Computer Security

2024-2025

Threats, exploits and attacks

A *threat* is any potential violation of security that could cause harm to the asset. It may be a person, an insecure service or some unacknowledged piece of information, service or system.

An *exploit* is any software or tool used to take advantage of vulnerabilities.

An *attack* is any intentional or unintentional event that harms, or tries to harm, an asset.

CVE (Common Vulnerability and Exposure).

Untargeted threats: worms and other malware trying to access any computer they find on the net. *Targeted attacks*: deliberate attack on a single, specific machine.

Bot network: network of hacked machines, owned by unknowing individuals but controlled by a single hacker, used for malevolent purposes such as DDoS attacks. Most common in the US where static IPv4 addresses are the norm. To mitigate this issue, ISPs try to identify both the single machines (by cutting off their connection and notifying the owners) and the control systems (to shut them down).

APT (Advanced Persistent Threat): a persistent, patient and well-prepared enemy that is constantly looking for a way to harm a specific asset. It can cause a high emotional impact. An APT may infiltrate, wait patiently while gathering information in the most discrete way possible, and finally unleash an attack only when fully ready.

Building a system that handles complexity in a simple way, and that is also safe, is a hard task. As the system grows in complexity, the amount of components that cannot be fully trusted increases, along with the necessity of redundancy to maintain it reliable.

A *risk* is the product of the probability of a threat of taking advantage of a vulnerability, multiplied by the size of the potential damage.

The value assessment of assets may be asymmetrical (for example, personal data is highly valuable to the owner and of very little value to attackers, except for ransom) or symmetrical (money is equally valuable to both the thief and the victim).

Insecure applications on the asset, conflicting security policies and insecurely configured systems enlarge the exposed surface. Having a myriad of internet-connected devices around us all day also enlarges the exposed surface. To solve this, strong authentication is needed on both sides, services and users. Authorization and access control systems need to be reliable. Abuse control (detection of suspicious activity) must be effective. Protocols, OSs and applications need to have a secure design. The implementations need a bug-free implementation. Security policies must be perfect. Users need to be "perfect" as in not falling for simple attacks.

IDS: Intrusion Detection System, log-based abuse control. Machine learning algorithms are beginning to be used for log checking, for their ability to detect patterns and discern normal patterns from malicious patterns.

In the real world, effective security protections are not deployed. Patches are not applied in a timely manner. Websites do not properly monitor and restrict access to their internal hosts and resources. Organizations do not allocate enough resources and manpower to security tasks. Users are uneducated about security issues. Sites do not implement the policies they define, if they even have any.

Absolute security does not exist. It is a tradeoff between functionality, security and usability. Rather than reducing attacks, it is more advisable to reduce their rate of success and limit the scope of their potential damage. An excess of security reduces the usability and performance of the system. Security has a cost and so does the lack of it.

Minimum viable security: intersection between what customers will buy and what the security team demands.

Non-technological security: logical security (access control), organizational security (assignment of critical roles), physical security (guards, locks, surveillance over the physical infrastructure).

Replacement cypher (monoalphabetic substitution)

Substitute each letter with another letter. Fixed cipher, each encrypted letter always corresponds univocally to the same unencrypted letter.

Brute force cipher breaking is highly complex, however:

- letters change their individual identity but not their groupings
- the cipher can be broken by matching the relative frequency of letters with the ones in the unencrypted alphabet

Nulls

Using least frequent symbols in positions that do not alter the meaning, e.g. instead of spaces. The symbol itself can be called through some sort of escape sequence, like repeating it twice.

Homophones

Use of a sequence of several symbols to substitute singular frequent characters. This changes the distribution of symbols to make it less recognizable.

Code words

Whole word substitution. The code word needs to be diffused along with the message, which might fall into the attacker's hands. No significant increase of protection over monoalphabetic substitution.

More complex approaches

Two possible lines:

- multiple encrypting alphabets
  - Leon Battista Alberti
  - Vigenère (predecessor of modern encryption)
- encryption of multiple letters as a unit
  - Porta
  - Playfair

Alberti cypher disk

Uses a rotating disk with letters both inside and outside the disk. Rotating the inner disk changes the cipher. For example, two different ciphers may be alternated for even and odd letters.

