Information Security (BTCS06001)

By
Prof. Pranita Binnar (Research Scholar, Ph.D. Pursuing)
Visiting Faculty, Dept. of Computer Engineering
NMIMS, Navi Mumbai Campus
Email Contact – pranitasadgir@gmail.com
Contact No. - 9699502992

Myself

- Ex-Scientific Officer, Cyber-Crime Forensic Lab, DFSL, Mumbai, GoM
- CEH, CHFI Certified by EC-council, USA
- PhD. Research Scholar research area Cyber security and Forensic
- CTF Player in IT/OT Domain
- Identifying vulnerability and reporting to respected vendor i.e Device VAPT.
- 02 best scientific research paper award securing 1st placed rank

Topics To Be Covered

- Introduction
- Security in Practices
- Confidentiality
- Integrity
- Availability
- Security violation and threats.

Learning Objectives

- Describe the key security requirements of confidentiality, integrity and availability
- Discuss the types security threats and attacks that must be dealt with
- Summarize the functional requirements for computer security

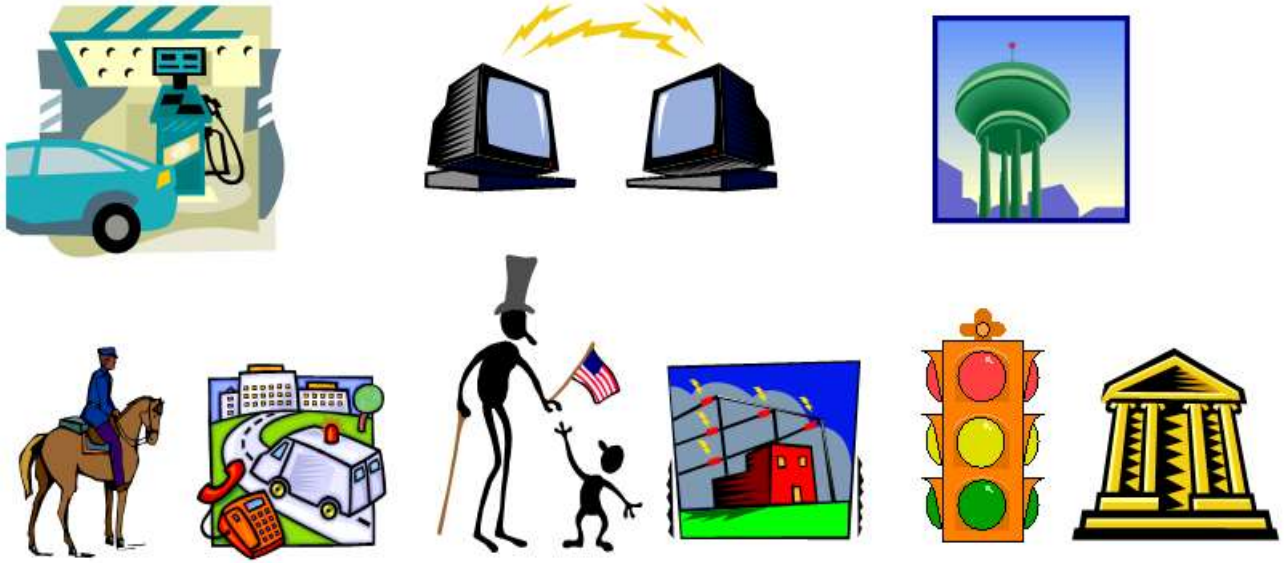

Information hiding
Security
Applications
Privacy
Negotiation
Access control
Integrity
Threats
DATA PROVENANCE
Biometrics
Semantic web security
Fraud
TRUST
POLICY MAKING
Encryption
Computer epidemic
Data mining
Anonymity
Formal models
System monitoring
Vulnerabilities
Network security

Security in Practices- **From CSI/FBI Report**

- 90% detected computer security breaches within the last year
- 80% acknowledged financial losses
- 44% were willing and/or able to quantify their financial losses.
- The most serious financial losses occurred through theft of proprietary information and financial fraud
- 34% reported the intrusions to law enforcement.
- 40% detected external penetration
- 40% detected denial of service attacks.
- 78% detected employee abuse of Internet access privileges
- 85% percent detected computer viruses.
- 38% suffered unauthorized access or misuse on their Web sites.
  [includes insider attacks]
- 12% reported theft of transaction information.
- 6% percent reported financial fraud

Critical Infrastructure Areas

- Include:
  - Telecommunications
  - Electrical power systems
  - Water supply systems
  - Gas and oil pipelines
  - Transportation
  - Government services
  - Emergency services
  - Banking and finance
  - …



