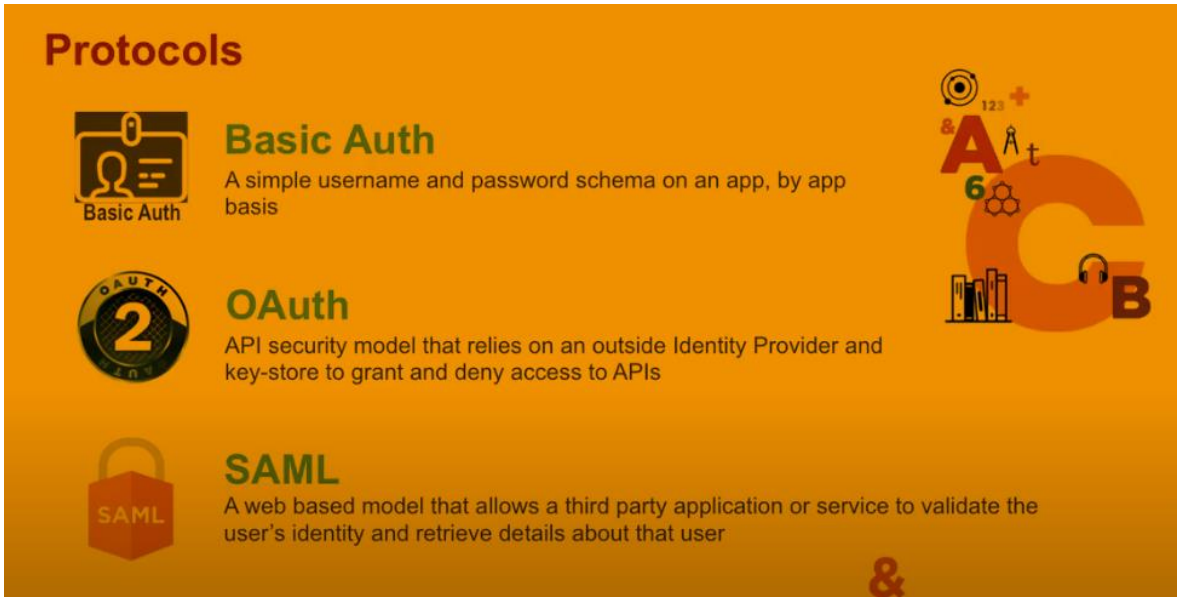
Defensive mechanisms on top of password usage: fraud detection (identify unusual activity and take actions when they happen), MFA (multi factor authentication – using mobile application for example) , security questions.

authentication based on **biometrics** (for example fingerprint) – what you are. Costly.

2. SSO – Single Sign On

the user authenticates once per session, and the system "forwards" this authenticated identity to any process that requires authentication. Of course, such a single-login is not more secure than a single login. The weak point is also that if someone unauthorized authenticates, he can abuse other services. Microsoft Active Directory – a solution for SSO. The advantage of this solution is that reduces the number of passwords required. Secondly, it is easier to deny access for a user to the system (just change in one place, the Active Directory).





3. Further Ideas what to talk about

- Risk of Undetected password theft
- Risk of Undetected password sharing
- Risk of Weakest – by finding out the weakest password other passwords can be revealed
- Dictionary attacks
- Risk of Online Guessing – based on exploiting best practices and personal information
- Risk of Off-Line Dictionary Attacks – checking against hash
- Risk of Password Replay
    1. The hash is sufficient for a dictionary attack unless a salt is used and kept secret.

2. The attacker does not even need to recover the password. Instead, the attacker can replay the hash of the password when needed.
3. send passwords from client to server encrypted using the server's public key !!

Risk of Server Spoofing

Risk of Password Reuse

Authentication Using Recognition of Symbols – pictures the user know

TOKEN-BASED AUTHENTICATION:

1. Card Entry Systems
2. Proximity and Touch Cards
3. Smart Cards and Dongles
4. Soft Tokens
5. One-Time Password Generators
6. Authentication Using Mobile Devices - SMS

BIOMETRIC AUTHENTICATION

CROSS-DOMAIN AUTHENTICATION - SAML

RELATIVE COSTS OF AUTHENTICATION

TECHNOLOGIES

4. Summary

Passwords are widely used in practice and will continue to be a dominant form of user authentication. There are many risks in deploying passwords, and a number of widely used password systems have serious vulnerabilities. Nonetheless, technical measures can mitigate the inherent vulnerabilities of passwords. Although it takes great skill and care, with our current understanding it is technically possible to build and deploy strong password-based authentication systems using commercial products. The truly inherent risks of undetected theft and undetected sharing can be largely mitigated by new technologies, such as intrusion detection systems. Undetected sharing may be deterred further by a system that couples high-value secret data, such as credit card account numbers, with passwords. Tokens are available to generate one-time passwords or to communicate directly with authentication systems. Although costs have been dropping, tokens are still not as widely deployed as early predictions suggested they would be. Biometric authentication has been implemented only infrequently and on a small scale but offers great potential, especially for high-security applications. Interesting new research and applications are extending the use of authentication (and authorization) over untrusted networks between federated organizations.

5. Salting

In cryptography, a salt is random data that is used as an additional input to a one-way function that hashes data, a password or passphrase.

Username	Salt value	String to be hashed	Hashed value = SHA256 (Password + Salt value)
user1	E1F53135E559C253	**password123**E1F53135E559C253	72AE25495A7981C40622D49F9A52E4F1565C90F048F59027BD9C8C8900D5C3D8
user2	84B03D034B409D4E	**password123**84B03D034B409D4E	B4B6603ABC670967E99C7E7F1389E40CD16E78AD38EB1468EC2AA1E62B8BED3A

A fixed salt is when a programmer uses the same salt for every hashed password.

While this will make current rainbow tables useless (if the salt is properly chosen), if the salt is hard-coded into a popular product that salt can be extracted and a new rainbow table can be generated using that salt.

Using a single fixed salt also means that every user who inputs the same password will have the same hash (unless the password hash is also dependent on the username). This makes it easier to attack multiple users by cracking only one hash.

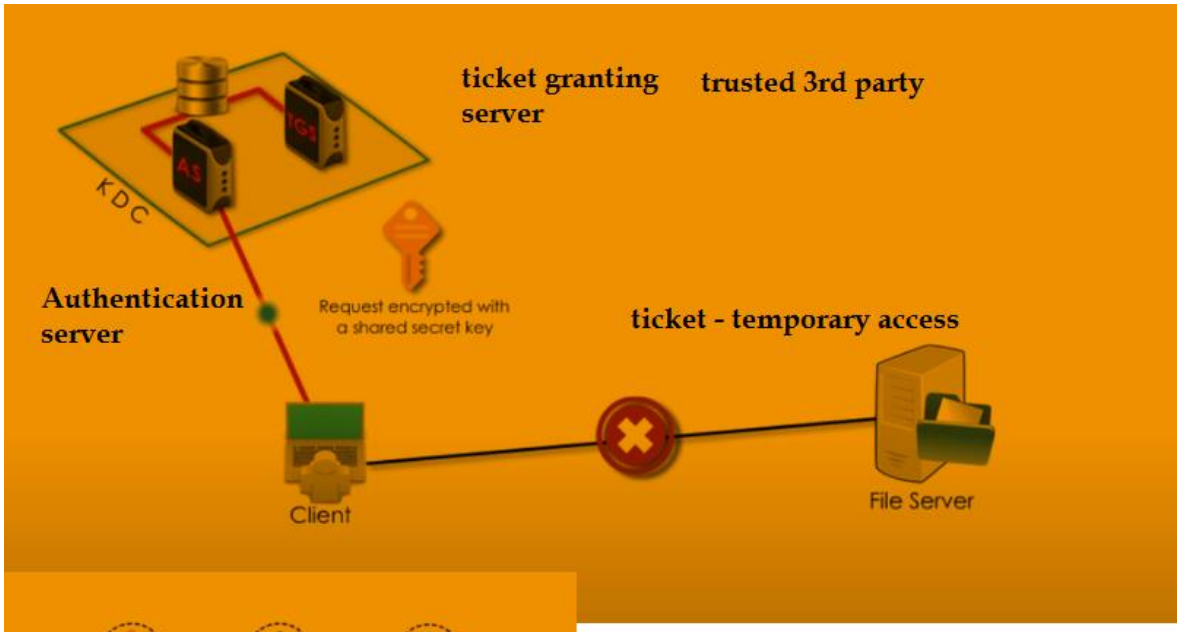
6. Pepper

In cryptography, a pepper is a secret added to an input such as a password prior to being hashed with a cryptographic hash function.

7. Kerberos

Kerberos is a computer-network authentication protocol that works on the basis of tickets to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.

Kerberos builds on symmetric key cryptography and requires a trusted third party, and optionally may use public-key cryptography during certain phases of authentication. Kerberos uses UDP port 88 by default. Kerberos authentication is a method for authenticating both explicit web proxy and transparent web proxy users.



Shared keys
no password sent through the network

4. Access control and security models

- A **subject** is an active entity, such as a process or a user.
- An **object** is a passive entity, such as a file.
- A **right** describes what a subject is allowed to do to an object; for example, the read right gives permission for a subject to read a file.
- The **protection state** of a system simply refers to the rights held by all subjects on the system.

- **Access control** – its goal is to prevent an unauthorized user (subject) from accessing the resource (s), including preventing the use of resources in an unauthorized manner.
- **Access Control Policy (ACP)** – describes what type of access is allowed and under what circumstances.
- **Audit** - independent review and review of system records and activities to test system controls. The aim is to identify breaches of security policy, changes in management and procedures.
- **Autorization –** process of validation if XY has rights to Z
- **Authentification** – validation of the identity of the subject

The system first verifies the identity of the user requesting access. Then the access control function determines whether this user has the required access allowed. The security administrator manages the authorization database in which he defines the permissions for each user on the basis of which the Access Control function grants access. The Audit function monitors and keeps a record of the access of individual subjects (users).

1. DAC, MAC and RBAC

Discretionary access control (DAC) – gives the owner (or anyone authorized to decide on access to the object) the freedom to decide about the object access control. The owner can assign access rights to the object and can decide which rights to assign. Rights can be assigned to the whole group, but also to individual users. Typically, DAC access rights can be changed dynamically. MAC and DAC can also be applied to an object at the same time. In this case, the MAC takes precedence over the DAC. This means that anyone who has permission to access the object through the MAC, and also has permission through the DAC, can actually access the object.

Mandatory access control (MAC) - means that security policy decisions are made outside the property owner. The central authority decides what information is accessible to whom and the

user cannot change the access rights in any way. Defines the access of subjects to objects based on the classification hierarchy of labels. Each object and subject in the system has its own designation. Access to objects is based on a comparison of the designations of the accessing entity and the given object. The control is statically secured. The obligation lies in centralized decision-making based on labeling. Entities cannot influence the decision.

Role-based access control (RBAC) - Responding to the problem of a large number of definitions of access rights due to the large number of objects and entities. Simplify administration, increase performance, simpler scalability of the system (adding, removing objects and entities). Users are assigned to roles. Objects are assigned to groups. Roles have defined rights and can be organized hierarchically with the support of inheritance rights

The special NIST RBAC model consists of four components, the RBAC core, hierarchically RBAC, SSD and DSD relationships. Where the RBAC kernel performs basic functions such as adding and removing users to roles. Creating new roles, etc. Hierarchically, RBAC allows the administrator to view all role assignments to individual users and assign roles to users directly or inheritably. SSD (Static Separation of Duty Relations) allows you to define a set of negative roles so that if a user is assigned to one role in a set, they do not have to be assigned to another role in the set. (role set, n) where there is a restriction that no user can belong to three or more roles in one group. DSD similar to SSD Dynamic Separation of Duty Relation, which restrict user privileges.

The subject is the active entity in the access process (requesting access). An object is a passive entity that is accessed (it is an access object). In general, access to objects is protected through access rights. Access control is provided by a reference monitor. The subject in the authorization process gains access rights to the object (which it can do with the object). The subject does not have to be authorized to access every object and also not all types of access that the object is able to provide.

2. Access control implementation

4.2.1. Access operations

The subject's access rights to the object are realized through access operations. At the lowest level, the subject may monitor the object or the subject may change the object. Accordingly, two access modes are defined:

- Tracking: the subject sees the object and its content.
- Change: the subject changes the content of the object.

4.2.1.1. Unix

expresses access rights to files through three access operations:

• Read: read from file / list the contents of directories

• Write: write to file / create or rename file in directory

• Execute: execute a (program) file / browse the directory

4.2.1.2. Windows NT

the standard access operations are:

• Reading control.

• Cancellation.

• DACL (access control list modification) entry.

• Owner registration (resource owner modification).

• Synchronization (for synchronizing multithreaded programs).

4.2.2. Protection rings

are a particularly simple example of a transitional layer of hardware control of subject access to objects. Each subject (process) and each object, depending on the "importance", is assigned a number. A typical example is the designation of processes with one of the following numbers:

• 0 - kernel OS

• 1 - OS

• 2 - utilities

• 3 - user process

The access control decision is made by comparing the "subject" number and the "object" number. (The outcome of the decision depends on the security policy enforced by the use of protection circuits.)

4.2.3. Access control matrix

Easy, but not scalable (in case we have large nuber of objects or subjects).

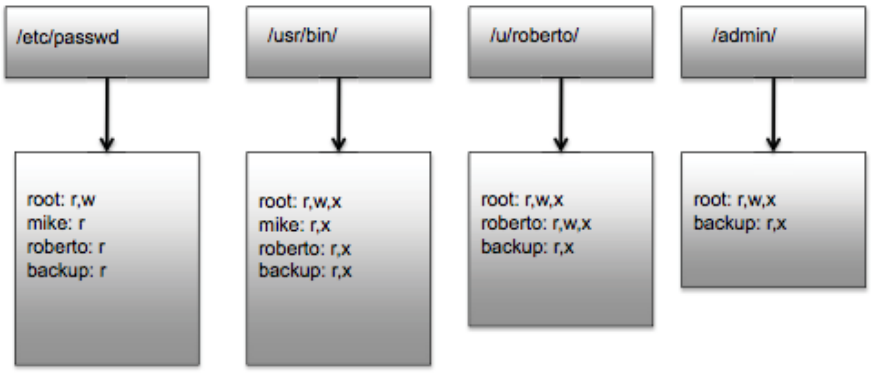
	Process 1	Process 2	File 1	File 2
Process 1	own	read	read, execute	read, write, own
Process 2	write	own	read, write, execute, own	read

EXHIBIT 9.1 Example Access-Control Matrix with Two Processes and Two Files

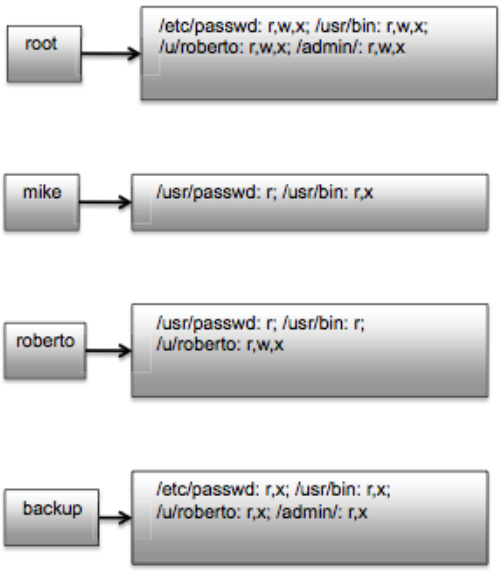
4.2.4. List based access control

List based on the objects.

