Co-funded by the Erasmus+ Programme of the European Union

InCyT Training Module

Information Security for Employees **Unit 2 Week 4** **Unit name: Malware and Most Common Attacks**	
Learning Activities	❑ **Webinar** ❑ **Exercises** ❑ **Workshop** ❑ **Presentation** ❑ **Other**
Learning Responsible	Claudio Fornaro
Learning Activity Goals	1. Introduction to the various types of malware 2. Knowledge of the most common types of attacks
Skills to be Gained	• Basic knowledge about what is malware and the most common attacks to computer systems Soft Skill 2 - SS2
Duration of Learning activity	1 week
Learning requirements	***What is needed?***
	• No previous specific knowledge is required

Co-funded by the
Erasmus+ Programme
of the European Union

Brief information about the module

When it comes to malware, the name already implies something malicious. This term generally refers to software designed intentionally to cause damage of various kinds, in particular (but not limited to): blocking of services and communications, interception of transmissions, theft of private information (personal, corporate or government) and of personal identities, unauthorized access to information or systems, deprivation of users of access to information.
After an introductory part, the fundamental concepts of malware are outlined, along with the description of the main types, the risks that they entail and how these can be mitigated.
The second part of the module intends to offer an overview of the most common attacks that a system can suffer, this list being by no means complete as new types of attacks are discovered every day. These attacks are attempts to gain access to a computer system or disrupt its services. The description starts from the most common and easiest attacks (Phishing) to more complex ones: Password Attacks, Buffer Overflow, SQL Injection, Cross-site Scripting (XSS), Cross-Site Request Forgery (CSRF), File Related Attacks, Denial of Service (DoS) and Distributed Denial of Service (DDoS), Man in The Middle (MITM), ARP Spoofing or ARP Poisoning, DNS Spoofing, Drive-by Attacks, Credential Reuse, Web Session Hijacking, TCP Session Hijacking, IP Spoofing, Zero-day Exploits, and the consequences of Leak of Code.

Content of the Learning Material

In computer science, "hacker" and "cracker" are two terms that are used to indicate people with a great knowledge of computer systems. The main difference between the two lies in the attitude with which they act. A **hacker** (the term was born in the second half of the 1900s at the MIT) acts by exploring systems in order to better understand how they work and trying to discover new flaws in order to solve them and therefore increase the security of a system. Some actions can only be demonstrative or for ethical reasons, but never for profit. A **cracker**, on the other hand, uses his abilities illegally trying to break into systems to profit by stealing data or in a destructive way.

Hackers are constantly trying to acquire new knowledge of computer systems in order to better understand them, or improve their functions, performance and security. Some of them also engage in demonstrative and ethical actions, but never for profit. Hacker actions are performed almost always without any permission, so the Law punishes them (with great differences among the countries), for this reason hackers usually hide behind nicknames. Sometimes, wise security or IT companies instead of prosecuting hackers, often hire them to take advantage of their high skills and creativity (possibly after paying part of their debt to Justice). They are then employed to test

applications, security systems and programs both in production and already on the market in order to block attacks, and to prevent problems that could arise in the future from system bugs.

Real hackers have a reputation as experts, and are highly respected in their community. Competitions are organized with the aim of "piercing" a system or program in the shortest possible time. These competitions are called *hackathons* and are often sponsored by the giants of the computer industry.

The term "cracker" is attributed to Richard Stallman (a well-known figure of the free software movement, creator of the GNU project and member of the hacker community since the 1970s). He wanted to avoid the abuse of the term "hacker", differentiating who acted without any willing to harm and who, on the contrary, acts in order to have profit or just give damage. A cracker is a cyber-criminal who tries to attack computer systems in an unauthorized and illegal way, with malice, trying to take fraudulent advantage of them by exploiting vulnerabilities not yet known and resolved. Crackers are often involved in industrial espionage, data theft, and blocking of networks and websites, vandalism, etc.

