Information Security

Dr. Nguyen Khanh Van
Dept. of Software Engineering
Hanoi University of Technology

Introduction

- First approach: What the system designer/security architect should know
  - Components of computer security
  - Threats
  - Policies and mechanisms
  - Assurance
  - Operational and Human issues
  - The security life cycle
- Second approach: the security engineer's

Basic Components

- Confidentiality
  - Keeping data and resources hidden
- Integrity
  - Data integrity (integrity)
  - Origin integrity (authentication)
- Availability
  - Enabling access to data and resources

Classes of Threats

- Disclosure
  - Snooping
- Deception
  - Modification, spoofing, repudiation of origin, denial of receipt
- Disruption
  - Modification
- Usurpation
  - Modification, spoofing, delay, denial of service

Snooping

- The unauthorized interception of information, is a form of disclosure.
  - Passive: some entity is listening to/reading communications …
  - (Passive) wiretapping is a form of snooping in which a network is monitored (wire: the network)
  - Confidentiality services counter this threat.

Modification

- Or alteration, an unauthorized change of information, covers three classes of threats.
  - Deception: incorrect information is accepted as correct/ wrong decision is made.
  - Disruption and usurpation: If the modified data controls the operation of the system
- Active wiretapping is a form of modification in which data moving across a network is altered.
  - Example: the man-in-the-middle attack
- Integrity services counter this threat.

The man-in-the-middle attack

- An intruder reads messages from the sender and sends (possibly modified) versions to the recipient,
  - Succeeds if the recipient and sender don't realize his presence.

Repudiation of origin

- A false denial that an entity sent (or created) something.
  - Example: suppose a customer ➔ a letter agreeing to pay for a product ➔ the vendor ships the product and then demands payment ➔ the customer denies having ordered the product and keep the unsolicited shipment without payment.
    - The customer has repudiated the origin of the letter. If the vendor cannot prove that the letter came from the customer, the attack succeeds.
  - Integrity mechanisms cope with this threat.

Denial of receipt

- A false denial that an entity received some information or message.
  - E.g. A customer orders an expensive product and pays in advance: customer pays ➔ vendor ships. The customer then falsly asks the vendor when he will receive the product➔ denial of receipt attack.
    - The vendor can defend against this attack only by proving that the customer did, despite his denials, receive the product.
  - Integrity and availability mechanisms guard against these attacks.

Denial of service

- A long-term inhibition of service, so a form of usurpation, although often used with other mechanisms to deceive.
  - The attacker prevents a server from providing a service. The denial may occur at
    - the source (by preventing the server from obtaining the resources needed to perform its function),
    - at the destination (by blocking the communications from the server), or along the intermediate path (by discarding messages from either the client or the server, or both).
  - Availability mechanisms counter this threat.

Policies and Mechanisms

- Policy says what is, and is not, allowed
  - This defines “security” for the site/system/*etc*.
- Mechanisms enforce policies
- Composition of policies
  - If policies conflict, discrepancies may create security vulnerabilities

Policies and Mechanisms

- Policy: may be expressed in
  - natural language, which is usually imprecise but easy to understand;
  - mathematics, which is usually precise but hard to understand;
  - policy languages, which look like some form of programming language and try to balance precision with ease of understanding

Policies and Mechanisms

- Mechanisms: may be
  - technical, in which controls in the computer enforce the policy, e.g. a user has to supply a password to authenticate herself before using
  - procedural, in which controls outside the system enforce the policy; e.g. , firing someone for bringing in a game disk from an untrusted source
- The composition problem requires checking for inconsistencies among policies

Goals of Security

- Prevention
  - Prevent attackers from violating security policy
- Detection
  - Detect attackers' violation of security policy
- Recovery
  - Stop attack, assess and repair damage
  - Continue to function correctly even if attack succeeds

Assurance

- Assurance is a measure of how well the system meets its requirements; i.e. how much you can trust the system to do what it is supposed to do.
- Assurance techniques:
  - Specification
  - Design
  - Implementation

Specification

- Specification
  - Arise from Requirements analysis
  - Statement of desired functionality: says what the system must do to meet those requirements. Can be
    - very formal (mathematical) or informal (natural language)
    - high-level or low-level
      - E.g. describing what the system as a whole is to do vs. what specific modules of code are to do

Design and Implementation

- Design: How to meet specification
  - Typically, the design is layered by breaking the system into abstractions, and then refining the abstractions (work down to the hardware).
  - An analyst must show the design matches the specification.
- Implementation
  - Actual coding of the modules and software components.
    - These must be correct (perform as specified), and their aggregation must satisfy the design.

Operational Issues

- Cost-Benefit Analysis
  - Is it cheaper to prevent or recover?
- Risk Analysis
  - Should we protect something?
  - How much should we protect this thing?
- Laws and Customs
  - Are desired security measures illegal?
  - Will people do them?

