Cybersecurity Fundamentals

Lecture 1

Introduction

Thomas Zacharias

University of Glasgow

Housekeeping
and
Ground rules

Course aims

- Course specification: https://www.gla.ac.uk/coursecatalogue/course/?code=COMPSCI4062
- Providing a broad overview of Cybersecurity.
- Explain the algorithms behind different cryptographic and communication solutions.
- Explain a number of different security protocols.
- Evaluate an existing or proposed system in terms of potential vulnerabilities.
- Recommend the most appropriate security solution to apply in a number of different scenarios.
- Implement an aspect of Cybersecurity.

Communications

- Use Moodle Discussion Forum for general questions.
- Use Moodle Coursework Questions forum for questions about coursework.
- DM or email the course coordinator or the lab assistants **only if** you have a problem that will affect your performance.
- Please find my office hours on Moodle page.

How to do well

- Attend all the classes and do the exercises and quizzes.
  - Learn as you go.
  - It is too difficult to try and learn the course just before the exam.
- Most marks are in the exam
  - You do well in the exam by learning the course as you go.

Code of conduct

- Please raise your hands if you would like to ask a question.
- We will all treat each other with respect and dignity.
- Any type of harassment will not be tolerated.

Course schedule

Lectures:

Tuesdays 09/01/2024 – 12/03/2024

Tutorials and Labs:

Tuesdays 16/01/2024 – 19/03/2024

(Please check Moodle page for details)

Assessment

- **Quizzes** (10%): open for 24 hours until 1pm on the next day of the lecture (every other week).
- **Written assignment** (10%): Due by 4pm 15th March.
- **Individual Exam** (80%): during April/May.

Plagiarism

In your written assignment, if you present someone else's work as your own work and/or without full acknowledgement, then this is plagiarism.

Introduction to Cybersecurity

What is Security?

*"The state of being free from danger or threat"**

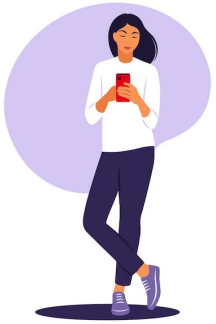
*Oxford Dictionaries

What is Cybersecurity?

The practice of protecting hardware, programmes, networks, and data from digital attacks

Protection targets (assets)

- Servers
- User devices
- Networks
- Applications
- Operating systems
- Databases
- Cyber-physical systems



Examples of cyberattacks

- Network attacks:
  - **Denial-of-Service**: a service becomes unavailable to its intended network users.
  - **DNS attacks**: exploitations of vulnerabilities in DNS.
    - **DNS amplification**: attackers use publicly accessible open DNS servers to flood a target system with DNS response traffic.
    - **Cache poisoning (spoofing)**: inject malicious data to the DNS servers' cache to redirect traffic to a malicious site.
    - **DNS tunneling**: encode data of other programs or protocols in DNS queries and responses (data exfiltration over DNS channel).

Examples of cyberattacks

- Software vulnerabilities:
  - **OS command injection**: the attacker injects arbitrary operating system (OS) commands on the target running the vulnerable application.
  - **SQL injection**: injects SQL commands that can read or modify data from a database.
  - **Cross-site scripting (XSS):** execute malicious scripts in a web browser of the victim by including malicious code in a web page or application.

Examples of cyberattacks

- Malware:
  - **Viruses**: malware with the ability to replicate itself and spread via an infected host file.
  - **Worms**: standalone malware that can spread itself across the network by self-replication.
  - **Trojans**: a legitimate software look-alike that is installed by tricking users.
  - **Rootkits**: provide the attacker with unauthorised access and control over a computer while hiding their presence.