Introduction - Information Security

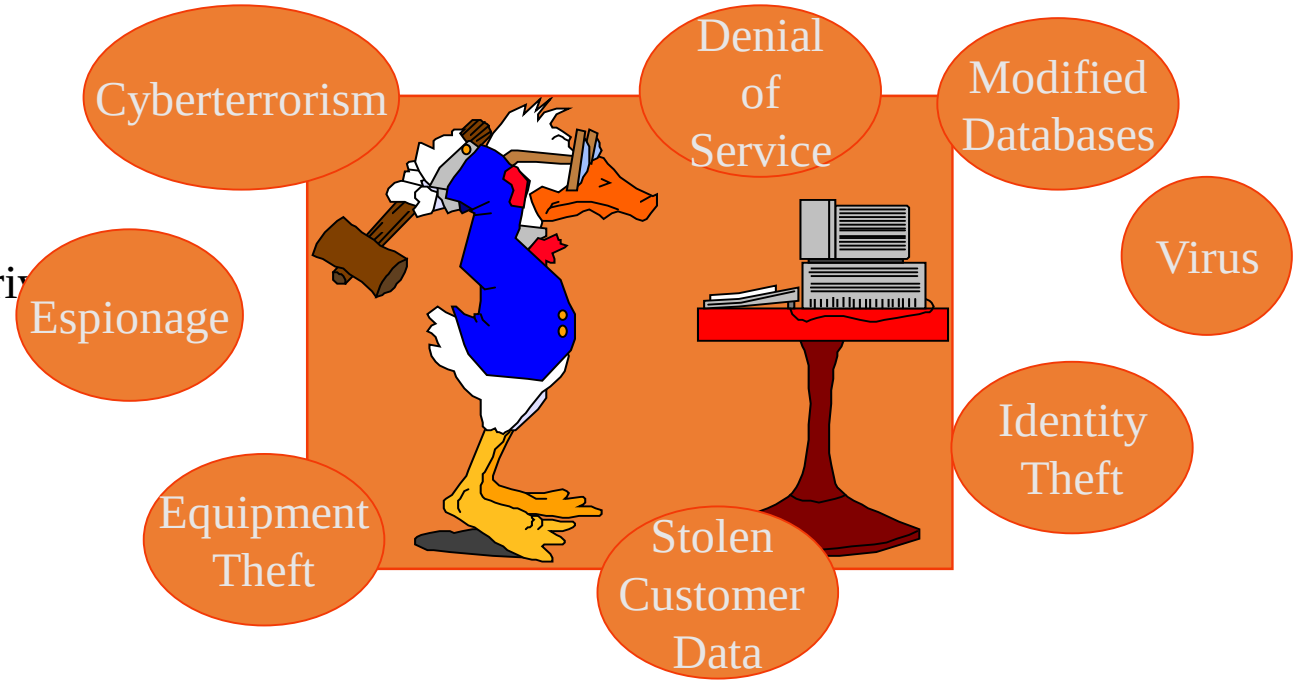
- The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability. By NIST

What is a “Secure” Computer System?

- To decide whether a computer system is “secure”, you must first decide what “secure” *means to you*, then identify the threats you care about.

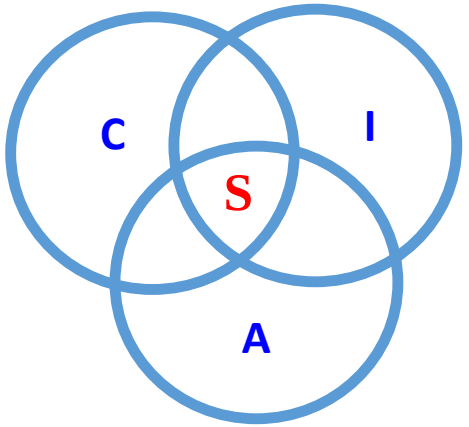
You Will Never Own a Perfectly Secure System!

- Threats(in Oval) - examples
  - Viruses, trojan horses, etc.
  - Denial of Service
  - Stolen Customer Data
  - Modified Databases
  - Identity Theft and other threats to personal pri
  - Equipment Theft
  - Espionage/spying in cyberspace
  - Hack-tivism
  - Cyberterrorism
  - …



Basic Components of Security: Confidentiality, Integrity, Availability (CIA)

- CIA
  - Confidentiality: Who is authorized to use data?
  - Integrity: Is data „good?”
  - Availability: Can access data whenever need it?



S = Secure

- CIA or CIAAAN... ☺

  (other security components added to CIA)
  - Authentication
  - Authorization
  - Non-repudiation
  - …

Ref: Security In Computing - Charles Pfleeger

Need to Balance CIA

- Example 1: C vs. I+A
  - Disconnect computer from Internet to increase confidentiality
  - Availability suffers, integrity suffers due to lost updates
- Example 2: I vs. C+A
  - Have extensive data checks by different people/systems to increase integrity
  - Confidentiality suffers as more people see data, availability suffers due to locks on data under verification)

Confidentiality

- "Need to know" basis for data access
  - How do we know who needs what data?

    Approach: access control specifies *who* can access *what*

  - How do we know a user is the person she claims to be?

    Need her identity and need to verify this identity

    Approach: identification and authentication

- Analogously: "Need to access/use" basis for physical assets
  - E.g., access to a computer room, use of a desktop i.e Confidentiality - concered with *access* to assets
- **Confidentiality**
  - **Data confidentiality**: Assures that confidential information is not disclosed to unauthorized individuals
  - **Privacy**: Assures that individual control or influence what information may be collected and stored

Risks

- Types Of Risk
  - Legal Risks
    - Fines, liability lawsuits, criminal prosecution
  - Financial Risks
    - Numerous costs involved including losing customer's trust, legal fees, fines
  - Reputational Risks
    - Loss of trust
  - Operational Risks
    - Failed internal processes – insider trading, unethical practices, etc.
  - Strategic Risks
    - Financial institutions future, mergers, etc.