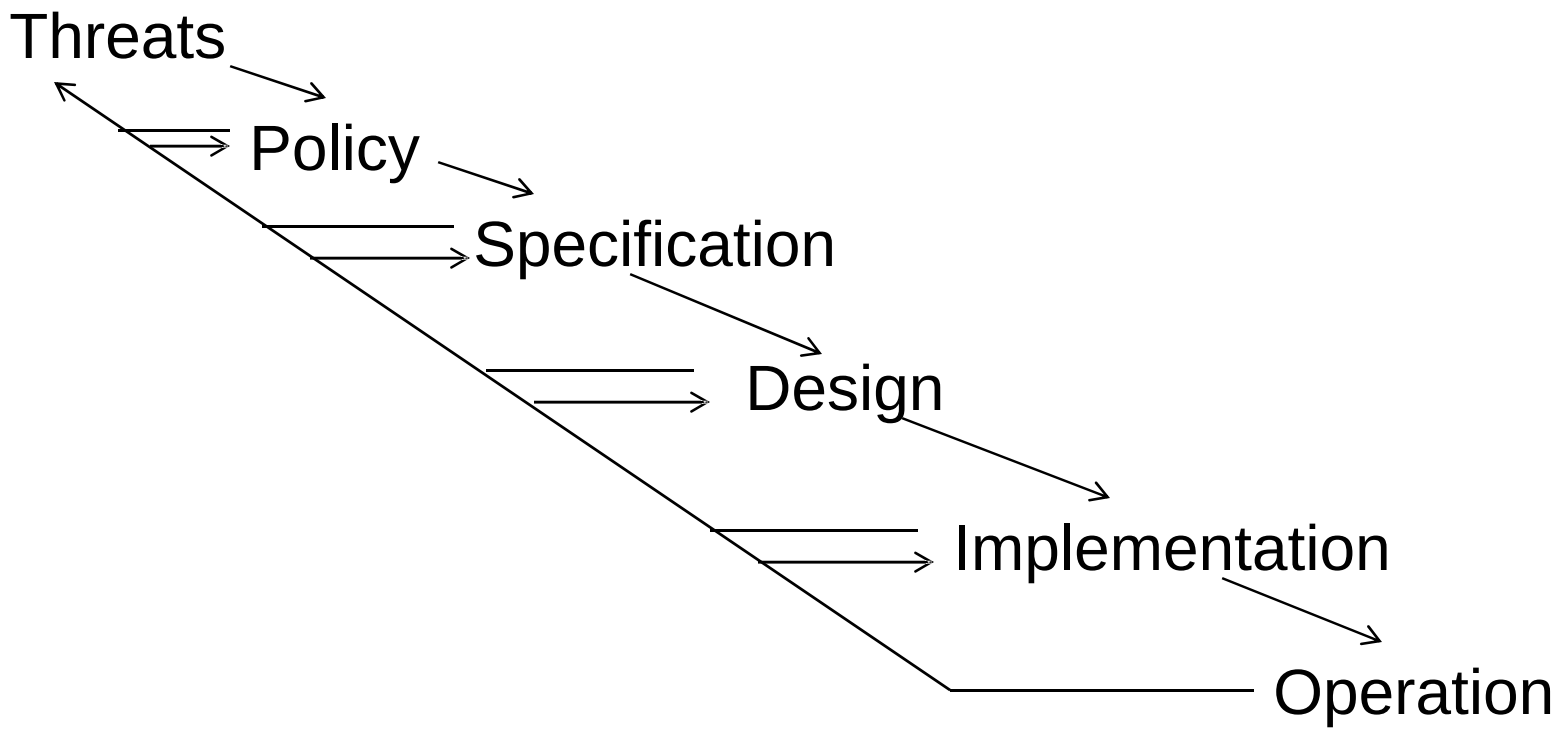
Human Issues

- Organizational Problems
  - Power and responsibility
    - those responsible for security have the power to enforce security (not responsibility without power or vice versa)
  - Financial benefits
    - Tricky: security is not a direct financial incentive, only appreciated when loss occurs

Human Issues

- People problems
  - Outsiders and insiders
  - Social engineering

The security lifecycle



Key Points

- Policy defines security, and mechanisms enforce security
  - Confidentiality
  - Integrity
  - Availability
- Trust and knowing assumptions
- Importance of assurance
- The human factor

EXAMPLE

- A major corporation decided to improve its security.
  - Hired consultants → determined the threats → created a policy → derived specifications that the security mechanisms had to meet → developed a design that would meet the specifications.
- During the implementation phase
  - discovered [modems to the telephones] → firewall → the design had to be modified to divide systems into two classes: outside or behind the firewall

EXAMPLE

- When deployed, the operation and maintenance phase revealed several unexpected threats.
  - sensitive data sent across the Internet in the clear → crypto is very difficult to use→ fixed implementation
  - several "trusted" hosts (allowed to log in without authentication) were physically outside the company'control
    - This violated policy, because of commercial reasons → modified the policy element about "trusted hosts"
  - Finally, the company detected proprietary material being sent to a competitor over electronic mail.
    - This added a threat that the company had earlier discounted. The company did not realize that it needed to worry about insider attacks.

SECURITY ENGINEER'S ANGLE

Security Goals

- Confidentiality (secrecy, privacy)
  - Assure that data is accessible to only one who are authorized to know
- Integrity
  - Assure that data is only modified by authorized parties and in authorized ways
- Availability
  - Assure that resource is available for authorized users

Terminologies

- Vulnerabilities (weaknesses)
- Threats (potential scenario of attack)
- Attacks
- Controls (security measures)

Methods of Defense

- Prevention
- Deterrence
- Reflection
- Detection
- Recovering

Controls

- Encryption
- Software controls
- Hardware controls
- Policies and procedures
- Physical controls

What is This Course About?

- Learn to think about security
  - Threats, defenses, policies
  - Software, human and environment factors
- Think as an attacker:
  - Learn to identify threats
- Think as a security designer:
  - Learn how to prevent attacks and/or limit their consequences
  - Understand and apply security principles
  - Learn tools that can defend against specific attacks, no silver-bullet solution

Agenda

- A gentle intro to Cryptography
- Operating systems security (access control mechanisms)
- Network security
- Software and Program security
- Database security
- Legal and ethical issues

Course Material

- Introduction to Computer Security, Matt Bishop, Addison-Wesley Professional
- Security in Computing, Charles P.Pfleeger, Prentice Hall
- Cryptography And Network Security: Principles and Practices, William Stallings, Prentice Hall
- Other material: Lecturer's website