Easy to implement, but it is hard to find which objects belongs to a given subject. (list all files where mike have write access)



The list can be based on the subjects. Harder to implement.



3. UNIX Access Control

4.3.1. Processes

Each process is identified by a PID (process ID). New processes are created with exec or fork statements. WITH

each process is linked to a real UID / GID and an effective UID / GID. The real UID is inherited from the parent

process, typically the UID of the logged-in user. The effective UID is inherited from the parent process or is

inherited from the file being executed.

4.3.2. Login and password

- in Unix, users are identified by username and authenticated with passwords. When

the boot system login process is started as root. When a user logs in to the system, login

the process verifies the username and password. If the verification is successful, the UID / GID changes according to the user and is performed

the shell from the user's login. On many Unix systems, the password length is limited to 8 characters. System

contains a tool for checking the quality of a password. The password is hashed by the crypt (3) algorithm. This algorithm

it repeats 25 times the slightly modified DES algorithm, where the data block is only 0 and the password is the encryption key.

Hash values are stored in the / etc / passwd file. Shadow password file - in more secure versions of Unix they are

passwords stored in the shadow password file /.secure/etc/passwd and this file is accessible only to root. This file

can be used for the security mechanism of "password aging" and after the expiration of the password to block the account.

Another safe. the mechanism for slowing down dictionary attacks is password salting. The salt is a 12 bit value,

which is added to the custom password and is stored in an open form.

4.3.3. Objects

Objects in access control represent files, directories, storage devices, and I / O devices. It is with everything treated as files. These are organized in a file system into a tree structure. Every a file entry in a directory is a pointer to a data structure called an inode. Each file has its owner and belongs to a group. File permissions are grouped into three triplets that define access read, write, and execute for owner, group, and others or world). A "-" indicates that the right has not been assigned. Their binary form e.g. 777. Remnant of older versions of Unix is a sticky bit. The original purpose of this bit was to ensure that the code segment of the program after the program ended, it remained in the memory swap space (and did not return to virtual memory). The system thus prevented the frequent transfer of program code of a frequently used program from virtual memory to physical memory. Today, the sticky bit is used to restrict the right to cancel a file.

4.3.3.1. Sticky bit

In computing, the sticky bit is a user ownership access right flag that can be assigned to files and directories on Unix-like systems.

When a directory's sticky bit is set, the filesystem treats the files in such directories in a special way so only the file's owner, the directory's owner, or root user can rename or delete the file. Without the sticky bit set, any user with write and execute permissions for the directory can rename or delete contained files, regardless of the file's owner. Typically this is set on the /tmp directory to prevent ordinary users from deleting or moving other users' files.

4. Security models

4.4.1. Bell-La Padula (BLP) model

want to preserve **CONFIDENTIALTY, MILLITARY ORIGINS, typical MAC.**

„no read up“ - A person is not allowed to read something which has higher security level than he has access to (an officer is not allowed to read the field marshall’s notes).

„no write down“ – A person is not allowed to publish something into security category lower than the information requires. (field marshall is not allowed to talk about the strategic decisions to the officer).

4.4.2. Biba model

wants to preserve **INTEGRITY**

„no read down“ – a higly secure datacenter cannot rely on the information from untrusted source.

„no write up“ – an untrusted source cannot publish information into secure datacenter.

4.4.3. Clark and Wilson model

The role of this integrity model is to ensure consistency between internal and external data requirements for this data. It introduces the concept of a well-formed transaction as a sequence of operations that they cause transitioning the system from one consistent state to another consistent state.

5. Malicious code – škodlivý kód

Definition of **malware** – Malware is a program, which is embedded in a system. It is usually hidden, with the intent of compromising the confidentiality, integrity, and availability of data, applications, or the OS.

Basic concepts in the field of malware and their description:

- **Adware** – advertisement integrated into software. Displays ad pop-ups or redirects web browser windows to ad pages.
- **Attack kit** – a toolkit that generates malware. Includes a wide range of advertising and payment mechanisms.
- **Auto rooter –** malicious hacking tool used to infiltrate devices by remote access
- **Backdoor Trapdoor –** mechanism that bypasses the security of the system and allows unauthorized access to the system.
- **Downloaders –** code that is part of malware and is used to download and install other malicious code in the system.
- **Drive by download –** an attack that uses malicious code placed on a web page exploits the web browser's vulnerability to attack the system
- **Exploits –** malware that was created for a specific vulnerability in the system
- **Flooders (dos clients)** – malware that generates a large number of requests and thus performs a DoS attack.
- **Keyloggers –** malware that detects and records keystrokes in the system
- **Macro virus –** malware that is created as a macro or script
- **Logic bomb –** malware code that executes when all logical conditions are met
- **Mobile code –** malware (macro, script, set of instructions) that is platform-independent
- **Rootkit –** a set of tools by which an attacker gains root-level privileges on the system
- **Spammers programs –** malware to send large amounts of spam
- **Spyware –** malware that records information in the system such as keystrokes, creates screenshots, monitors network communication and the files in the system etc.
- **Trojan horse –** malware that has a useful function, but also contains malicious code that bypasses the security mechanisms of the system by using legitimate authorization in the system.
- **Virus –** malware that replicates to executable files that become a means of activation. These files are called infected.

- **Worm –** malware that replicates and runs independently, most often its task is to find all software vulnerabilities and thus execute malicious code.
- **Zombie Bot, robot, drone –** malware that runs on the system without the user's knowledge and causes attacks such as DoS, DDoS.

1. Classification of malware

According to replication:

1. **Malware that replicates** – virus, worm
2. **Malware that does not replicate** – trojan horse, spam

According to dependency:

1. **Independent malware –** worm, trojan horse, bots
2. **Dependent malware –** virus

2. Virus

A program or code that infects executable files or documents. It is replicated to other executable files, documents.

Components of a virus:

1. **Infectious mechanism –** the was the virus spreads in the system, allowing it to replicate. This mechanism is also referred to as an infectious vendor.
2. **Trigger –** an event or condition that determines when a payload is activated, sometimes referred to as a logic bomb.
3. **Payload –** the function that the virus performs, in addition to replication. Includes the execution of malicious code.

Phases of a virus:

1. **Dormant phase –** phase in which the virus is inactive
2. **Propagation phase –** the virus places its replica code in other executable files, documents, or system partitions. The copy of the virus may not be the same as the original version, viruses often morph (change smoothly from one image to another by small gradual steps) to avoid detection.
3. **Triggering phase –** if the conditions are met, the virus is activated to perform the function for which it was created.
4. **Execution phase –** the virus performs its functionality

3. Compression logic of viruses

The virus can be easily detected because the infected file is larger than the uninfected one. One way to work around this protection is to compress an uninfected file and then infect it, to have the original size.

4. Classification according to the target of the attack

- **Boot sector –** infects the main boot sector. The virus is activated when the system boots.
- **File infector –** infects the files that the system considers executable.
- **Macro virus –** infects files (mostly documents) with macros or scripts and is executed directly by the interpreter.
- **Multipartite virus –** infects files in various ways. It can infect different types of files.

5. Classification according to cover and confidentiality

1. **Encrypted virus –** part of the virus generates an encryption key using the mutation engine that encrypts the rest of the virus. If an infected file is executed, the virus is decrypted and executed. The encryption key always changes during virus replication.
2. **Stealth virus –** a virus that proves to hide its existence in the system even from antivirus protection. It can hide itself as a whole, not just part of the code. It achieves this by code mutation, compression, and rootkit techniques.
3. **Polymorphic virus –** a virus that mutates at each replication. It is not possible to determine its unique identification signature. However, it is functionally equivalent.
4. **Metamorphic virus –** a virus that mutates at each replication. It completely changes the structure and content of the code, which increase the complexity of detection and determination of the signature. It is visually and functionally non-equivalent.

6. Macro and script viruses

- Used in the 90’s
- Platform independent
- Infect documents
- Easy to spread, most often using email
- Often used with MS Office documents and Adobe PDF documents
- Antivirus software has strong detection of this type of virus, so their number has dropped rapidly
- Microsoft has implemented the Macro Virus Protection tool to protect its documents

7. Worm

Program that actively searches for multiple devices that it infects. each infected device serves as an automated machine to attack other machines. It exploits software vulnerabilities on client or server programs to gain access to the system. Worms replicate themselves as whole, fully functional versions. It can be spread very quickly via the Internet, emails (as attachments), storage media (USB, CD, DVD etc.). The worm replicates to other systems as much as possible and then performs malicious functionality. The first implementation of the worm was **XEROX** in Alto Labs in 1980.

8. Worm replication

1. **Via email or instant messenger facility –** The worm accesses the mailer, creates its own replica, inserts it in, and sends it to another device.
2. **File sharing –** The worm replicates to removable storage media such as USB, CD, DVD, or infects files, documents.
3. **Remote execution capability –** The worm creates its copy using an explicit remote device or finds a program error in a network service.
4. **Remote file access or transfer capability –** The worm uses a remote file access system and infects other systems.
5. **Remote login capability –** The worm logs on to the remote system as a user and uses commands to infect the system.

9. Phases of worm

1. **Dormant**
2. **Propagation**
3. **Triggering**
4. **Execution**

10. Worm technology

1. **Multiplatform –** The worm is not platform limited. It is created for Windows and Unix distributions or as a macro and script in documents.
2. **Multi-exploit –** The worm penetrates the system in different ways, using different techniques. It uses web servers, email, real-time chat, browsers, shared files, network-oriented applications, or portable storage media
3. **Ultra-fast spreading –** The worm has the ability to reproduce quickly and find as many vulnerabilities in the system as possible in the shortest possible time.
4. **Polymorphism –** The worm can be polymorphic to avoid detection or filtering.

5. **Metamorphism –** In addition to appearance, the pattern of behaviour also changes.
6. **Transport vehicles –** are suitable for spreading large amounts of malicious code such as: DoS Bot, Rootkit, Spam Email Generator, Spyware etc.
7. **Zero-day exploit –** To achieve the maximum effect of the attack and distribution of the worm, it should use an unknown vulnerability, which is revealed only after the attack.

11. Mobile worm

The first was the Cabin in 2004. Reproduction via Bluetooth or MMS. The target of such worms are mobile phones. They cause complete malfunction of the phone, physical damage to electronic circuits and components. They violate data confidentiality, availability, and integrity.

12. Drive by download

It exploits a web browser vulnerability to download and install malware on a system while a user is browsing a web page.

13. Social engineering

Cheating users to unknowingly provide the information needed to attack their own system, or to provide sensitive information.

Common social engineering malicious codes:

- **Spam –** Unwanted email containing an advertisement, an attachment such as a malicious code document, or an installation file containing malicious code. This is the most used phishing attack – redirecting to fake websites and obtaining sensitive information.
- **Trojan horse –** A useful or seemingly useful program or tool that contains hidden malicious code that has unwanted or malicious functions. It hides behind an official utility program that is not downloaded from the official website (most often associated with a spam attack). It can also be presented as a system update or an antivirus program.

The three types of trojan horse:

1. It continues to execute the original program and additionally executes malicious code.
2. It continues to execute the original program but modifies some of its functions.
3. It executes the malicious code and has completely replaced the execution of the original program

14. Agent BOTS attack

They infect multiple systems on the network and use them to launch and manage an attack on one specific system. Most often a type of DDoS attack. Infected systems are called **BOT (ROBOT),**

DRONE, ZOMBIE. They carry out the attack simultaneously and in a coordinated manner. They disrupt data integrity and availability.

BOTNET is a group of infected systems in a network that attack cooperatively and in coordinated manner.

Distribution of BOTS according to usage:

1. **DDoS – Distributed denial of services –** an attack causing the service not to be made available to customers
2. **Spamming –** Using BOTNET, a huge amount of unsolicited email is sent
3. **Sniffing traffic –** BOT uses sniffer packets to monitor network traffic
4. **Spreading new malware –** BOTNET spreads in a large number of other BOTs which infect other systems.
5. **Installing advertising Add-ons and Browser Helper Objects (BHOs)** – AD attacks in web browser, spoofing fake websites and automatic clicks
6. **Attack on IRC chat –** BOTNET methods of attack on IRC chat, popular is CLONE attack
7. **Manipulation of online voting, polls, games –** Each infected system has its own identification on the network, so each vote has the same credibility as a real person's vote.

15. Remote control facility

The BOT is controlled from a central control module. Uses IRC server. The BOTs are connected to a common channel on the IRC server and send commands as messages. They use HTTP communication.

16. Information thief keyloggers and spyware

- **Keylogger –** Records keystrokes to obtain sensitive information. It can also contain a filter that can combine and analyse words, which it then sends to the attacker.
- **Spyware –** Monitors a wide range of activities in the system such as network communication, the content of files in the system, the history in web browsers etc.

17. Information thief phishing

Phishing attack – Uses social engineering to multiply user confidence by pretending to be a trustworthy resource.

Spam email – Contains false information and a link to a fake page that mimics the login page of a bank, online games, social networks. Puts pressure on the victim, as an urgent action to change the password for security reasons or account recovery that requires re-entering sensitive data. In principle, it collects personal data about the user and can use it to take over the user's identity.

Spear phishing attack – Very dangerous attack. Email is created specifically for a particular person, often containing more information already found, which increases credibility.

18. Stealthing backdoor

It is also known as **trapdoor**. A secret entry point into a system that bypasses all security mechanisms. Legitimately used to test and debug systems, it is also called as **maintenance hook**. The attacker gains unauthorized access to the system.

19. Stealthing rootkit

A set of programs in the system that have root rights. They hide the proof of their existence as much as possible. It allows access to all functions and services of the OS. Attacker with root privileges has full control over the system.

Classification of rootkit:

1. **Persistent –** activates each time the system boots
2. **Memory based –** stored in temporary memory
3. **User mode –** intercepts API calls in user mode and activates
4. **Kernel mode –** intercepts API calls in kernel mode and activates
5. **External mode –** the malware is out of operating mode

Higher layer malware detection in user mode is possible using antiviruses. Detection at the lower layer of the kernel mode is not easy because the antivirus does not have access to the kernel.

20. Counter-measuring malware

The ideal solution to the malware threat is prevention.

4 basic elements of prevention:

1. **Policy**
2. **Awareness**
3. **Vulnerability mitigation**
4. **Threat mitigation**

When prevention fails, safety mitigation mechanisms can be used:

1. **Detection**
2. **Identification**
3. **Removal –** Malicious software will be removed, if it cannot be removed, the infected files will be replaced with new ones from the system backup

Malware security mechanisms should meet the following requirements:

- **Generality –** the chosen mechanism should handle all types of malware
- **TimeLines –** respond as soon as possible to prevent further dissemination and corruption of the password
- **Resiliency –** the resilience of the mechanism to malware hiding techniques
- **Minimal DoS cost –** minimum reduction in the number of normal operation services
- **Transparency –** the software should not require modification of the system or hardware
- **Global and local coverage –** global and local availability, dealing with attacks from both inside and outside

21. Generations of antivirus systems

1. **Generation of simple scanner –** only known malware was detected, it needed unambiguous identification of the malware
2. **Generation of heuristic scanner –** uses heuristic rules to detect the virus. Did not rely on specific identification
3. **Generation of activity traps –** identification of malware by its actions
4. **Generation of full-featured protection –** combining a number of virus detection techniques. It incorporates all the previously mentioned techniques.

