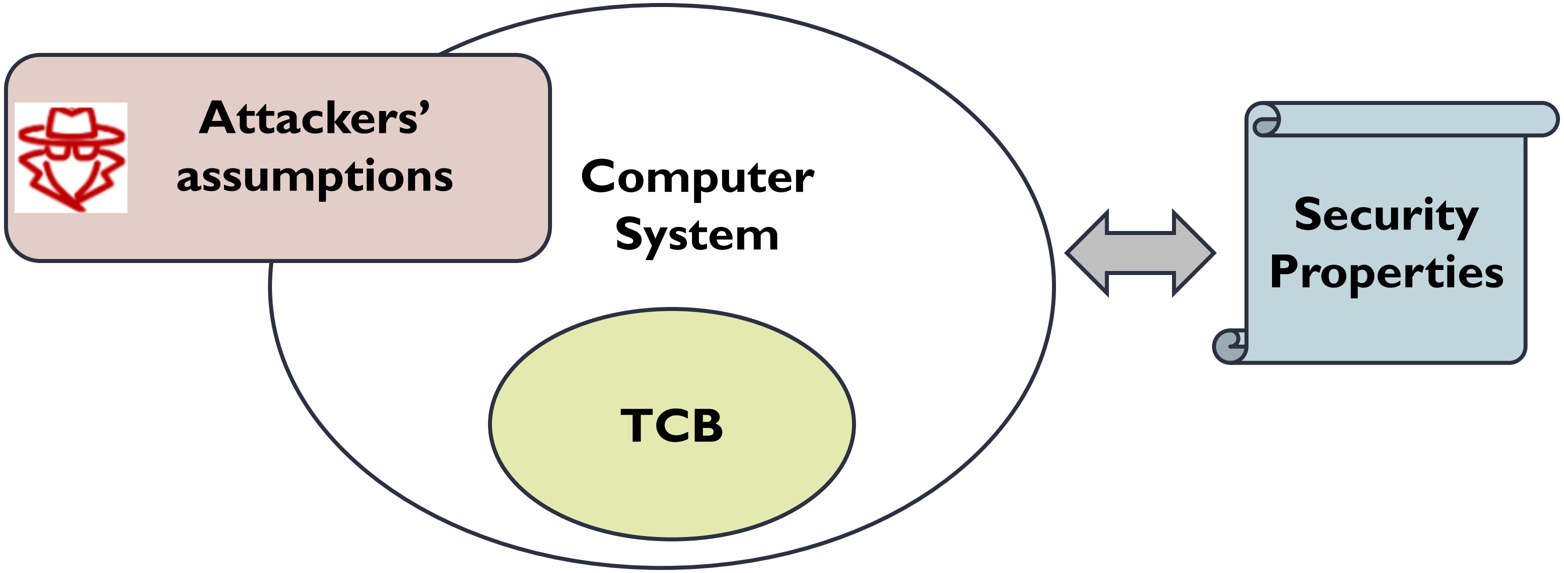
Threat Model

Describe the adversaries and threats in consideration

- What is trusted and what is not trusted (TCB).
- For the untrusted entities, what resources, capabilities and knowledge they have; what actions they can perform.
- What security properties the system aim to achieve.



Trust

The degree to which an entity is expected to behave:

- What the entity is expected to do:
  - Anti-malware can detect malicious programs;
  - System can prevent illegal account login, etc.
- What the entity is expected not to do:
  - The website will not expose your private data to third parties;
  - An application will not inject virus into your system.

Security cannot be established in a cyber system if no entities are trusted.

It is important to make clear what should be trusted. Otherwise, the designed security solutions may fail in practice.

Trusted Computing Base (TCB)

A set of components (e.g., software, OS, firmware, hardware) that need to be trusted to ensure the security of the cyber system

Components outside of the TCB can be malicious and misbehave.

When we design a security solution, we need to

- Assume all the components inside the TCB are secure <u>with valid justifications</u>.
- Prevent any damages from any components outside of the TCB.

TCB Design

Design principles

- Unbypassable (completeness): there must be no way to breach system security by bypassing the TCB.
- Tamper-resistant (security): TCB should be protected against other parts outside the TCB. These parts cannot modify the TCB's code or state.
- Verifiable (or correctness): it should be possible to verify the correctness of TCB.

Size of TCB

- A system with a smaller TCB is more trustworthy and easier to verify (we do not need to make too many assumptions, which may be violated). This follows the KISS (Keep It Simple, Stupid) principle
- Designing a secure system with a smaller TCB is more challenging (we need to consider more malicious entities)

Attacker's Assumption

Type of attacker

- Active: manipulate or disrupt the systems, e.g., modifying data, injecting code
- Passive: observing and gathering information without interfering system

Attacker's knowledge

- Know the system's design, architecture, source code, etc. ,
- Lack the detailed knowledge and must rely on probing or trial and error

Attacker's capability

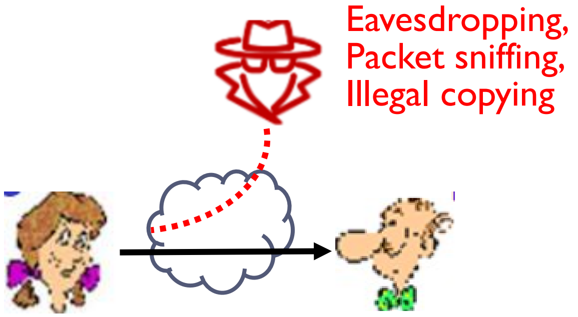
- How much computing resources can the attacker leverage?
- What parts of the system can the attacker interact with?
- Does the attacker have unlimited time or need to act quickly?

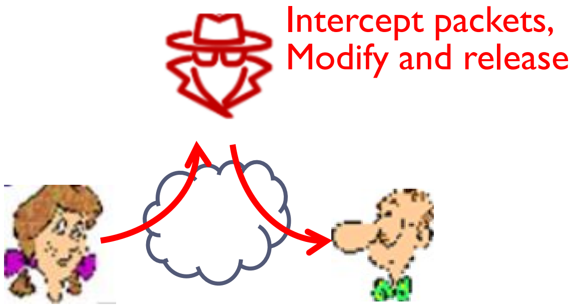
Security Properties

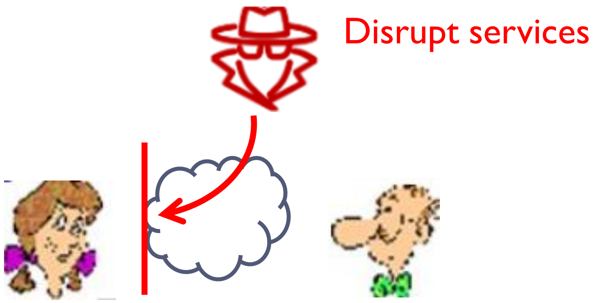
The security goals that we aim to achieve for the system.

Common security properties (CIA model)

- Confidentiality (C): prevent unauthorized disclosure of information. Sensitive information should not be leaked to unauthorized parties
- Integrity (I): prevent unauthorized modification of information. Critical system state and code cannot be altered by malicious parties
- Availability (A): prevent unauthorized withholding of information or resources. The resources should be always available for authorized users







Security Properties

Other properties

- Accountability: actions of an entity can be traced and identified
- Non-repudiation: unforgeable evidence that specific actions occur
- Authenticity: ensure the communicated entity is the correct entity.
- Anonymity or privacy: hide personal information and identity from being leaked to external parties.
- Verifiability: the system's operations can be independently verified.
- Freshness: the data or communications are current and not reused or replayed.
- Fault tolerance: the system can continue to function correctly despite failures.