Chapter 1: Introduction

- Components of computer security
- Threats
- Policies and mechanisms
- The role of trust
- Assurance
- Operational Issues
- Human Issues

Basic Components

- Confidentiality
- Integrity
- Availability

Basic Components

- Confidentiality
  - Keeping data and resources hidden - Why?
    - Use of computers in sensitive fields such as government and industry
    - First formal work in computer security was motivated by Military's attempt to implement controls to enforce a "Need to know" principle
  - Access control mechanisms support confidentiality
    - Cryptography

Basic Components

- Confidentiality also applied to the existence of data which is sometimes more revealing than the data itself
  - The number of people who distrust a politician may be less important than knowing that such a poll was taken by the politician's staff.
  - How a particular government agency harassed citizens in its country may be less important than knowing that such harassment occurred.
- Resource hiding is another aspect of confidentiality
  - Sites often wish to conceal their configurations as well as what systems they are using
  - Organizations may not wish others to know about specific equipment

Basic Components

- Integrity
  - Prevents improper use of data
  - Trustworthiness of data or resources

  - Data integrity (integrity) – Content of information
  - Origin integrity (authentication) – Source of information

  - Integrity mechanisms
    - Prevention Mechanism
    - Detection Mechanism

Basic Components

- Integrity
  - Source of the information may bear on its accuracy and credibility and on the trust that people place in the information
  - Example: A newspaper may print information obtained from a leak at the White House but attributes it to the wrong source.
    - The information is printed as received (preserving Data Integrity)
    - But the source is incorrect (corrupting Origin Integrity)

Basic Components

- Integrity
  - Prevention mechanism seek to maintain the integrity of data by blocking
    - Any unauthorized attempt to change the data (1)
    - Any attempt to change the data in unauthorized ways (2)

What is the difference?

Basic Components

- Integrity
  - Prevention mechanism seek to maintain the integrity of data by blocking
    - Any unauthorized attempt to change the data (1)
    - Any attempt to change the data in unauthorized ways (2)

(1) Occurs when a user tries to change data which he/she has no authority to change

(2) Occurs when a user authorized to change data tries to change data in other ways

Basic Components

- Integrity
  - Prevention mechanism seek to maintain the integrity of data by blocking
    - Any unauthorized attempt to change the data (1)
    - Any attempt to change the data in unauthorized ways (2)

    Example:

    An accounting system on a computer.
    Someone breaks into the system and tries to modify the accounting data (1).
    An accountant hired by the firm transfers money to her account (2).

Basic Components

- Integrity
  - Detecting mechanism do not prevent violation of integrity
  - They simply report that data's integrity is no longer valid / trustworthy
  - How does the detecting mechanism work
    - Analyze System Events (user/system activities)

Basic Components

- Confidentiality VS Integrity
  - Working with Integrity is different from working with Confidentiality
  - With Confidentiality → Data is either compromised or it is not
  - Integrity includes → Both correctness and the trustworthiness of the data
    - Origin of the data (how and from whom it was obtained)
    - How the data was protected before it arrived

Basic Components

- Availability
  - Refers to the ability to use the information or resource desired
  - Important aspect of reliability
  - Disabling access to data and resources
  - Attempt to block availability is called *Denial of Service Attacks*

Threat

- A potential violation of security
- The violation need not actually occur for there to be a threat
- Violation might occur means that those actions that could cause it to occur must be guarded against (or prepared for)
  - Those actions are called attacks
  - Those who execute such actions are called attackers
- Can be categorized into four broad classes
  - Disclosure
  - Deception
  - Disruption
  - Usuruption

Classes of Threats

- Disclosure (unauthorized access to information)
  - Snooping
    - Unauthorized interception of information
      - Passive
      - Wiretapping

- Deception (acceptance of false data)
  - Modification
    - Some entity relies on the modification of data
    - Active
    - Man-in-the-middle attack
  - Spoofing (or masquerading)
    - Impersonation of one entity by other
    - Difference between Delegation and Masquerading
  - Repudiation of origin
    - A false denial that an entity sent something
  - Denial of receipt
    - A false denial that an entity received some information or message

Classes of Threats

- Disruption (interruption or prevention of correct operation)
  - Modification
- Usurpation (unauthorized control of some part of a system)
  - Modification
  - Delay
    - Temporary inhibition of service
  - Denial of service
    - A long-term inhibition of service
    - Denial may occur at
      - the source
      - the destination
      - along the intermediate path

