# Computer Security (CM2025)

**Course Notes**

Felipe Balbi

October 13, 2020

# Contents

# Week 1

Key Concepts

- Understand the central goals and aspects of computer security.
- Understand and explain the differences between a range of malware types.
- Identify key examples of malware and their historical significance.

## Reading: Ethics in information security

Read the following article:
Ethics in information security, IEEE Security & Privacy, vol.15 May/June 2017, pp.3–4.
You should also familiarise yourselves with the ACM ethics guidelines.

## Defining computer security

With the advent of the Internet, computer security became a very important field of research.

Computer Security is the *protection of computer-related assets against danger, loss or loss of control of something valuable.*

Security has three main goals:

**Prevention** The safeguarding of assets from threats;

**Detection** Systems that alert if malicious activity is, or is about to, take place;

**Reactiion** Definition of procedures that enables us to deal with an attack.

Security can also be split into three main components:

**Policy** deals with confidentiality, integrity and availability of data;

**Threat Model** assumptions about those involved with malicious activity;

**Mechanism** The SW/HW used to make sure the policies are enforced using the assumptions made during Threat Modelling.

There are five important terms that need to be defined:

**Attack** activities harmful to computer systems, data, software, and hardware;

**Risk** the possibility of damage or loss of digital assets in case of an attack;

**Zero-day (vulnerability)** a vulnerability used by an attacker before its discovery by the developer of the SW;

**Exploit** SW used to take advantage of a bug or vulnerability;

**Hacker** Subdivided into three groups

**White Hats** Find vulnerabilities with a goal of fixing them before its discovery by an attacker. They work under the permission of the owner of the computer system being attacked;

**Black Hats** Try to penetrate the system to gain unauthorized access. Often, their motivation is harm operations or steal sensitive operations;

**Gray Hats** They fall somewhere in-between White Hats and Black Hats and work with varying combinations of good and bad intentions.

## Types of malicious software 1

Malware is a piece of softwarte designed to disrupt, damange and destroy an information system. There are many types of malwares, some of which are discussed in the following subsections

### Viruses

They self-replicate by inserting themselves into other files, programs, documents, etc. Can spread through emails, USB sticks and downloads from unknown sources.

The *Creeper* is considered to be one of the first computer viruses. Developed in 1971, it infected computers connected to the ARPANET, the internet prototype.

Another early virus was the *Elk Cloner*. Written by a high-school student to infect Apple II computers using floppy disks in 1982. Every 50th time the computer booted, it would display a poem written by the hacker.

Not all viruses are harmless. The *I Love You* virus in 2000 caused around $10 billion worth of damages by affecting nearly 10% of all computers around the globe.

### Worms

They can replicate without attaching themselves to existing software. The *Stuxnet* is a well-known worm, considered to be one of the most destructive worms ever created. It was designed to attack Programmable Logic Controllers by Siemes.

PLC devices are used for the automation of processes in machinery. In this case, it targetted centrifuges in Iranian nuclear power plants and altered the speed of the machine, causing it to tear itself apart.

It's estimated that *Stuxnet* destroyed 20% of Iranian's nuclear power plant centrifuges.

### Adware

These display advertisements on your screen during browsing or online shopping. Possibly the most visible form of malware one can encounter. Its main purpose is to collect user data.

### Trojans

Trojans are named after the famous ancient Greek tale of the invasion of the city of Troy by the Greek during the Trojan War.

After trying and failing several times to get access to the city, the Greeks came up witha plan where squatter soldiers would hide in the joint wooden status of a horse presented as a gift.

During the night, after entiring the city within the horse, the Greeks broke out of the horse to attack the city.

Trojans have a similar way of working: they hide themselves inside an application or program data and spread based on specific user action.

### Spyware

Designed to spy on the target machine or the user, it collects information and sends it back to the hacker for further use or for sale on the dark web.

The *Dark Hotel* spyware is one famous case which used Hotel Wi-Fi to target the personal systems of government officials, business tycoons and political leaders to extract sensitive information.

## Types of malicious software 2

We have a look at *Keyloggers*, *Ramsonware*, *Botnets*, and *Rootkits*.

A *Keylogger* records every keystroke from the user. This may include messages being typed, emails, confidential information such banking credentials, users, passwords, etc.

*Olympic Vision* is a keylogger used for Business Email Compromise (BEC) attacks. It also uses several other pieces of malicious software to steal sensitive information and spy on business transactions. Nowadays it's very easy to get a hold of a keylogger.

In a *Ramsonware* attack, the victim's data is encrypted, backup files are deleted and the people responsible demand money in exchange for decryption of this data. In other words, the victims are held to *ramsom* for renewed access to their data. The *WannaCry* attack is a recent example that took place in May 2017.

In a *Botnet* attack, computers connected to the internet are taken over by an attacker which remotely controls the computers using a Command And Control (CNC) server to carry out Distributed Denial Of Service (DDOS) attacks. A DDOS attack is when a given server is flooded with so much traffic at one moment, that it collapses.

*EchoBot* is a botnet used to exploit over 59 known vulnerabilities and launch a number of attacks, such as DDOS attack, steal sensitive information, conduct corporate espi-

onage, and infects a wide range of Internet Of Things connected devices. Furthermore, it also scans for old vulnerabilities in legacy systems for future exploitation.

Finally, we have *Rootkits*. These remain hidden in a target computer and activate in secret. They can perform several activities rangin from giving attackers remote access to a computer all the way to stealing sensitive information such as a password or credit card details. They can also use a compromised computer to launch any of the other attacks described before.