The terms hacker and cracker have subtle bounds that divide them. For example, an intrusion can be considered an illegal act for some people and not for others, and the Law reflects these opinions; but this document is not about ethic or legislative aspects. Instead, we must consider that today, unfortunately, the two terms are considered synonymous.
In this document, we will *unhappily* use the more common and misused term hacker for both categories, being sure that the wise reader will be quite able to distinguish between a *hacker* and a *cracker*.

“Hackers” can be roughly classified into several categories, the most widely used are the following.

Black hat hackers use their knowledge for malicious or criminal purposes, being careful not to be traced.

White hat hackers test systems in order to identify possible vulnerabilities and eliminate them to prevent unwanted access. They often are security experts payed for this purpose.

Gray hat hackers are similar to white hats, but they often conduct research on system vulnerabilities without authorization and once these vulnerabilities are discovered they use them to force vendors to fix the problem, usually for free. Some vendors reward them with money, others prosecute them.

Suicide hackers are usually less competent than black/white hats, but they act for ideological reasons and have no interest in hiding their identity.

State-sponsored hackers are paid by a Government to launch cyber attacks against another Country; they have access to large resources as provided by the State as well as protection from prosecution for the offenses committed. They act to subtract confidential information, plans or projects, or to create generalized disorders and damage the population, the Army, the Government management, hospitals, plants for the production of energy, water and power supply, etc. These actions are considered real war actions.

Script kiddies do not have or have not yet acquired great skills, they usually make use of scripts and tools created by other hackers. All hackers before they acquire great skills are script kiddies.

Cyber terrorists are individuals or groups with hostile intentions that generally aim to create chaos in a Country; they are similar to state-sponsored hackers but do not have (yet) a Government to support them, or the support is not evident.

Malware

Malware is the acronym of "**mal**icious soft**ware**", a generic term that identifies any malicious program harmful to a system. Malware is the software hackers create to pursue their goals: making money, stole data, disrupt activities, etc.

One of the most common malware is the so called a **Trojan horse**[1] (or simply **Trojan**), a program that seems useful software and often act as such, but carries a hidden function that is executed to do along with the normal activated when the application is started. These are often shareware programs or cracked programs that are distribute modified in order to carry the Trojan horse. In other cases, it is just malware with a misleading name that makes it seem a useful software. Trojans are often spread by emails or by downloading programs through the Web and usually do not spread by themselves to other computers. Malicious code examples are *keyloggers* (programs that steals passwords and send them to the hacker), *backdoors* (programs that work as a terminal server and allows the remote hacker to access the computer in some ways bypassing the normal authentication procedures), *cryptominers* (programs that mines cryptocurrency), *bots* (short for robots, used for example for a distributed denial of service attack).

Rootkits are software installed by malware that try to hide the presence of a malware itself from detection; they achieve this by modifying the operating systems.

The malware that infect other software can be a virus or a worm. These get their name from the way they spread rather than from any specific types of behavior. A computer **Virus** is software similar to Trojan horse but can produce copies of itself by infecting (introducing itself into) other

[1] The term is derived from the Ancient Greek story of the Trojan horse used to invade the city of Troy by stealth

software, including the operating system files. The virus is spread to other systems when infected software is copied and run on these systems.

Similar to a virus is a **Worm**, this is a stand-alone software that operates on a computer and infects other computers through the network, but without infecting files.

A **Ransomware** is a type of malware that restricts access to the device it infects, requiring a ransom to be paid to remove the restriction. The main types of ransomware are 1) *Screen-locking ransomware* that blocks screens with an intimidating accusation of illegal behavior and asking the victims to pay a fee; 2) *Encryption-based ransomware*, where the malicious software some type of files (pictures, documents) or all files on an infected machine. When the full encryption is done, a pop-up usually informs the user that (all) files have been encrypted and that a fee must be payed to recover them, usually in cryptocurrency. Examples of encryption-based ransomware are CryptoLocker and WannaCry. A variant known as a **Wiper** looks like a ransomware but just destroys the data, even if a payment is often required.

