C1 Intro

1. Security Fundamentals

1.1 Security Components & Goals

Components

- Confidentiality
  - Content + Existence
- Integrity
  - Correctness of Content + Origin (Authentication)
- Availability
  - not a finite property
  - cannot be treated probabilistically for security

1.2 Security Breaches

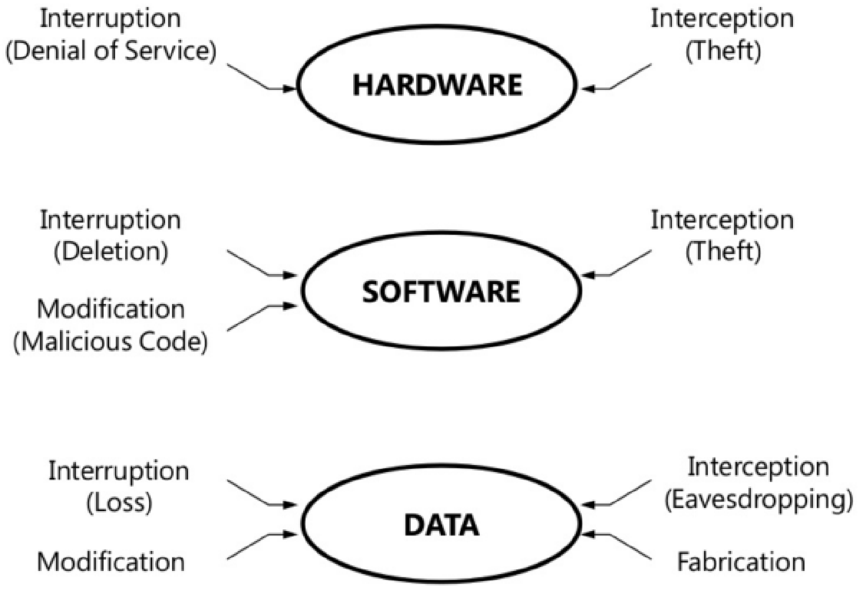
Assets

- Hardware, Software, Data

Threats

- **Interruption:** an asset of the system becomes lost, unavailable or unusable
- **Interception:** some unauthorized party has gained access to an asset
- **Modification:** an unauthorized party not only access but tampers with an asset

- **Fabrication:** an unauthorized party might make counterfeit objects on a computing system



1.3 Security Aspects

Communications (network) security

- Addresses security of the communications links

Computer Security

- Addresses security of the end systems (Software security fits in here)

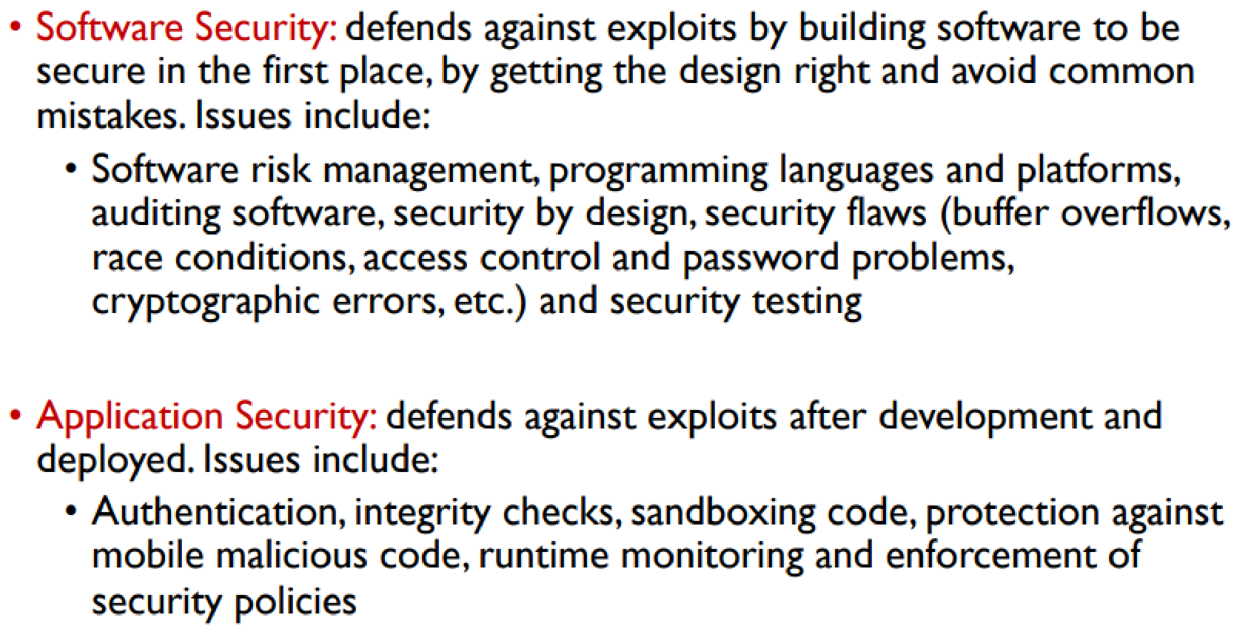
Application Security

- Relies on both to provide services securely to end users

Security management

- How to deploy security technologies

Software Security vs Application Security



- Software Security: defends against exploits by building software to be secure in the first place, by getting the design right and avoid common mistakes. Issues include:
  - Software risk management, programming languages and platforms, auditing software, security by design, security flaws (buffer overflows, race conditions, access control and password problems, cryptographic errors, etc.) and security testing

- Application Security: defends against exploits after development and deployed. Issues include:
  - Authentication, integrity checks, sandboxing code, protection against mobile malicious code, runtime monitoring and enforcement of security policies

1.4 Malware

Virus

- Self-replicating code that spreads by embedding itself in executable files or system areas in memory
- Not structured to exist by itself (needs a host that it executes together with)

Worm

- Self-contained self-replicating program; does not need to be part of another program to propagate itself

Trojan

- Malign program disguised as legitimate software, not intended to replicate itself

1.5 Vulnerabilities

1.6 Security Strategies

Prevention

- Take measures that prevent assets from being damaged

Detection

- Take measures so that you can detect when, how and by whom an asset has been damaged

Reaction

- Take measures so that you can recover your assets or to recover from a damage to your assets

Countermeasures for Vulnerabilities

- **Prevention**

    - Avoid vulnerabilities in new code
    - Eliminate vulnerabilities from existing code base
    - Harden execution environment so that attempts to exploit vulnerable code are stopped

- **Detection & Reaction**
    - Virus / malware scanners
        - Canaries (run-time checks)
    - IDPS

2. Secure Software

2.1 Why secure software

Secure products are quality products

- Security is a subset of quality, a product that is not appropriately secure is inferior to competing products

Media and Competitors leap on security issues

- You do not want your products in the headlines due to security issues

People shy away from products that don't work as advertised

- People will begin to shy away and start looking for solutions from competitors

Don't be a victim

- You do not want your product to be a trophy on someone's wall

Security vulnerabilities are expensive to fix

- Fixes are expensive to make late in the development process

Secure Software

- Secure software $\neq$ Software with security features
    - Security is not a feature you can add to a system at any time
    - Security is a behavioural property of a complete system in a particular environment
- A system that is secure enough in one environment may be insecure when placed in another

2.2 Design for security

- Security should be considered during all phases of the development cycle and should deeply influence a system's design
- Security model
    - How data flow between components

- Any users, roles and rights, either explicitly stated or implicitly included in the design
- The trust relationships of each component
- Any potentially applicable solution to a recognised problem