22. Generic descryption

Used for easy and fast detection of polymorphic viruses. The scanner contains:

- **CPU emulator –** software that creates a virtual machine
- **Virus signature scanner –** checks the code for a unique identification of the malware
- **Emulator control module –** controls the entire execution. If the malware is encrypted, it decrypts it. If it is hidden, it detects it. Malicious code cannot perform a malicious operation because it is integrated in a fully managed environment.

23. Host based behaviour blocking software

Integrated with the OS. Monitors program behaviour in real-time against malicious code. Blocks a potentially harmful action before it has a chance to hit the system. It blocks the program in real-time, so it has an advantage over antivirus programs.

24. Worm protection mechanisms

1. **Signature based worm scan filtering –** the filter detects the worm based on its signature

2. **Payload classification-based worm containment –** based on network technology, it checks packets for worms
3. **Filter based worm containment –** the filter detects the worm based on its code, not the signature
4. **Threshold random walk scan detection –** random detection in the system
5. **Rare limit –** slows down the network connection of a potentially compromised system
6. **Rare halting –** blocks the network connection of a potentially compromised system

6. Program security – programova bezpecnost

1. Categories of software errors

1. **Unsecured interaction between components**
2. **Risky resource management**
3. **''Holey'' defence (lyukas xd)**

2. Safe programming

A form of design that will ensure the operation of the program despite unexpected use. Requires attention to all aspects of program implementation. Requires the examination of all potential problems and the verification of all potential inputs. We never expect a given function or library to work as we expect, so we ensure it in the code. It calls for a return to traditional programming methods. The programmer must find out what errors arise and reduce the chance of their occurrence. This method conflicts with the business (there we try to complete the product as quickly as possible).

3. Security as design

There is a high tolerance for security errors in software development today. Nevertheless, there are many quality standards.

4. Program input

Improper work with inputs is a common problem. We need to identify all data sources. The input is any external data source. Its value is not explicitly known to the programmer at the time of writing the code. We need to explicitly validate assumptions about the size and type of the value before use.

5. Stack overflow

Happens, when the allocated container size is not verified - the result may be its overflow. Test inputs may not detect this crack. Safe programming considers each input dangerous. The process tries to store data beyond the defined boundaries, the result is an incorrect program run. The data is also stored beyond the defined boundaries. Most often it is the result of an error. Consequences: a code may be executed that should not normally be executed. This is the most common way to gain unauthorized access to data. Gives the attacker the opportunity to run any application, which is a high security risk. Protection – allocate the stack dynamically, use secure functions, do not run the program with root rights. This mainly concerns the C language, which has no overflow protection.

6. Interpretation of input

Binary or textual. The binary interpretation depends on the coding and is normally application specific. Input must be validated. Incorrect validation can cause security crack. The input data should be compared with what we expected. They can also be compared with known dangerous values. By accepting only safe inputs there is a better chance that the program will remain safe.

7. SQL injection

It is a technique of attacking the DB layer of a program by inserting code through untreated input and executing its own modified SQL query. This unwanted behaviour arises when the application layer is connected to the DB layer (these are almost always two different programs). Many web applications get user input from a form. Often this input is used immediately to construct an SQL query that is sent to the DB. For example:

SELECT user FROM table WHERE name = 'user_input';

An SQL injection attack consists of inserting SQL statements into an input.

8. Cross site scripting (XSS)

Method of disrupting web pages by using security holes in scripts (untreated inputs). The attacker pushes his Javascript code into the page. May damage the appearance of the site, disable it, or obtain sensitive information about visitors.

9. Fuzzy input

Input testing technique. Uses randomly generated data as program inputs. The range of inputs is huge. The goal is to determine if the program will correctly evaluate the abnormal inputs. It is a simple and cheap solution that helps both security and reliability of the program. It can also use templates to generate classes with problem inputs.

10. The risks of writing safe code

- **Correct implementation of the algorithm**
- **Correct processor instructions for the algorithm**
- **Valid data manipulation**

11. Comparison of machine language with algorithm

We assume that the compiler / interpreter generates / executes code that validly implements language commands – it a mistake. It is necessary to compare the code of the machine

with the original source, which is difficult and time consuming. There are computer systems with such a level of security that they require such control.

12. Data interpretation

The interpretation depends on the executed instruction of the machine. Languages offer different possibilities for restricting and validating data in variables. Strongly typed languages are more restrictive but safer. Other languages allow the program to explicitly change the interpretation of the data.

13. OS interaction

Multi-user concept. The files are user-owned and have a wide range of rights for different groups of users. What if multiple programs work with a shared file at the same time? – it needs to be ensured in the system. Synchronization mechanisms are needed, for example a **lock** on a shared file. The process must create and own a **lockfile** to gain control of the share file.

14. Environment variables

It is a collection of string process values inherited from its ancestor. It can affect the behaviour of the executed process. They are included in memory during build. They can be modified by the program – the modifications will also affect the descendants. They are a source of unreliable inputs. The most common use – the local user is trying to gain privileges. Programs can be vulnerable due to manipulation with PATH variable (LD_LIBRARY_PATH during dynamic linking).

15. Root/Admin privileges

Programs with root privileges are most often the target of an attack. Root privilege is needed to manage protected system resources. This privilege is often needed at system start-up (not later). Good design is the division of complex programs into smaller modules, which provides isolation between components and reduces damage if the protection of one module is violated.

16. System calls and standard library functions

Programs use them to perform normal operations.

17. Safe temporary files

Many programs use them. They are usually in a known, shared system directory. They must be unique, not shared with others. They usually use a name consisting of the process ID, which is unique but predictable – the attacker can try his own temporary file. Security requires the creation of temporary files with random names.

18. Program outputs

Binary or textual. Displayed, saved, sent. It is important that only valid output is provided. The character set used should be defined.

7. OS security – Bezp. OS

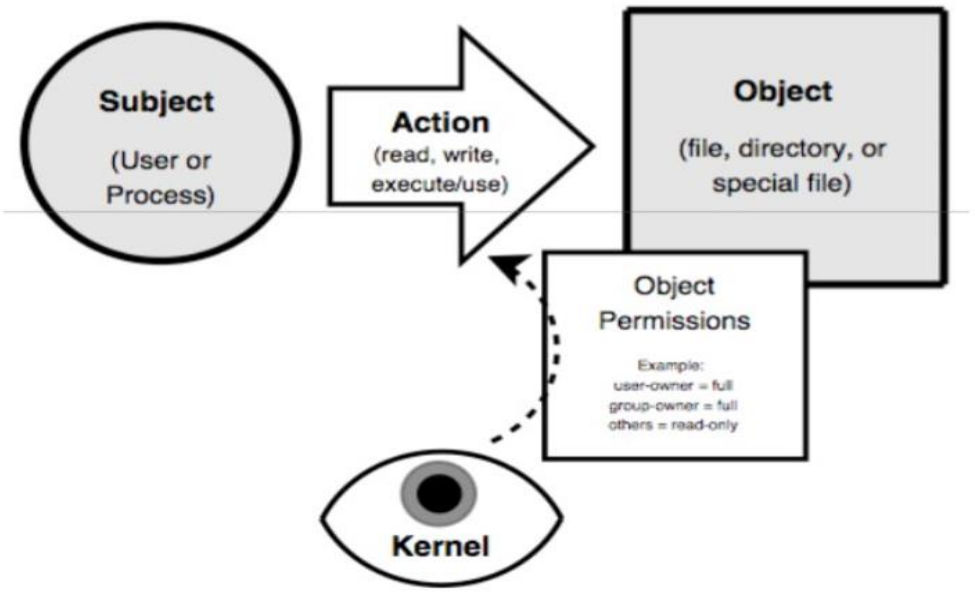
1. Unix

Unix security architecture

Although most OSs have a security architecture that explains how security is enforced and where security-relevant data is stored – **security features** were built into Unix when such a need arose (security was not the original design goal). Unix was originally designed for small multi-user computers in a network environment and was later scaled for commercial servers and even later scaled for PCs. Unix, like the Internet, was developed for friendly environments (non-malicious users), such as research labs and universities. Its security mechanisms were weak. As Unix evolved, new security measures were built into it to reinforce existing measures. When introducing new security measures, the designers made an anxious effort to minimally interfere with the existing Unix structure. Unix design philosophy assumes that security is managed by an experienced administrator and not a regular user. Therefore, security management tools are often in the form of scripts and command lines.

Access control

Superuser (SU) management – the root account is used by the OS to perform its basic tasks, but also for certain other system administration tasks. SU privileges are a major weakness of Unix. If an attacker gains SU status, he takes control of the entire system. Therefore, access to SU status must be very carefully protected. In Linux, everything is a file: memory, device drivers, named pipes, other system resources, the data files themselves, folders. Access control is as shown:



Users and groups

A user is someone who can use files. It can be a human, or a process, a thread. The group is a list of users. Each user has assigned at least 1 group, but a user can be part of more groups too. Users and groups are not files. All user data is in /etc/password and groups in /etc/group.

Access management

Files have two owners: user and group. Linux uses Discretionary Access Control (DAC), a type of access control based on which it determines to which group or user an object (file) belongs. Access to files is therefore determined for the user entity, group, and others. 3 types of rights can be specified for each entity: read, write, execute. The rights for the file are set with chmod (change mode) command. Rights for a file may look as shown:

-rwxrw-r—

On the first place: can be ‘’d’’ or ‘’-’’. In case of ‘’d’’, the file is a directory. This is followed by three triplets of characters r, w, x. The first three – the owner, the second three – the group, the third three – the others. If there is a ‘’-’’, the right is denied. So, if we have the right to a file as shown above, it means that the owner has the right to read, write and execute. The group has the right to read and write only. Others have the right to read only. But root can do anything, regardless of file rights. There is a possibility to set the folder with so called **Sticky Bit**. The file can only be deleted by the owner, or root. This is reflected in the file by adding a ‘’t’’ of ‘’T’’:

drwxrw-r-T

The numeric notation is calculated as the sum of the values of the individual rights for each entity separately. The values are r – 4, w – 2, x – 1. Thus, the notation:

-rwxrw-r—

can otherwise be written as 7 (4+2+1), 6 (4+2), 4 (4) i.e. 764

SetUID and **SetGID** are special file rights settings. It means ‘’run as’’. If SetUID is set, it means ‘’run as owner’’. SetGID – ‘’run as group’’. It is dangerous if the owner is root – it means full access rights from the running process (it does not matter who executes it). Used only for executable (x) files. The entry then looks like:

-rwsrw-r—(setuid)

-rwxrwsr—(setgid)

Chroot is setting for program. It provides the program with access to only a certain part of the file subsystem and maps it as ‘’/’’. This is useful for some processes: FTP should only have access to certain files. Files outside the chroot are not visible or reachable.

System security

Security begins with the installation. It is important not to leave services and programs installed and running on the system that are not needed. They can be the target of an attack. It is generally recommended to disable: X Window system, RPC services, R-services, inetd, SMTP daemons, telnet. The initial configuration should include:

- **Setting the root password**
- **Creating non-root users**
- **Setting up the firewall**
- **SELinux permission**
- **Setting up regular updates and patches**

The update should be downloaded automatically, but not installed automatically. The system must first be tested with new updates.

A rootkit is a set of computer programs and technologies that can be used to mask the presence of malicious software on a computer. It does it by hiding the directories in which they are installed, API calls, Windows registry entries, processes, network connections, and system services so that the presence of malicious software is not detectable by commonly available system resources.

MAC

Mandatory Access Control (MAC) is a type of access control in computer security, according to which the OS has the ability to give an entity or initiator access or generally have it perform some kind of operation on an object or target. In practice, the subject is usually a process or thread. The objects are files, folders, ports, shared memory segments, and more. Subjects and objects each have a set of security attributes. Whenever an entity attempts to access an object, an operating rule is initiated by the OS kernel, which checks security attributes and decides whether to grant access rights. Any operation performed by any entity on any object will be tested against a set of authorization rules, leading to a decision as to whether the operation will be allowed. MAC is a security policy controlled centrally by an administrator.

SELinux

SELinux (Security-Enhanced Linux) is an extension of the Linux kernel in computer science with MAC to increase computer security. SELinux allows us to define permissions to perform a particular operation at the level of individual processes, users, sockets, and so on, by placing calls to its control routines at critical points in the kernel. This increases system overhead, but it is possible to prevent the program from performing a potentially dangerous action that could lead to elevation of authority.

In connection with SELinux, special sandboxes called domains are being introduced, which also prevent programs under SU rights from leaving this domain. If an attacker manages to take control of a program, he can perform only those operations that the program was allowed to do, even in the case of programs under SU privileges.

SELinux is not a Linux distribution, but a set of adjustments and modifications to the system kernel itself, including the addition of user tools. SELinux implements MAC in addition to the classic DAC authorization system used by Unix systems as well as many Windows NT systems.

SELinux users and roles are not related to current users and roles of the system itself. For each regular user and process, SELinux assigns 3 basic pieces of information such as role, username, and domain (or type). This system is more flexible than normal. The rules for when a given user can get to a certain domain are defined in the SELinux policy.

In SELinux, **''what is not explicitly allowed is forbidden''**.

AppArmor

AppArmor is an extension of the Linux kernel in computer science with MAC to increase computer security. AppArmor provides protection against viruses and other malware. SELinux is very often compared to it. However, AppArmor does not load the system as much as SELinux (SELinux slows down system performance by up to about 7%, while AppArmor only slows down by 1-2%).

2. Windows

Unlike Linux, Windows basically offers more specific file access control. It is possible to define all users and their rights for each file. In Linux, file rights can only be determined based on owner, group, and rights for others. Windows ME (Millennium Edition – basically the Windows 2000) and earlier versions did not offer any form of access control. Windows XP implemented a

DAC, but most applications still needed to be ''run as an administrator''. Windows Vista has an improved rights control model called **UAC**.

User Account Control (UAC) is a security technology first introduced in the Microsoft Windows Vista OS. It increases Microsoft Windows security by restricting application permissions to the user level until the admin confirms that the application has been elevated. Only applications that the user trusts will be given higher privileges, and malware is prevented from obtaining privileges to damage the system. In other words, the user account may have administrator privileges, but the applications launched by the user may not. Only applications for which higher permissions have been set in advance, or those that the user will temporarily allow at the system query.

Registry is a central database for OS configuration data. DB entries are called keys (not encryption related keys!!!). Using the Registry Editor (regedit.exe, regedt32.exe), it is possible to modify the database and display entries. At the highest level, the registry has five important predefined keys:

- **HKEY_CLASSES_ROOT –** contains an association to the name of the file extension, it is possible to specify that .doc files will be processed by MS Word
- **HKEY_CURRENT_USER –** contains the config information of the currently logged in user
- **HKEY_LOCAL_MACHINE –** contains config information about the local machine
- **HKEY_USERS –** contains all actively loaded user profiles on the system
- **HKEY_CURRENT_CONFIG –** contains information about the hardware profile used on the local machine when starting the system

Registry hive is a group of keys, subkeys, and values in a registry. Using the registry, the system can be tailored to user requirements and predefined protections can be set. By modifying registry keys, an attacker can modify the behaviour of the OS, for example:

- The registry key may point to a location (path) where the OS automatically searches for certain executable files.
- If the set access rights to this key are weak (write access for everyone), then an attacker can insert malicious code by modifying the key.
- It is an absolute necessity to protect the integrity of the registry data.
- Removing the registry editor from all machines that are not used for system management is a good safety practice.