Policies and Mechanisms

- Policy says what is, and is not, allowed
  - This defines "security" for the site/system/*etc*.
  - A *security policy* is a statement of what is, and what is not allowed.
  - A *security mechanism* is a method, tool, or procedure for enforcing a security policy

Policies and Mechanisms

- Mechanisms enforce policies
  - Mechanism can be non-technical, such as requiring proof of identity before changing a password
  - Suppose a university's computer science laboratory has a policy that prohibits any student from copying another student's homework files.
  - The computer system provides mechanisms for preventing others from reading a user's file.
  - Anna fails to use these mechanisms to protect her homework files, and Bill copies them.
  - A breach of security has occurred, because Bill has violated the security policy.
  - Anna's failure to protect her files does not authorize Bill to copy them.

Policies and Mechanisms

- Policies may be presented mathematically as a list of allowed (secure) and disallowed (non-secure) states
- Policies should be unambiguous
  - Consider the homework policy. If someone looks through another user's directory without copying homework files, is that a violation of security?
- Composition of policies
  - If policies conflict, discrepancies may create security vulnerabilities

Goals of Security

- Prevention
  - Prevent attackers from violating security policy
  - Involve implementation of mechanisms that users cannot override and that are trusted to be implemented in a correct, unalterable way
  - Preventive mechanisms are cumbersome
- Detection
  - Detect attackers' violation of security policy
  - Most useful when an attack cannot be prevented
- Recovery
  - Stop attack, assess and repair damage
    - If the attacker deletes a file, file can be restored from backup
    - In some cases, retaliation is part of recovery
  - Continue to function correctly even if attack succeeds
    - Difficult to implement

Assumptions and Trust

- How to determine if the policy correctly describes the required level of security
- Security rests on assumptions specific to the type of security required and the environment on which is to be employed
- Example: Opening a door lock requires a key.
    - Assumption: The lock is secure against lock picking
- Policies (set of axioms)
  - Unambiguously partition system states
  - Correctly capture security requirements
- Mechanisms
  - Assumed to enforce policy
  - Support mechanisms work correctly

Assumption and Trust (contd.)

- Designers of policies always make two assumptions
  - The policy correctly and unambiguously partitions the set of system states into secure and non-secure states
  - The security mechanisms prevent the system from entering into a non-secure state
- The first assumption asserts that the policy is correct description of what constitutes a secure system
  - For example: a bank's policy may state that officers of the bank are authorized to shift money among accounts

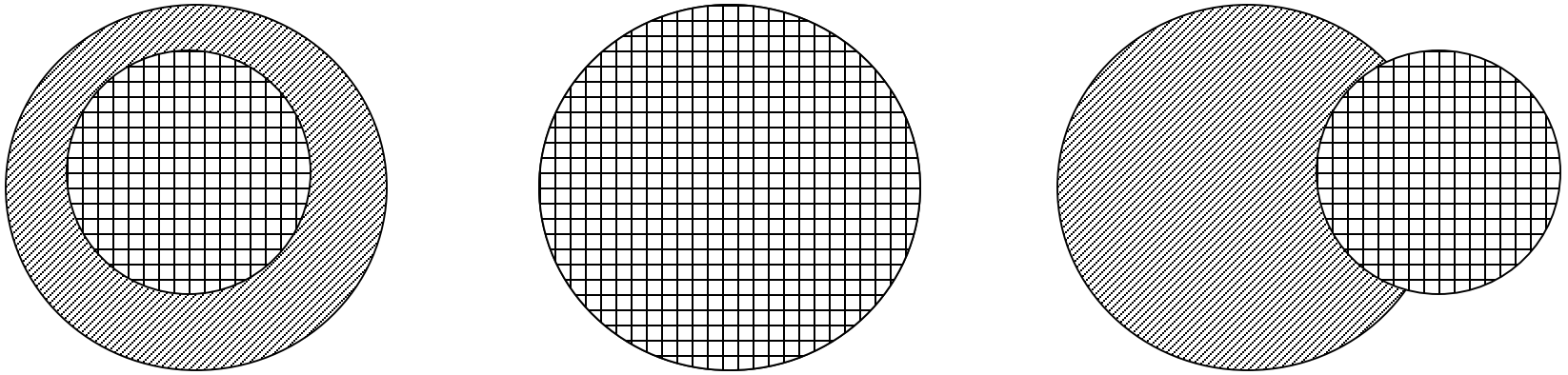
Assumptions and Trust

- The second assumption says that the security policy can be enforced by security mechanisms
- These mechanisms are either
  - Secure,
  - Precise, or
  - Broad

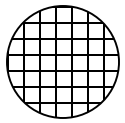
Assumptions and Trust (contd.)

