

Foundations of Computer Security

Andrea Pintér-Husztí

Contents

- 1 Basic computer security concepts
 - A Model for Computer Security
 - Security Design Principles
 - Threats, Attacks

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ▶ ↺ 🔍 ↻

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

Computer Security Terminology - Asset, Threat

System Resource(Asset) Users and owners wish to protect:
hardware, communication facilities and networks,
software, data.

Threat A potential for violation of security, a threat is a **possible danger** that might exploit a vulnerability.

Computer Security Terminology - Vulnerability

Vulnerability A **flaw or weakness** in a system's design, implementation, or operation and management that could be exploited to violate the system's security.

Categories:

- System resource can **be corrupted**: it does the wrong thing or gives wrong answers, pl. stored data values have been improperly modified
- System resource can **become leaky**: someone who should not have access to the information obtains such access
- System resource can **become unavailable** or very slow

Computer Security Terminology - Risk

Risk An expectation of loss expressed as the **probability** that a particular threat will exploit a particular vulnerability with a particular harmful result.

Goal: to minimize the risk

Computer Security Terminology - Attack

Attack An assault on system security that derives from an intelligent threat. **A threat that is carried out**, if successful, leads to a violation of security.

Based on the origin:

- ① **Inside** attack: The insider is authorized to access system resources but uses them in a way not approved by those who granted the authorization.
- ② **Outside** attack: An unauthorized or illegitimate user of the system attacks.

In practice most of the attacks are inside ones. In case of an outside attack mostly the adversary would like to get access to the system and carry out an inside attack.

Computer Security Terminology - Attack

Attack An assault on system security that derives from an intelligent threat. **A threat that is carried out**, if successful, leads to a violation of security.

Based on the way:

- ① **Active** attack: An attempt to alter system resources or affect their operation.
- ② **Passive** attack: An attempt to learn or make use of information from the system that does not affect system resources.

Computer Security Terminology - Countermeasure

Countermeasure An action, device, procedure, or technique that reduces a threat, a vulnerability.

Plane of application:

- ① Physical: doors, secure facilities, fire extinguishers, flood protection, and air conditioning
- ② Administrative: the organization's policies, procedures, and guidelines
- ③ Technical/algorithmic: firewalls, authentication systems, intrusion detection systems, and file encryption

Computer Security Terminology - Countermeasure

Countermeasure Functionality:

- Preventive controls: Preventive controls try to prevent security violations (physical, administrative, or technical) e.g. security updates, encryption
- Detective controls: When preventive controls have failed, detective controls are in place to detect security violations and alert the defenders. e.g. cryptographic checksums, audit trails and logs
- Corrective controls: Although a violation occurred, not all is lost, so it makes sense to try and fix the situation. (administrative, technical) e.g. backup

Countermeasures may cause new vulnerabilities.

Model

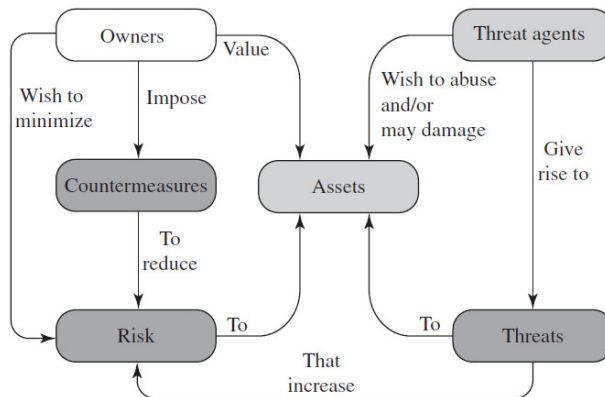


Figure 1.1 Security Concepts and Relationships

Security Design Principles

Security Design Principles - Least Privilege

- Do not give any more privileges than absolutely necessary to do the required job.
- The principle of least privilege is a preventive control, because it reduces the number of privileges that may be potentially abused and therefore limits the potential damage.
- Applicable in all information systems environments.
- Examples
 - ➊ Giving users only read access to shared files if that's what they need, and making sure write access is disabled.
 - ➋ Not allowing help desk staff to create or delete user accounts if all that they may have to do is to reset a password.
 - ➌ Not allowing software developers to move software from development servers to production servers.

Security Design Principles - Minimization

- The cousin of the least privilege principle and mostly applies to **system configuration**.
- Do not run any software, applications, or services that are not strictly required to do the entrusted job.
- Also improves performance, saves storage space, and is a good system administration practice in general.
- Example:
Computer whose only function is to serve as an e-mail server should have only e-mail server software installed and enabled. All other services and protocols should either be disabled or not installed at all to eliminate any possibility of compromise or misuse.