# Week 2

Key Concepts

- Explain the key differences between static and dynamic analysis.
- Explain the usage of sandboxes in malware analysis.
- Understand the need for a variety of methods of malware analysis.

## Malware analysis and techniques

Static Malware Analysis is one of the techniques used to analyze and combat the types of malware discussed previously.

More generally, Malware Analysis is a set of processes and techniques that help a Security Analyst understand the functionality, origin, impact, and intent of malicious software.

The goal of this activity is find the *Indication Of Compromise* (IOC) that depicts the behavior of malicious software. IOCs are also used to develop signatures of malware. The two techniques used for such analysis are Static Malware Analysis and Dynamic Malware Analysis.

In Static Malware Analysis, the executable files are examined without being executed. We can determine if a file is clean or malicious and also discover information about its functionality.

With the information collected during Static Malware Analysis, allows us to determine signatures, which are a collection of distinguishing features that can be used to recognize malware.

Two essential techniques used in static analysis are antivirus scanning and hashing. Antivirus scanning is the traditional method of running a file through an antivirus scanner to try to detect whether the file contains malware. Hashing is an algorithm that produces a value referred to as a *Hash* which is a unique fingerprint for a given file. Any modification to a file will result in a different hash fingerprint, including infection by a malware.

While static analysis methods are useful as a starting point for some more basic types of attack, it can be powerless against more recent and advanced types of malware. These have found ways to circumvent the detection methods used during static analysis.

The solution to this is Dynamic Malware Analysis.

## Malware analysis 2

Dynamic Analysis or Behavioral Analysis is where we execute the malware in a controlled environment known as a sandbox. This allows a Security Professional to observe the behavior of malicious software, help to understand its functionality and, hopefully, find the Indicator Of Compromise.

This method overcomes some of the limitations of Static Analysis with regards to catching the more advanced and adaptive forms of malware.

Dynamic Analysis is an efficient method for analyzing malware because it helps uncover the functionality of the malware, which is not entirely possible with static analysis.

A Sandbox is a virtualized environment that contains a virtual network, services, drives, etc, to ensure that the malware behaves exactly as it would in a real environment.

There are two main types of sandboxes: Agent-based and Agent-less. Agent-based sandboxes require software to be installed on every computer that needs to be monitored. Well known examples are cuckoo, threat expert, bit blase, and Comodo. Conversely, an Agent-less sandbox monitors computers on the network from afar without needing to be installed on every device. Popular examples are VMRay, Analyzer, and SNDBOX.

Security Researchers use both types of sandboxes, but some research suggests that agent-less sandboxes are more efficient.

Another common tool for dynamic analysis on Windows machines is process monitor (Procmon). It's used to monitor the registry, file system, network, running processes, etc.

## Ethics

Ethics is really important in the field of Computer Security.

Because we frequently work with computers, we may be exposed to security issues and vulnerabilities. When that occurs, we may be able to fix the issue ourselves by setting a rules in our *iptables* or blocking traffic from a certain port for instance.

There may be, however, wider ramifications to the problem we have discovered. It could be something worth mentioning to the maker of the faulty software.

That's where **Responsible Disclosure** comes into the picture. The term itself is somewhat subjective, but there may be legal ramifications related to the disclosure of a security issue.

In a situation where we find an issue in a popular software package, e.g. a popular Operating System, it may be the case that many other users are affected. This means that we have some responsibility to the Company or Service Provider in terms of disclosing the issue.

It is common practice to identify the bug to the provider and offer them enough time to fix the issue before disclosing the problem publicly. There are considerations about transparency.

Some companies may refuse to patch issues or even seek legal action against those exploiting or even simply highlighting the issue.

We should also think about security in a distributed way. For example, cloud computing services offer on-demand computing. These services may not be hosted in the locality of the developer or the client. As such, we must consider conforming to the law in **all** of these locations.

As a final thought, actions have consequences. Identity theft and distributed denial of service attacks have real world consequences. People's lives can be destroyed if we do something that can cause harm.

## Passwords

Designing truly secure systems is very hard, only made harder by data leaks happening periodically as can be seen in the news. Data is becoming more and more valuable and there are places in the *Dark Web* where one can buy leaked data.

One of the most prolific leaks utilised a simple encryption method where the same encryption key was utilized. This resulted in a one-to-one mapping between plain-text and encrypted passwords. In other words, there was a situation where different users with the same password would end up with the same encrypted string stored in the database.

What this means is that if one password is cracked, all other users who happen to be using the same password were also compromised. To make matters worse, some users used password hints, which were also stored in the database.

Reusing passwords is also a common problem. If a user's password for one service is leaked, there is a probability that the same user employed the exact same password on multiple services which renders all of such services compromised.

There are ways to design more secure systems, however that also has implications. We could require longer passwords and encourage two-factor authentication, but a user may loose their phone or forget that longer, more complex password.

A good system design balances accessibility with security and usability.

## Social Engineering

Insecure designs can have far reaching ramifications, however a system is only as secure as its safest link.

Social Engineering is one of the most common attack vectors and it does not rely on technical subtleties of attacks. These attacks rely on the fact that not all staff are properly trained in security and, as such, attackers may exploit gaps in their knowledge.

A company may have a robust security policy in place to handle access control, however **compliance** is a different matter.

Phishing emails are a good example of Social Engineering attacks. They try to trick you into thinking the email comes from a reliable source and convince you to give them the information they're after. Some of these emails may look fairly authentic and we may have to look deeper to determine their authenticity.