**1.1 General Terms:**

Al-Mustanseriya University Class: Third Year

Engineering College Subject: Ciphering & Data Security

Computer and Software Engineering Dep. Lecture -1-

* Data security

Data security is the means of ensuring that data is kept safe from corruption and that access to it is suitably controlled.

* Computer Security

The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users.

Malware: malicious software

* + - includes computer viruses, worms, trojan horses, most rootkits, spyware, dishonest adware,
* Network Security

Protect the network and the network-accessible resources from unauthorized access, consistent and continuous monitoring and measurement of its effectiveness. Cryptography has a long history. Its original and main application is to enable two parties to communicate in secret, across an unsecured (public) channel.

* *Cryptography*: science of secret writing with Ciphers
* *Cryptography* (from Greek kryptós, "hidden", and gráphein, "to write") is, traditionally, the study of means of converting information from its normal, comprehensible form into an incomprehensible format, rendering it unreadable without secret knowledge — the art of encryption*.*
* *Past:* Cryptography helped ensure secrecy in important communications, such as those of spies, military leaders, and diplomats.
* In recent decades, cryptography has expanded its remit in two ways
	+ mechanisms for more than just keeping secrets: schemes like digital signatures and digital cash, for example.
	+ in widespread use by many civilians, and users are not aware of it.
* *Cryptanalysis*: science of breaking ciphers
* *Cryptology*: both of above
* *Encryption*: transforming *plain text* to *cipher text*
* *Decryption*: recovering *plain text* from *cipher text*
* *Encryption scheme, cipher, cryptosystem*: a mechanism for encryption and decryption Cryptography and cryptanalysis are sometimes grouped together under the umbrella term *cryptology*, encompassing the entire subject.
* *Steganography*

the study of hiding the very *existence* of a message, and not necessarily the *contents* of the message itself (for example, microdots, or invisible ink)

**1.2 Attacks**

When you test any computer system, one of your jobs is to imagine how the system could malfunction. Then, you improve the system's design so that the system can withstand any of the problems you have identified. In the same way, we analyze a system from a security perspective, thinking about ways in which the system's security can malfunction and diminish the value of its assets.

**1.2.1 Vulnerabilities, Threats, Attacks, and Controls**

A computer-based system has three separate but valuable components: **hardware**, **software,** and **data**. Each of these assets offers value to different members of the community affected by the system. To analyze security, we can brainstorm about the ways in which the system or its information can experience some kind of loss or harm. For example, we can identify data whose format or contents should be protected in some way. We want our security system to make sure that no data are disclosed to unauthorized parties. Neither do we want the data to be modified in illegitimate ways. At the same time, we must ensure that legitimate users have access to the data. In this way, we can identify weaknesses in the system.

**Vulnerability** is a weakness in the security system, for example, in procedures, design, or implementation that might be exploited to cause loss or harm. For instance, a particular system may be vulnerable to unauthorized data manipulation because the system does not verify a user's identity before allowing data access.

A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm. To see the difference between a threat and vulnerability, consider the illustration in Figure 1. Here, a wall is holding water back. The water to the left of the wall is a threat to the man on the right of the wall: The water could rise, overflowing onto the man, or it could stay beneath the height of the wall, causing the wall to collapse. So the threat of harm is the potential for the man to get wet, get hurt, or be drowned. For now, the wall is intact, so the threat to the man is unrealized.



Figure 1-1. Threats, Controls, and Vulnerabilities

However, we can see a small crack in the wall a vulnerability that threatens the man's security. If the water rises to or beyond the level of the crack, it will exploit the vulnerability and harm the man.

There are many threats to a computer system, including human-initiated and computer-initiated ones. We have all experienced the results of inadvertent human errors, hardware design flaws, and software failures. But natural disasters are threats, too; they can bring a system down when the computer room is flooded or the data center collapses from an earthquake, for example.

A human who exploits vulnerability perpetrates an **attack** on the system. An attack can also be launched by another system, as when one system sends an overwhelming set of messages to another, virtually shutting down the second system's ability to function. Unfortunately, we have seen this type of attack frequently, as denial-of-service attacks flood servers with more messages than they can handle.

How do we address these problems? We use a **control** as a protective measure. That is, a control is an action, device, procedure, or technique that removes or reduces a vulnerability. In Figure 1-1, the man is placing his finger in the hole, controlling the threat of water leaks until he finds a more permanent solution to the problem.

To devise controls, we must know as much about threats as possible. We can view any threat as being one of four kinds: *interception, interruption, modification, and fabrication*. Each threat exploits vulnerabilities of the assets in computing systems; the threats are illustrated in Figure 1-2.



Figure 1-2. System Security Threats

* An **interception** means that some unauthorized party has gained access to an asset. The outside party can be a person, a program, or a computing system. Examples of this type of failure are illicit copying of program or data files, or wiretapping to obtain data in a network. Although a loss may be discovered fairly quickly, a silent interceptor may leave no traces by which the interception can be readily detected.
* In an **interruption**, an asset of the system becomes lost, unavailable, or unusable. An example is malicious destruction of a hardware device, erasure of a program or data file, or malfunction of an operating system file manager so that it cannot find a particular disk file.
* If an unauthorized party not only accesses but tampers with an asset, the threat is a **modification**. For example, someone might change the values in a database, alter a program so that it performs an additional computation, or modify data being transmitted electronically. It is even possible to modify hardware. Some cases of modification can be detected with simple measures, but other, more subtle, changes may be almost impossible to detect.
* Finally, an unauthorized party might create a **fabrication** of counterfeit objects on a computing system. The intruder may insert spurious transactions to a network communication system or add records to an existing database. Sometimes these additions can be detected as forgeries, but if skillfully done, they are virtually indistinguishable from the real thing.