- Some security-relevant keys cannot even be changed by the registry editor, but only by a specific utility.

Identification and authentication

Security-relevant user information is stored by the SAM in the user account DB. User accounts are edited using the User Manager for Domains utility. The following fields can be defined in the user account:

- **Username –** unique name used for login
- **Full name –** the name of the user who owns the account
- **Expiration date –** the default value is no expiration
- **Password dates –** time of the last pw change, time of pw expiration, time from when the pw should be changed, it is also possible to set whether the pw should be changed, it is also possible to set whether the user can change the pw independently
- **Logon hours and workstations –** it is possible to specify when and from which workstation the user can log in
- **User profile path and logon script name –** the profile specifies the user's workstation environment (program groups, network connections, screen colours, etc.). A logon script is a batch file (executable file) that is automatically executed when a user logs on.
- **Home directory –** It is possible to specify whether the home directory is on a local machine or on a network server.
- **Local and global groups –** groups of which the user is a member

The OS supports security management through pre-set accounts. There are three types of pre-set user and group accounts:

- **Predefined accounts –** installed by the OS
- **Embedded accounts –** installed by the OS, applications, or services
- **Default accounts –** are created by default when accessing network resources

Pre-set users and groups created by the OS can be modified, but not deleted, e.g. LocalSystem is a built-in account used to perform system processes and handle system-level tasks. Only certain processes can log in to this account. The predefined Administrator accounts (cannot be cancelled or blocked, have full access to all objects) and Guest (occasional access) are installed locally.

Access control management

Access control in Windows is more complex than access control in a typical file system (objects are files, registry keys, Active Directory objects, etc.). The policy structuring tools in Windows 2000 are groups, roles, and inheritance.

Subjects in the OS are active entities: local user, alias, domain user, group, or machine. Subjects have a human-readable name (username) and a machine-readable name SID (security identifier). SID is an individual subject. A global group is a SID file managed by a DC. The group also has its own group SID – groups can be nested. A group member can use the privileges of the group. (The group created an intermediate – transitional layer of control.) Object permissions are assigned to the group. Users gain access to objects by becoming members of this group.

An **alias** is a set of user and group SIDs managed by a DC or a local LSA.

The **security identifier (SID)** is constructed when creating a user account and is unchanged for the entire duration of the account. The SID format is S-R-I-SA-SA-SA-N, where S – letter S, R – revision number (now 1), I – authority identifier (48 bits), SA – sub-authority (32 bits), N – relative identifier which is unique in the authority namespace.

Privileges control access to system resources. They are uniquely identified by their program name. Privileges are different from access rights. Typical privileges are: backup files and directories, generate security audits, manage and audit security log, take ownership of files and other objects, bypass traverse checking, enable computer and user accounts to be trusted for delegation, shut down the system.

Objects are passive entities in access operations. Windows 2000 recognizes:

- Executable objects such as processes and threads
- File system objects such as files or directories
- Other objects are registry keys and devices such as a printer
- Lockable objects have a security descriptor. For embedded objects, security descriptors are managed by the OS. For private objects, security descriptors must be managed by the application software.

Authorization is authorization to perform a certain operation on an object.

Access rights correspond to operations that can be performed on an object. The standard access rights that are applicable to most objects are: DELETE, READ_CONTROL, WRITE_DAC, WRITE_OWNER, SYNCHRONIZE.

Security aspects of DLL

Dynamic linking allows the module to include and use only the necessary information from the exported DLL. When an application loads a DLL without a specified path name, Windows tries to load that DLL from a predefined set of directories in a specific order. If an attacker gains control of one of these directories, he can place his malicious copy of the DLL in it. This attack is called a **DLL preloading attack,** and if the application runs with admin rights, the attacker gains these rights in this way.

3. Boot sequence

The process of loading the OS into memory. When we turn on the computer, the code stored in the firmware known as the BIOS (Basic Input Output System) is executed. On modern systems, the BIOS loads a secondary boot loader into memory, which takes care of loading the OS itself into the memory. Protection can be provided by setting a BIOS password that prevents a secondary boot.

Hibernation

Modern computers can be put into hibernation. When entering this state, the OS saves the entire contents of the memory to disk. Upon the ''awakening'' of the computer, this content is loaded back into memory. An attacker could copy this content to access passwords or application states stored in memory.

Password protection

To break the password, a dictionary attack is first used, i.e. testing the most used passwords. A dictionary of 500 000 words is enough to break most passwords. For better protection, passwords are ''salted''.

Salt is a random string of characters that is added to a password (at the beginning or end). This will artificially increase the strength of the password. For example, if we have the password 12345, we usually have its hash stored in the DB. If we use the MD5 algorithm, the DB will contain the string 827ccb0eea8a706c4c34a16891f84e7b. But this hash is well-known, so the attacker immediately knows that the password is 12345. Although, if we add the salt „KYXrWc7W8dsa8KNN“ to the password, then the password will be 12345KYXrWc7W8dsa8KNN, and the hash in the DB will be bdc4f5bee2cbc8e02af084c880a260cc. This hash is no longer known, so the attacker only has to guess the password using brute force (try all possible combinations of letters and numbers). For the

most efficient implementation, it is necessary to store a random string for each user separately. Of course, salt is only effective if passwords are hashed.

4. Virtual OS

The OS can be run in another OS environment. For example, Windows on OS X. This ensures the isolation of the guest OS and thus the security is increased. The guest system can allow or deny access to certain system resources (memory, processor) and devices (USB).

8. Security of database systems – Bezp. databazovych sys.

1. Introduction to relational databases

Database concept

A database is a set of data and a set of rules that organize data by determining certain relationships between data. Through these rules, the user describes a logical format for the data. Data items are stored in a file, but the exact physical file format does not have to bother the user. A db administrator is a person who defines the rules that organize data and manages who should have access to what part of the data. The user works with the db through a program called a database manager or a database management system (DBMS), informally known as the front end.

Components of db:

- **Record –** The db file consists of records, each of which contains one group of related data. Each record contains fields or elements, i.e. its own basic data elements.
- **Scheme –** The logical structure of a db is called a scheme. A specific user can have access only to a part of the db, the so-called sub-scheme. Schemes and sub-schemes can only be used to present to users those elements that they wish or need to see. Db rules identify named columns. The name of each column is called the db attribute. A relationship is a set of columns and defines clusters of related data values in much the same way as a ''someone's mother'' relationship specifies a relationship between pairs of people.
- **Queries –** Users communicate with db admins through DBMS commands that search, modify, add, or delete fields and records from the db. The commands are called queries. DBMSs have precise syntax rules for queries. Most query languages use notifications similar to English, and many are based on SQL, a structured query language originally developed by IBM.

Advantages of using dbs:

A db is a simple set of data stored and maintained in one central location that many people can access as needed. However, the current implementation may include some additional physical storage or access arrangement. The essence of a good db is the fact that users are not aware of the physical layout, that all they see is a single logical layout. As a result, the database offers many advantages over simple file systems:

- **Shared access** so that many users can use one common, centralized data set.

- **Minimal redundancy** so that individual users do not have to collect and manage their own data groups.
- **Consistency of data** so that changing the value of data affects the data values of all users.
- **Data integrity** so that data values are protected against accidental or dangerous, unwanted changes.
- **Controlled access** so that only authorized users can view or change data values.

The **DBMS** is designed to provide these benefits effectively. However, the individual goals conflict with each other. For example, security interests may conflict with performance, because security considerations often increase the demands on a computer system in size or complexity. Surprisingly, however, security interests can also reduce the system's ability to provide data to users by limiting some queries that might otherwise seem harmless.

2. Safety requirements

- **Physical integrity of the db –** The db data is immune to physical problems such as power outages and it is possible to reconstruct the db if it is destroyed in a disaster.
- **Logical integrity of the db –** The structure of the db is preserved, e.g. with logical database integrity, changing the value of one field does not affect other areas.
- **Integrity of elements –** The data contained in each element is accurate.
- **Verifiability –** It is possible to track who or what accessed or modified elements in the db.
- **Access control –** The user is only allowed access to his accessible data.
- **User authentication –** Each user is uniquely identified to monitor progress and to allow access to certain data.
- **Availability –** In general, users can access the db and all the data to which they are authorized.

3. Access control

Key words for access control are **GRANT** and **REVOKE**. DBs are often **logically separated according to user access rights**, e.g. all users can access general data, but only the HR department can receive salary records, only the marketing department can obtain sales data. DBs are very useful because they centralize data storage and maintenance. Limited access is both a responsibility and a benefit of this centralization.

The **db admin** determines who should be allowed access to data, views, relations, fields, records, or elements. The DBMS must enforce this policy, provide access to all listed data, or deny

access when it is needed. In addition, there are many ways to access it. The user or program may have the right to read, modify, delete entire fields or records, or reorganize the entire db.

DB access control looks like access control for Oss or other components of a computing system. However, the db problem is more complex. OS objects such as files are unrelated items, but records, fields and elements are related. Although the user cannot determine the contents of one file by reading others, he may be able to identify one data element by reading another. The problem of obtaining data from other values is called **inference/deduction**. The data can be accessed by judgment without the need for direct access to the object itself. Restriction of deduction may also mean prohibiting certain pathways to prevent possible deduction. However, restricting access to deduction control restricts queries from users who do not intend to gain unauthorized access to values. Attempts to control the required approaches to prevent possible unacceptable deductions can degrade DBMS performance.

Size or graininess varies between OS and db objects. Implementing an ACL for several hundreds of files is much easier than implementing an ACL for a db with several hundreds of files (each with about one hundred fields). Size affects processing efficiency.

Role-based access control (RBAC)

- **Application owner** - The end user who owns the db objects (tables, columns, rows) as part of the application. This means that db objects are generated for applications or are ready for use in applications.
- **End user other than the application owner** – An end user who works with the db objects through a specific application but does not own any of the db objects.
- **Administrator –** A user who has proper responsibility for part or all of the db.

The RBAC db must provide the following options:

- Create and delete roles
- Define permission for the role
- Assign and unassign users to roles

Inference attack

Inference is a technique used to attack databases, where malicious users steal a wealth of sensitive information from complex, high-level dbs. Essentially, inference is a **data mining** technique used to search for information hidden from ordinary users. Deriving an attack can compromise the integrity of the entire db. The more complex the db, the greater security should

be implemented. If the problems are not solved effectively, sensitive information may be accessible to foreigners.

Inference channel is an access path outside the access authorization control system, i.e. unauthorized access to data through which sensitive data can be stolen (in short, we bypass the security gateways and we access directly to the db).

4. Statistical database (SDB)

A db that uses mathematical operations such as SUM or AVG to provide data to users. There are 2 types of SDB:

1. **(Only) Statistical dbs –** this type of db only stores statistics
2. **Ordinary dbs with statistical access -** this type of db contains separate records. The db supports a population of nonstatistical users who are allowed access to selected parts of the db through DAC, RBAC, or MAC.

5. Integrity and reliability

The db groups data from many sources. Users expect DBMS to access data reliably. When developers claim that software is reliable, it means that the software runs for a very long time without errors or failures. Users expect the DBMS to be reliable because data is usually critical to a business or organization. In addition, they entrust their data to a DBMS and legitimately expect data protection from loss or damage. Reliability and integrity are general security issues more visible in dbs. DBMS protects against loss or damage in several ways. However, no control can prevent authorized users from inadvertently entering acceptable values, even if they are incorrect. The 3 aspects of integrity:

1. **Database integrity –** Concern that the db as a whole is protected against corruption, disk failure, or db index. These concerns are addressed in the OS by integrity checks and recovery procedures
2. **Elementary integrity –** Concern that the value of a certain data is written/modified only by an authorized user. The right access policy protects the db from damage by unauthorized users.
3. **Elementary accuracy –** Concern that only the correct values are written to the db. Checks on element values can prevent the insertion of impermissible values. It is also possible to introduce conditions and/or constraints detecting incorrect values.

Two-phase update

A serious problem for the db admin is the **failure of the computer system during data editing**. When the data item to be modified is long, half of the field may show a new value, while the other half will contain the old one. Although these errors are easily observable, a more serious problem occurs when several fields are updated, and no simple field looks wrong. The solution to this problem is a two-phase update, which governs most DBMSs:

- **The first phase (Intent Phase) –** The DBMS gathers the resources it needs to perform the update. It is possible to collect data, create fake records, open files, block other users. Everything needed for the update is done, but no changes are made to the db. The phase is repeatable indefinitely because it has no lasting consequences. If the system fails during the execution phase, nothing happens because all steps can be repeated after the system is restored.
- **At the end of the first phase** the committing of changes takes place, which includes entering **commit flag** into the db. Commit flag means that the DBMS has passed a point from which no return is possible. After confirmation, the DBMS will start making permanent changes.
- **The second phase –** Makes permanent changes. It is not possible to repeat any actions of the previous phase, but the update can be repeated as often as necessary. If system fails, the db may contain incomplete data, but the system can correct it by performing all second-phase activities. After the end of the second phase, the db is complete again.

When using two-phase confirmation, so-called shadow values are stored for the key data. The shadow values are computed and stored locally during the first phase and copied to the current db during the second phase.

Redundancy/Internal consistency:

DBMSs maintain additional information to reveal internal data inconsistency.

- **Error detection –** One form of redundancy is error detection and correction codes, e.g. parity bits, Hamming codes, and cyclic redundancy check (CRC). These codes can be applied to some field, records, or entire dbs. Each time data is placed in the db, control codes must be calculated and stored. When retrieving data items, it is necessary to calculate the control code and compare it with the stored value. If the values are different, this fact shall be reported to the DBMS. Some of these codes point to the location of the error, others

indicate what the correct value should be. More information = more space needed to store codes.

- **Shadow fields –** All attributes or entire records can be duplicated in the db. If the data is unrepeatable, this copy may be provided for immediate exchange if an error occurs. Redundant arrays require significant storage space.
- **Recovery –** The DBMS can maintain a record of user accesses, and in part it can maintain changes too. In case of failure, the db is repopulated from a backup copy and all subsequent changes are then applied from the log.

Concurrency and consistency:

DB systems are often multi-user systems. Access by two users sharing the same db must be restricted so that they do not interfere with each other. Simple locking is performed in the DBMS.

If two users try to **read the same data items**, there is no conflict because they both get the same value.

If two users try to **modify the same data items**, it is often assumed that there is no conflict because everyone knows what to write. The value entered does not depend on the previous value. However, this assumption is not entirely accurate, e.g. if two independent agents try to reserve the same seats for different passengers. The DBMS solves this problem by **using the entire query-update cycle as an atomic operation**. The command therefore looks like this: ''I read the current value of the passenger seat to xyz, if it is free, I assign it to the passenger.''. This cycle must be performed continuously without the possibility for other passengers to access the seat.

