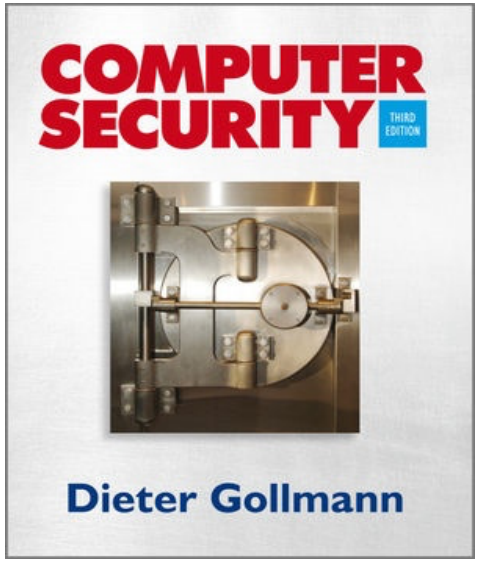
502049 – Introduction to Information Security

Chapter 2: Foundations of Computer Security



Ngoc Tu Huynh, PhD
huynhngoctu@tdtu.edu.vn

Agenda

- Security strategies
  - Prevention – detection – reaction
- Security objectives
  - Confidentiality – integrity – availability
  - Accountability – non-repudiation
- Fundamental Dilemma of Computer Security
- Principles of Computer Security
- The layer below

Security Strategies

- Prevention: take measures that prevent your assets from being damaged.
- Detection: take measures so that you can detect when, how, and by whom an asset has been damaged.
- Reaction: take measures so that you can recover your assets or to recover from a damage to your assets.
- The more you invest into prevention, the more you have to invest into detection to make sure prevention is working.

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.

Example 2 – E-Commerce

- Prevention: encrypt your orders, rely on the merchant to perform checks on the caller, don't 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.

Security Objectives

- Confidentiality: prevent unauthorised disclosure of information
- Integrity: prevent unauthorised modification of information
- Availability: prevent unauthorised withholding of information or resources
- Authenticity: “know whom you are talking to”
- Accountability (non-repudiation): prove that an entity was involved in some event

Confidentiality

- Prevent unauthorised disclosure of information (prevent unauthorised reading).
- Secrecy: protection of date belonging to an organisation.
- Historically, security and secrecy were closely related; security and confidentiality are sometimes used as synonyms.
- Do we want to hide the content of a document or its existence?
  - Traffic analysis in network security.
  - Anonymity, unlinkability

Privacy

- Privacy: protection of personal data (OECD Privacy Guidelines, EU Data Privacy Directive 95/46/EC).
- "Put the user in control of their personal data and of information about their activities."
- Taken now more seriously by companies that want to be 'trusted' by their customers.
- Also: the right to be left alone, e.g. not to be bothered by spam.

Integrity

- Prevent unauthorised modification of information (prevent unauthorised writing).
- Data Integrity - The state that exists when computerized data is the same as that in the source document and has not been exposed to accidental or malicious alteration or destruction. (Integrity synonymous for external consistency.)
- Detection (and correction) of intentional and accidental modifications of transmitted data.

Integrity ctd.

- Clark & Wilson: no user of the system, even if authorized, may be permitted to modify data items in such a way that assets or accounting records of the company are lost or corrupted.
- In the most general sense: make sure that everything is as it is supposed to be.

  (This is highly desirable but cannot be guaranteed by mechanisms internal to the computer system.)
- Integrity is a prerequisite for many other security services; operating systems security has a lot to do with integrity.