Greyware are a different category of malware: these programs can worsen the performance of computers and may cause security risks, behave in an annoying or undesirable manner, but usually are less dangerous (which is a relative opinion) than the previous categories of malware. Spyware (e.g. keyloggers, programs that record all the keys pressed on the keyboard), adware (advertisements windows on the screen or in the browser, fraudulent dialers, etc.), Potentially Unwanted Programs (PUPs, applications that the user installed just because of a mistake in a windows of agreement during another program installation) are typical form of greyware. They can be stand-alone programs or plugins for other programs, for example browsers. They sometime limits themselves to change the settings of other programs, for example in order to bring the user to a specific ad website or show an ad page each time the browser is started or occasionally.

Antiviruses

As antivirus engines enhance their power to detect malware, a combination of various techniques are used to try to avoid detection and analysis. More recent malware is able to adopt sophisticated countermeasures such as:

- analyzing and detecting by fingerprinting the environment to adopt specific hiding methods;
- trying to obfuscate themselves in order to confuse antivirus detection based on malware signature;
- adopting a timing-based evasion in order to fool detection of anomalous behavior, the malware is normally quiescent and activates itself in very specific periods (for example during the boot process) or following certain actions taken by the user.

A very hard to discover technique for evasion is *Fileless malware* or *Advanced Volatile Threats*. This kind of malware runs in memory and does not require a file to operate. It uses existing system tools to perform its activity. The absence of any file on the file system makes them quite difficult to detect and analyze, except, partially, for those monitoring tools that check the resource usage and behavior of the system. The increase of this type of attacks is very high.

Risks

Vulnerable software can be exploited to gain privileges or bypass defenses, until the problem has been patched; this is the main reason that software should be up-to-date, in particular the Operating System. Firewalls and Intrusion Detection Systems monitor traffic patterns on the LAN.

The rule of the least privilege says that users and programs should have no more privileges than they require, otherwise malware can take advantage of this.

Mitigation

Antivirus software block and possibly remove some types of malware, they can provide real-time protection against the installation of malware software on a computer and a full scan of files and configuration (e.g. Windows registry) looking for potential threats.

Personal firewalls analyze any network connection and can spot unusual activity.

Sandboxing is running an application isolated from the rest of the system, some application can run a specific application while in more complex systems a virtual machine could be more appropriate.

Website vulnerability scans is performed by some antivirus software against a list of well-known dangerous sites and prevent the connection to them.

Network Segregation dividing a network in parts and enable only that traffic between them that is known to be legitimate, this prevents malicious software to spread across the network. Modern Software Defined Networking techniques are able to implement realize this segregation.

“Air gap” isolation or “parallel network” brings Network Segregation to the limit by completely isolate a portion of a network and provide enhanced controls over the entry and exit of software and data from the outside world. Even this solution is not the definitive one, as software and data must enter the isolated system, for example for patching. A well-known example of breaching an isolated network is the famous **Stuxnet** malware that was introduced to the target environment via USB drives “casually” forgotten nearby, picked up by the plant staff and inserted into the plant computers to see the contents. Stuxnet has three modules: a worm that controls the attack; a link file that automatically executes the worm when inserted in a Windows computer; and a rootkit

component responsible for hiding all malicious files and processes, to prevent detection. Other ways of crossing air gaps analyze electromagnetic, thermal and acoustic emissions.

Most Common Types of Attacks

This part describes some of the most common attacks to a computer system. It is not and cannot be complete as the variety of attacks is extremely huge, each one has many variants and many attacks and variants are created and discovered (and solved) every day. This means that the aim of this part is not to provide a complete classification, but to give an idea of the high level of knowledge required to both perpetrate and counter attacks.

Phishing

Phishing attacks are very common and easy to perform. They consist of sending e-mails that appear to come from a reliable sender, such as a colleague or a bank, with the aim of tricking the victim into doing a certain type of operation.

The attacker tries to persuade the victim to open an attachment or provides a link that leads to a page very similar to the original one, for example a bank home page, but created by the attacker, then to ask the victim to enter their credentials. To optimize this type of attack there are a whole series of techniques that are part of the so-called social engineering, since, in this case, the vulnerability of the computer system is to be identified in the user himself, who is deceived and unintentionally provides his credentials.