- P : the set of all possible states.
- Q : the set of secure states
- R: the set of states restricted by the security mechanisms
- Obviously $R \subseteq P$
- Definition: A security mechanism is
  - secure if $R \subseteq Q$
  - precise if $R = Q$
  - broad if there are states r such that $r \in R$ and $r \notin Q$

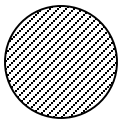
Types of Mechanisms



secure precise broad



set of reachable states



set of secure states

Example

Policy says that the students must take the online tests during the lab hours using only the lab computers. Students cannot take the online test outside the lab. Classify the following mechanisms as secure, precise, or broad.

– Students will write their names on a signup sheet before they enter lab to take the test. The instructor checks the signup sheet to verify that everybody is taking the test inside the lab.
– The instructor restricts the online submission of tests using IP address mask which ensures that only machines which match the IP mask are used to access the test during the lab hours.

Assumptions and Trust (contd.)

- Trusting the mechanisms work requires several assumptions
  - Each mechanism is designed to implement one or more parts of the security policy
  - The union of the mechanisms implements all aspects of the security policy
  - The mechanisms are implemented correctly
  - The mechanisms are installed and administered correctly

Assurance

- How to quantify "trust"
- System specification, design and implementation can provide a basis of determining "how much" to trust a system (this is called "assurance")

- Example: In US, aspirin from a nationally known and reputed manufacturer, delivered to the drugstore in safety-sealed container, and sold with the seal still in place, is considered trustworthy by most of the people. Why?
  - Certification
  - Manufacturing standard
  - Preventive Seal

Assurance (contd.)

- Assurance in the computing world requires specific steps to ensure that the computer will function properly
  - Detailed specification of the desired behavior
  - An analysis of the design of hardware, software, and other components
  - An argument of proof that the implementation, operating procedures and maintenance procedure will produce the desired behavior

- **Definition**: A system is said to ***satisfy*** a specification if the specification correctly states how the system will function

Assurance (contd.)

- Specification
  - A statement of the desired functioning of the system
    - Formal or Informal
  - Requirements analysis
  - Example: A company is purchasing a new computer for internal use. “The system cannot be attacked over the Internet”
- Design
  - Translates the specifications into components that will implement them
  - Should satisfy the specifications
  - Example: For the above company, the design of the system could be “a computer system with no network interface card, no modem cards and no network drivers”
- An analyst can determine whether a design satisfies a set of specifications
  - By mathematical proofs
  - By convincing and compelling arguments

Assurance (contd.)

- Implementation
  - Given a design, the implementation creates a systems that satisfies the design
  - If the design also satisfies the specification, then by transitivity the implementation will satisfy the specifications
  - How do we prove that a program correctly implements the design

Assurance (contd.)

- **Definition**: A program is ***correct*** if its implementation performs as specified
- Proof of correctness
  - Require each line of source code to be checked for mathematical correctness
  - Each line can be seen as a function transforming the input (constrained by preconditions) into some output (constrained by postconditions)
  - Each routine can be seen as composition of functions

Assurance (contd.)

- Drawbacks of mathematical proof
  - Complexity of programs makes it difficult
  - Preconditions derived from environment are difficult to specify
  - Assumes that the programs are compiled correctly, linked and loaded correctly, and executed correctly

Assurance (contd.)

- As the formal proofs of correctness are time-consuming, ***testing*** has become widespread
- Testing ranges from supplying input to ensure that all the execution paths are exercised
- Simpler but does not provide same degree of assurance that formal methods do

Operational Issues

- Cost-Benefit Analysis
  - The benefits of computer security are weighed against their total cost (including the cost if the system is compromised)
  - Is it cheaper to prevent or recover?
  - Example 1: A database provides salary information to a second system that prints checks. If the data in the database is altered, the company would suffer financial loss.
  - Example 2: Suppose the company has several branch offices, and everyday the database downloads a copy of the data to each branch office. The branch offices use the data to recommend salaries for new employees. The main office makes the final decision using the original database.

