Week 1- Introduction to Security Engineering

THE UNIVERSITY OF SYDNEY

Dr. Thilini Dahanayaka

School of Computer Science, The University of Sydney

Agenda

- What is Security Engineering?
- A Security Engineering Framework
- Security Goals
- Security Attacks and Other Key Definitions
- Applying the Security Framework: A Banking Case Study
- Why Security Is a Process?

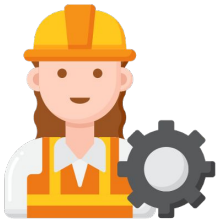
Recommended Reading

- Security Engineering - 3rd Edition by Ross Anderson
  - **Chapter 1** – What is security engineering?
  - **Chapter 2** - Who is the opponent?

- Cryptography & Network Security - 7th Edition by William Stallings
  - **Chapter 1** – Computer network and security concepts

1. What is Security Engineering?

- Before we define security engineering, let's first define **what engineering is**



Engineering is the application of science and maths to solve problems. While scientists and inventors come up with innovations, it is engineers who apply these discoveries to the real world.

What is Security Engineering?

- Cyber threats are increasing and have many forms
  - Ransomware
  - Data Breaches
  - DDoS
  - Virus/Worms/Trojans
  - Phishing, Spear Phishing, Vishing, Smishing
  - Business Email Compromises (BEC)
- Cost businesses/individuals millions of money

4.4M USD - The global average cost of a data breach in 2025.

-  Structured thinking about security is required

An Early Definition of Security Engineering

- Quote from: Programming Satan's Computer, R. Anderson, R. Needham. In: Computer science today: recent trends and developments, 1995.

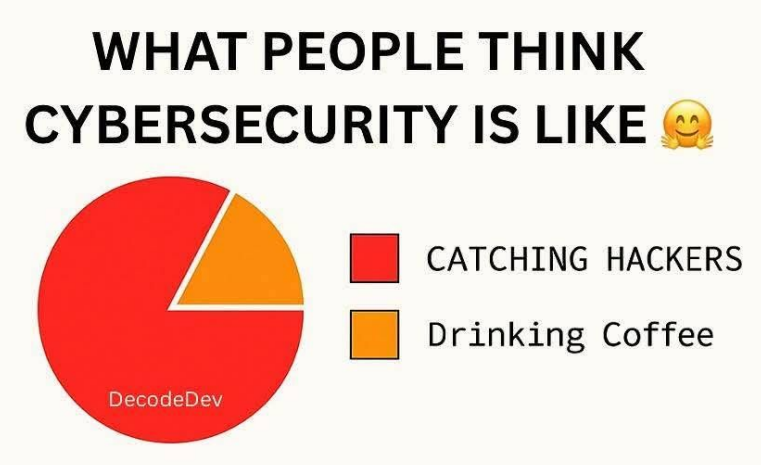
> *"Our task is to program a **computer that gives answers which are subtly and maliciously wrong** at the most inconvenient possible moment." [...]*
>
> *"The problem is the presence of a **hostile opponent** [...]"* **.**

What is Security Engineering?

Security engineering means conceiving, designing, and implementing hardware and software that produces only the expected answers, even when confronted with malice, error, or mischance.

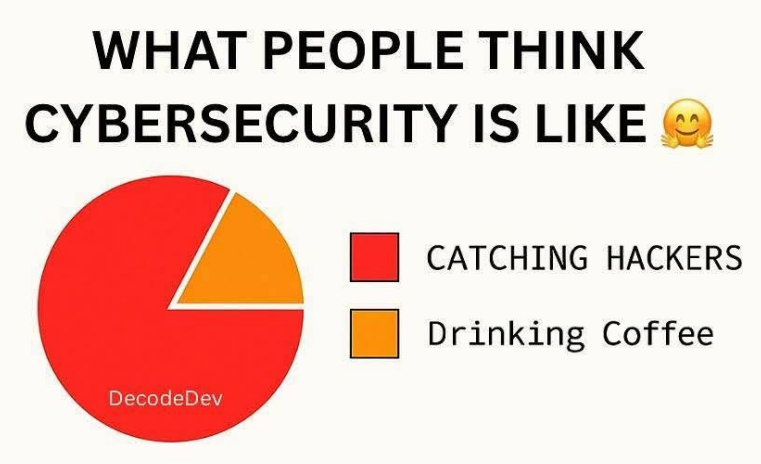
- It goes beyond simple fault tolerance.
  - Safety Engineering
  - Reliability Engineering
- A secure system tolerates the activities of the opponent.
- But we will see. There is no such thing as a perfectly secure system!
- Security is multi-faceted – We need to conceptualise it.
  - It also requires cross-disciplinary expertise