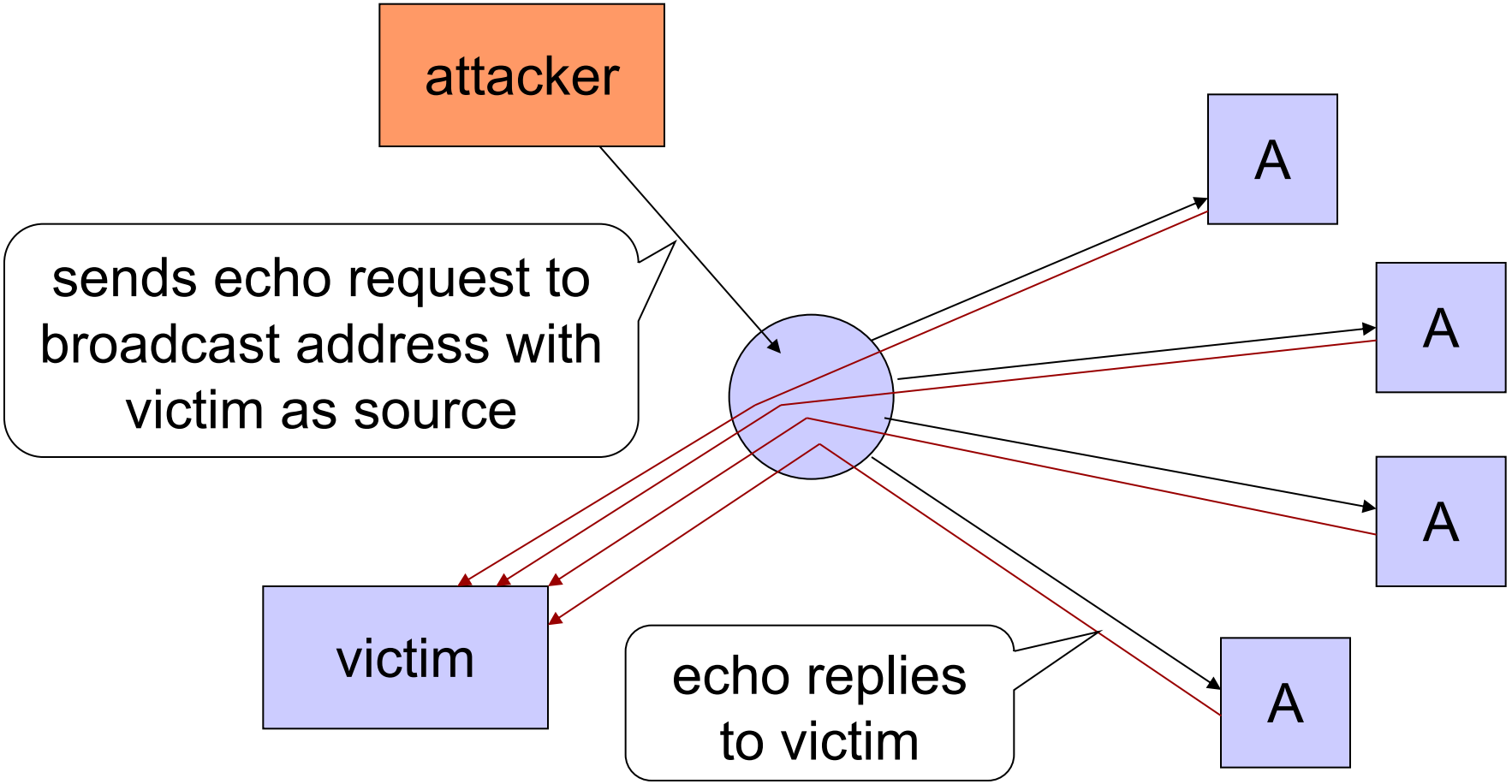
Availability

- The property of being accessible and usable upon demand by an authorised entity.
- Denial of Service (DoS): prevention of authorised access of resources or the delaying of time-critical operations.
- Maybe the most important aspect of computer security, but few methods are around.
- Distributed denial of service (DDoS) receives a lot of attention; systems are now designed to be more resilient against these attacks.

Denial of Service Attack (smurf)

- Attacker sends ICMP echo requests to a broadcast address, with the victim's address as the spoofed sender address.
- The echo request is distributed to all nodes in the range of the broadcast address.
- Each node replies with an echo to the victim.
- The victim is flooded with many incoming messages.
- Note the amplification: the attacker sends one message, the victim receives many.

Denial of Service Attack (smurf)



Accountability

- At the operating system level, audit logs record security relevant events and the user identities associated with these events.
- If an actual link between a user and a “user identity” can be established, the user can be held accountable.
- In distributed systems, cryptographic non-repudiation mechanisms can be used to achieve the same goal.

Non-repudiation

- Non-repudiation services provide unforgeable evidence that a specific action occurred.
- Non-repudiation of origin: protects against a sender of data denying that data was sent.
- Non-repudiation of delivery: protects against a receiver of data denying that data was received.
- Danger – imprecise language: has mail been received when it is delivered to your mailbox?

