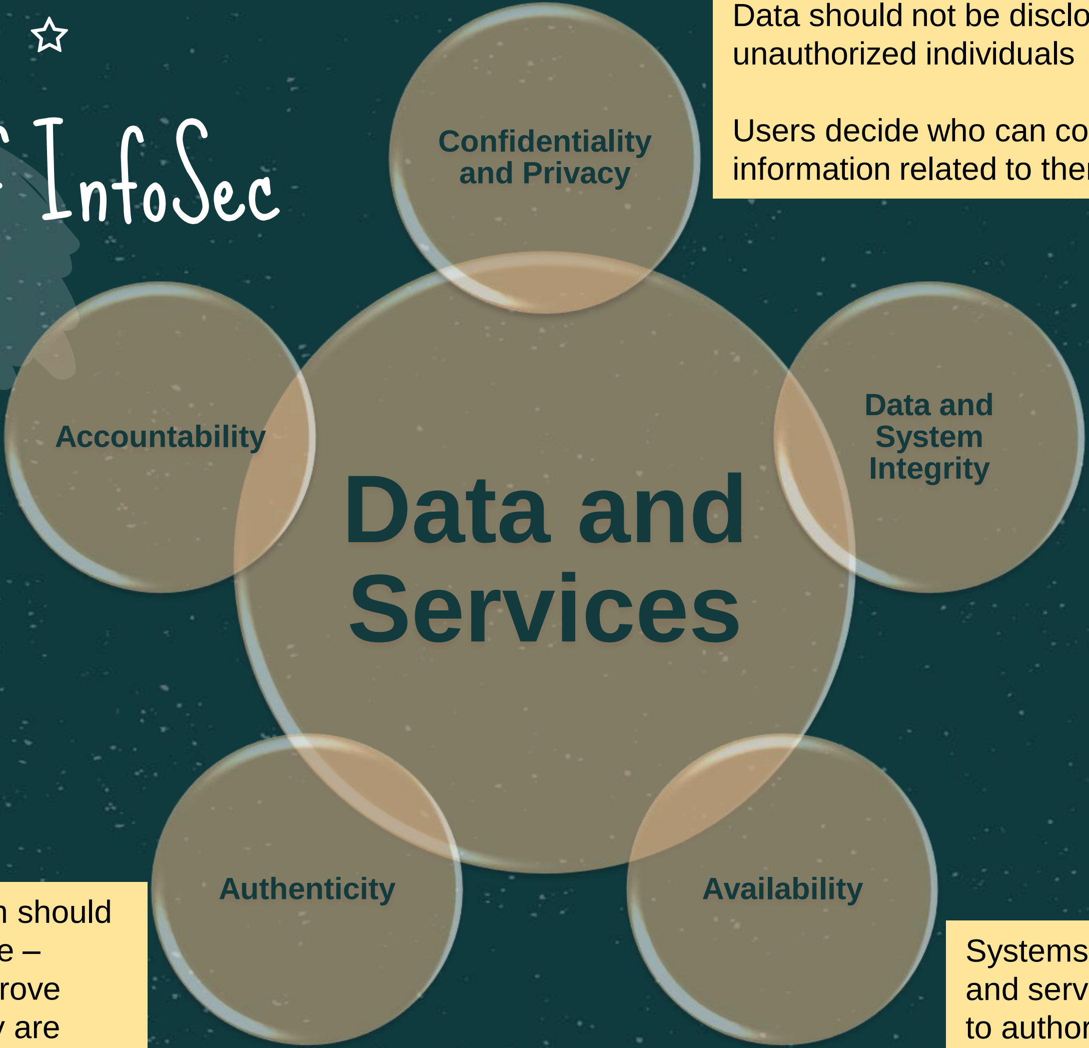
Computer Security Concepts

Information Security – Lecture 02

Aadil Zia Khan

Pillars of InfoSec



Data and Services

Confidentiality and Privacy

Data and System Integrity

Availability

Authenticity

Accountability

Data should not be disclosed to unauthorized individuals

Users decide who can collect what information related to them

Data should be accurate, complete, and consistent

Sender/receiver should not be able to deny that the data was from it (principle of non repudiation)