What Security Actually Looks Like

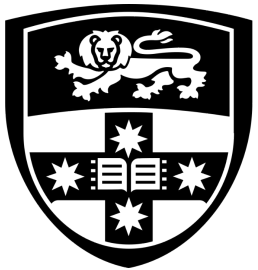


Source: https://www.facebook.com/groups/it.humor.and.memes/posts/31443305618601822/

What Security Actually Looks Like



Source: https://www.facebook.com/groups/it.humor.and.memes/posts/31443305618601822/

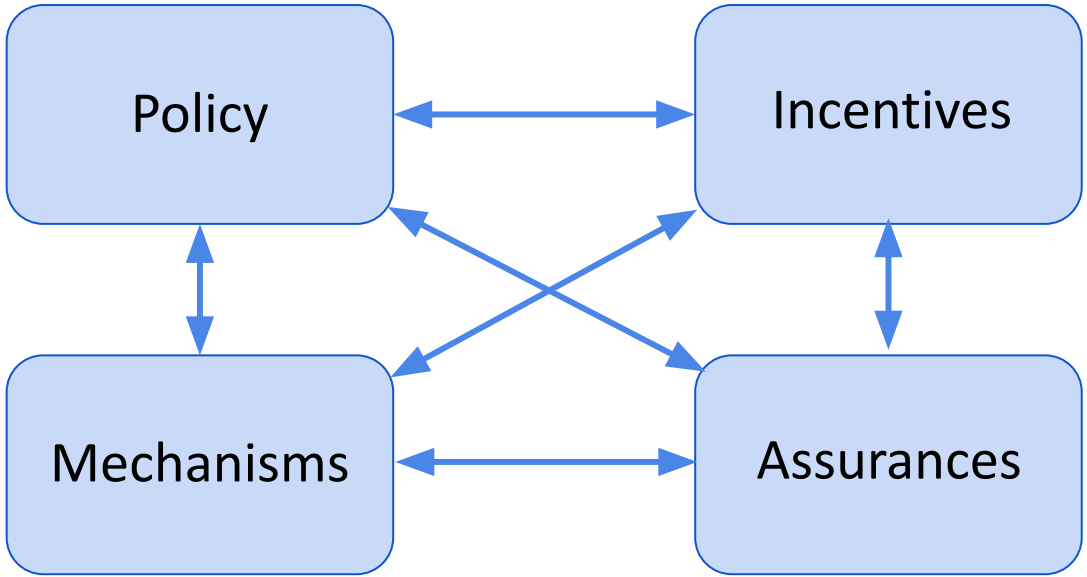


THE UNIVERSITY OF
SYDNEY

2. A Framework for Security Engineering

- Provides a reference for designing and assessing distributed system security.
- Helps investigate incidents and understand why security failures occur.

A Framework for Security Engineering



- **Policy** – What we want to achieve
- **Mechanisms** – Tools and machinery at our disposal to enforce policy
- **Incentives** – Motivations of system defenders and attackers
- **Assurance** – Confidence in mechanisms' reliability

Security Framework - Policy

Policies define what we mean to achieve. In other words, what it means to keep the system secure.

- Often higher management set the policies that are
  - Ensuring business continuity
  - Comply with various regulations and standards
- Some example policies are
  - *"Customer data must be stored only within our data centres."*
  - *"Only authorized employees must be able to access the personal information of clients."*
- Note that the policies do not define how to achieve or implemented.
  - Rather they define what we want to achieve

Security Framework - Mechanisms

Mechanisms are the machinery we have to implement policies

- Example for mechanisms
  - Encryption for data confidentiality
  - Tamper-resistant hardware for physical security
  - Cryptographic hashes for data integrity
- Security engineers must be aware and familiar with various mechanisms that are required to implement policies.