Porta cipher

It encrypts letters in pairs through a 26×26 encryption matrix that assigns each pair to its cell number. The key can be changed by changing the cipher numbers (even by substituting them with symbols) or the letters on row and column headers.

Vigenère cipher

Treat letters as numbers and sum them the cipher, mod 26.

If the key has t letters, the text will be stripped from blank spaces and broken up into groups of t letters. After that, each letter will be summed (mod 26) to the corresponding letter in the key.

It was considered unbreakable for a long time. The possible keys are 26^t where t is the key size.

It's resistant to frequency analysis. Babbage and Kasiski realized that the length of the key can be deduced from repeating letters.

Grids

Girolamo Cardano. Even-sized grid with holes in certain points. By alternatively rotating the grid in its four possible positions, the message can be read.

Playfair cipher

5×5 matrix, with the key at the beginning and the rest of the alphabet, in order, following it. It considers letters in pairs. Digrams with repeating letters add an "X" in between. Each digram marks the vertices of the diagonal of a rectangle on the matrix. The digram gets substituted by choosing the letters on the other diagonal.

It's better than a monoalphabetic cipher but easy to break using digram frequency analysis.

One-time pad (Vernam Cipher)

Perfect cipher: the encrypted and unencrypted messages are independent. The key is a set of n random independent binary bits, that are XOR-ed with the message itself, also n-sized.

Cipher disks and rotors

Cipher disks based on the Alberti cipher were the state of the art until the First World War. By the end of it, the first electromechanical rotors were built. Couples of metal disks that are connected by electrical wires with a permutation. One unit is similar to the Alberti cipher, but many interconnected rotors, rotating for each set of letters, are way safer. With 26 disks and n rotors, the permutations are 26^n.

Engima machine

It was based on three rotors, the configuration of which was changed every day. The rotors were connected to a "reflector", a disk that sent the signal back through the rotors making the cipher symmetrical. A keyboard was used for input, and the encrypted result was shown through lights representing letters.

The first rotor (furthest from the reflector) rotates at each character. The second rotates every 26 rotations of the first, unless the third has to rotate too, the third rotates on each turn of the second. That makes $26 \times 25 \times 16 = 16900$ disk permutations. On top of that, 6 couples of letters were connected to each other on a connector board, increasing the number of permutations.

Every day the new key was sent. The key was sent twice. Weaknesses:

- no letter encrypts itself
- letters do not encrypt contiguous letters
- the cipher is symmetrical
- keys are short

Some words were in a fixed position (e.g. `WETTER`, "weather" in weather reports), giving an advantage to the Allies in code breaking. Two "decoding bombs", Victory and Agnus Dei.

The Kriegsmarine used an 8-rotor machine with a rotating reflector and did not use stereotypical formulas.

Traditional ciphers are text-based, unlike modern ciphers that need to be digital to cover all kinds of information.

Honeypot: bait resource within a secure system that authorized users do not access - whoever accesses it is an attacker that fell into the trap.

Symmetrical ciphers

Both writer and reader use the same key.

Kerckhoffs' principle: the security of a cryptographic system must depend only on the security of the key and not on the security of the system.

4 attacks:

1. known ciphertext attack
2. known plaintext attack
3. chosen plaintext attack
4. chosen ciphertext attack

Known ciphertext / plaintext : encrypted or unencrypted messages are captured by the attacker.

Chosen plaintext / ciphertext: the attacker can decide which messages are sent, before or after encryption, without knowing the key.

The Vigenère cipher is easily broken with a known ciphertext attack. By finding repeating patterns, their distance is a multiple of the key length.

Coincidence index: probability of two random characters in a string being the same. It changes depending on the language. It can be used to find the length of the key.

Mutual coincidence string: probability of two randomly chosen characters, one from a first string and the other from a second string, being equal. It can be used for finding the correct key given its length.

Block vs stream ciphers: in blocks vs character-by-character. The cleartext is broken up into equally sized blocks, which are then encrypted one at a time.

Feistel block encryption (1973): based on permutations and substitution within blocks. Principles (derived from Shannon's 1949 work):

- diffusion: mutiple unencrypted characters can lead to the same encrypted character
- confusion: hard to detect statistical relationships between encrypted text and keys

Increasing block size enhances security at the expense of speed. The same trade-off applies to key size.