Security Design Principles - Defense in Depth

- Having more than one layer or type of defense
- The reasoning behind this principle is that any one layer or type of defense may be breached, no matter how strong and reliable you think it is, but two or more layers are much more difficult to breach.
- All types of controls should be used together: preventive, detective, corrective controls
- Example:
Works best when you combine two or more different types of defense mechanisms: such as using a firewall between the Internet and your LAN, plus the IP Security Architecture (IPSEC) to encrypt all sensitive traffic on the LAN. In this scenario, even if your firewall is compromised, the attackers still have to break IP Security to get to your data flowing across the LAN.

Security Design Principles - Open design

- A security mechanism should be open rather than secret.
- Algorithms can then be reviewed by many experts, and users can therefore have high confidence in them.
- 1883 Auguste Kerckhoffs' principle: "the enemy knows the system, *i.e.* one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them" (reformulated by Claude Shannon)
- Example:
Although encryption keys must be secret, encryption algorithms should be open to public

Security Design Principles - Compartmentalization

- The use of compartments, also known as zones, jails, sandboxes, and virtual areas
- Limits the damage and protects other compartments when software in one compartment is malfunctioning or compromised.
- Applications run in different compartments are isolated from each other.
- Example:
 - 1 The compromise of web server software, for example, does not take down or affect e-mail server software running on the same system but in a separate compartment.
 - 2 Solaris 10 Operating System: *zones*. A zone is a virtual operating system environment created within a single instance of the Solaris operating system. Zone partitioning technology can be used to create local zones that behave like virtual servers.

Security Design Principles - Keep Things Simple

- Complexity is the worst enemy of security.
- They are difficult to design, implement, test, and secure.
- When you have to choose between a complex system that does much and a simple system that does a bit less but enough, choose the simple one.

Security Design Principles - Psychological acceptability

- If security mechanisms hinder the usability or accessibility of resources, users may opt to turn off those mechanisms.
- Security mechanisms should be transparent to the users of the system
- Security procedures must reflect the user's mental model of protection.

Threats, Attacks

Threats and Assets

- Hardware** Major threat is the threat to availability. Threats include accidental and deliberate damage to equipment as well as theft. Theft of pendrives, tablets, DVDs can lead to loss of confidentiality. Physical and administrative security measures are needed.
- Software** Availability: application software, is often easy to delete, can also be altered or damaged to render it useless, technical control: Backup
Integrity: software modification, computer viruses
Software piracy: the problem of unauthorized copying of software has not been solved.

Threats and Assets

- Data Storage, transmitting, processing
 - Confidentiality: unauthorized reading of data files, databases
 - Integrity: altering data
 - Availability: deleting data files accidentally or maliciously

Attack Trees

An attack tree is a hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities.

- The goal of the attack is represented as the *root node*.
- *Subnodes* define subgoals.
- *Leaf nodes* represent different ways to initiate an attack.

Bank Account Compromise - Internet banking authentication application

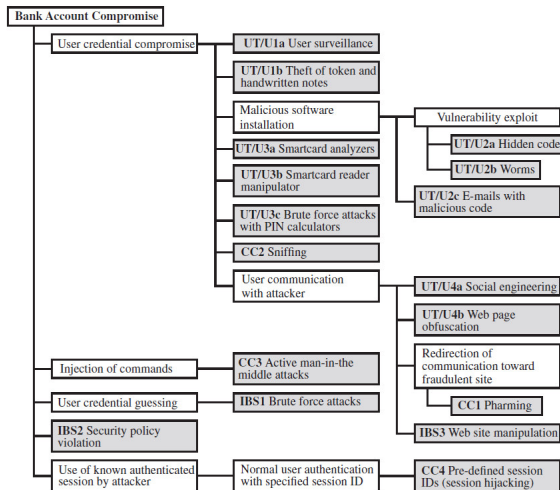


Figure 1.4 An Attack Tree for Internet Banking Authentication

Bank Account Compromise - Internet banking authentication application

- User terminal and user (UT/U): These attacks target the user equipment, including the tokens that may be involved, such as smartcards or other password generators, as well as the actions of the user.
e.g. *worm*: Malware computer program that replicates itself in order to spread to other computers.
- Communications channel (CC): This type of attack focuses on communication links.
e.g. *sniffing*: obtain credential information via the communication channel
- Internet banking server (IBS): These types of attacks are offline attacks against the servers that host the Internet banking application.

Bank Account Compromise - Internet banking authentication application

- Injection of commands: The attacker is able to intercept communication between the UT and the IBS. Various schemes can be used to be able to impersonate the valid user and so gain access to the banking system.
- User credential guessing: Brute force attacks against some banking authentication schemes are feasible by sending random usernames and passwords. The attack mechanism is based on distributed zombie personal computers, hosting automated programs for username- or password-based calculation. *Zombie* is a computer connected to the Internet that has been compromised and can be used to perform malicious tasks under remote direction.

Bank Account Compromise - Internet banking authentication application

- *Pharming*: A cyber attack intended to redirect a website's traffic to another, fake site.
- *Session hijacking* is the exploitation of a valid computer session — sometimes also called a session key — to gain unauthorized access to information or services.
- *Social engineering*: Psychological manipulation of people into performing actions or divulging confidential information.