For most beginners in security engineering, it is all about machinery. It is important to understand that it is only one aspect of our framework.

Security Framework - Assurance

Assurance refers to how much reliance you can place on each particular mechanism and how well they work together.

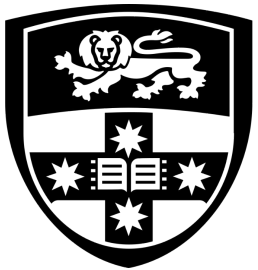
- Examples
  - How much time is required based on current computing capacities to break RSA encryption?
  - How much time is required to brute-force a 10-character password?
- Security engineers know/must be able to conduct professional estimates of assurances given by commonly used security mechanisms.
- Note that some of the security mechanisms we conder secure under current hardware may not be secure in future
  - **E.g.** It would take trillions of years for a classical computer to break RSA-2048. However, breaking RSA-2048 will take around 10 seconds* in a quantum computer.

Security Framework - Incentives

Incentives refer to the motive that the people guarding and maintaining the system have to do their job and the motive that the attackers have to try to defeat your policies.

- Depending on the business, we may get attacked by different attackers
  - State actors, Cybercriminals/ransomware groups, Script kiddies, Hacktivists
- The methodical process of understanding the possible threats and prioritising defence mechanisms is called **“Threat Modelling”**.

Security costs money – Understanding incentives allows us to understand the degree of security we need.



THE UNIVERSITY OF
SYDNEY

3. Security Goals and Design Principles

3a. Security Goals

- Security goals are used to define the security policies we discussed earlier.
- We are going to discuss some of them (not exhaustive).
- Also, note that sometimes there are contextual overlaps between some of these goals.

- Confidentiality
- Data/Message Integrity
- Authenticity
- Authorization
- Accountability
- Non-repudiation
- Deniability
- Availability
- Privacy

Confidentiality (and Secrecy)

Confidentiality refers to the effect of mechanisms that limit who can access data (in plain text form)

- That would mean access control is a form of confidentiality/secrecy.
- Secrecy can be understood to refer solely to the effect
- Confidentiality adds the obligation to protect secrets

- **Other use:** confidentiality is the goal of protecting the secret content of a message or data, commonly by encryption.
  - We do not distinguish between secrecy and confidentiality
  - We rarely need to refer to the obligation aspect of confidentiality; hence we simplify our terminology

Data/Message Integrity

Integrity refers to the effect of a mechanism to allow a verifier to check whether a message has been modified in transit or data has been changed since the last honest write.

- In general, modification of data in transit/messages cannot be defended against effectively.
    - The attacker may have a useful position in the network and can flip bits.
    - The important effect is that we can detect such changes!
    - Very close in meaning to tampering-evident.
- Not the same as tampering-resistance.
    - The latter refers to the capability to withstand attempts at modification.
    - More commonly used for hardware.

Authenticity

Authenticity refers to the effect of allowing a verifier to determine the origin/originator of a message or data item.

- Effective implementation as a mechanism usually implies integrity as well.
- Not the same as authentication - that is the process of identifying a party we are communicating with.
- In cryptographic protocols, the term also implies freshness.
    - When trying to authenticate, we need integrity.
    - We also need freshness proof that the other party is active in the process and not someone else replaying recorded messages.

Authorisation

Authorisation refers to the effect that a verifier can determine whether an entity is allowed to execute some action or access some data.

- Authorisation often makes use of authentication but does not imply it.
    - **E.g.** When you enter a cinema, you only need to show authorisation, not who you are.
- Authorisation is closely linked to Access Control and Accountability.

Accountability

Accountability refers to the effect that a verifier can determine which entity (or which category of authorised entities) is responsible for a certain action.

- Implies that one can trace the actions of an entity through the system.
- Implies authorisation and access control.
- May imply authentication.
- Is closely related to non-repudiation.

- Accountability does in general not stop there.
  - Needs some form of logging.
  - Needs to be auditable (the verifier can access and rely on the logs).
  - Needs protection for the logs.

Non-repudiation

Non-repudiation means that the entity who has caused an action cannot successfully deny responsibility.

