



Riferimenti

- R.Anderson, **Security Engineering: a guide to building dependable distributed systems** , 2nd ed., John Wiley and Sons 2008
- C.Pfleeger, S.L.Pfleeger, **Sicurezza in Informatica**, 2nd ed., Pearson 2008
- W.Stallings, **Cryptography and Network Security: Principles and Practices**, 5th ed., Prentice Hall 2011
- J.Viega, G.McGraw, **Building Secure Software**, Addison-Wesley 2002
- M.Bishop, **Computer Security: Art and Science**, Addison-Wesley 2003

1



Security Breaches - Terminology

- Assets – h/w, s/w, data
- Vulnerability
 - a weakness in the system that can be exploited
- Threat
 - Potentiality for loss or harm
 - Human attacks, natural disasters, errors
- Attack
 - Realization of a threat
- Control
 - a protective measure/action/procedure to remove or reduce a vulnerability

2



Types of Security Breaches

- Interruption
 - Example: DOS (Denial of Service)
- Interception/Disclosure
 - Peeping eyes
- Modification
 - Change of existing data
- Fabrication
 - Addition of false or spurious data

3



Computing System Vulnerabilities

- Hardware vulnerabilities
- Software vulnerabilities
- Data vulnerabilities
- Human vulnerabilities ?

4



Hardware Vulnerabilities

- Destroyed hardware
- Stolen hardware
- Substituted hardware
- Altered (but still functioning) hardware
 - Keystroke
 - Information leaks

5



Software Vulnerabilities

- Destroyed (deleted) software
- Stolen (pirated) software
- Altered (but still running) software
 - Logic bomb
 - Trojan horse
 - Virus
 - Trapdoor
 - Information leaks

6



Data Vulnerability

- Destroyed (deleted) data
- Stolen (pirated) data
- Altered (but still usable) data
- Fabricated (false) data

- **Principle of adequate protection**


7



Other Exposed Assets

- Storage media (often included in h/w)
 - backup
- Networks
 - usual problems enhanced
- Access
 - leads to three types of vulnerabilities
- Key people
 - whom can you trust?

8



What is "Security"?

To decide whether a computer system is "secure", you must first decide what "secure" *means* and which threats are a concern.

9



Security is ...

- **Confidentiality:**
 - Assets of your system are accessible only by authorized parties (a.k.a., secrecy/privacy).
 - Prevent / detect / deter the improper disclosure of information or of resources

10



Security is ...

■ **Integrity:**

- Assets can be modified only by authorized parties or only in authorized ways.
- Data Integrity and Origin Integrity
- Tied to *trustworthiness*
- Prevent / detect / deter improper modification of information (precise, unmodified, consistent, modified by authorized process only and only in an acceptable way, ...)

11



Security is ...

■ **Availability:**

- Assets are accessible in expected ways to authorized parties (a.k.a., denial of service).
- Prevent / detect / deter improper denial of access to services provided by the system of information
 - Timely response
 - Fair allocation

12



Security is ...

- **Accountability:**

- Ability to map between action in a system and responsibility for the action.
(note - emerging and not universally accepted as a distinct issue)
- Prevent / detect / deter improper use of the resources (Hardware, Software, Data, Network)
- (a.k.a. usage)

13



Commercial Example

- **Secrecy** (Confidentiality)- An employee should not come to know the salary of his/her manager
- **Integrity** - An employee should not be able to modify his/her own salary
- **Availability** - Paychecks should be printed on time as stipulated by law
- **Usage** – An employee personal information should not be used for illegal purposes

14



Achieving Security

- Policy
 - Different “languages”: natural, math., ad-hoc
 - What is and is not allowed
- Mechanism
 - technical or procedural
 - How to enforce the policy
- Assurance
 - How well the system meets its requirements

15



Trust and Assumptions

- Underlie *all* aspects of security
- Policies
 - Unambiguously partition system states
 - Correctly capture security requirements
- Mechanisms
 - Assumed to enforce policy
 - Support mechanisms work correctly

16



Assurance

- Specification
 - Requirements analysis
 - Statement of desired functionality
- Design
 - How system will meet specification
- Implementation
 - Programs/systems that carry out design

17



Security strategies/goals

- **Prevention:** take measures that prevent violation of security policy (your assets from being damaged)
- **Detection:** take measures so that you can detect when, how, and by whom a violation of security policy takes place (an asset has been damaged)
- **Reaction:** take measures to stop the attack, to recover your assets or to recover from a damage to your assets. Keep on ticking ...

18



Example 1 – Private Property

- **Prevention:** locks at doors, window bars, walls round the property
- **Detection:** stolen items are missing, burglar alarms, closed circuit TV
- **Reaction:** call the police, replace stolen items, make an insurance claim ...
- **Footnote:** Parallels to the physical world can illustrate aspects of computer security but they can also be misleading.

19



Example 2 – E-Commerce

- **Prevention:** encrypt your orders, rely on the merchant to perform checks on the caller before accepting order, do not use the Internet (?) ...
- **Detection:** an unauthorized transaction appears on your credit card statement
- **Reaction:** complain, ask for a new card number, etc.
- **Footnote:** Your credit card number has not been stolen. Your card can be stolen, but not the number.

20



Security Mechanism

- Prevention is more fundamental
 - Detection seeks to prevent by threat of punitive action
 - Detection requires that the audit trail be protected from alteration
- Sometime detection is the only option
 - Accountability in proper use of authorized privileges
 - Modification of messages in a network

21



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?