Non-repudiation

- 'Bad' but frequently found definition: Non-repudiation provides irrefutable evidence about some event.
- Danger – imprecise language: is there anything like irrefutable evidence?
- Non-repudiation services generate mathematical evidence.
- To claim that such evidence will be "accepted by any court" is naïve and shows a wrong view of the world.

Non-repudiation

- Typical application: signing emails; signatures in S/MIME secure e-mail system.
- Are such signatures analogous to signing a letter by hand?
- In the legal system, hand written signatures (on contracts) indicate the intent of the signer.
- Can a digital signature created by a machine, and maybe automatically attached to each mail, indicate the intent of a person?

Reliability & Safety

- Reliability and safety are related to security:
  - Similar engineering methods,
  - Similar efforts in standardisation,
  - Possible requirement conflicts.
- Reliability addresses the consequences of accidental errors.
- Is security part of reliability or vice versa?
- Safety: measure of the absence of catastrophic influences on the environment, in particular on human life.

Security & Reliability

- On a PC, you are in control of the software components sending inputs to each other.
- On the Internet, hostile parties provide input.
- To make software more reliable, it is tested against typical usage patterns:
  - "It does not matter how many bugs there are, it matters how often they are triggered."
- To make software more secure, it has to be tested against 'untypical' usage patterns (but there are typical attack patterns).

Dependability

- Proposal for a term that encompasses reliability, safety, and security
- Dependability (IFIP WG 10.4):

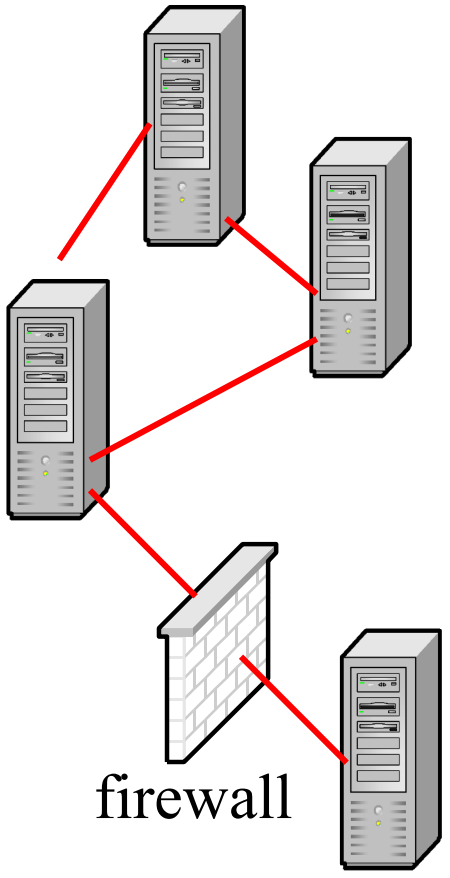
  The property of a computer system such that reliance can justifiably be placed on the service it delivers. The service delivered by a system is its behaviour as it is perceived by its user(s); a user is another system (physical, human) which interacts with the former.

A Remark on Terminology

- There is no single definition of security.
- When reading a document, be careful not to confuse your own notion of security with that used in the document.
- A lot of time is being spent – and wasted – trying to define an unambiguous notation for security.
- Our attempt at a working definition of security:
  - Computer security deals with the prevention an detection of unauthorized actions by users of a computer system.
  - Computer security is concerned with the measures we can take to deal with intentional actions by parties behaving in an unwelcome fashion.

Distributed System Security

- Distributed systems: computers connected by networks
- Communications (network) security: addresses security of the communications links
- Computer security: addresses security of the end systems; today, this is the difficult part.
- Application security: relies on both to provide services securely to end users.