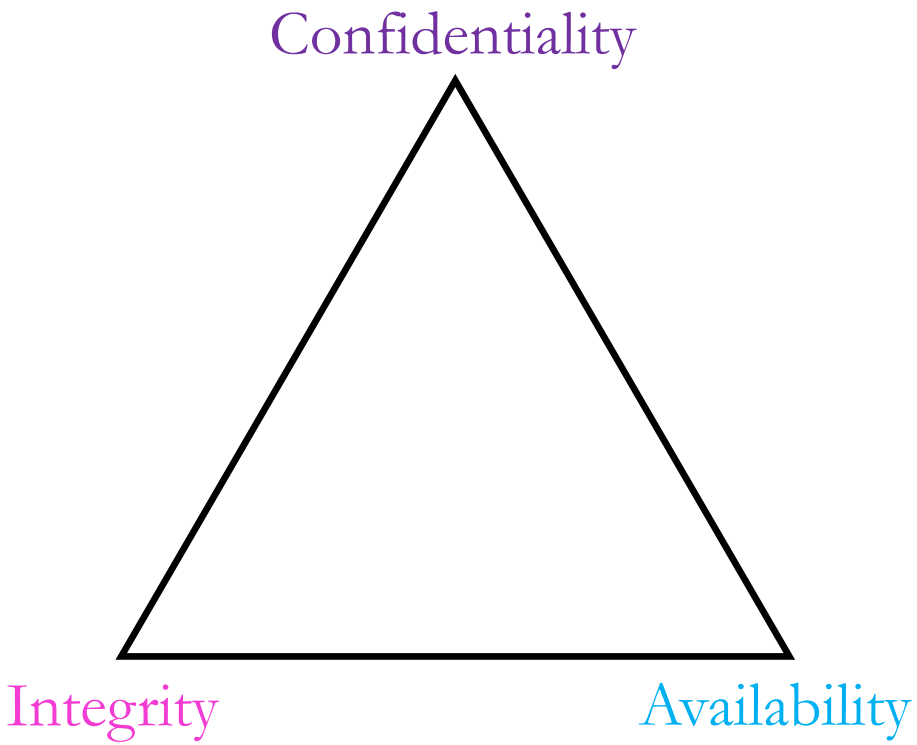
Examples of cyberattacks

- Social engineering:
  - **Weak/old password interception**: 123456, 123456789, qwerty, password, 12345, qwerty123, 1q2w3e, 12345678, 111111, 1234567890,… (from cybernews.com)
  - **Phishing attacks**: an attacker, masquerading as a trusted entity, deceives a victim into opening an email, instant message, or text message.

Protection mechanisms

- Anti-malware software
- Firewalls
- Intrusion-detection systems
- Cryptography
- User education and awareness

Key objectives of Cybersecurity



Key objectives of Cybersecurity

- Confidentiality:
  - **Data confidentiality**: private or confidential information is not made available or disclosed to unauthorised individuals.
  - **Privacy**: individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
- Integrity:
  - **Data integrity**: Assures that information and programs are changed only in a specified and authorised manner.
  - **System integrity**: a system performs its intended function in an unimpaired manner, free from unauthorised manipulation of the system.
- Availability: systems work promptly and service is not denied to authorised users.

The key objectives in terms of requirements

- Confidentiality: preserving authorised restrictions on information access and disclosure, including means for **protecting personal privacy and proprietary information**.
- Integrity: guarding against improper information modification or destruction, including ensuring information **non-repudiation and authenticity**.
- Availability: ensuring **timely and reliable access** to and use of information.

An additional objective

- Truly secure systems are not yet an achievable goal.
- We must be able to trace a security breach to a responsible party.
- Accountability: the requirement for actions of an entity to be traced uniquely to that entity. This objective supports:
  - non-repudiation
  - deterrence
  - fault isolation
  - intrusion detection and prevention
  - recovery and legal action

Low level impact of security breaches

- Definition of low level impact:
  - the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced.
  - minor damage to organizational assets.
  - minor financial loss.
  - minor harm to individuals.
- Examples:
  - Confidentiality: lists of faculty or departmental lists (typically published on a school's Web site).
  - Integrity: anonymous online poll (the inaccuracy of such polls is well understood).
  - Availability: online telephone directory lookup application (hardcopies offer another way of accessing the information).

Medium level impact of security breaches

- Definition of medium level impact:
  - the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced.
  - significant damage to organizational assets.
  - significant financial loss.
  - significant harm to individuals that does not involve loss of life or serious, life-threatening injuries.
- Examples:
  - Confidentiality: student enrollment information (usually protected information, yet seen by more people on a daily basis).
  - Integrity: a forum Web site to discuss some specific topic (if used for the enjoyment of the registered users, then the Web master may experience some data, financial, and time loss).
  - Availability: public Web site for a university (although not a critical component, its unavailability will cause some embarrassment).