System should not be tampered with by unauthorized personnel and it should verifiably work properly

Systems should work promptly and service should not denied to authorized users

Input arriving at the system should come from a trusted source – source should be able to prove they are who they say they are

System should be able to trace a security breach to the responsible party

Systems must keep records of their activities to trace security breaches or to aid in disputes

Threats and Attack Types that Cause them

- **Unauthorized disclosure** (*threat to confidentiality*): adversary gains access to private data
  - **Exposure**: sensitive information is leaked either deliberately or through error
  - **Interception**: communication traffic (on broadcast or dedicated links) can be sniffed and sensitive information stolen
  - **Inference**: sensitive information is derived from limited available data – e.g., if you know the department's average salary, you can ask your colleagues about their salaries and determine the pay your boss is getting ☺
  - **Intrusion**: adversary can overcome system's access control to gain access to data

Threats and Attack Types that Cause them

- **Deception** (*threat to integrity*): false data may be received and accepted
  - **Masquerade**: unauthorized user may gain access by posing as an authorized user, or a Trojan may trick user into running malicious code
  - **Falsification**: valid data may be altered with false information, e.g., changing grades
  - **Repudiation**: user sends or receives data but later denies it

Threats and Attack Types that Cause them

- **Disruption** (*threat to system integrity and availability*): correct operations of a system may be interrupted
  - **Incapacitation**: system services may be halted by damaging the hardware/software
  - **Corruption**: malicious software causes the system to behave in an unintended manner
  - **Obstruction**: either communication is prevented, or the system is overloaded through excessive communication traffic or processing – e.g., through DDoS

Threats and Attack Types that Cause them

- **Usurpation** (*threat to system integrity*): adversary takes control of the system
  - **Misappropriation**: unauthorized use of another's identity without permission, resulting in harm to that person
  - **Misuse**: unauthorized use of a system

Examples of Threats to Assets

Assets\Threat	Availability	Confidentiality	Integrity
Hardware	electricity meter is stolen or broken	unencrypted USB is stolen	electricity meter is tampered with to reduce bill
Software	program is deleted	pirated copy of the software is made	program is corrupted to do unintended task
Data	file is deleted	file is read unauthorized or information inferred	fabricated data is added to the file
Communication	Denial of Service - packets/links destroyed - wireless jammed	packets are sniffed and read or traffic analyzed	packets are modified, delayed, or fabricated/replayed

Security Design Principles

- Impossible to develop security design and implementation techniques that exclude security flaws and prevent all unauthorized actions

- We do have widely agreed design principles that can guide the development of secure system

Security Design Principles - Economy of Mechanism

- Design should be small and simple
- Why?
  - Easier to test and verify thoroughly
  - The more complex the design, the greater the chance to miss a weak point
  - Easier to configure
- E.g.,
  - The more third party components your system uses, the more devices/data/resources it exposes, and the more connections it has => the more vulnerable it becomes

Security Design Principles - Failsafe Default

- If the system fails for whatever reason, it will default to a safe outcome
- By default, the access to the system/resource is forbidden
- Why?
  - In case of failure – system would opt for the safer outcome
- E.g.,
  - When you swipe a credit card, the credit card company analyzes the requested purchase in context of your recent purchases and spending trends - if it seems suspicious, the transaction is denied

Security Design Principles - Complete Mediation

- Every access must be checked against the access control mechanism
  - Systems should not rely on access decisions retrieved from a cache
- Why?
  - If a user had access to some resource, but that access was revoked, the new situation should be reflected in the system
- E.g.,
  - A user should not be able to access company data once he has left the job