- At its core, this is a legal concept
  - Cannot deny sending or receiving a transaction.
  - Note that in law we do not need cryptographic proof, but only plausible evidence to make a case.
- Accountability can support non-repudiation
- In cryptography it implies
  - Authenticity and integrity
  - Secure timestamping of the action (extremely hard)

Deniability

Deniability is the capability to (successfully) reject the notion of being responsible for some action.

- In many ways, the opposite of non-repudiation.
- Often more useful.
- Examples: highly confidential communication.

- It can be achieved up to a certain degree by cryptographic means.
- Understanding in security is not necessarily the same as the interpretation by a court.

Availability

Availability refers to a system's capability of providing its services when needed, even during an attack, error, or mishap.

- Includes availability/correct functioning of all relevant security mechanisms and network channels.
  - May trade lower level of security against higher availability.
  - **E.g.,** In warfare, the availability of communication channels is more important than having confidentiality on them - one may be willing to switch off encryption if needed.
- One of the most important properties in security.
- Very hard to achieve.
  - Many systems are asymmetric by design (an exploitable property).
  - Attackers often have a huge attack surface to choose from

Privacy

Privacy refers to an entity's right to determine what information relating to themselves they want to release or hide.

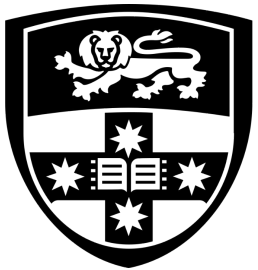
- Many different aspects as the term is related to society, law, and technology. So there can be other definitions.
  - Privacy is related to anonymity. But they are not the same
  - Privacy may also imply secrecy/confidentiality. Again not the same
- Technology can support the goal of privacy to a degree
- **Example:** Different aspects of privacy
  - Privacy of static data (released data sets)
  - Privacy of dynamic data (DB queries)
  - Privacy when moving through the network

3b. Fundamental Security Design Principles

- **Economy of mechanism:** The design of security measures embodied in both hardware and software should be as simple and small as possible
- **Fail-safe defaults:** Access decisions should be based on permission rather than exclusion
- **Complete mediation:** Every access must be checked against the access control mechanism
- **Open design:** The design of a security mechanism should be open than secret
- **Separation of privilege:** A practice in which multiple privilege attributes are required to achieve access to a restricted resource
- **Least privilege:** Every process and user of the system should operate using the least set of privileges necessary to perform the task
- **Least common mechanism:** The design should minimise the functions shared by different users, providing mutual security

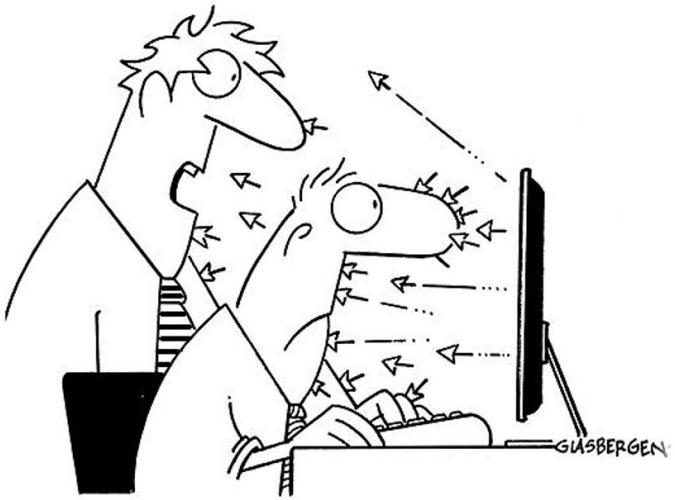
3b. Fundamental Security Design Principles

- **Psychological acceptability:** Security mechanisms should not interfere unduly with the work of users while still meeting the needs of authorised access/user
- **Isolation:** For example, critical resources should be isolated from public access
- **Encapsulation:** A specific form of isolation based on object-oriented functionality
- **Modularity:** Refers to the development of security functions as separate, protected modules and to the use of a modular architecture for mechanism design and implementation
- **Layering:** Multiple, overlapping protection approaches address the people, technology, and operational aspects of information systems.
- **Least astonishment**: means that a program or user interface should always respond in the way that is least likely to astonish the user.



THE UNIVERSITY OF
SYDNEY

4. Security Attacks



"I'm no expert, but I think it's some kind of cyber attack!"