The problem with concurrent access is **read/write**. Have one user edit the value and another want to read it. When a read is performed during writing, it may be a partially updated entry. Therefore, the DBMS does not allow read requests to be processed until the write is completed.

6. Sensitive data

Sensitive data is data that should not be made public. The identification of sensitive fields and items depends on the **specific db and the meaning of the data.** Some dbs, such as public book catalogues in libraries, do not contain sensitive data, and conversely, state defence dbs are sensitive. In this case, access to the db as a whole can be prevented. More complicated is the case when not all elements of the db are sensitive. **Different degrees of sensitivity** should also be considered. One of the security requirements may be the fact that some people are authorized to

see the fields but cannot see them all. The task of this problem is to restrict access so that users can only access data to which they have legitimate access. Several factors affect data sensitivity:

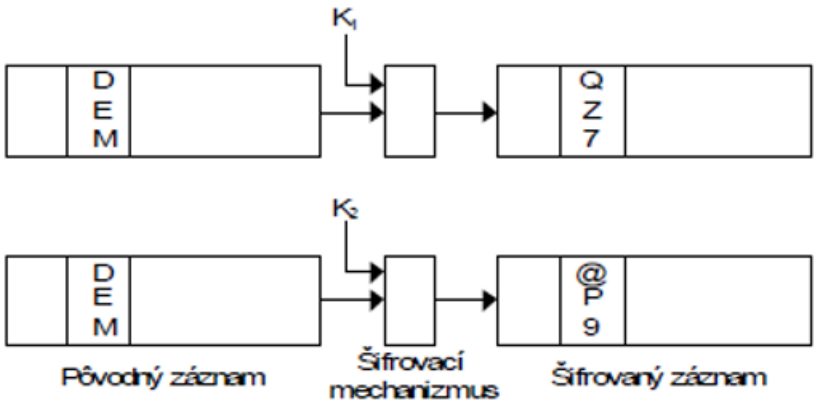
- **Inherently sensitive –** The value itself can be revealing, making it sensitive, such as the location of defensive missiles or middle income for barbershops and hairdressers in a city with one barber shop.
- **From a sensitive source –** The data source indicates that it needs to be kept confidential, for example an informant whose identity would be compromised if the information were disclosed.
- **Declared sensitive –** The admin or data owner may declare the data sensitive, such as classified military data or the name of an anonymous artwork donor.
- **Part of sensitive attributes or records –** The entire attribute or record may be classified as sensitive, for example, payroll attributes or a record describing a secret mission.
- **Sensitive in relation to information already published –** Some data become sensitive in the presence of other data, for example, the longitude of a secret location reveals little, but longitude in cooperation with latitude points to a specific point.

7. Multilevel databases

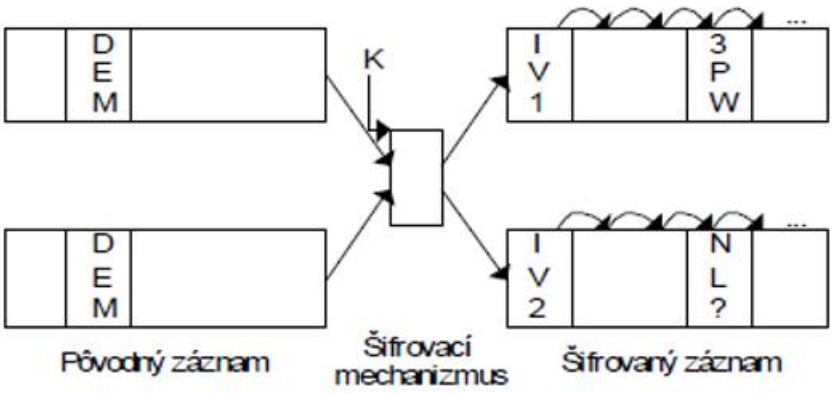
Data – considered in two categories: **sensitive and not sensitive.** Some are more sensitive than others, but most are only allowed yes-or-no access. The implementation of multilevel dbs is complicated due to the **control of low granularity of items**. Maintaining the confidentiality of data can lead to redundancy. One record can appear multiple times with different degrees of confidentiality. Separation is needed to restrict access. There are several separation mechanisms that can help implement multilevel security in dbs:

- **Segmentation –** Method of managing multilayer dbs. The db is divided into separate dbs, each with its own degree of sensitivity. However, it eliminates the basic benefits of dbs (elimination of redundancies, increased accuracy), because only one record is modified. In addition, it does not address the issue of a higher tier user who needs access to lower level data combined with higher level data.
- **Encryption –** If sensitive data is encrypted, the user who accesses it cannot interpret it. Thus, each data sensitivity level can be stored in a table encrypted with a unique key for that sensitivity level. Encryption can have disadvantages, for example, if a user obtains encrypted records, he can add a new record to the table and compare the resulting encrypted record with other encrypted records, which can lead to their decryption. If the authentication data is encrypted, the user can replace it with his own, which not only gains

control over specific users, but also prevents access by a legitimate user. Using these different encryption keys for each record can eliminate these problems, where each record item will be **encrypted with a different key**:



Another solution is to cryptographically bind all blocks of items as with a block cipher:



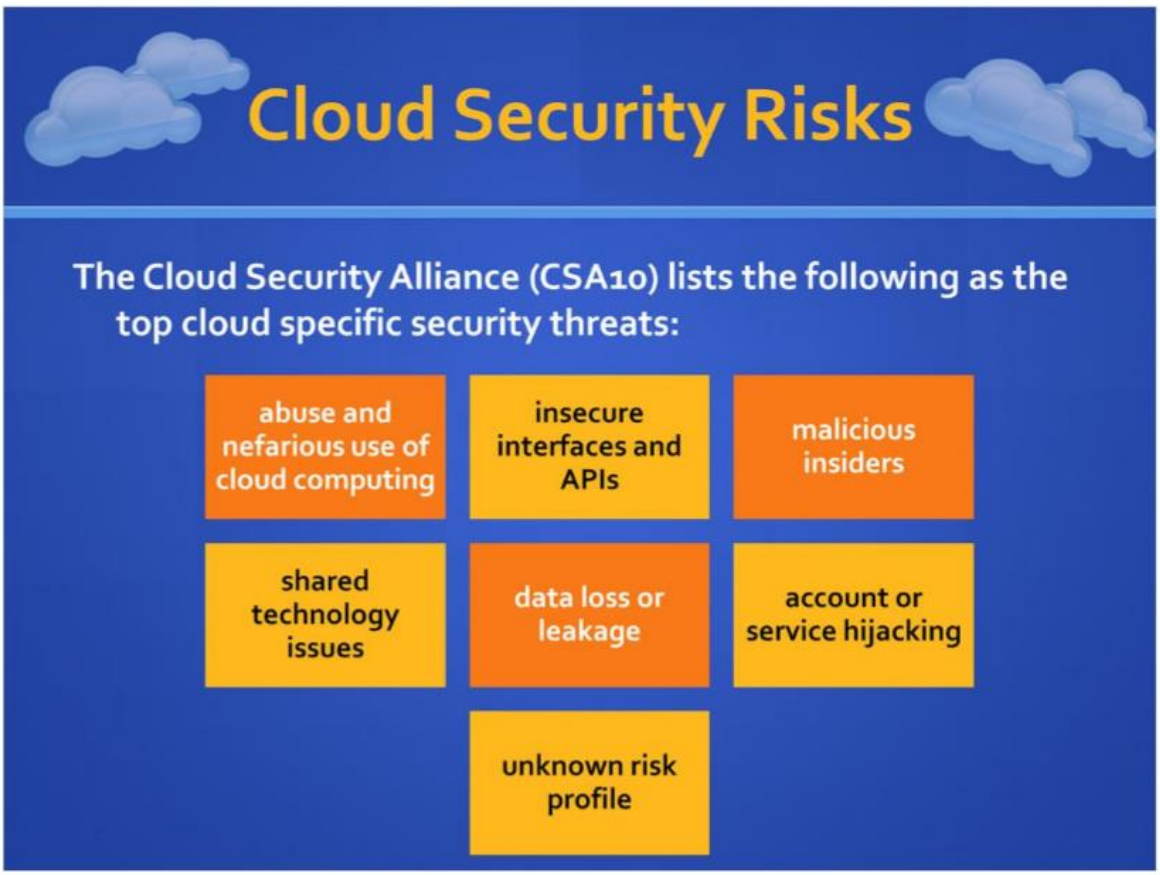
The disadvantage is that if the user performs correct operations, each item must be decrypted, which increase the operation time. Therefore, encryption is not often used when implementing separation

- **Integrity lock –** Lock is a way to ensure integrity and limited access for the db. The operation is sometimes called a spray because each element is ‘‘painted’’ with a colour representing sensitivity. Colouring is applied to items, not the main db table. For efficiency, items are stored as readable text. Each element consists of three parts:
    1. **Current data**
    2. **Sensitivity marker –** defines the degree of sensitivity of the data. Can be **non-covalent**, when the intruder cannot create a new level of sensitivity for the item, and can be **unique,** when the intruder cannot copy the sensitivity level from another element. It can also be **hidden,** when an intruder cannot determine the degree of sensitivity for any element.

3. **Checksum –** Error detection code. It is calculated from the data and the sensitivity marker in order to avoid unauthorized modification of the data and their degree of sensitivity. The uniqueness of the checksum for each element is a guarantee of maintaining the value of the data and the sensitivity marker. An appropriate checksum contains something unique to the item (such as an index), something unique to a field without items (such as a field name), a data value, and an item's sensitivity classification that protects items from modification, copying, and moving. The checksum can be calculated by a strong encryption algorithm or a hash function.

8. Cloud security

Today, many organizations and companies are moving their technologies to the Cloud to be accessible form anywhere. This leads to various security risks even at the db level.



Dangers:

- **Abuse and criminal use of cloud computing** – it is easy to register and start using could services, which allows attackers to get inside the cloud to carry out various attacks such as spam, malicious code, and DoS.

- **Unsecured interfaces and APIs –** APIs that customers use to manage and work with cloud services. The security and availability of universal cloud services depends on the security of these basic APIs.
- **Malicious infiltration –** direct control over many aspects of security is given by an unprecedented level of trust in securities. Cloud architectures require certain roles that are extremely high risk. Examples include security system admins of CP systems and managed providers.
- **Common technology challenges –** IaaS vendors provide their services in a scalable way of sharing infrastructure. MRs typically approach this risk by using isolated virtual machines for individual clients. This approach is still vulnerable to attack by both insiders and outsiders, and so can only be part of an overall security strategy.
- **To data loss or leakage –** the most devastating consequences of a breach is data loss or leakage.
- **Account or service hijacking –** usually with stolen credentials, remains a major threat.
- **Unknown risk profile –** when using cloud infrastructures, the client will necessarily transfer control of cloud providers to a number of issues that may affect security.

9. Security in computer networks

1. Threats in computer networks

An independent home user or a smaller company is usually not the target of attacks. However, the risk is growing because the network and separate environments are different:

- **Anonymity -** An attacker could attack the system from any distance without coming into direct contact with the system itself, its administrators, or users. The potential attacker is thus safe behind the electronic screen between him and the infected system. The attack can take place through a number of other users in an attempt to disguise the attacker.
- **Lots of sources and targets of attacks -** The computer system is a separate unit. The confidentiality of his data can be maintained through access control. However, if the file is stored on a remote network drive, it can pass through multiple devices (guests, systems) until it reaches the user. The administrator of one guest / system device can enforce a strict security policy, while others cannot. The user is thus dependent on the access control mechanisms on the several guests / systems. An attack can come from any guest to any guest, so a large network offers, in addition to various benefits, a number of potential vulnerabilities.
- **Sharing -** Because networks allow you to share resources and workloads, more users can access more network systems. Access control of individual systems may thus be insufficient.
- **Complexity of the system -** Reliable security of large operating systems is challenging, especially if the system is not specifically designed for security. In addition, the network connects two or more different operating systems, so the network management system is more complex than the control system of a stand-alone computer system. The administrator can easily overlook the places through which the attacker penetrated the system / network. Conventional workstations today have a higher computational power, which the intruder can also use to perform (calculate) an attack so that the user does not know it.
- **Unknown circuit -** Network extensibility means the unreliability of its peripheral parts, because one host can be a part (node) of two different networks. The resources of one network are thus also accessible to users of another network. Although wide accessibility is an advantage, this unknown, uncontrolled group of potential attackers is a security disadvantage. It sets a similar problem for new guests, so each network node must respond to new, unreliable guests. User on network D does not need to know about the connection

of users of networks A and B. The guest connecting networks A and B actually belongs to networks C and E. If there are different security rules on these networks, a particular guest may violate security under the security rules of one network. rules of the second network.

- **Unknown route -** There can be more than one road from one place to another. If user A wants to send a message to user B, the message can be routed through guest C or D. Guest C can provide acceptable security, guest D does not have to. Ordinary network users do not have control over the routing of their messages, so they cannot even influence their path.

Threats before transmission

- **Port scan -** an easy way to get network information. It is a program that records for individual IP addresses, which ports respond to messages and which vulnerabilities the system has. An attacker learns three facts: which standard ports or services are running on the monitored system, what operating system is installed on the monitored system, and which applications and application versions are present. This information is easily accessible, anonymously, without identification or authentication, and thus without attracting attention. Tools for performing this activity are not illegal, some can be downloaded for free from the Internet (eg nmap, netcad). The most popular commercial scanners are Nessus, CyberCop Scanner and Cisco Secure Scanner. Port scanning provides an external view of the system.
- **Social engineering -** In addition to the external view of the system (port scan), the attacker needs to know the internal architecture of the system, which can be achieved by skilful use of social communication. The main goal is to persuade the victim to trust the attacker and to be useful to him, e.g. through a telephone conversation. An attacker often impersonates someone inside the company whose name can be easily found on the company's website, e.g. high-ranking man, or the person responsible for managing the computers / network. The victim can ask for the address of the network, gateway, network device ... The attacker thanks the victim, so that he achieves that the victim does not report the event, so no one finds out about the attack.
- **Research -** The attacker tries to learn as much as possible about his goal. Mostly it is about collecting discrete bits from different sources and putting them together like a puzzle. This also includes spying or searching for garbage.
- **OS and application fingerprints -** Network protocols are standard and independent of manufacturers, and at the same time each manufacturer implements its code independently, so there are only minor changes in the interpretation and management of network protocols, which can characterize the manufacturer, and thus the individual

application / operating system. Sometimes an application identifies itself when an attacker sends an insignificant message to a port that responds that it cannot react to the message and identifies itself at the same time (e.g., port 80-http, 25-smtp, 110-pop, 21-ftp).

Transmission threats: eavesdropping and ''stabbing''

There are many ways in which an attacker can compromise your computing environment: loss of credibility, integrity, or availability of data, hardware or software, processes, or other important resources. The simplest form of attack is eavesdropping. An attacker can intercept the content of a communication without disturbing it and without suspicion, without making any effort. The administrator can also legally listen while monitoring the network. A worse form is "stabbing" - wiretap, which means intercepting communication with the use of effort. There are two forms - passive and active. Passive stabbing is similar to eavesdropping and is merely listening by some means. Active puncture means introducing something into the communication, e.g. replace the communication of one party with another communication or create a communication with a purpose and pretend that the other party is a specific user or process. None of the participants has any idea that the communication has been disrupted.