Threats to Confidentiality

- Access to confidential information by any unauthorized person
- Intercepted data transfers
- Physical(HDD, Pendrive, etc) loss of data
- Privileged access of confidential information by employees
- Social engineered methods to gain confidential information
- Transfer of confidential information to unauthorized third parties
- Compromised machine where attacker is able to access data thought to be secure

Cont…

- Scheduling information regarding national level speakers/sensitive private meetings highly restricted
- Concerns over unauthorized access as a result of leaks – includes leaks to press as well as opposition/protest groups
- Concerns over “leaks” via IT from opposition groups within the national organization
- Loss of trust in decisions made at event
  - Can include public exposure of sensitive data

Cont…

- Common theme: leaking private data
- Strict access controls are crucial to protecting the confidential information
- Those who should have access to the confidential information should be clearly defined
  - These people must sign a very clear confidentiality agreement
  - Should understand importance of keeping the information private

Integrity

- Concerned with unauthorized *modification* of assets (= resources)

- **Integrity**
  - **Data integrity**: assures that information and programs are changed only in a specified and authorized manner
  - **System integrity**: Assures that a system performs its operations in unimpaired manner(Not weakened or damaged)

Availability

- **Availability**: assure that systems works promptly and service is not denied to authorized users

Revision

- What is Information Security?
- What are the Key elements of security?
- Explain the key elements ?
- What is the threads ? List some threads

Vulnerabilities, Threats, and Controls

- Understanding Vulnerabilities, Threats, and Controls
  - Vulnerability = a weakness in a security system (i.e., in procedures, design, or implementation), that might be exploited to *cause loss or harm*.
  - Threat = circumstances that have a *potential* to cause harm

    -a potential violation of security
  - Controls = means and ways to block a threat, which tries to exploit one or more vulnerabilities.
- How do we address these problems?
  - We use a **control** as a protective measure.
  - That is, a control is an action, device, procedure, or technique that *removes or reduces a vulnerability*.

[Pfleeger & Pfleeger]