Security Attacks

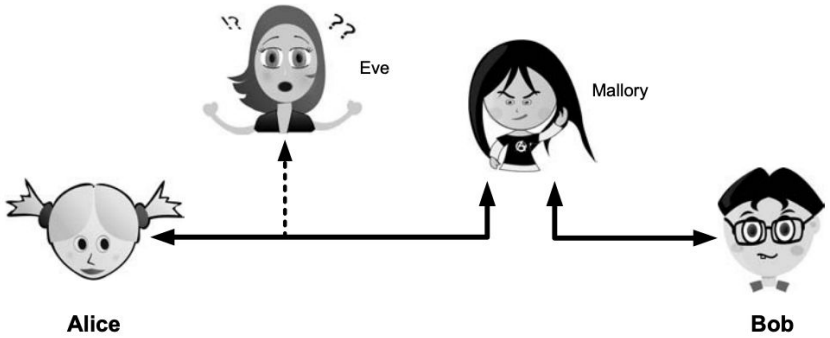
Security attack: Any action that compromises the security of information owned by an organisation.

- A useful means of classifying security attacks is defined in ITU-T Recommendation X.800, Security Architecture for OSI.

Passive attacks are in the nature of eavesdropping on, or monitoring of transmissions. The goal of the attacker is to obtain information that is being transmitted.

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: replay, masquerade, modification of messages, and denial of service.

Passive and Active Attacks



Source: TCP/IP Illustrated
Volume 1 Second Edition The Protocols

Passive Attacks

- Release of message content
- Traffic Analysis

Active Attacks

- Replay
- Data Modification
- Masquerade
- Denial of Service

Attack Vector

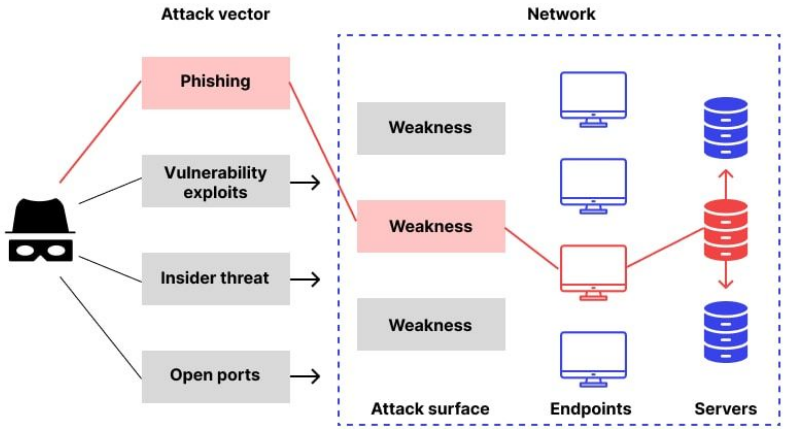
Attack Vector: Any action A specific method or pathway through which a hacker or malicious actor gains unauthorized access to a computer or network system

- **Examples**
  - **Phishing:** Using fraudulent emails or messages that appear to be from a trusted source to trick individuals into revealing sensitive information or clicking on malicious links.
  - **Malware:** Software designed to harm or exploit any programmable device, service, or network. Attackers can introduce malware through email attachments, software downloads, or operating system vulnerabilities.
  - **Person-in-the-Middle (PitM) Attacks:** Interception of the communication between two parties to steal or manipulate the data being exchanged.

Attack Surface

Attack Surface: Consist of the reachable and exploitable vulnerabilities in a system.