22



Security by Obscurity

- **Security by obscurity** says that if we hide the inner workings of a system, then the system it will be secure

- Less and less applicable due to
 - widespread vendor-independent open standards
 - widespread computer knowledge and expertise
 - widespread transfer of information

23



Security by Legislation

- **Security by legislation** says that if we instruct our users on how to behave, then we can secure our systems:
 - Users should not share passwords
 - Users should not write down passwords
 - Users should not type in their password when someone is looking over their shoulder
- User awareness and cooperation is important, but cannot be the principal focus for achieving security

24



Methods of Defense

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

25



Encryption

- Encryption is likely the most powerful tool available - but does not solve all problems.
- Application domains include
 - Data Transmission
 - Digital Signatures
 - Data Base Security techniques
 - Electronic Commerce/Internet/Intranet

26



Software Controls

- Software controls include:
 - Internal Program Controls
 - Operating System Controls
 - Independent Control Programs
 - Development Control

Software controls are usually the 1st aspects of computer security that come to mind.

27



Policies and Procedures

- Policy controls can be simple but effective
 - Example: frequent changes of passwords
- Legal and ethical controls
 - Gradually evolving and maturing

28



Other forms of Control

- Hardware Controls
 - smartcards,
 - locks,
 - devices to verify identities,
 - boards to control access to disks
- Physical Controls (lock on doors, backups)

29



Effectiveness of Control

- Controls must be used to be effective:
 - Must be efficient (time, memory space, human activity)
 - Must be easy to use
 - Must be appropriate
- Overlapping Controls (layered defense)
 - several controls for single vulnerability
- Problem awareness
- Periodic reviews

30



Access Control

- One of the main aspects of security is controlling access to information
- Controlling access to **information** may be elusive and need to be replaced by controlling access to **data**
- The distinction between data and information can be subtle but causes some of the more difficult problems in computer security

31



Data vs. Information

- **Data** are physical phenomena chosen by convention to represent certain aspects of our conceptual and real world. The meanings we assign to data are called **information**.
- **Covert channels**: response time or memory usage may signal information (more later)
- **Inference in statistical databases**: combine statistical queries to get information on individual entries (more in the database security course)

32

1st Fundamental Design Decision

Where to focus security controls

The focus may be on **data – operations – users**;
e.g. integrity requirements may refer to rules on

- Format and content of **data items** (internal consistency): account balance is an integer
- **Operations** that may be performed on a data item: credit, debit, transfer, ...
- **Users** who are allowed to access a data item (authorised access): account holder and bank clerk have access to account

33

2nd Fundamental Design Decision

Where to place security controls

applications
services (middleware)
operating system
OS kernel
hardware

34



The Man-Machine Scale

- Visualize security mechanisms as concentric **protection rings**, with hardware mechanisms in the centre (more generic) and application mechanisms at the outside (more specific)
- The **man-machine scale** for security mechanisms combines our first two design decisions
 - Person-oriented: specific, complex, focus on users
 - Machine-oriented: generic, simple, focus on data

35



3rd Fundamental Design Decision Complexity or Assurance

- Often, the location of a security mechanism is related to its complexity. Generic mechanisms are **simple**, applications seek **feature-rich** security functions.
- *Do you prefer simplicity - and higher assurance - to a feature-rich security environment?*
- Dilemma: simple generic mechanisms may not match specific security requirements. To choose the right features from a rich menu, you have to be a security expert. Security unaware users are in a no-win situation.

36



Example: Security Evaluation

- Security evaluation checks whether products deliver the security services promised. State the
 - **function** of the security system
 - required degree of **assurance (trust)** in its security
- To achieve high assurance, the security system has to be examined exhaustively and in close detail.
- Obvious trade-off between complexity and assurance. The higher the assurance level wanted, the simpler your system ought to be.
- *Feature-rich security and high assurance do not match easily*

37



4th Fundamental Design Decision

centralize or decentralize control

- If single entity in charge of security, then easy to achieve uniformity but central entity may become a performance bottleneck.
- A distributed solution may be more efficient but added care to guarantee that different components enforce a consistent policy.
- *Should the tasks of defining and enforcing security be given to a central entity or should they be left to individual components in a system?*

38

5th Fundamental Design Decision

Blocking access to layer below

- Attackers try to bypass protection mechanisms. Every protection mechanism defines a **security perimeter (boundary)**. The parts of the system that can disable the mechanism lie within the perimeter, the parts of the system that can malfunction without compromising the mechanism lie outside.
- There is an immediate and important corollary to the second design decision: *How do you stop an attacker from getting access to a layer below your protection mechanism?*

39

The Layer Below - Examples

- **Recovery tools**, like Norton Utilities, restore the data by reading memory directly and then restoring the file structure. Such a tool can be used to circumvent logical access control as it does not care for the logical memory structure
- **Unix** treats I/O devices and physical memory devices like files. If access permissions are defined poorly, e.g. if read access is given to a disk containing read protected files, then an attacker can read the disk contents and reconstruct the files.

40



More examples

- **Object reuse:** in a single processor system, when a new process becomes active, it obtains access to memory positions used by the previous process. You have to avoid **storage residues**, i.e. data left behind in the memory area allocated to the new process.
- **Backup:** whoever has access to a backup tape has access to all the data on it. Logical access control is of no help and backup tapes have to be locked away safely to protect the data.

41



Key Points

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

42