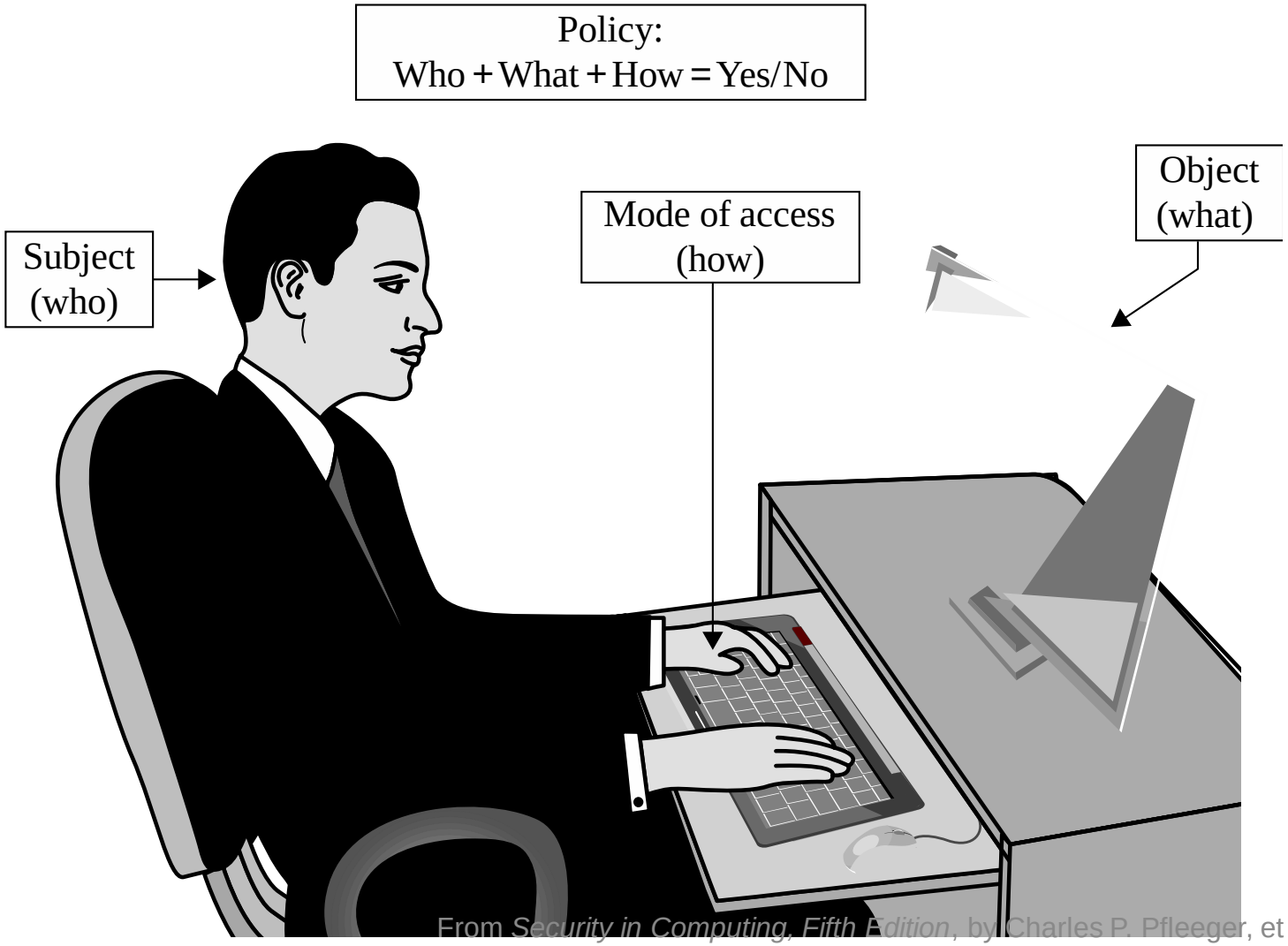
Cont…

- Relationship among threats, controls, and vulnerabilities:

  A threat is blocked by control of a vulnerability.

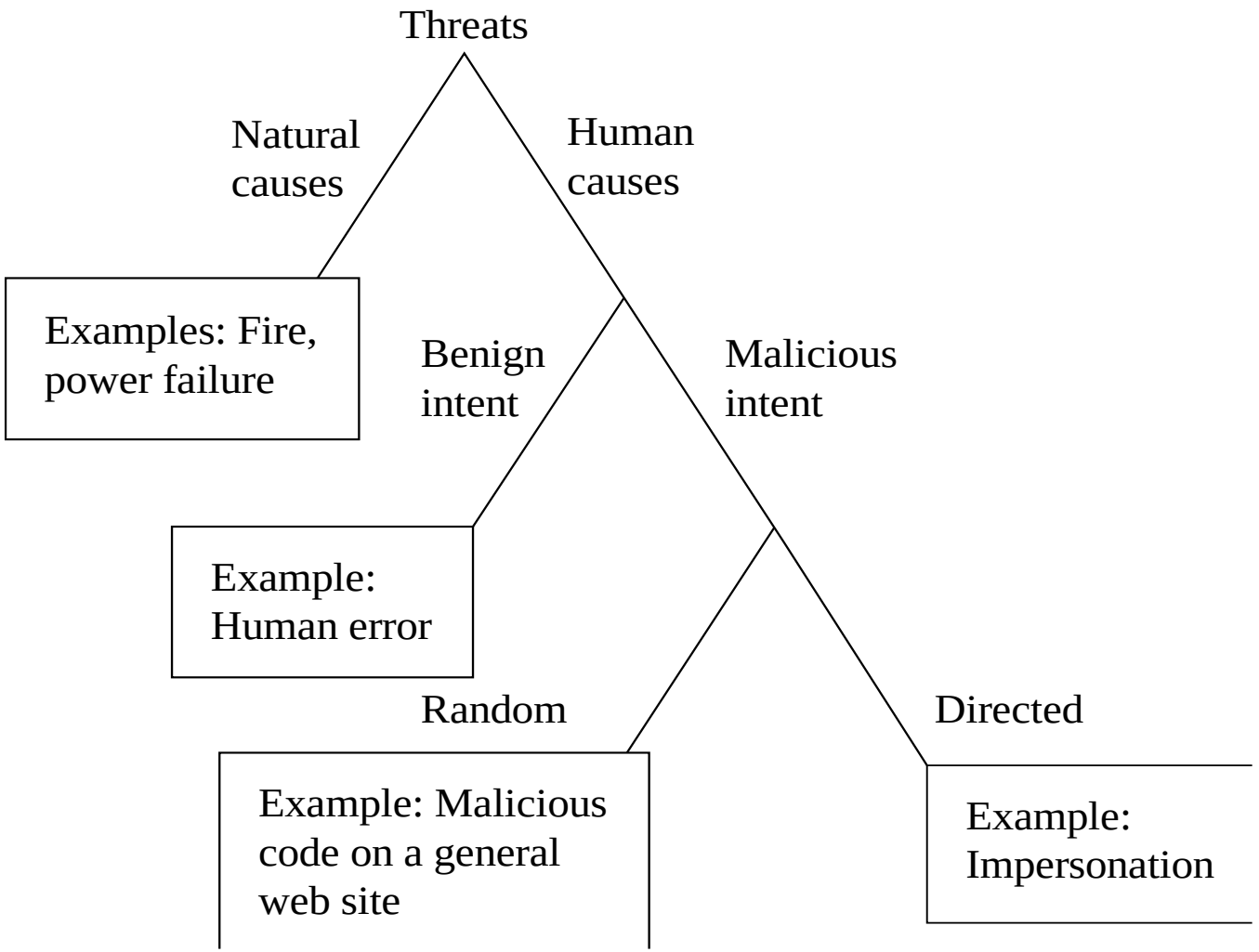
  To devise controls, we must *know as much about threats as possible*.