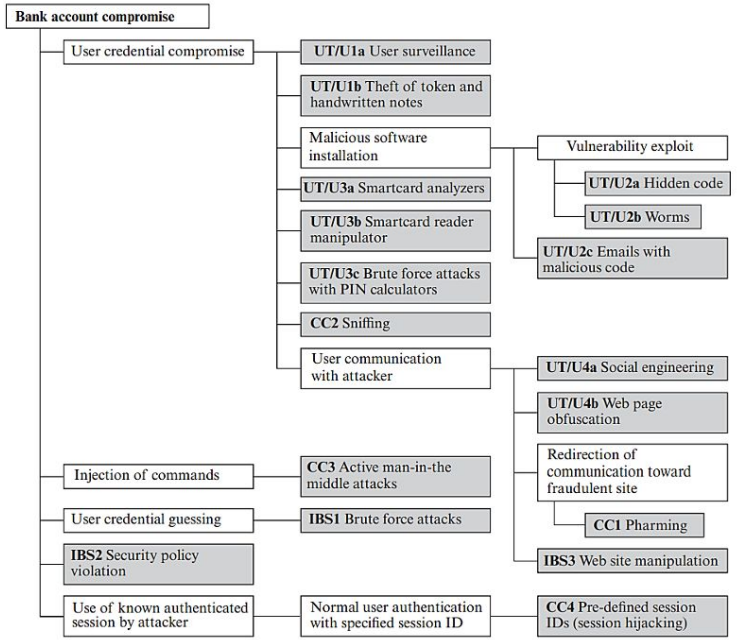
- Can be categorised as follows
  - Network attack surface
  - Software attack surface
  - Human attack surface



Attack Tree

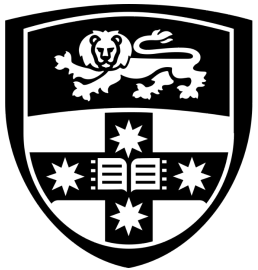
Attack Tree: A branching, hierarchical data structure that represents a set of potential techniques for exploiting security vulnerabilities

Example: An attack tree for internet banking authentication



Who is the opponent

- Criminals: Botnet herders, Malware developers, Spam senders, Bulk account compromise, Targeted attackers, cash-out gangs, ransomware
- Malicious insiders
- State actors
- Whistle-blowers
- Hacktivists and hate campaigns
- AI (Soon)



THE UNIVERSITY OF
SYDNEY

5. Example: Bank Security

Policy

- **Overall goal:** Correct account balances.
- Many different applications run together.
- Many people working together.
  - Huge attack surface!
  - Insider attacks!
- Many sub-goals - too many to list!
  - ATMs - integrity, confidentiality, availability
  - Online banking
  - Bank-to-bank transfers

Mechanisms

- Strong bookkeeping procedures and access control, especially in the backend.
- Anomaly detection systems! (e.g., unusual transfer).
- Four-eyes principle: 2+ signatures for large transfers.
- ATMs use strong cryptography (authentication!).

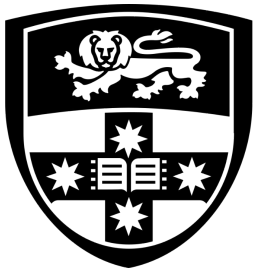
5. Example: Bank Security

Assurance

- Depends strongly on the mechanism
  - The biggest threat is probably insiders (petty theft!)
  - Neither bookkeeping nor cryptography defends against phishing the customer being robbed
  - Or, alternatively, the bank employee (spear-phishing!)
- Cryptography for integrity and confidentiality
- Anomaly detection is difficult—unusual transfers can still be legitimate!

Incentives

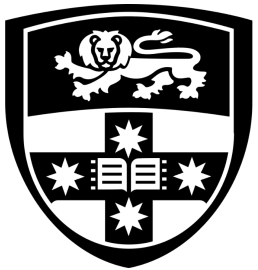
- For attackers
  - Get at the customer's money without the risk of detection
  - Involves a series of transfers so the trail of the money is lost or the money is irrecoverable (paid out).
- How about the employees?
  - Need to be carefully screened and vetted
  - Quite a common attack vector!



THE UNIVERSITY OF
SYDNEY

6. Security is a Process. It is not a Product

- History teaches us that all products come with flaws.
- The many facets of security mean that it is not enough to build (or buy a product) and then ‘have security’.
- Any number of other factors come into play.
  - Users, admins, insiders
  - Need to link systems
- The environment is dynamic, not statics.
  - Change of business value brings a change in attacker incentive
  - Need to update systems and work with others, …
  - The huge complexity of systems that are meant to be secured by a ‘security product.’



THE UNIVERSITY OF
SYDNEY

Recap

- **We discussed:**
  - What Security Engineering is
  - An overview of the preferred security framework
  - Key security goals and their relationship to the framework
  - The opponents, what attack surfaces and attack trees are
  - Applying the discussed security framework in a practical scenario
  - Why security is a process, not a product