To defend against these attacks, a careful inspection of the link is needed (it is near impossible to distinguish between lowercase L 'l', an uppercase I 'I', and the digit one '1', or to notice the difference between .it and .lt in a URL). Moreover, one should be aware that serious companies never put links in their email that bring to a login page.

Password Attacks

To get passwords from a victim a hacker can try social engineering, as seen in the previous paragraph, or launch a password attack. Passwords are usually stored as hashes, and hashes are very hard to invert. There are essentially two cases: the encrypted password is available or is unknown and this type of attack can be performed in different ways:

- **Brute force attack:** in this case the program will try all the possible combinations of characters out of a set;
- **Attack with dictionary:** the program will test a sequence of commonly used passwords (dictionary);

- **Attack with rainbow files:** a rainbow table is a precomputed table listing the output of hash functions of password. Use of a key derivation that employs a salt makes this attack infeasible.

System managers should test the password files of the users of their systems on a regular basis with programs capable of performing password attacks, both by brute force and by dictionary.

To protect against these attacks, a system should check the quality of the password before allowing the user to set it, requiring the use of a minimum number of characters (e.g. 10) from different sets (upper and lower case letters, digits, punctuation characters). If the password entered is incorrect for some subsequent attempts, it is appropriate to adopt an account lockout or delay policy on the next attempt, thus preventing the attacker from quickly trying many combinations.

Buffer Overflow

Buffer overflow (or overrun) occurs when you send more bytes to a buffer than it can hold, these bytes overflow the buffer intended to hold them and change other bytes or, depending on the Operating System and file type, even inject executable machine code. While modern Operating Systems may prevent running an executable file with mixed data and code, this is not true for the simpler Operating Systems often found in embedded systems, where in many cases there is no Operating System at all.

SQL Injection

A very common attack is SQL Injection. The SQL language is used in relational databases; hence, these types of attacks are aimed at information theft. In essence, due to the interpreted nature of the SQL query, malicious SQL code is sent to a web page that is know to use an SQL query to gather the data to present to the user. The malicious SQL code modifies (sometimes just adding other syntactic parts) the predefined SQL query that results is a modified query that exposes other data than those the original query was written for.

While preventing this type of attacks is not complex (for example preventing the access to other tables of the database besides the ones needed to perform the original query), SQL injection is still one of the most common attacks.

Cross-site Scripting (XSS)

In a cross-site scripting (XSS) malicious executable scripts is injected into the code of a trusted web application (plugin) or website page, with the result that it is downloaded by other people (viewers). When the server sends back the compromised web page, the user's web browser considers it been delivered from a trusted source, and then the injected code is not identified as being malicious. The active content is usually a script (programming code that is executed by the browser) that can allow an attacker to get access to sensitive page content, session cookies, or other information the

browser keeps on behalf of the user. With XSS attacks, it is possible to steal sensitive data of all users who access infected sites. For example, stealing the SESSIONID of a website session (which is located in the user browser cookies that in turn are used to avoid that each connection to the same website require authentication beyond the first one), could allow an impostor to bypass the login phase and illicitly operate on behalf of the user.

Cross-Site Request Forgery (CSRF)

A CSRF is an involuntary http request (wanted by the attacker) to a website performed by a user already authenticated to that website; the website at this point will execute the request without further verification, allowing the attacker to carry out his attack. This type of attack allows modifying user data or performing unwanted transactions, such as unwanted purchases, if it were for example an e-commerce site. The attacker must have very good knowledge of the site in question to carry out a CSRF attack.

An example of a CSRF attack is performed by changing an URL. For example the source code of the web page is analyzed by the malware to find the "Change password" link and verify that the http GET method is used, then a new link with a new password (chosen by the attacker) can be formed and sent to the victim, for example via an email attachment. If the link is clicked and it happens that the victim is already authenticated to the website, a hidden start of password changing procedure is activated, effectively giving the account to the hacker. Good password changing methods ask to insert the old password too (and the GET method is not used). Moreover, the use of a CAPTCHA method (a test to verify whether the user is a human or a computer) is quite effective.