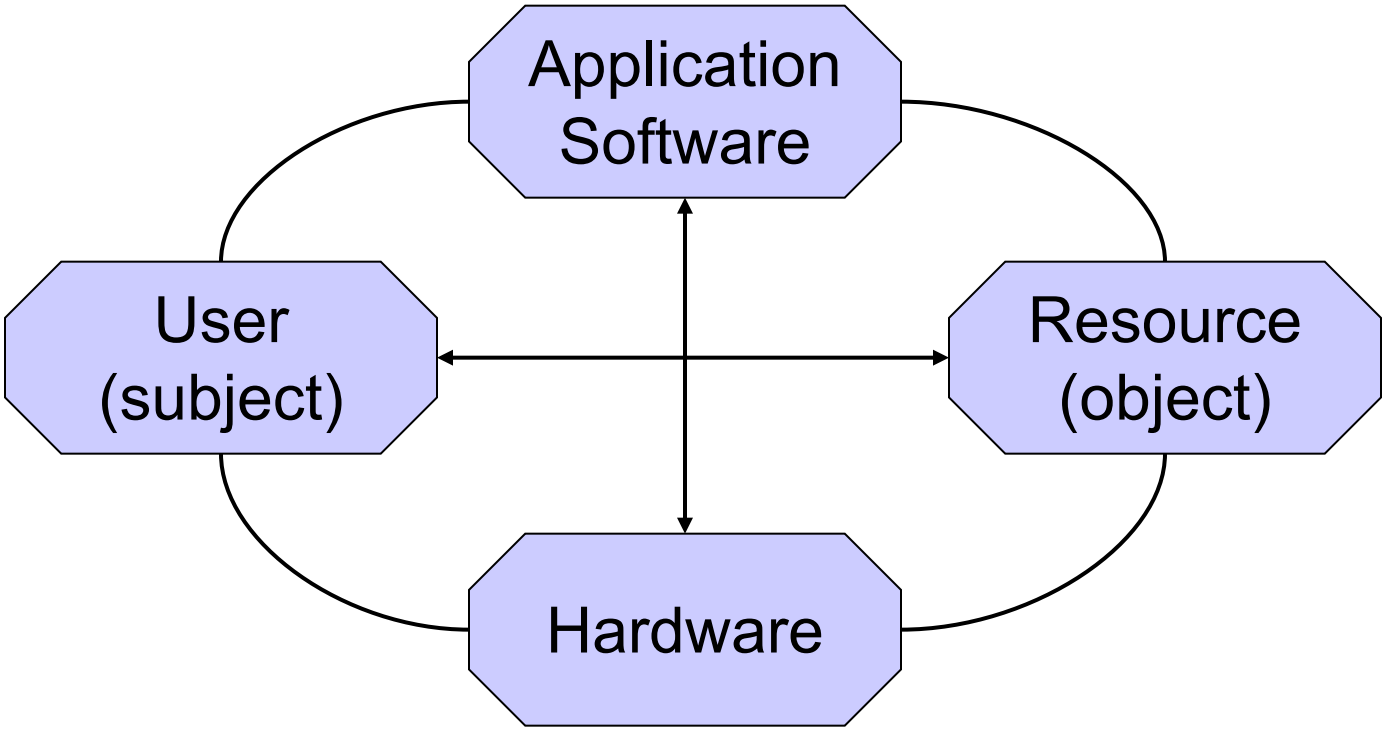
Fundamental Dilemma of Computer Security

Security unaware users have specific security requirements but no security expertise.

- If you provide your customers with a standard solution it might not meet their requirements.
- If you want to tailor your solution to your customers' needs, they may be unable to tell you what they require.

Principles of Computer Security

Dimensions of Computer Security



1st Fundamental Design Decision

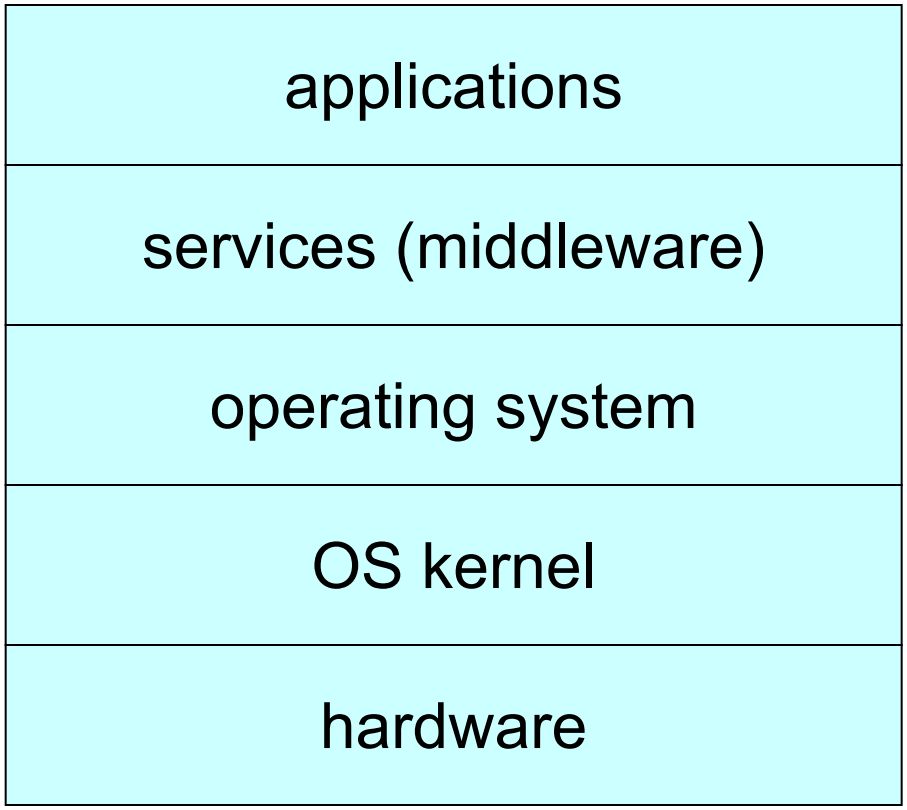
Where to focus security controls?