Visual explanation of basic access control terms



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Types of Threats



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

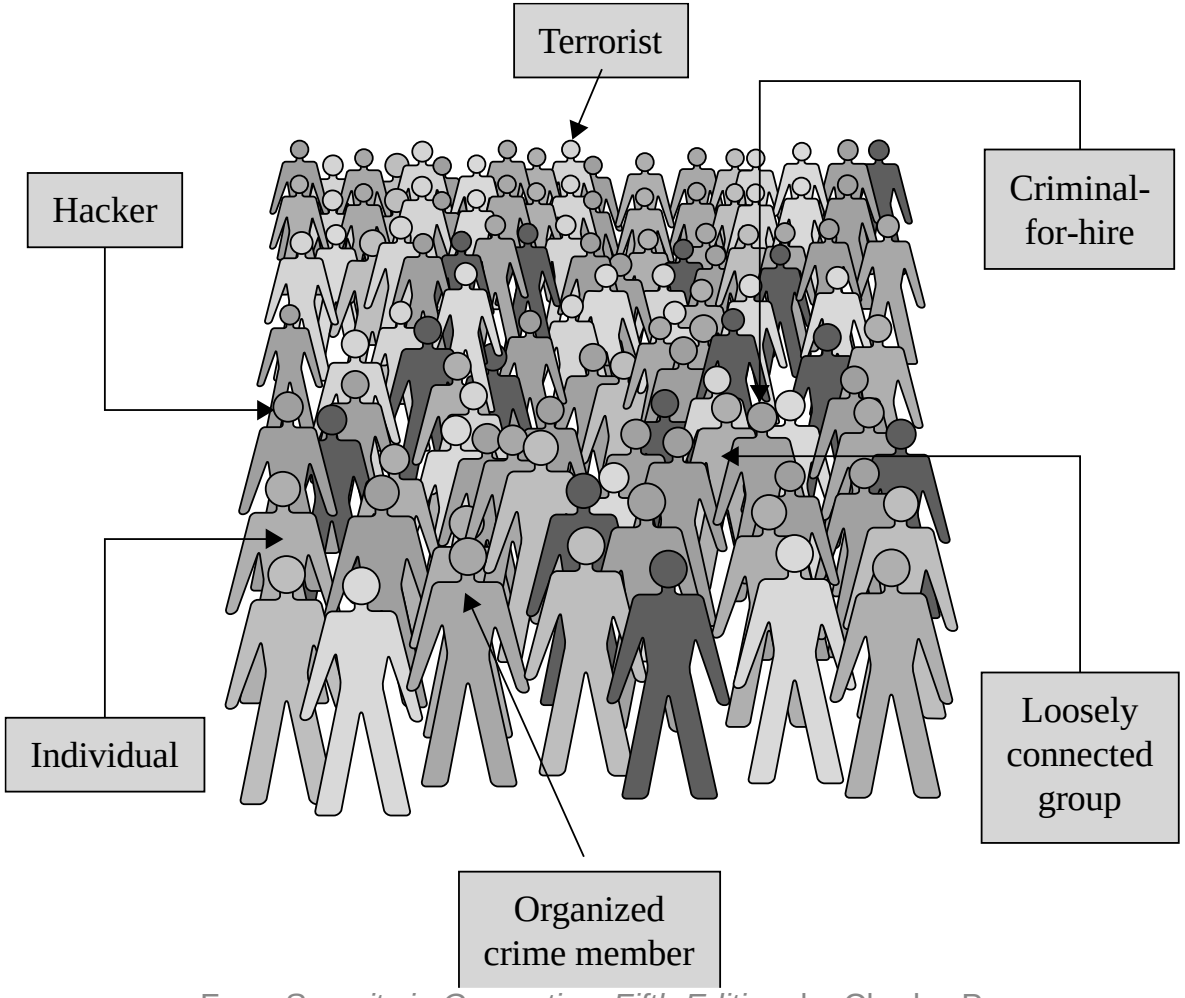
Advanced Persistent Threat (APT)

- APT is a special type of threat that has only been taken seriously by the broad security community over the past decade. In general, security experts believe that no one who becomes a high-priority target can truly be safe from APT.

- Types of APT
  - Organized
  - Directed
  - Well financed
  - Patient
  - Silent

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Types of Attackers



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

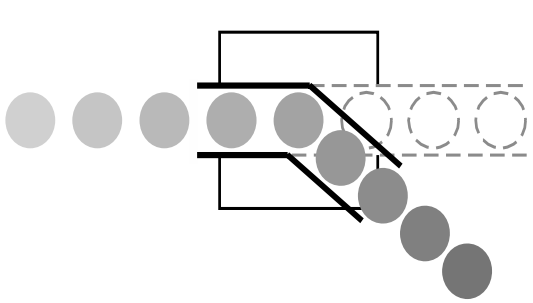
- Attack
  - = exploitation of one or more vulnerabilities by a threat; tries to defeat controls
    - Attack may be:
      - *Successful*
        - resulting in a breach of security, a system penetration, etc.
      - *Unsuccessful*
        - when controls block a threat trying to exploit a vulnerability

[Pfleeger & Pfleeger]

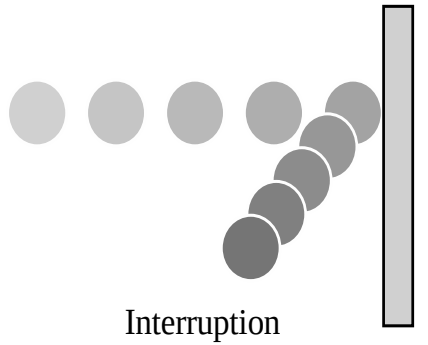
Kinds of Threats

- Kinds of threats:
  - Interception
    - an unauthorized party (human or not) gains access to an asset
  - Interruption
    - an asset becomes lost, unavailable, or unusable
  - Modification
    - an unauthorized party changes the state of an asset
  - Fabrication
    - an unauthorized party counterfeits(imitate something authentic, with the intent to steal, destroy, or replace the original,) an asset