Protocol cracks

Internet protocols are public. Each accepted protocol is known by its Request for Comment (RFC) number. Many protocol issues have been identified and fixed before they were recognized as standards. However, protocols are defined and controlled by people who can be wrong, which also applies to their implementation. E.g. TCP connections are made through a sequence of numbers - the client sends a sequence of numbers to open communication and the server responds with these numbers and its own number. The client then responds with the number sent by the server. These sequences of numbers are incremented regularly, so it is relatively easy to predict the next number. The intruder can estimate and send a sequence of numbers on behalf of the client.

Spoofing – guiding you to the wrong action

Spoofing is an activity similar to a trick, in which the intruder unauthorizedly operates communication on the network - it pretends to be the second party communicating with the user. Simply put: Spoofing is the effort to pretend to be someone I am not.

- **Masquerade -** A guest pretends to be another guest, e.g. changes URL. The domain name can be easily changed, with frequent typos or suffixes (com, org, net ...). The masquerade can also be applied to domains that the names are often mistaken by users, e.g. citibank.com Vs. citybank.com. An attacker can adjust the look of his site to be identical to

the one he wants to pretend to be. Users will try to log in to this domain, thus entering their confidential information (account number, pin, password ...). Then they will be redirected to the real page. This method of masquerade is called phishing. Another way could be to exploit the cracks on the web server and interfere with its pages.

- **Session hijacking -** piracy of a connection - represents the interception of a connection made by another entity and the continuation of its execution, e.g. if the administrator connects to the privileged account of the remote system via Telnet, the connection can be intercepted and the intruder can enter commands as if they were entered from the administrator.
- **Man-in-the-Middle attack -** the intruder acting between the two participants in the communication from the beginning of the communication, not after it started. If the communication participants use public key encryption, the attacker intercepts User A's request for User B's public key and requests it himself. It then sends user A its own public key. The intruder is thus able to decrypt and read the message from user A and then encrypt and send it user B using his public key. User B can then decrypt the message and respond to it without him or user A knowing that the communication is being disrupted.
- **DHCP Spoofing -** connecting an unauthorized DHCP server to the network. It may be a malicious activity. Many times, however, it is more about negligence (own access point, laptop with network software, etc.). Messages from the client are usually sent as broadcasts, so they are all received in a common VLAN, not just the server. The message from the server can be sent as a broadcast or the destination MAC does not have to be known on the switch, so it can again get to the wrong stations. New DHCP servers can be added to the network without any problems, often unnoticed. DHCP messages can be spun or nonsensical. In addition, if the client alternates between different MACs and requests a new IP for each, it can run out of free IP addresses on the DHCP server. A suitable protection against DHCP DoS is DHCP Snooping, which monitors and manages the flow of DHCP messages, recognizes trustworthy and untrustworthy ports, and can limit the maximum number of received DHCP messages per second on the port using Limit Rate.
- **IP Spoofing -** stealing another station's IP address. (An intruder attempts to send packets from one IP address that appear to come from another.) A combination of DHCP Snooping and IP Source Guard is a suitable protection.
    1. **Blind Spoofing –** attack from any source
    2. **Non-blind Spoofing –** attach from the same subnet

- **ARP Spoofing -** sending unsolicited ARP messages in which the selected IP is mapped to other than the actual MAC address. Suitable protection is a combination of DHCP Snooping and DAI (Dynamic ARP Inspection), or ARP ACL.

Packet Sniffers

- Packet sniffers "read" information passing through a network
    1. They capture network packets, possibly also using ARP cache poisoning
    2. They can be used as legitimate network analysis tools
    3. They can also be used maliciously
- Packet sniffers can be either software-based or hardware-based - they depend on the network settings

ARP

The Address Resolution Protocol (ARP) connects the network layer to the data layer by converting IP addresses to MAC addresses. ARP works by "broadcasting" requests and "caching" responses for future use.

ARP Spoofing

ARP The table is updated whenever an ARP response is received. Requests are not tracked. ARP notifications are not verified. The devices trust each other. An "infected" device can "spoof" other devices.

ARP Poisoning

By standard, almost all ARP implementations are stateless. The ARP cache is updated every time it receives an ARP response ... Even if no ARP request has been sent! It is possible to "poison" the ARP cache by sending unreasonable ARP responses. Using static records solves the problem, but it's almost impossible to manage!

NAT

Used to alleviate IPv4 address space congestion. It relies on the translation of internal network addresses, to external addresses that are used for communication to and from the outside world. NAT is usually implemented by placing a router between the internal private network and the public network. It saves IP address space, because not every terminal needs globally unique IP addresses, you are only organizationally unique. While NAT should really be transparent to all high-

level services, this is unfortunately not true because many high-level communications use things on IP.

Denial of Service (DoS)

The most significant attacks in a computer network are attacks on reachability / availability, which are also called DoS attacks or denial of service. DoS is an attack technique on Internet services or sites that overwhelm a request, crash, or malfunction / unavailability for other users. Availability threats:

- **Transmission error -** Communication error, for example due to transmission media cut-off, high noise, network device failure, network device congestion. The intruder can also attack physically, causing a denial of service. This method of attack is quite common.
- **Connection congestion -** The most primitive DoS attack, when the intruder sends an innumerable amount of data, thus preventing the user from being able to receive new data or thus degrading his communication. Smarter DoS attacks use parts of network protocols:
    1. **Echo-Chargen -** Attack between two guests. Chargen is a packet streaming protocol used to test network capacity. The intruder sets this process on station A to flood station B with enormous amount of packets, which responds with echo packets. If the source and destination of the packets is the same address, the guest can overwhelm himself.
    2. **Ping of Death (Ping Flood) -** A simple attack using the fact that ping requires a response from the recipient. The intruder floods the user with pings, overwhelming him. The basic premise for this attack is the fact that the attacker has a faster connection than the target of his attack. This is an ICMP attack.
    3. **Smurf -** It consists of a faulty system configuration that allows packets to be sent to all computers connected to the network via a broadcast address. Then it is enough for the sent packet to have a sufficient size and not filtered out, and it has to be received and processed by all the computers in the network. If the attacker sets the victim's address as the return address, the victim will be flooded with echo responses from across the network. This is an ICMP attack.
    4. **Traffic redirection -** A router is a network device that has information only about neighbouring routers and based on this information, knows which one is the most convenient for communication with a specific destination. If the router is set to be the best route for all neighbouring routers, all traffic will be routed through it. This

will cause any communication to be thwarted, because it will go the wrong way and at the same time the router will be congested, or it will block most of its flow.

5. **SYN Flood -** The recommended way to defend against this attack are the so-called SYN cookies, which internally modify the behaviour of the TCP protocol so that the server's own resources are accessed only after verifying the validity of the address. This procedure can very effectively prevent a SYN flood attack. The implementation of this defence is common on Solaris and Linux systems.
6. **DNS attack -** It is a group of attacks based on the concept of a DNS server. Domain name server (DNS) converts domain names into addresses, which can be misused to store fake addresses, and thus redirect traffic to incorrect addresses or prevent access to specific domains.

Distributed Denial of Service (DDoS)

Amplification of DoS attack in a distributed manner. An attacker can for example use any attack to place a Trojan horse on the machine. The attacker repeats this process on several machines, making each of them a so-called zombies. They then attack several attackers at the same time. The victim then must defend himself against several attacks at once, which can be different types of attacks.

DNS

DNS is an application layer protocol for mapping domain names to IP addresses. DNS provides a distributed database over the Internet that stores various resource records, including:

- **Address (A) record:** IP address associated with the host name
- **Mail exchange (MX) record:** mail server domains
- **Name server (NX) record:** authoritative server for the domain

DNS Caching

If the path in the DNS tree was to be traversed with each query, there was too much network traffic. Therefore, DNS servers "cache" results for a period of time.

Pharming – DNS Hijacking

Change the IP address associated with the server maliciously.

DNS Cache Poisoning

The basic idea is to give the DNS server fake records and "cache" them. DNS uses a 16-bit request identifier to match queries to responses. The cache can be poisoned when the name server scans identifiers, has predictable identifiers, or accepts unsolicited DNS records.

Prevention from DNS Cache Poisoning

- **Use random identifiers for queries**
- **Always check identifiers**
- **Random port for DNS requests**
- **DNSSEC deployments**

DNSSEC

- **Guarantees:**
    1. **Credibility of the origin of DNS responses**
    2. **Integrity of the response**
    3. **The authenticity of the denial of existence**

This is accomplished by signing DNS responses at each step of the method. Uses public key cryptography to sign responses. Usually, confidential anchors, records in the operating system are used to implement the process.

2. Network security control

To ensure security in computer networks, it is possible to use similar principles as in the case of OS security:

- **Segmentation**
- **Redundancy**
- **Single point of failure**
- **Encryption (link encryption, end-to-end encryption) – PKI and certificates, SSH encryption, SSL/TLS (Secure Sockets Layer/Transport Layer Security) encryption**
- **VPN**
- **Content integrity**
    1. The integrity of the content is automatically preserved during encryption because no one can change the encrypted data to make sense without breaking the cipher itself. However, changing one bit of encrypted data can affect the overall result after decryption. There are three potential threats: data modification, where data

makes sense; a modification that changes the content to an incomprehensible form (it does not have to be intentional); a modification that changes the content so that no one finds out (unintentionally).

2. **Error detection and correction codes -** Used against modification during transmission, which first reveals that an error has occurred, then tries to correct the errors without the need for the original message. They are mainly used to detect unintentional changes. The simplest error detection code is parity check, where one bit is added to an existing data group according to their sum or XOR function. Even (not odd) parity means that the added bit has a value of 0 if the number of data is even. If the number of data is odd, the added bit has the value 1, so the sum of all data + added bit always has an even result. Odd parity means the same thing, but the total is odd. Parity can signal whether a change has occurred on one bit, while it cannot signal on which bit an error has occurred. Another way to detect an error is hash codes, which can detect errors on multiple bits and specify which bits have been changed.
3. **Cryptographic checksum -** The ingenious modification must be controlled to prevent the attacker from modifying the change detection mechanism and the bits themselves. A cryptographic checksum, sometimes called a message digest, is a feature used to protect against code tampering and maintain integrity during transmission. The function calculates the fixed length of the message and at the same time determines whether its hash value has changed

3. Firewalls

A set of HW and SW resources, the task of which is to separate the local network from the Internet. Firewall is a device whose task is to filter traffic between the internal protected network and the less reliable external network. There are several types of firewalls:

- **Packet filtering gateway -** the simplest type of firewall. They are characterized by high speed. This type, based on the rules defined in the ACL, the source and destination IP addresses, and the protocol type, decides which packets are forwarded to / from the network and which are not.
- **Stateful inspection firewall -** packet filtering gateways process packet by packet and do not keep any picture of the state, or context and relationships between individual packets. This species preserves these relationships and context
- **Application proxy -** the types of firewalls mentioned above only look at packet headers and do not examine their contents. Application proxy firewalls act as an intermediary between

the sender and the recipient and simulate a specific application to examine the contents of packets. There is a proxy (http, FTP) for each service. Only allows services that allow proxy.

- **Guard –** similar to application proxy, but more sophisticated
- **Personal Firewall -** an application that runs on a local computer and monitors the operation of this machine.

Stateless vs. Stateful Firewall

- **Stateless -** does not maintain any memorized context (or "state") with respect to the packets it processes. Instead, it checks each packet that passes through it separately, regardless of the packets it processed previously. Stateless firewalls can be quite restrictive to prevent most attacks.
- **Stateful -** can tell when packets are part of a legitimate session from a trusted network. Maintains tables containing information about each active connection, including IP addresses, ports, and packet sequence numbers. Using these tables, the stateful firewall can only allow incoming TCP packets that are in response to a connection initiated directly from the internal network.

Advantages of firewall

- **the entire communication of the local network with the surroundings passes through one point**
- **local network coverage**
- **creation of audit records**
- **prevention of information export**
- **translation of local addresses**

Disadvantages of firewall

- **does not protect against attacks from within**
- **does not prevent the creation of an alternative route**
- **if the protection mechanism is exceeded, the network is unsafe**
- **its performance is critical to the network**

IPSEC

IPSec defines a set of protocols for providing the confidentiality and authenticity of IP packets. Each protocol can operate in one of two modes:

- **Transport mode -** the additional IPsec header information is inserted before the data of the original packet, and only the payload of the packet is encrypted or authenticated.
- **Tunnel mode -** the new packet is constructed with IPsec header information, and the entire original packet, including its header, is encapsulated as a payload for the new packet.

Virtual Private Networking (VPN)

Virtual Private Network (VPN) is a technology that allows private networks to be securely extended over long physical distances using a public network, such as the Internet, as a means of transportation. VPNs guarantee data confidentiality, integrity, and authentication, despite the use of an untrusted network for transmission. 2 basic types of VPN:

1. **Remote access VPN -** allows authorized clients to access a private network called an intranet
2. **Site-to-site VPN** - is designed to provide a secure bridge between two or more physically distant networks.

4. Intrusion detection systems

Depending on the method and place of deployment:

1. **HIDS**
2. **NIDS**
3. **Hybrid**

Depending on the method of detection:

- **Signature based IDS**
- **Anomaly based IDS**
- **Security policy based**
- **Honeypot based IDS**

5. Secure mail

Email is an everyday part of today's company's communication. The fact that it is publicly accessible to everyone increases the risk of its misuse and therefore it is necessary to secure email communication. It is necessary to ensure:

- **confidentiality of the message**
- **integrity of the message**
- **authenticity of sender**

The two most important tools are used to ensure email communication:

1. **PGP (Pretty Good Privacy):**
    (a) Creation of a random session key for symmetric encryption.
    (b) Encryption of the message using a session key (to ensure the confidentiality of the message).
    (c) Session key encryption using the recipient's public key.
    (d) Generate a message digest or message hash. Signing a hash by encrypting with the sender's private key (ensuring message integrity and authenticity).
    (e) Attaching an encrypted session key to an encrypted message and hash.
    (f) Transmission of the message to the recipient.
    (g) The recipient will perform these steps in reverse order.
2. **S/MIME (Secure Multipurpose Internet Mail Extensions) -** the difference between PGP and S / MIME is in the way the key is distributed. PGP relies on the distribution of the key through individual users. S / MIME uses hierarchically verified certificates in X.509 format. Thus, with S / MIME, the sender and recipient of the email do not have to exchange a key with each other if they have the same certification authority. In addition to PGP, this type of security also secures email attachments.

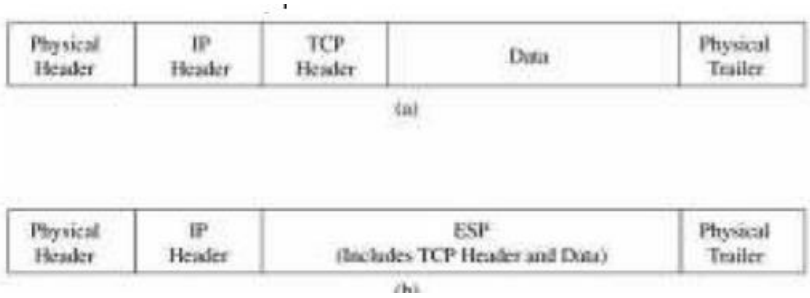
6. TCP-IP security

IP, TCP, and UDP do not define security mechanisms for data encryption or authentication. Telnet, FTP, SMTP, POP, http protocols also do not provide reliability and authentication.