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.

2nd Fundamental Design Decision

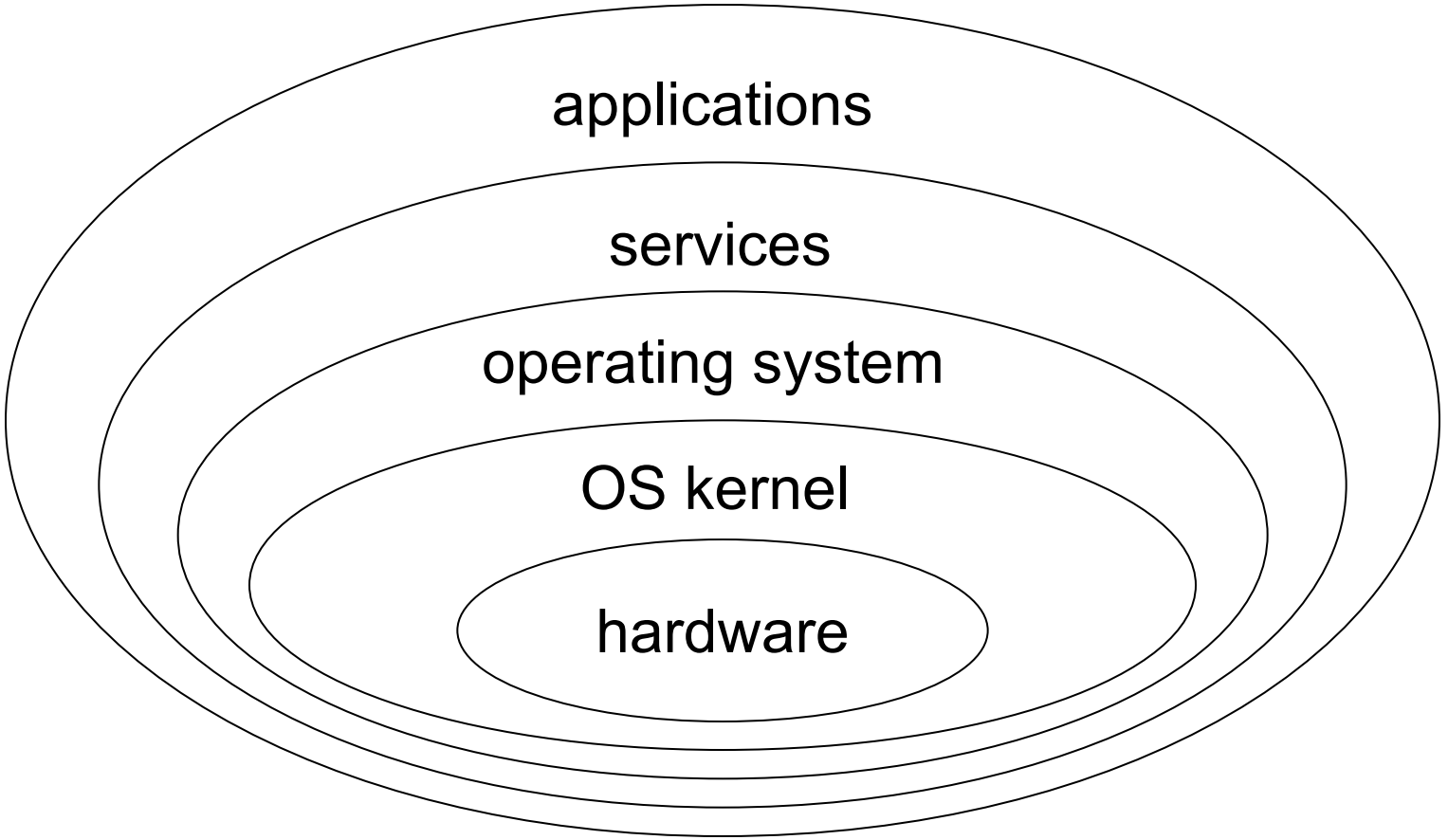
Where to place security controls?