File Related Attacks

There are many types of attacks on files. The most known vulnerabilities are:

- **Insecure Direct Object References:** occurs when an application requests a resource from a server calling it by its name, but allowing the user to modify the latter, in order to request another resource.
- **Local File Inclusion (LFI):** exploits the vulnerability of the GET and POST functions of the PHP language to insert malicious files into the code.
- **Path Traversal:** allows a malicious user to access directories with limited access and execute commands. Often, they are used following an LFI to get deep into the system and obtain the necessary privileges.

To protect against these types of attacks, it is important to prevent file uploads to servers, disable remote code execution, restrict PHP access to the file system, and update software periodically.

Denial of Service (DoS) and Distributed Denial of Service (DDoS)

Maybe the most know network attacks, DoS attacks basically consist in sending many packets to block the possibility to receive other packets, thus preventing access to the machine for regular users, blocking the services offered.

The difference between a DoS and a DDoS attack lies in the fact that in the first case a single host is used to carry out the attack, while in the second the attack is carried out with many hosts (called zombies) at the same time. These hosts have been previously infected by a malware and at the command of the attacker start the connection to the victim machine.

Among the DoS attacks, we can remember:

- **Syn-Flood**: consists of sending a large amount of connection requests. The machine that receives a request (a SYN packet) from a client will respond to the client by sending a message, allocating some resources for this connection, and waiting for the acknowledge packet (that will never arrive). Therefore, the connection request remains active until a timeout occurs. If all the possible connections have been started, the service provided by the machine will no longer be available. The timeout occurs within seconds but the huge rate of requests keeps the machine clogged.
- **Smurf**: this attack is a combination of IP spoofing and ICMP requests. IP spoofing refers to the sending of an IP packet in which the sender's IP is forged. By continuously sending *ICMP echo requests* (the message sent by the ping command, used to verify if a machine is reachable) to broadcast IP addresses after having substituted the sender IP address with victim's IP address, the victim machine will receive a response for every request from any host active on the network, thus blocking the connection (flood). To prevent these attacks from occurring, it is possible to disable in the routers the forwarding of broadcast IP addresses.
- **Ping of death**: consists in sending a fragmented packet with a total IP size greater than the maximum allowable to a victim machine, once the machine reassembles the packet it can incur buffer overflows.

It is common for a DDoS attacks to use a **Botnet** that is thousands or millions of computers infected with malware, which at the command of the attacker, execute a Syn-Flood attack in unison on a victim server.

Man in The Middle (MITM)

With MITM we mean a whole typology of attacks that aim to intercept, analyze and retransmit a communication between two parties who believe they are communicating directly to each other.

This type of attack is usually carried out using IP spoofing: network sniffers eavesdrop a connection and new packets created by changing the IP address of the sender and recipient so that the sender sends data to the attacker who in turn it will re-send them to the recipient and vice-versa.

There are many ways to defend against these types of attack; the one most effective and used today is adopting *end-to-end encryption*, where the end subjects perform the encryption/decryption and all the intermediate systems can only see encrypted data.

ARP Spoofing or ARP Poisoning

The Address Resolution Protocol (ARP) enables network communications to reach a specific device on the local network. It is used to map IP addresses to Media Access Control (MAC) addresses (used to identify the hosts on a LAN). A host must use the ARP protocol to know the MAC address of the router/gateway in order to connect to the Internet. Each host maintain an *ARP cache* to map already requested and known IP/MAC pair. If the host does not know the MAC address for a certain local IP address, it sends out a broadcast ARP request packet, asking all the other hosts on the local network who has the IP address. The host with that IP address will answer, thus providing its MAC address.

An ARP spoofing/poisoning attack can be classified as a local network MITM attack. The attacker must have access to the local network and scans it to determine the IP addresses of the router and another host. The attacker sends forged ARP responses that make the router and the host to believe that the attacker's MAC address is the one of the host and the router respectively. This fools both router and workstation to connect to the attacker's machine, instead of to each other. As the two devices update their ARP cache entries, host and router communicate with the attacker instead of directly with each other.