- **Application layer -** If we integrate communication security on the application layer, then we must do so for each application. The application has access to user information, so it is advisable to use role-based access control. In addition, the application can use interfaces with other security services to directly control security.
- **Transport layer -** At this layer, the SSL / TLS protocol is used for security. Alarm management can still be implemented in an application, but more often it is handled by a separate alarm management process because the application may not be ready to manage security events. The SSL / TLS protocol was developed to secure communication between a web browser and a server. The so-called security tunnel is created in which encrypted communication takes place:
    1. Before using SSL, the client requests an SSL session.
    2. The server responds with its public key certificate so that the client can verify the authenticity of the server.

3. The client returns a symmetrically session key encrypted with the server's public key.
4. Subsequently, a secure tunnel is created in which encryption communication takes place with a symmetric session key.

- **Network layer -** If we integrate security on this layer, then we lose even more contact with the application, and therefore we have to rely on higher layers, which must provide us with the necessary information to evaluate the security event. Interaction between layers in relation to security should be enshrined in the rules of security policy. One of the protocols that acts as protection at the network layer is the **IPsec** protocol. IPsec is a protocol designed by the IETF that serves to ensure the confidentiality, authentication, and integrity of data at the network layer. It provides two basic types of protection, **authentication, and encryption**. The basic concept of this protocol is a **security association**, which includes a set of security parameters for a secure communication channel. The basic data structure of IPSec is **AH (authentication header) and ESP (encapsulated security payload)**. ESP replaces the flood TCP header and data in the packet.



- **Connection and physical layer -** On these layers, we are completely isolated from any application information. Separate hardware encryption circuits or private interconnection channels can be used for security.

10. Web Security

Secure socket layer (SSL) provides security services between TCP and applications that use TCP. The Internet standard version is called transport layer service (TLS).

SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code.

SSL/TLS includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use.

Secure electronic transaction (SET) is an open encryption and security specification designed to protect credit card transactions on the Internet.

1. TLS / SSL

Transport Layer Security (TLS), and its now-deprecated predecessor, Secure Sockets Layer (SSL),[1] are cryptographic protocols designed to provide communications security over a computer network.[2] Several versions of the protocols find widespread use in applications such as web browsing, email, instant messaging, and voice over IP (VoIP). Websites can use TLS to secure all communications between their servers and web browsers.

The web secured with TLS have these properties:

1. The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session (see § TLS handshake). The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted. The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).
2. The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).
3. The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

2. Attacks

10.2.1. Phishing

Create a website that looks similar (mbe even identical than the original one (Facebook, PayPal). Use url obfuscation methods to create similar URLs as well. Then send the obfuscated url to the victims (mbe via email). Once they will try to login via the phishing site the attackers can get their passwords.

10.2.2. URL Obfuscation – Manipulation with the URL

Actual URL different from spoofed URL displayed in address bar

1. URL escape character attack

Old versions of Internet Explorer did not display anything past the Esc or null character

Displayed vs. actual site
http://trusted.com%01%00@malicious.com

2. Unicode attack

Domains names with Unicode characters can be registered Identical, or very similar, graphic rendering for some characters E.g., Cyrillic and Latin “a”

10.2.3. Image Crash

Browser implementation bugs can lead to denial of service attacks.

By creating a simple image of extremely large proportions, one can crash Internet Explorer and sometimes freeze a Windows machine

```
<HTML>
  <BODY>
    <IMG SRC="./imagecrash.jpg" width="9999999" height="9999999">
  </BODY>
</HTML>
```

Variations of the image crash attack still possible on the latest IE version

10.2.4. JavaScript Click Jacking attack

<a onMouseUp="window.open('http://www.evilsite.com')"href="http://www.trustedsite.com/">Trust me!

10.2.5. Mobile Code

Mobile code is an executable program sent via a computer network. It is executed at the destination. Examples: JavaScript, ActiveX (windows specific staff for internet explorer), Java Plugins, Integrated Java Virtual Machines

Java Applets - Platform-independent via browser plugin Java code running within browser Sandboxed execution Support for signed code Applet runs only on site where it is embedded Applets deemed trusted by user can escape sandbox. (quite old staff)

10.2.6. Cookies

Cookies are a small bit of information stored on a computer associated with a specific server. When you access a specific website, it might store information as a cookie Every time you revisit that server, the cookie is re-sent to the server Effectively used to hold state information over sessions Cookies can hold any type of information Can also hold sensitive information. This includes passwords, credit card information, social security number, etc. Session cookies, non-persistent cookies, persistent cookies Almost every large website uses cookies

10.2.7. XSS

Attacker injects scripting code into pages generated by a web application

Script could be malicious code JavaScript (AJAX!), VBScript, ActiveX, HTML, or Flash

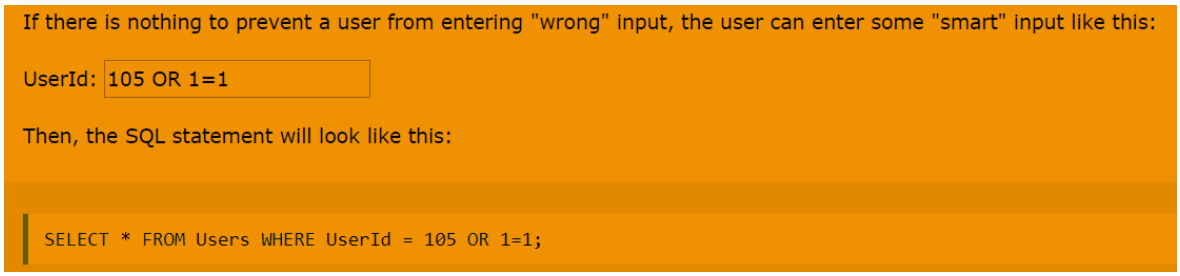
Threats:

Phishing, hijacking, changing of user settings, cookie theft/poisoning, false advertising , execution of code on the client, ...

An example how to protect can be disabling JavaScript. The websites will may look uglier, but they will be safer.

3. My Ideas

- SQL Injection - SQL injection is a code injection technique that might destroy your database.



- Cross Site Request Forgery - In a CSRF attack, an innocent end user is tricked by an attacker into submitting a web request that they did not intend. This may cause actions to be performed on the website that can include inadvertent client or server data leakage, change of session state, or manipulation of an end user's account. Prevention using unique and unpredictable tokens.
- Security Missconfiguration – debugging on on production, not up to date wordpress version, etc...

4. DNS Cache Poisoning

DNS cache poisoning is the act of entering false information into a DNS cache, so that DNS queries return an incorrect response and users are directed to the wrong websites. DNS cache poisoning is also known as 'DNS spoofing.' IP addresses are the 'room numbers' of the Internet, enabling web traffic to arrive in the right places. DNS resolver caches are the 'campus directory,' and when they store faulty information, traffic goes to the wrong places until the cached information is corrected. (Note that this does not actually disconnect the real websites from their real IP addresses.)

Because there is typically no way for DNS resolvers to verify the data in their caches, incorrect DNS information remains in the cache until the time to live (TTL) expires, or until it is removed manually

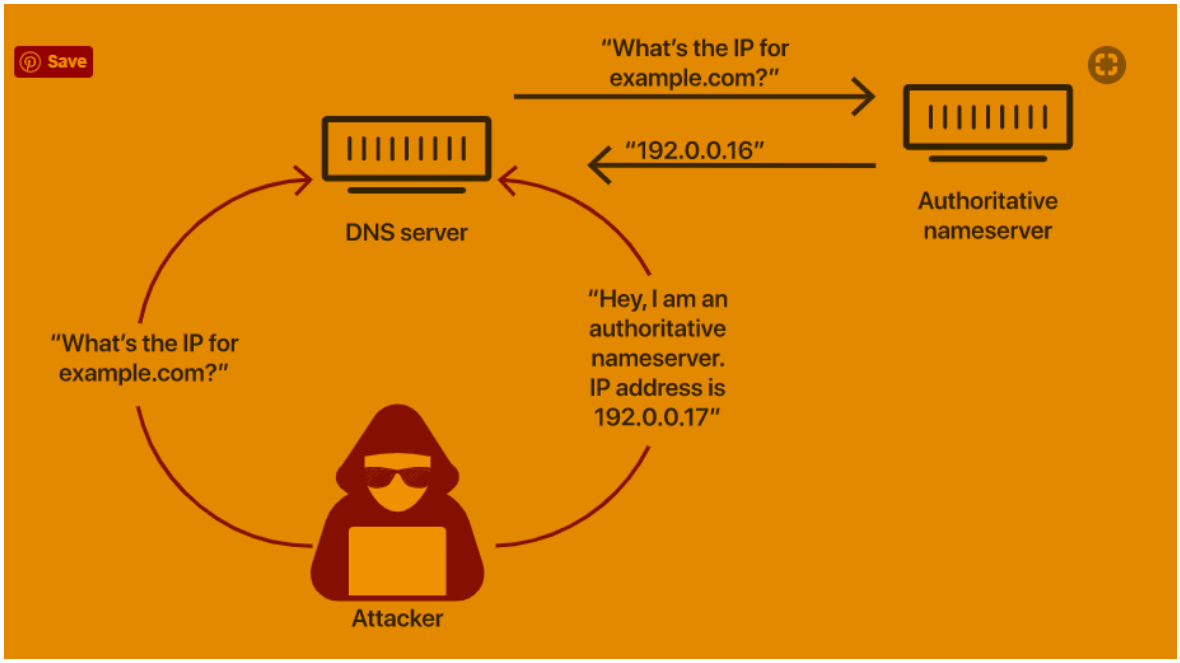
A more secure DNS protocol called DNSSEC aims to solve some of these problems, but it has not been widely adopted yet.

Attackers can poison DNS caches by impersonating DNS nameservers, making a request to a DNS resolver, and then forging the reply when the DNS resolver queries a nameserver. This is possible because DNS servers use UDP instead of TCP, and because currently there is no verification for DNS information.

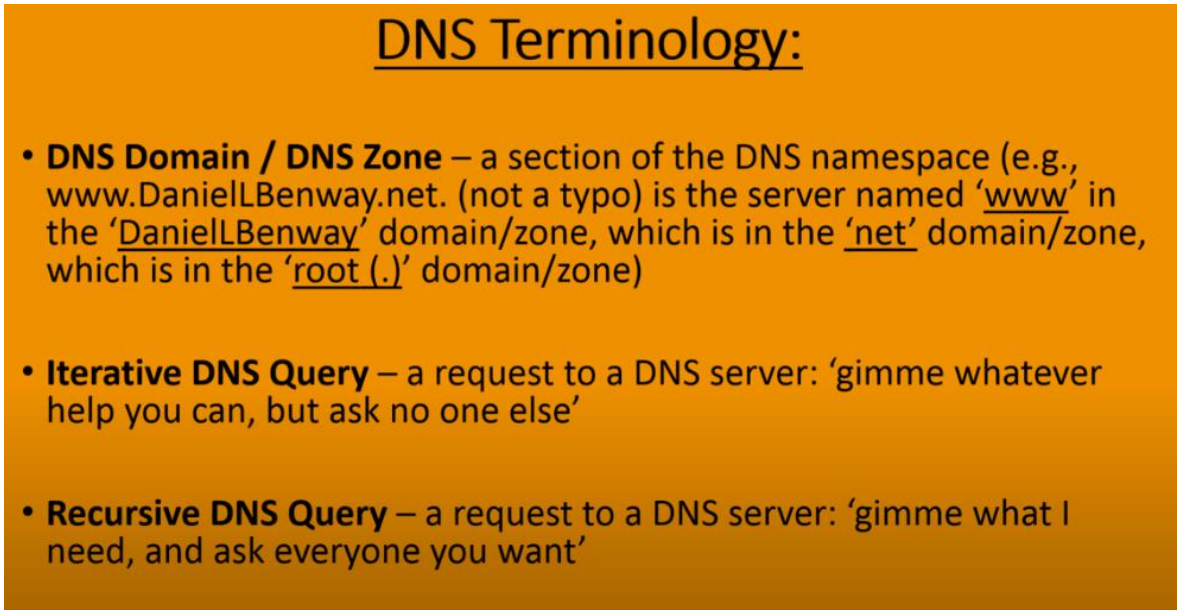
If a DNS resolver receives a forged response, it accepts and caches the data uncritically because there is no way to verify if the information is accurate and comes from a legitimate source. DNS was created in the early days of the Internet, when the only parties connected to it were

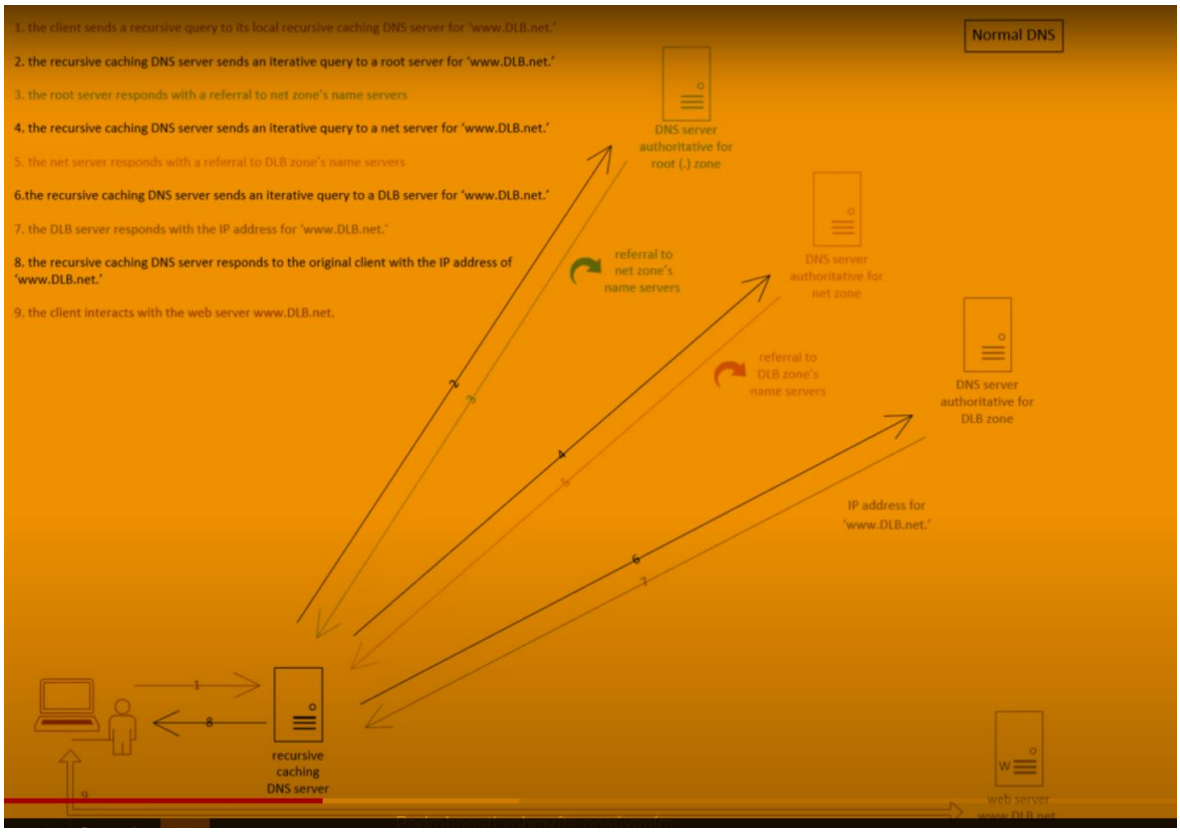
universities and research centers. There was no reason to expect that anyone would try to spread fake DNS information.