Man-Machine Scale

- Visualize security mechanisms as concentric protection rings, with hardware mechanisms in the centre and application mechanisms at the outside.
- Mechanisms towards the centre tend to be more generic while mechanisms at the outside are more likely to address individual user requirements.
- The man-machine scale for security mechanisms combines our first two design decisions.

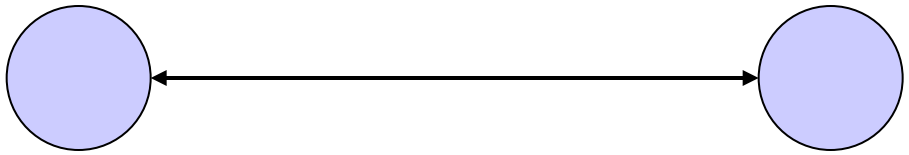
Onion Model of Protection



Man-Machine Scale

specific
complex
focus on users

generic
simple
focus on data



man
oriented

machine
oriented

Data & 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.
- Information and data lie on the two ends of the man-machine scale.
- The distinction between data and information can be subtle but causes some of the more difficult problems in computer security.

Data & Information

- Controlling access to information may be elusive and need to be replaced by controlling access to data.
- If information and corresponding data are closely linked the two approaches give very similar results, but this is not always the case.
- Covert channels: response time or memory usage may signal information.
- Inference in statistical databases: combine statistical queries to get information on individual entries.

3rd Fundamental Design Decision
Complexity or Assurance?

- Often, the location of a security mechanism on the man-machine scale is related to its complexity.
- Generic mechanisms are simple, applications clamour for feature-rich security functions.
- Do you prefer simplicity – and higher assurance – to a feature-rich security environment?

3rd Fundamental Design Decision
Complexity or Assurance?

- Fundamental 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.
- Feature-rich security and high assurance do not match easily.

4th Fundamental Design Decision

Centralized or decentralized control?

- Within the domain of a security policy, the same controls should be enforced.
- Having a single entity in charge of security makes it easy to achieve uniformity but this central entity may become a performance bottleneck.
- A distributed solution may be more efficient but you have to take added care to guarantee that different components enforce a consistent policy.
- Should a central entity define and enforce security or should these tasks be left to individual components in a system?