DNS Spoofing

The Domain Name System (DNS) is the naming system used to convert Internet names (more properly FQDN - Fully Qualified Distinguished Names, e.g. www.server.net) to the corresponding IP address. The system is hierarchical and decentralized with some top-level servers. Each time a system attempts to reach a network resource, a connection is made to a DNS server in order to convert the name of the resource to an IP address. The resource records usually contains administrative information and other data used for example by mail servers.

A hacker will want to alter DNS records in order to send traffic to a fake or "spoofed" website instead of the legit one. The spoofed machine could then do MITM attacks, spoofing, etc. To prevent DNS spoofing, DNS servers must be kept up-to-date.

Drive-by Attacks

In a drive-by attack, malicious code is embedded into an insecure website. When the infected page is visited, the script is automatically executed on the victim computer and compromises it, there is no need to do anything on the site or enter any information. In this case, the use of an updated

recent and decent antivirus and/or personal firewall is the only protection as they can detect from their database if a site is unsafe before a user visits it.

Credential Reuse

The need to have credentials for several services brings users to reuse some of their passwords. Once an attacker has a list of usernames and passwords of compromised websites or services (for example obtained from the Dark Web), it is possible that the same user could use the same user/passwords pair on other systems. This will greatly reduce the number of password to check with a dictionary attack. In order to contrast this type of attack, the same countermeasures adopted to counter password attacks are effective. When many complex passwords are needed, some people find password managers useful.

Web Session Hijacking

A Session Hijacking attack consists of the exploitation of session tokens, used by web browser as session control mechanisms. The web server needs to relate different connections from the same user, as for each web page a TCP connection is created for each element (text, images, etc.). The most used method is based on a token that the Web Server sends to the client browser after a successful client authentication. A session token is just an identifier, a long unique string, and can be used to avoid the need to perform an authentication procedure for each connection. This attack is performed by stealing or predicting a valid session token to gain unauthorized access to the Web Server.

TCP Session Hijacking

TCP connections must guarantee to deliver packets in the right sequential order, in order to achieve this it uses acknowledgement packets and sequence numbers. These acknowledgement packets are just numbers that are incremented in some way after the 3-way initial handshake. The goal of the TCP session hijacker is to create a state where the client and server are unable to exchange data; enabling him/her to forge acceptable packets for both ends, which mimic the real packets. Thus, the attacker is able to gain control of the session. If the sequence can be predicted, an attacker can send forged packets that are recognized as valid from the victim host.

IP Spoofing

An IP spoofing is usually performed with a DoS attack: a huge number of connections blocks the legitimate server and, at the same time, a fake server sets its IP address with the number of the legitimate server. Therefore, the user is driven to the fake server that sends messages with an IP address indicating that the message is coming from a trusted host.

Blind Hijacking: In cases where source routing is disabled, the session hijacker can also use blind hijacking where he injects his malicious data into intercepted communications in the TCP session. It is called blind because he cannot see the response; though the hijacker can send the data or commands, he is basically guessing the responses of the client and server.

Zero-day Exploit

Strictly speaking, zero-day exploit is not a specific attack as cyber-criminals learn of a vulnerability that has just been discovered in certain software and operating systems and then target organizations who are using that software before a fix becomes available.

Leak of Code

The source code of applications can sometimes contain sensitive information and give valuable insights for a cyber-attack, in addition to the intrinsic economic value of the code itself.

Questions:

In separate file.

Attachments:

In separate file.

References:

Every topic has its own page on Wikipedia, sometimes already too thorough for the purposes of this course; books are too specialized and are useful only after a very deep training on the specific subject.

1. Malware, or malicious software
   URL: https://www.malwarebytes.com/malware
2. Cybersecurity for everyone.
   https://www.avast.com/c-malware
3. What are Cyber Security Threats?
   URL: https://www.imperva.com/learn/application-security/cyber-security-threats/
4. 21 Top Cybersecurity Threats and How Threat Intelligence Can Help
   URL: https://www.exabeam.com/information-security/cyber-security-threat/