DES (Data Encryption Standard)

DES uses 64 bit blocks and 56 bit keys. AES uses 128 bit blocks and 128 or 256 bit keys.

DES and Blowfish are based on Feistel rounds.

Many cryptography algorithms are based on multiple rounds. This increases security. For instance, AES has never been broken for high round numbers. The default number of rounds for AES-256 is 14 rounds.

Even with hardware support for encryption, with the possibility of executing a whole round in a single clock cycle, there is still a need to repeat the procedure for a number of clock cycles equal to the number of rounds. This is not a problem for modern consumer devices but it may be in some other domains.

DES was requested by NBS (NIST) in 1973 and published in 1977.

The key is 64 bits long - 8 bytes of which, for each one, the 8th bit is a *parity bit* (a `xor` of the previous 7).

The cleartext is passed through:

- an IP permutation (8×8 permutation matrix)
- 16 iterations of key scheduling on 48-bit blocks
- an exchange
- a reverse IP^{-1} permutation

The following scheme represents a single iteration:

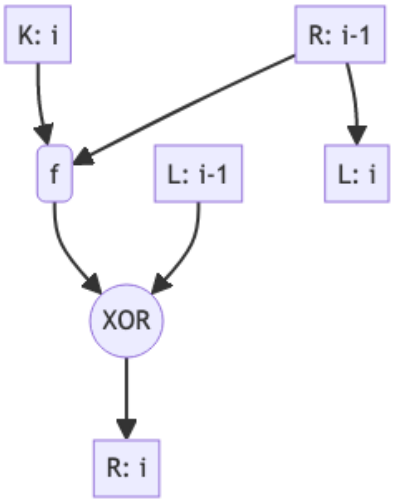

Figura 1: diagram

The function f used in iterations works in this way:

- the 32-bit input is *expanded* to 48 bits by repeating half of them twice, following an expansion table which specifies in which position to copy each bit
- the expanded 48-bit block is `xor`-ed with a 48-bit subkey
- the result is divided in 8 6-bit block and each block is passed through an *S-Box*

- the S-boxed output is reunited and de-expanded to go back to 32 bits

S-box (NBD, 1976):

- each row is a permutation of integers 0-15
- no S-box is a linear function of its inputs
- changing one bit in the input causes at least two output bits to change
- works in 6 bit blocks
  - the first and last bit specify the row
  - the center bits specify the column
  - the output is 4 bits from the table cell
- for each S-box and each 6-bit input, the S-box output and the S-box output of the number `xor`-ed with `001100` differ for at least two bits
- for each S-box, the number of inputs for which the output bit is 0 is around the same for which it is 1

The subkeys are the output of a *key scheduling* operation:

- the 64-bit key passes through a PC-1 permutation
  - table-based like the IP
  - bits 8, 16, 24, 32, 40, 48, 56 and 64 are parity bits and do not appear
- the 56-bit output is left-shifted by a specific number of bits (1 or 2) for each iteration, with a total shift of 28 bits on 16 iterations
- the left-shifted results are passed through a PC-2 permutation to produce the 48-bit subkeys for each iteration
  - table-based
  - suppression of bits 9, 18, 22, 25, 35, 38, 43, 54

Decryption uses the same process as encryption but reverses the order of the subkeys.

A key is called *weak* if it produces the same subkey for all 16 iterations. A couple of keys is said to be *semi-weak* if the subkeys are two, each used 8 times. There are 6 semi-weak key couples and 4 weak keys.

DES enjoys the *complement property*: the DES encryption of the complement of the input and the complement of the key is the complement of the output. This makes a bruteforce chosen plaintext attack take only 2^{55} attempts instead of 2^{56}.

DES was broken in the late 90's through a series of paid challenges open to private users and companies. These were brute-force attacks with 2^{56} (or 2^{55}, for the optimized version working with complements) possibilities. More sophisticated attacks, based on linear (2^{43} attempts) and differential (2^{47} attempts) crypto-analysis, were made through a plaintext + ciphertext attack.

Encryption modes are possible answers to the question "how do I organize encryption for messages that are longer than a single block?". Possible options (all called by a three-letter acronym):

- Electronic Codebook (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)
- Galois Counter Mode (GCM)

The last one was added later, while the first five come from the original *DES Modes of Operation* of 1977.