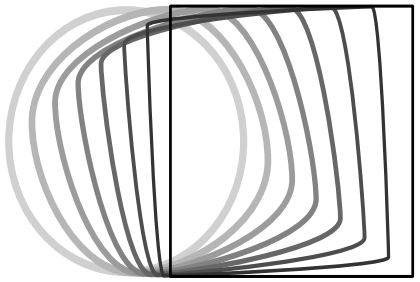
[Pfleeger & Pfleeger]



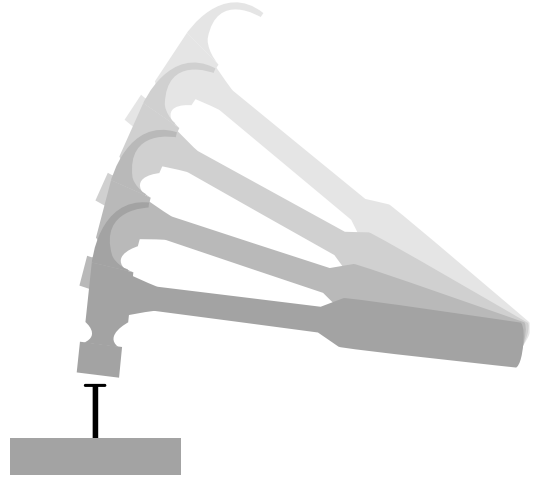
Interception



Interruption



Modification



Fabrication

Levels of Vulnerabilities / Threats

- D) for other assets (resources)
    - including. people using data, s/w, h/w
- C) for data
- B) for software
- A) for hardware

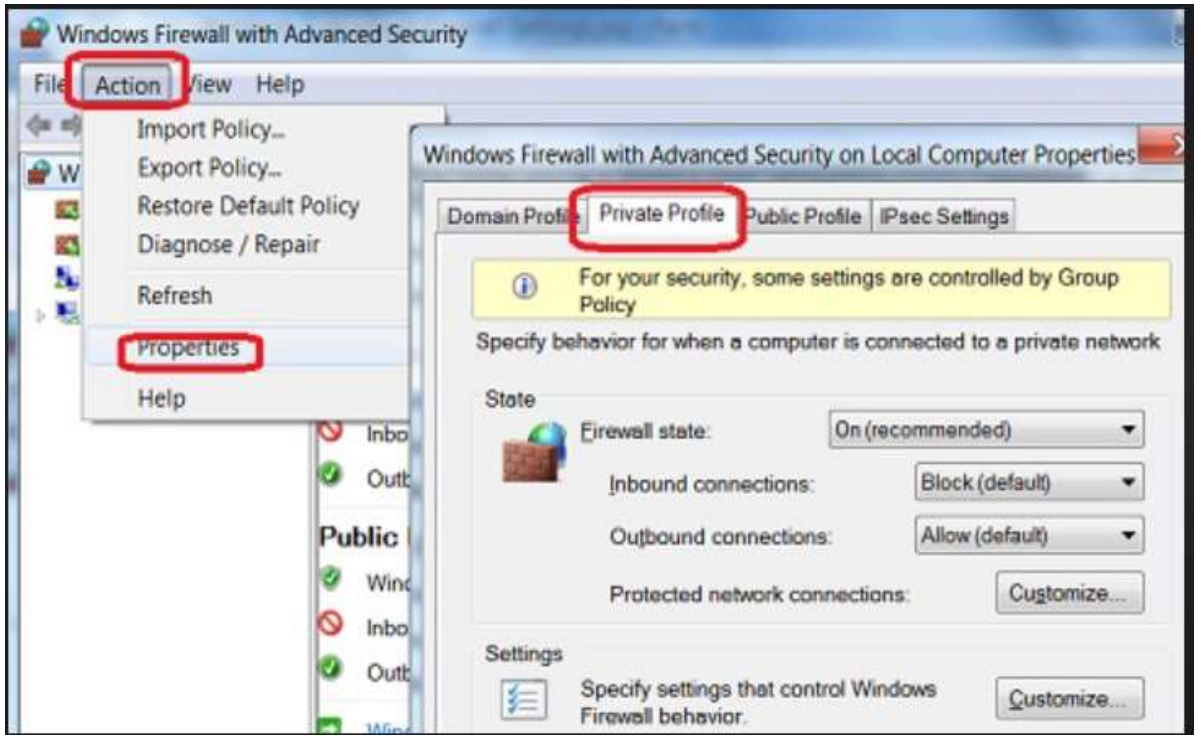
[Pfleeger & Pfleeger]

Topics to be covered

- Security policy and procedure
- Assumptions and Trust
- Security Assurance
- Implementation and Operational Issues
- Security Life Cycle

Security policy and Procedures

- A *security policy* is a statement of what is, and what is not, allowed.
- A *security procedure* is a method, tool, or mechanism for enforcing a security policy.
- Mechanisms can be nontechnical, such as requiring proof of identity before changing a password; in fact, policies often require some procedural mechanisms that technology cannot enforce.