Despite these major points of vulnerability in the DNS caching process, DNS poisoning attacks are not easy. Because the DNS resolver does actually query the authoritative nameserver, attackers have only a few milliseconds to send the fake reply before the real reply from the authoritative nameserver arrives.



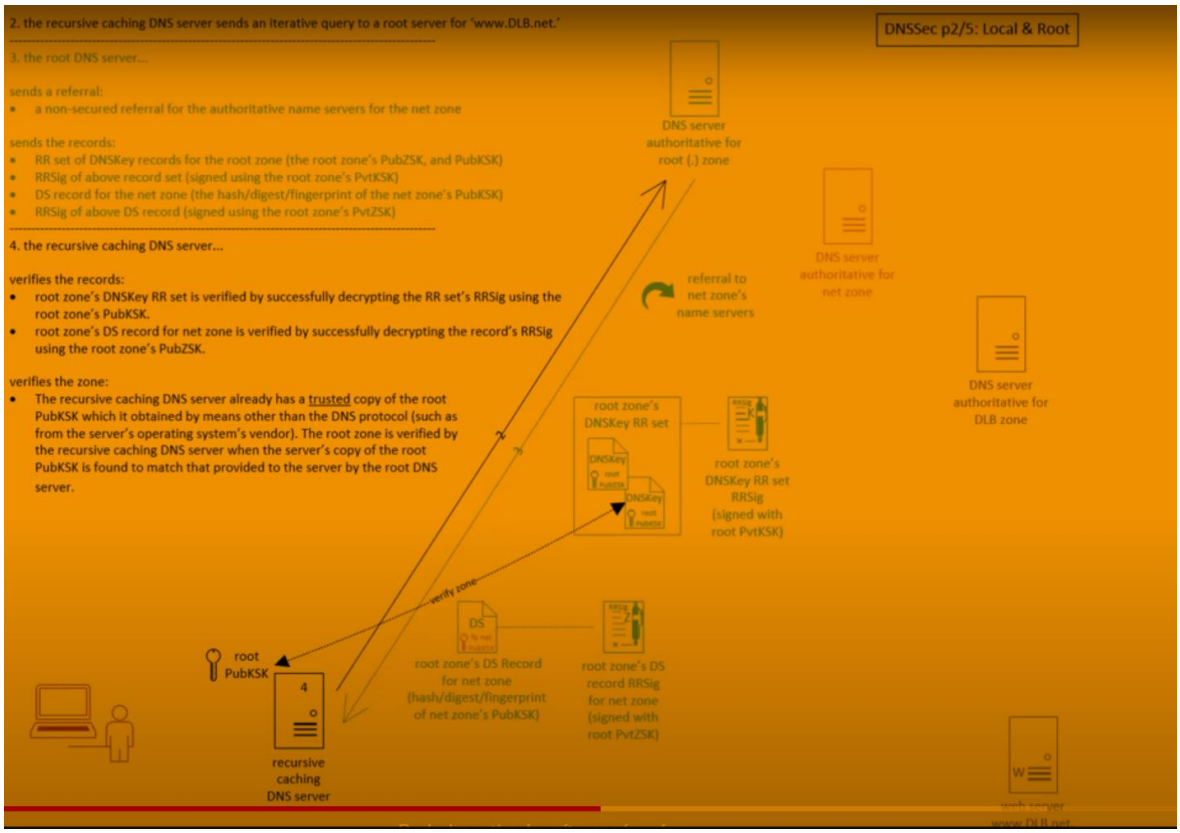
5. DNS SEC





- **KSK – Key Signing Key** – used to sign or verify a domain's/zone's keys
- **ZSK – Zone Signing Key** – used to sign or verify a domain's/zone's non-key records

- **RRSet – Resource Record Set** – a set of records with the same type and same domain/zone
- **RRSig – Resource Record Signature** – a record containing an RRSet's digital signature
- **DS Record – Delegation of Signing** – a record containing the hash/digest of a child domain's/zone's PubKSK (the fingerprint of a child's PubKSK)


DNSSec p2/5: Local & Root
2. the recursive caching DNS server sends an iterative query to a root server for 'www.DLB.net.'
3. the root DNS server...
sends a referral:
a non-secured referral for the authoritative name servers for the net zone
sends the records:
RR set of DNSKey records for the root zone (the root zone's PubZSK, and PubKSK)
RRSig of above record set (signed using the root zone's PvtKSK)
DS record for the net zone (the hash/digest/fingerprint of the net zone's PubKSK)
RRSig of above DS record (signed using the root zone's PvtZSK)
4. the recursive caching DNS server...
verifies the records:
root zone's DNSKey RR set is verified by successfully decrypting the RR set's RRSig using the root zone's PubKSK.
root zone's DS record for net zone is verified by successfully decrypting the record's RRSig using the root zone's PubZSK.
verifies the zone:
The recursive caching DNS server already has a trusted copy of the root PubKSK which it obtained by means other than the DNS protocol (such as from the server's operating system's vendor). The root zone is verified by the recursive caching DNS server when the server's copy of the root PubKSK is found to match that provided to the server by the root DNS server.
DNS server authoritative for root (.) zone
DNS server authoritative for net zone
DNS server authoritative for DLB zone
referral to net zone's name servers
root zone's DNSKey RR set
root zone's DNSKey RR set RRSig (signed with root PvtKSK)
verify zone
root PubKSK
root zone's DS Record for net zone (hash/digest/fingerprint of net zone's PubKSK)
root zone's DS record RRSig for net zone (signed with root PvtZSK)
recursive caching DNS server

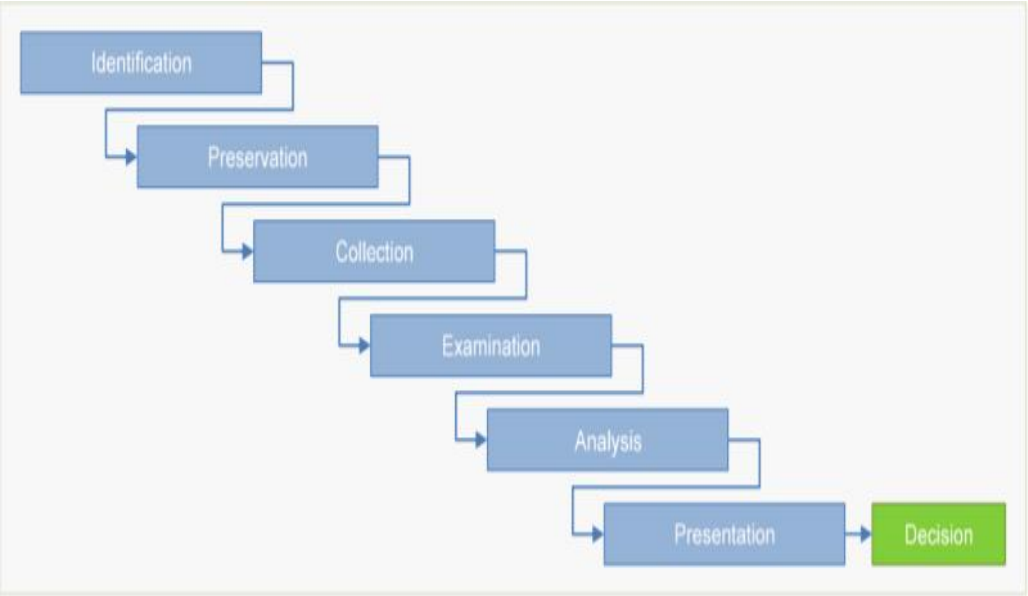
11. Forensic Analysis

Forensic analysis is a comprehensive investigation for identifying the pieces of evidence, consequences, reasons and culprits of a security incident or violation of rules of the state laws or organizations. Digital forensic analysis is essentially a type of forensic analysis. It is considered as a particular domain of information and communication technologies (ICT), that is applied when a criminal act, involving ICT is committed. To provide evidence for criminal acts committed on digital devices (usually on smartphones and computers), digital forensic analysis is being used. It is a technique of recovering digital evidence under forensically sound conditions, using different methods.

The specialists who perform the forensic analysis are so-called forensic investigators since forensic analysis is part of every criminal investigation. These forensic experts are responsible for collecting the pieces of digital evidence and for following every security and data protection procedures when they are handling sensitive financial or private information (such as documents, videos, or images). Forensic experts need to have the following competencies:

- Comprehensive IT knowledge
- Analytical skills
- Excellent problem-solving skills and creativity
- Good communication skills
- Attention to detail

A fundamental rule of the forensic analysis is to sustain the integrity of the original data and to protect it from any contagion that would interfere with its approval in court, so this is something that the examiner needs to take care of. To fulfil this rule, the forensic analysis needs to be completed between forensically acceptable conditions. In forensically acceptable conditions, when a forensic analysis is being performed, if any changes occur during an examination, the nature, the extent, and the reason for all changes need to be accurately documented. Furthermore, the application of forensic tools and techniques should be completed regarding the proper rules of evidence.



Forensic analysis consists of 5 phases:

11.1.1. Identification

The first phase of digital forensic analysis is all about identifying the purpose of the investigation and identifying the required resources a forensic investigator needs for the execution of analysis. In this phase, the investigator might ask himself – what are the best sources of potential evidence that is needed to be accessed for collection? If we consider an example of a case, when the digital device is a smartphone, the appropriate method of data gathering is being considered and chosen by the investigator.

11.1.1. Preservation

This step is not a phase, rather just a rule. When dealing with potential pieces of evidence of a case, preservation of the evidence holder device is a must. The forensic investigator needs to make sure to eliminate any activities, any potentially harmful elements that may harm or even destroy them. Pieces of digital evidence need to be protected. They must be precisely the same how they were found, without any alteration, so that it can be later analysed. The rule of preservation applies to any other phases of digital forensic analysis.

11.1.1. Collection

In the third phase of digital forensic analysis, the potential digital evidence is being collected. This is the first phase when the investigator gains physical access to the digital device and its related items. Firstly, the investigator searches for potential pieces of digital evidence. When he recognizes potential evidence, he collects it. It is critically important that all the collected potential pieces of evidence must be preserved, cannot be changed. If we consider the case of a mobile

phone, a forensic image copy of the phone's internal or/and external memory is being created by the forensic investigator. The forensic image is a bit-by-bit copy of the phone's memory. It contains all the phone's operating system data and the user-generated data. The usage of such forensic images allows the expert, to work with the gathered digital information, without the possibility of (accidentally) modifying some of the original data, so during a forensic analysis, the investigator never works directly with the original image of the phone. Additionally, the investigator needs to document the initial state of the digital information properly.

11.1.1. Examination and analysis

As mentioned before, digital forensic analysis is an in-depth examination of electronically stored information, intending to identify information that may help criminal investigations. This phase includes a comprehensive systematic search of evidence related to the incident being investigated. The examination process needs to be well organised to prevent crucial data from being lost, which may be required and critical to a court case. In the examination and analysis phase, the examiner needs to understand, recreate, and analyse discretionary events based on the files, that have been gathered from the device. It is also advised that the forensic investigator of the device or mobile phone, secure proper use of snapshotting and video equipment to record or document the state of the device throughout several stages in the examining processes. Three types of forensic examination exist: limited, partial, and full examination. The limited examination wraps the areas of the filesystem, which are based on interviews or specified by legal documents. This is the most used and the least time-consuming process.

The partial examination is focusing on – as its name tells us – a specific part of the filesystem. These specific parts are such parts that usually contain the most significant number of available pieces of evidence. This type of examination is based on general search criteria, that are established by forensic investigators. Such specific part of the filesystem can be for example the, registry and system files, log files, cookies, user directories, slack space, and deleted files.

The full examination of a filesystem is the most time consuming and the least used method of data examination. The forensic investigator looks at every single bit of the filesystem to find potential pieces of evidence.

Knowing the nature of the crime helps to identify what type of data or what file types the forensic examiner should be searching for and where that information might be found in the filesystem. The investigator should also know the times and dates when the incident or crime might have happened. This knowledge helps to narrow the examination. For example, in the case of suicide, it is critically needed to know when the individual was last seen or when was his device last

used. The forensic examiner should know the keywords he needs to look for. These keywords can be named, e-mail addresses, nicknames, city names etcetera.

Several software tools exist to help and fasten the work of forensic investigators. These tools can be used in this phase to help with file extraction, file filtering, text indexing, e-mail, and NAT recovery and with many other essential processes. These tools help the investigators to understand the evidence better, though they need to be valid. The examiner must only use valid forensic software to gather accurate information. Additionally, usage of non-validated forensic software tools - in any forensic investigation - is not accepted by the court of law.

Once the critical information has been uncovered, the forensic investigator needs to put together a scenario of what might have happened. The investigator builds a hypothesis based on the findings. This phase of analysis should use the scientific method of hypothesis, which consists of 5 steps:

1. **Identify and research a problem.**
2. **Formulate a hypothesis.**
3. **Conceptually and empirically test the hypothesis.**
4. **Evaluate the hypothesis with regards to the results.**
5. **If the hypothesis is acceptable, evaluate its impact. If not, re-evaluate the hypothesis.**

After the forensic examination and analysis are completed, the forensic investigator makes a conclusion for the device, based on the information he learnt from it and documents it with all other steps he took during this phase so that the evidence collected and findings reached can be defended in court.

11.1.1. Presentation (reporting)

In this significant last phase, all the findings are summarized in the documentation, which is the product of the complete forensic analysis and is presented in the court of law. It’s a formal written report that is viewed by multiple types of audiences, such as human resources, law enforcement, lawyers, judges etcetera, so it must be written clearly and concisely but still needs to contain an acceptable amount of details of the forensic analysis. Since the report is intended for several types of audiences, it is essential to segregate the report into several parts that appeal to each. This report must contain the executive summary, the analysis details section, the appended reports, and the conclusion section.

The report must always begin with an Executive Summary because executives do not have time and sufficient technical knowledge to read the full report, including every detail of the analysis.

For this purpose, this section is dedicated to them. It is written in non-technical language that can be understood by persons, that does not have sufficient technical background. It contains high-level summarization and description of forensic analysis findings.

The Analysis details section contains all the details of forensic analysis, including photos, charts, and diagrams. It is written in technical language and is dedicated to those parties, that have sufficient technical background, who are frequently other forensic investigators. This section is meant to defend the findings defined in the Executive Summary by precisely describing the whole process of forensic analysis.

The Appended Reports section further supports the analysis of relevant information through the presentation of highly detailed technical information, including evidence that can produce an enormous amount of data, such as text message or email analysis.

The Conclusion section provides subjective analysis and expert opinions. This part wraps up the whole forensic analysis in a directly and concisely. It should never contain new information.