High level impact of security breaches

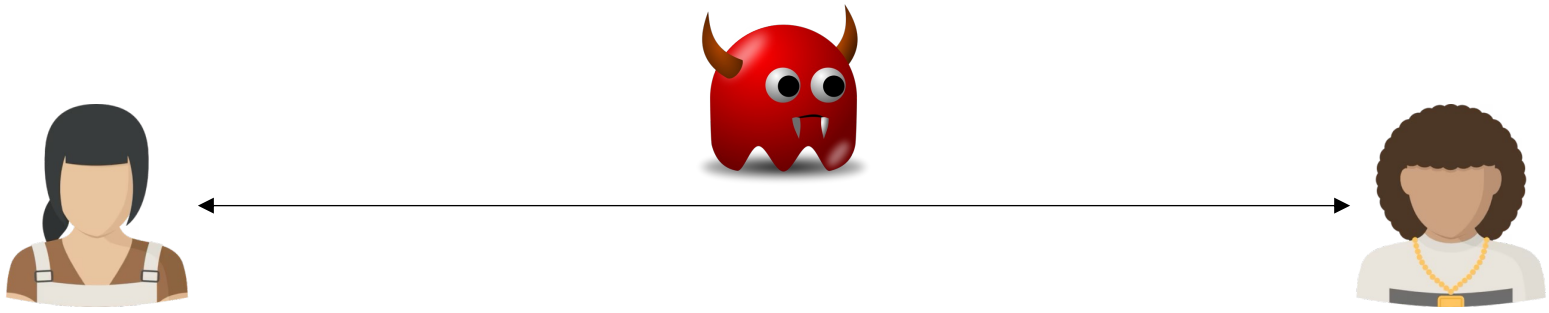
- Definition of high level impact:
  - the organization is not able to perform one or more of its primary functions.
  - major damage to organisational assets.
  - major financial loss.
  - severe or catastrophic harm to individuals involving loss of life or serious life-threatening injuries.
- Examples:
  - Confidentiality: student grade information (available only to students and employees that require the information to do their job).
  - Integrity: a patient's allergy information (inaccurate information could result in serious harm or death to a patient and expose the hospital to massive liability).
  - Availability: authentication service for critical systems, applications, and devices (interruption of service results in the inability for users to access computing resources and/or perform critical tasks).

Challenges of Cybersecurity

- In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features.
- Typically, a security mechanism is complex, and it is not obvious that such elaborate measures are needed.
- Security mechanisms typically involve more than a particular algorithm or protocol (e.g., participants may be in possession of some secret information).
- The great advantage that the attacker has is that they need to only find a single weakness.
- Security is not often by design; instead, it is considered only after the system design is complete.
- Strong security is often seen as an impediment to efficient and user-friendly operation.

Types of attacks

- **Passive attacks**: an attempt to learn or make use of information from the system that does not affect system resources.



Example: eavesdropping

Types of attacks

- **Active attacks**: an attempt to alter system resources or affect their operation.

Example: man-in-the-middle

Origin of an attack

- **Inside attack**: Initiated by an entity inside the security perimeter (an "insider"). The insider is authorised to access system resources but uses them in a way not approved by those who granted the authorisation.
- **Outside attack**: Initiated from outside the perimeter, by an unauthorized or illegitimate user of the system (an "outsider"). On the Internet, potential outside attackers range from amateur pranksters to organized criminals, international terrorists, and hostile governments.

Countermeasures

- A countermeasure is any means taken to deal with a security attack.
- Ideally, a countermeasure can be devised to prevent a particular type of attack from succeeding.
- When prevention is not possible, or fails in some instance, the goal is to detect the attack and then recover from the effects of the attack.

Threats and assets: Hardware

- A major threat to computer system hardware is the threat to availability.
- Threats include accidental and deliberate damage to equipment as well as theft.
- Theft of data storage media (e.g., DVDs and USB flash drives) can lead to loss of confidentiality.