Security policy and Procedures

- Policies may be presented mathematically, as a list of allowed (secure) and disallowed (non secure) states.

- In practice, policies are rarely so precise; they normally describe in English what users and staff are “allowed” to do or “disallowed” to do.

- security policy’s specification of “secure” and “nonsecure” actions, these security mechanisms can prevent the attack, detect the attack, or recover from the attack.

Security policy and Procedures

- Policy correctly divides world into secure and insecure states.
- Mechanisms prevent transition from secure to insecure states.
- The strategies may be used together or separately.
- *Prevention* means that an attack will fail. For example, if one attempts to break into a host over the Internet and that host is not connected to the Internet, the attack has been prevented.

Example

- 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 files.
- 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.
- In this example, Anna could easily have protected her files. In other environments, such protection may not be easy. For example, the Internet provides only the most rudimentary security mechanisms, which are not adequate to protect information sent over that network.

Assumptions and Trust

- How do we determine if the policy correctly describes the required level and type of security for the site?
- This question lies at the heart of all security, computer and otherwise.

- Assumption Example
- Opening a door lock requires a key. The assumption is that the lock is secure against lock picking. This assumption is treated as an axiom and is made because most people would require a key to open a door lock. A good lock picker, however, can open a lock without a key. Hence, in an environment with a skilled, untrustworthy lock picker, the assumption is wrong and the consequence invalid.

Assumptions and Trust

- Trust Example
- If the lock picker is trustworthy, the assumption is valid. The term “trustworthy” implies that the lock picker will not pick a lock unless the owner of the lock authorizes the lock picking. This is another example of the role of trust.

Assurance

- Evidence of how much to trust a system

- Evidence can include
  - System specifications
  - Design
  - Implementation

Assurance

EXAMPLE: In the United States, aspirin from a nationally known and reputable manufacturer, delivered to the drugstore in a safety-sealed container, and sold with the seal still in place, is considered trustworthy by most people. The bases for that trust are as follows.

1. The testing and certification of the drug (aspirin) by the Food and Drug Administration(FDA).
2. The manufacturing standards of the company and the precautions it takes to ensure that the drug is not contaminated.
3. The safety seal on the bottle.

- The three technologies (certification, manufacturing standards, and preventative sealing) provide some degree of assurance that the aspirin is not contaminated.
- The degree of trust the purchaser has in the purity of the aspirin is a result of these three processes.

Assurance

- Assurance in the computer world is similar. It requires specific steps to ensure that the computer will function properly.

- The sequence of steps includes detailed specifications of the desired (or undesirable) behavior; an analysis of the design of the hardware, software, and other components to show that the system will not violate the specifications; and arguments or proofs that the implementation, operating procedures, and maintenance procedures will produce the desired behavior.

Specification

- *A specification* is a (formal or informal) statement of the desired functioning of the system.

- The specification can be low-level, combining program code with logical and temporal relationships to specify ordering of events.

- EXAMPLE: A company is purchasing a new computer for internal use. They need to trust the system to be invulnerable to attack over the Internet. One of their (English) specifications would read “The system cannot be attacked over the Internet.”

- Specifications are used not merely in security but also in systems designed for safety.

Design

- The *design* of a system translates the specifications into components that will implement them.

- The design is said to *satisfy* the specifications if, under all relevant circumstances, the design will not permit the system to violate those specifications.

- EXAMPLE: A design of the computer system for the company mentioned above had no network interface cards, no modem cards, and no network drivers in the kernel. This design satisfied the specification because the system would not connect to the Internet. Hence it could not be attacked over the Internet.

Implementation

- Given a design, the *implementation* creates a system that satisfies that design. If the design also satisfies the specifications, then by transitivity the implementation will also satisfy the specifications.

- The difficulty at this step is the complexity of proving that a program correctly implements the design and, in turn, the specifications.

- A program is *correct* if its implementation performs as specified.

Security Assurance

- Furthermore, testing relies on test procedures and documentation, errors in either of which could invalidate the testing results.

- Although assurance techniques do not guarantee correctness or security, they provide a firm basis for assessing what one must trust in order to believe that a system is secure.

Security Assurance

- Trust cannot be quantified precisely.
- System specification, design, and implementation can provide a basis for determining “how much” to trust a system. This aspect of trust is called *assurance*.
- It is an attempt to provide a basis for bolstering (or substantiating

or specifying) how much one can trust a system.