Security Design Principles - Open Design

- Design of a security mechanism should be open rather than secret
- Why?
  - Algorithms can then be reviewed by many experts, and users can therefore have a high confidence in them
- E.g.,
  - Encryption keys are private but the algorithms are well known

Security Design Principles - Separation of Privilege

- Multiple steps are required to access a restricted resource
- Why?
  - Added defense – if one step fails, there are other lines of defense
- E.g.,
  - Multifactor user authentication – credit card use requires swiping the card followed by entering the password

Security Design Principles - Least Privilege

- Every user of the system should operate using the least set of privileges necessary to perform the task
- Why?
  - Limits the damage that can result from an accident or error or malicious intent
- E.g.,
  - Suppose you hire a housekeeper, you would give him the keys to your gate, and maybe front door, but not your room and certainly not your cupboard where you keep your valuables

Security Design Principles - Least Common Mechanism

- Minimize the functions shared by different users
- Why?
  - Easy to identify the perpetrator in case of malicious activity or error
  - Reduce the number of unintended communication paths
  - Can separate the users
- E.g.,
  - Suppose a website is shared between multiple users, if one user send too many requests (DoS), the website would be unable to serve others

Security Design Principles - Psychological Acceptability

- Security mechanisms should not interfere unduly with the work of users – ideally they should be transparent not intrusive nor burdensome
- Why?
  - People would get tired and also make mistakes
- E.g.,
  - Suppose the UMT admin asks you to change your password everyday – you would not like that and may also forget the password

Security Design Principles - Isolation / Encapsulation

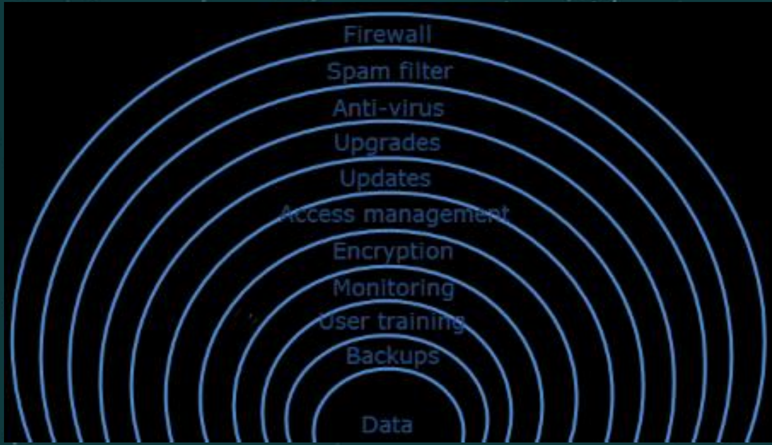
- 1) public access systems should be isolated from critical resources, 2) processes and files of individual users should be isolated from one another except where it is explicitly desired, 3) security mechanisms should be isolated
- Why?
  - Prevents disclosure or tampering or unauthorized access
- E.g.,
  - On a shared machine (very common in cloud computing), if different users are allowed to access the same memory location, they would be able to read each other's private data

Security Design Principles - Modularity

- The design should be modular – each security feature should be a separate module
- Why?
  - Individual parts of the security design can be upgraded without the requirement to modify the entire system
- E.g.,
  - If some communication protocol requires encryption, that should be a separate module so that if someone wants to change the encryption algorithm or key type, the entire system doesn’t need to be changed

Security Design Principles - Layering

- Use multiple, overlapping protection approaches - often referred to as ***defense in depth***
- Why?
    - Failure or circumvention of any individual protection approach will not leave the system unprotected
- E.g.,
    - Use access control, encryption, firewalls, anti-virus, etc.



Security Design Principles - Least Astonishment

- System should behave in a way that most users will expect it to behave
- Why?
  - Lesser chances of error
- E.g.,
  - A website should have an input field that gets focus automatically after the page loads – like Google Search, or the username field of a login form