Operational Issues

- Risk Analysis
  - Should we protect something?
  - How much should we protect this thing?
  - The level of protection is a function of the probability of an attack occurring and the effects of the attack should it succeed
  - If an attack is unlikely, protecting against it has a lower priority than protecting against a likely one
  - Example: Consider the company with the salary database that transmits salary information over a network to a second computer that prints employees' checks. The risk of unauthorized changes in the data occurs in three places
    - On the database
    - On the network
    - On the printing system
  - Risk is a function of environment
  - Risk changes with time

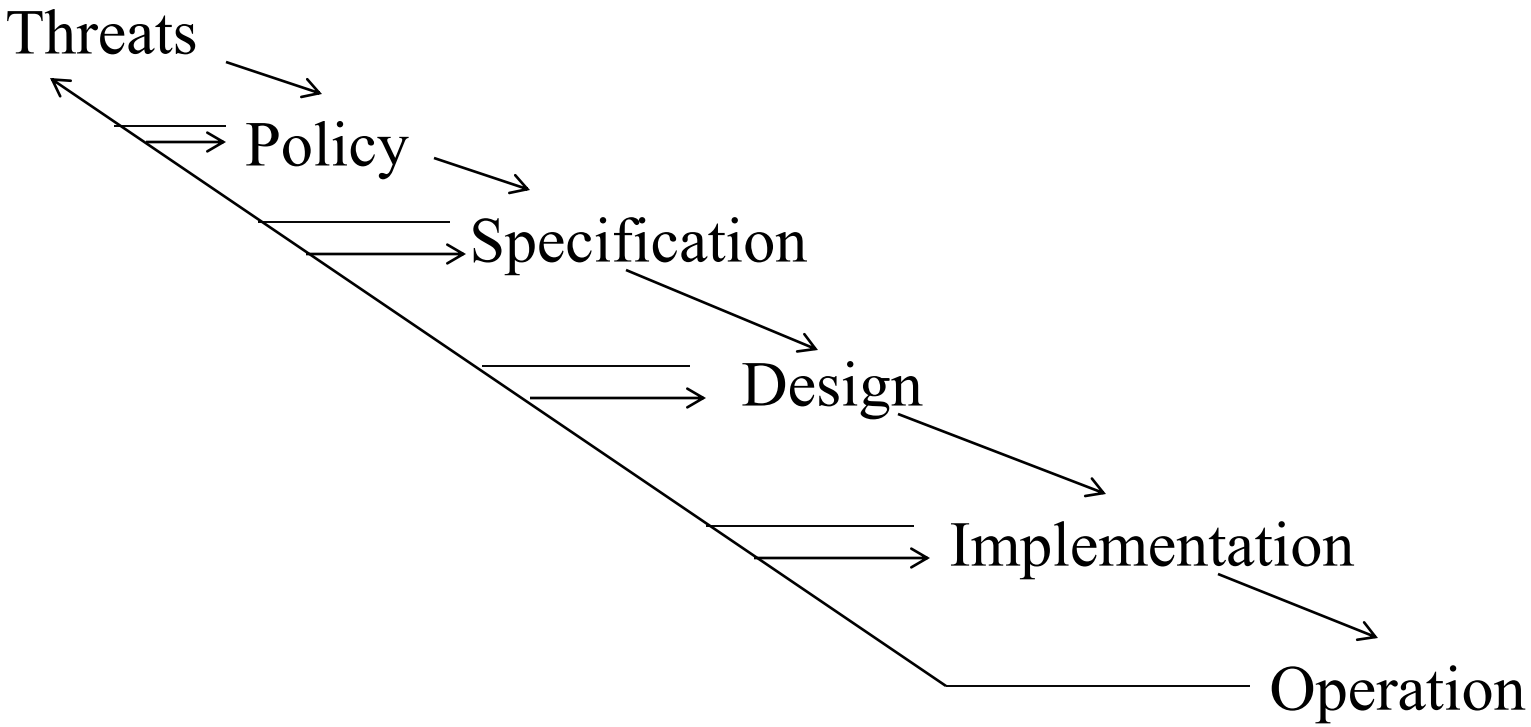
Operational Issues

- Laws and Customs
  - Are desired security measures illegal?
  - Law restricts the availability and use of technology and affect procedural controls
  - Example 1: In the 1990s, the laws of France require companies sending enciphered data out of the country to register their cryptographic keys with the government.
  - Example 2: Suppose the law makes it illegal to read a user's file without the user's permission. An attacker breaks into the system and begins to download users' files. If the system administrator notice this and observe what attacker is reading, they will be reading the victim's files without his permission.

Human Issues

- Organizational Problems
  - Power and responsibility
  - Financial benefits
- People problems
  - Outsiders and insiders
  - Social engineering

Tying Together



Key Points

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