Operational Issues

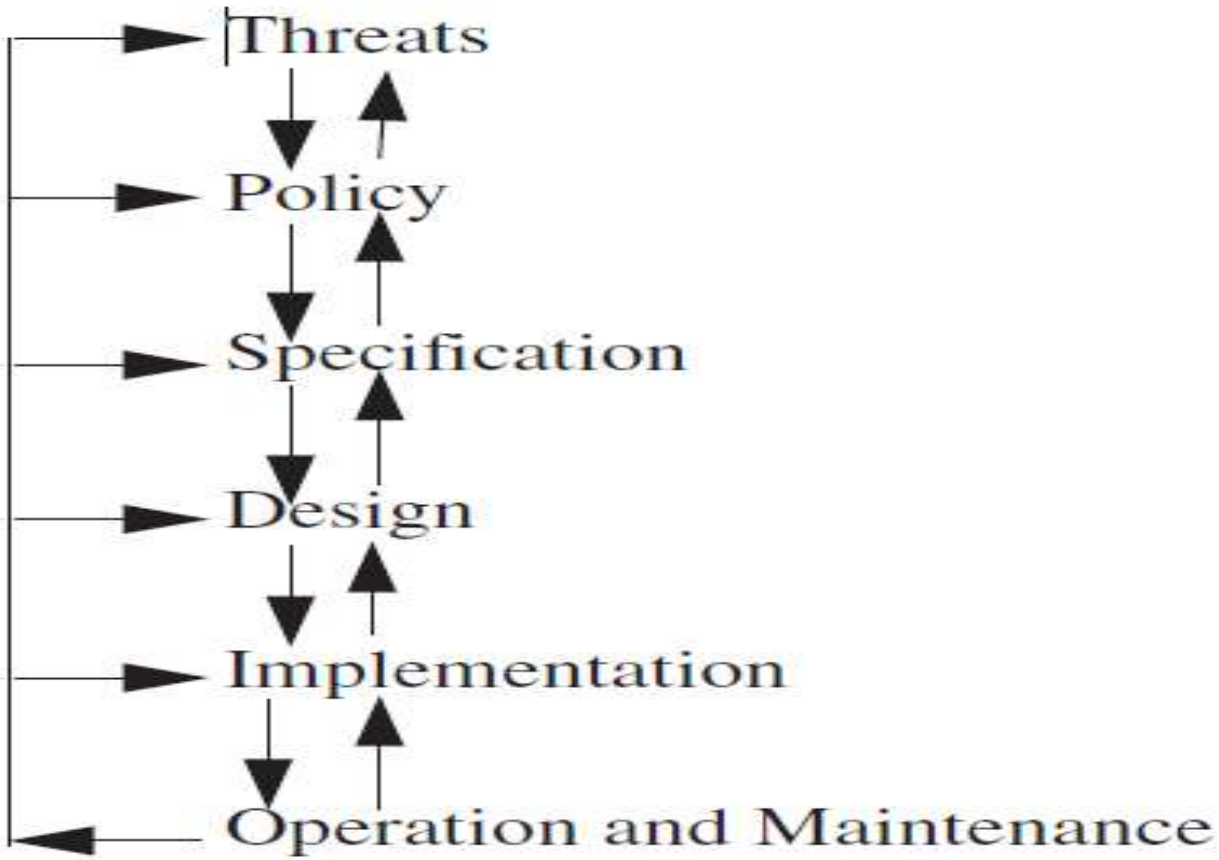
- Any useful policy and mechanism must balance the benefits of the protection against the cost of designing, implementing, and using the mechanism. This balance can be determined by analyzing the risks of a security breach and the likelihood of it occurring.
- Such an analysis is, to a degree, subjective, because in very few situations can risks be rigorously quantified.

Cost-Benefit Analysis

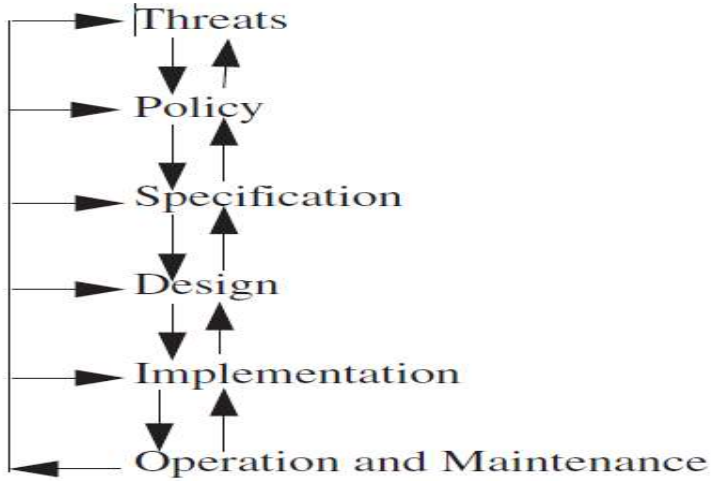
- If the data or resources cost less, or are of less value, than their protection, adding security mechanisms and procedures is not cost-effective because the data or resources can be reconstructed more cheaply than the protections themselves. Unfortunately, this is rarely the case.

- EXAMPLE: A database provides salary information to a second system that prints checks. If the data in the database is altered, the company could suffer grievous financial loss; hence, even a cursory cost-benefit analysis would show that the strongest possible integrity mechanisms should protect the data in the database.

The security life cycle



Cont…



- The operation and maintenance stage is critical to the life cycle.
- EXAMPLE: A major corporation decided to improve its security. It hired consultants, determined the threats, and created a policy.
- From the policy, the consultants derived several specifications that the security mechanisms had to meet.
- They then developed a design that would meet the specifications.
- During the implementation phase, the company discovered that employees could connect modems to the telephones without being detected.
- The design required all incoming connections to go through a firewall. The design had to be modified to divide systems into two classes: systems connected to "the outside," which were put outside the firewall; and all other systems, which were put behind the firewall. The design needed other modifications as well.
- When the system was deployed, the operation and maintenance phase revealed several unexpected threats.

Thank you