Threats and assets: Software

- A key threat to software is an attack to availability.
  - Application software is easy to delete.
  - Software can also be damaged to render it useless.
- A more difficult problem to deal with is attacks to integrity:
  - Software modification that results in a program that still functions but behaves differently than before.
  - Computer viruses fall into this category.
- Software privacy: the problem of unauthorized copying of software has not been solved.

Threats and assets: Data

- Destruction of data files is the main concern in terms of availability.
- Data confidentiality:
  - A widely explored problem is the preservation of the unauthorised reading of data files or databases.
  - Data analysis over (multiple) databases used for statistical analysis purposes (aggregate information).
- Data integrity is a major concern in most installations, as modifications to data files can have consequences ranging from minor to disastrous.

Threats and assets: Networks

- **Passive attacks**: eavesdropping/monitoring network traffic.
  - **Release of message contents** (e.g., in email, conversations, or transferred files) may include confidential information.
  - **Traffic analysis** (release of metadata such as the location and identity of communicating hosts and the frequency and length of messages being exchanged).
  - Encryption protects the communication contents but does not suffice to prevent traffic analysis!
  - Passive attacks are very difficult to detect because they do not involve any alteration of the data.

Threats and assets: Networks

- **Active attacks**: involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories:
  - **Replay attacks**: data retransmission that creates an unauthorised effect.
  - **Masquerade**: one entity pretends to be a different entity (typically, having extra privileges).
  - **Modification of messages**: messages are altered, delayed, or reordered.
  - **Denial-of-Service (DoS) attacks**: preventing the normal use or management of communication facilities (a specific target or an entire network).

It is quite difficult to prevent active attacks absolutely, because of the large number of possible vulnerabilities. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.

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.
  - Small design is easier to test and verify thoroughly.
  - The more complex the mechanism, the more likely it is to possess exploitable flaws.
  - Updating or replacing a simple mechanism becomes a less intensive process.
- Open design: the design of a security mechanism should be open rather than secret.
  - Cryptographic algorithms should be open to public scrutiny (only the keys must remain secret).
  - Reviewing by experts leads to high confidence.

Security design principles

- Least privilege: every process and every user of the system should operate using the least set of privileges necessary to perform the task.
  - Unless permission is granted explicitly, the user or process should not be able to access the protected resource.
  - Any access control system should allow each user only the privileges that are authorised for that user.
- Psychological acceptability: security mechanisms should not overly hinder the usability or accessibility of resources.
  - Otherwise, users may opt to turn off security mechanisms.
  - Security procedures must reflect the user's mental model of protection (this highlights the importance of security training).
  - If the protection procedures do not make sense to the user, the user is likely to make errors.

Security design principles

- Isolation:
  - Public access systems should be isolated from critical resources (physically or logically) to prevent disclosure or tampering.
  - The processes and files of individual users should be isolated from one another except where it is explicitly desired.
  - Security mechanisms should be isolated in the sense of preventing access to those mechanisms (e.g., logical access control may protect cryptographic keys).
- Modularity: the development of security functions as separate, protected modules and to the use of a modular architecture for mechanism design and implementation.
  - Common cryptographic algorithms serve as modules that can be invoked by numerous protocols and applications.
  - Each security mechanism should be able to support migration to new technology or upgrade of new features without requiring an entire system redesign.

Attack surfaces

- **Attack vector**: a specific way to breach the Cybersecurity of a system (e.g., insider threats, phishing mails, compromised credentials).
- **Attack surface**: the total number of attack vectors of a system. Namely, all the reachable and exploitable vulnerabilities in the system.
  - **Network attack surface**: vulnerabilities over an enterprise network, wide-area network, or the Internet.
  - **Software attack surface**: vulnerabilities in application, utility, or operating system code. A particular focus in this category is Web server software.
  - **Human attack surface**: vulnerabilities created by personnel or outsiders, such as social engineering, human error, and trusted insiders.

End of Lecture 1

The slides content is covered by Chapter 1 of "Computer Security Principles and Practice (3rd Edition)" by Stallings and Brown