Security Perimeter

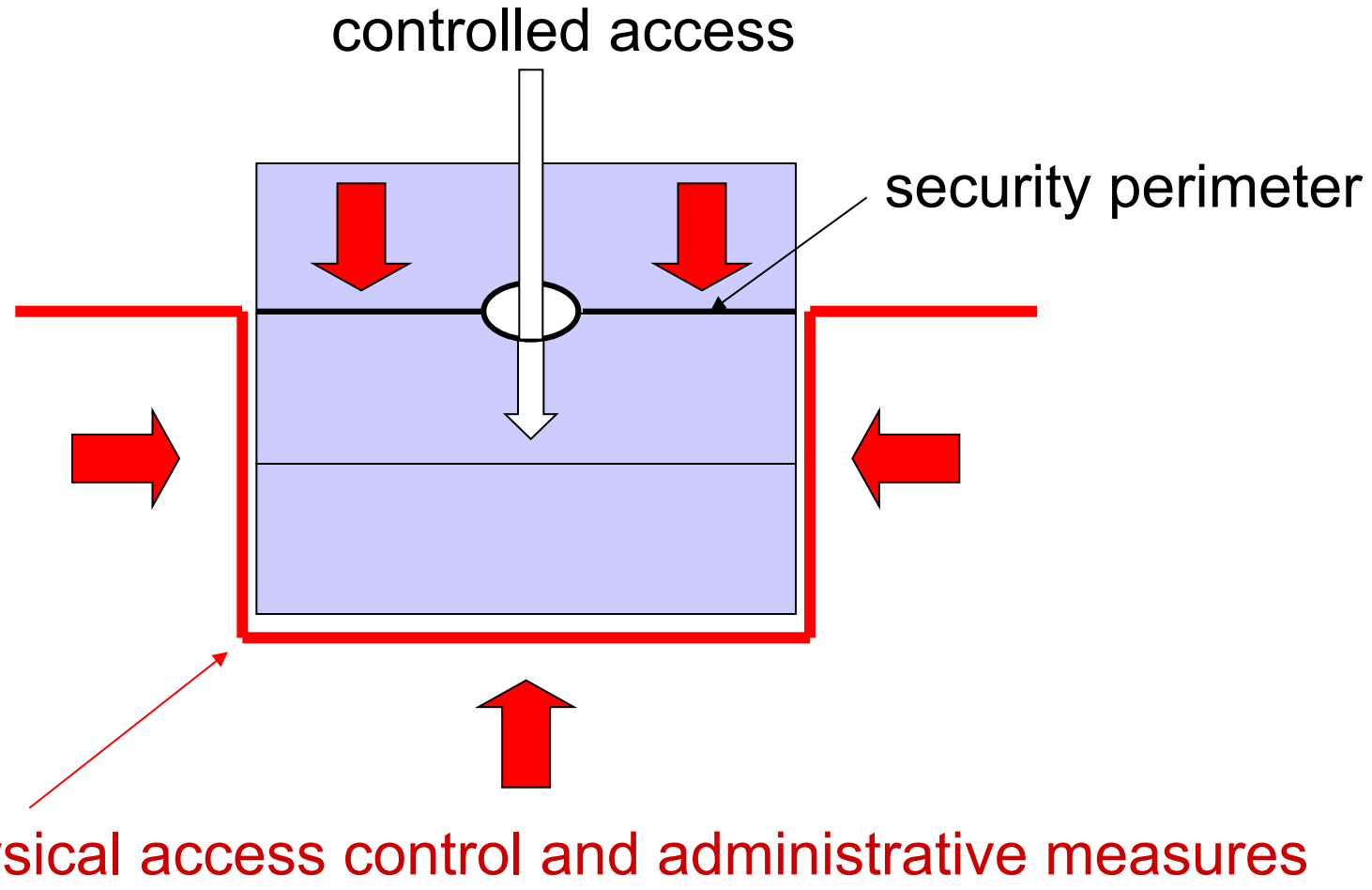
- Every protection mechanism defines a security perimeter (security boundary).
- The parts of the system that can malfunction without compromising the mechanism lie outside the perimeter.
- The parts of the system that can disable the mechanism lie within the perimeter.
- Note: Attacks from insiders are a major concern in security considerations.

5th Fundamental Design Decision

Blocking access to the layer below

- Attackers try to bypass protection mechanisms.
- 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?

Access to the Layer Below



The Layer Below – Examples

- Recovery tools restore 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; with badly defined access permissions, e.g. if read access is given to a disk, an attacker can read the disk contents and reconstruct read protected files.
- Buffer overruns: a value assigned to a variable is too large for the memory buffer allocated to that variable; memory allocated to other variables is overwritten.

More Examples – Storage

- Object reuse: in single processor systems, when a new process is activated it gets access to memory positions used by the previous process. 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.
- Core dumps: same story again

More Examples – Time

- Side channel analysis: smart cards implement cryptographic algorithms so that keys never leave the card; keys may still be obtained by observing side channels (power consumption, timing behaviour).
- SSL: error messages are encrypted to defend against certain guessing attacks; attacks are still possible if the timing of the reply depends on the nature of the error message.

The Layer Above

- It is neither necessary nor sufficient to have a secure infrastructure, be it an operating system or a communications network, to secure an application.
- Security services provided by the infrastructure may be irrelevant for the application.
- Infrastructure cannot defend against attacks from the layer above.
- Fundamental Fallacy of Computer Security:
  Don't believe that you must secure the infrastructure to protect your applications.

Summary

- Security terminology is ambiguous with many overloaded terms.
- Distributed systems security builds on computer security and communications security.
- Two major challenges in computer security, are the design of access control systems that fit the requirements of the Internet and the design of secure software.
- In security, understanding the problem is more difficult than finding the solution.