POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

Automatic Malware Signature Generation

Relatori
prof. Antonio Lioy
ing. Andrea Atzeni

Michele Crepaldi

Anno accademico 2020-2021

Thanks...

Summary

Summary...

Acknowledgements

Aknowledgments...

Contents

Chapter 1

Introduction

Introduction...

Chapter 2

Background

2.1 Malware

Malware, short for ***malicious software***, is a general term for all types of programs designed to perform harmful or undesirable actions on a system. In fact in the context of IT security the term *malicious software* commonly means [1]:

> *Software which is used with the aim of attempting to breach a computer system's security policy with respect to Confidentiality, Integrity and/or Availability.*

Malware consists of programming artefacts (code, scripts, active content, and other software) designed to disrupt or deny operation, gain unauthorized access to system resources, gather information that leads to loss of privacy or exploitation, and other abusive behaviour. Malware is not (and should not be confused with) defective software - software that has a legitimate purpose but contains harmful bugs (programming errors).

Different companies, organizations and people describe malware in various ways. For example **Microsoft** defines it in a generic way as:

> *Malware is a catch-all term to refer to any software designed to cause damage to a single computer, server, or computer network* [2].

The **National Institute of Standards and Technology** (**NIST**), on the other hand, presents multiple definitions for malware, describing it as "hardware, firmware, or software that is intentionally included or inserted in a system for a harmful purpose" [3].

In another more specific definition **NIST** affirms that Malware is:

> *A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or of otherwise annoying or disrupting the victim* [3].

Notice that, since the attacker can use a number of different means - such as executable code, interpreted code, scripts, macros etc. - to perpetrate its malicious intents, the term ***software*** should be understood in a broader sense in the above definitions.
Moreover, the computer system whose security policy is attempted to be breached is usually known as the ***target*** for the malware. Instead, the cybercriminal who originally launched the malware with the purpose of attacking one or more targets is generally referred to as the "*initiator* of the malware". Furthermore, depending on the malware type, the initiator may or may not exactly know what the set of targets is [1].
According to the above definitions software is defined as malicious in relation to an attempted breach of the target's ***security policy***. In other words, software is often identified as malware based on its *intended use*, rather than the particular technique or technology used to build it.

2.1.1 Why is Malware used

Generally, cybercriminals use malware to access targets' sensitive data, extort ransoms, or simply cause as much damage as possible to the affected systems.

More generally malware serves a variety of purposes. For example, the most common cybercriminals' uses of malware are: [4]

- **To profit financially (either directly or through the sale of their products or services)**. For example, attackers may use malware to infect targets' devices with the purpose of stealing their credit account information or cryptocurrency. Alternatively, they may sell their malware to other cybercriminals or as a service offering (*malware-as-a-service*).
- **As a means of revenge or to carry out a personal agenda**. For example, Brian Krebs of Krebs on Security was struck by a big DDoS attack in 2016 after having talked about a DDoS attacker on his blog.
- **To carry out a political or social agenda**. Nation-state actors (like state-run hacker groups in China and North Korea) and hacker groups such as Anonymous are a perfect example.
- **As a way to entertain themselves**. Some cybercriminals enjoy victimizing others.

Obviously there are also reasons for non-malicious actors to create and/or deploy some types of malware too - for example it can be used to test a system's security.

2.1.2 Malware types

There are numerous different ways of categorizing malware; one way is by *how* the malicious software spreads. Another one is by what it *does* once it has successfully infected its victim's computers (i.e. what is its payload, how it exploits or makes the system vulnerable).

By how they spread

Terms like *trojan*, *virus* and *worm* are commonly used interchangeably to indicate generic malware, but they actually describe three subtly different ways malware can infect target computers [5]:

- ***Trojan horse***. In broad terms, a *Trojan Horse*, commonly referred to as "Trojan", is any program that disguises itself as legitimate and invites the user to download and run it, concealing a malicious payload. The payload - malicious routines - may take effect immediately and can lead to many undesirable effects, such as deleting the user's files or further installing malicious or undesirable software. Trojan horses known as *droppers* are used to start off a worm outbreak, by "injecting" the worm into users' local networks. Trojans may hide in games, apps, or even software patches, or they may rely on social engineering and be embedded in attachments included in phishing emails.

  Trojan horses cannot self-replicate. They rely on the system operators to activate. However, they can grant the attacker remote access permitting him to then perform any malicious activity that is in their interest. Trojan horse programs can affect the host in many different ways, depending on the payload attached to them [17].

  ***Example**: **Emotet** is a sophisticated banking trojan that has been around since 2014. It is hard to fight Emotet because it evades signature-based detection, is persistent, and includes spreader modules that help it propagate. The trojan is so widespread that it is the subject of a US Department of Homeland Security alert, which notes that Emotet has costed US state, local, tribal and territorial governments up to $1 million per incident to remediate.*
  *Other notable trojan strains include: **Trickbot**, **Ryuk**, **Sodinokibi**, **Mirai** and **SamSam**.*

- ***Virus***. The term "computer virus" is used for describing a self-replicating malicious program. When run it spreads to other executables (and/or boot sectors) by embedding copies of itself into those files. A virus, in fact, in order to spread from one computer to another, relies on the infected files possibly ending up, by some means or another, in the target system. Viruses are therefore passive. The mean of transport (file, media file, network file, etc.) is often referred to as the virus *vector*. Depending on how complex the virus code is, it may be able to modify its copies upon replication. For the transport of the infected files to the target system(s), the virus may rely on an unsuspecting human user (who for example uses a USB drive containing the infected file) or initiate itself the transfer (for example, it may send the infected files as an e-mail attachment) [1].

  Usually spread via infected websites, file sharing, or email attachment downloads, a virus will lie dormant until the infected host file or program is activated. Once that happens, the virus is able to replicate itself and spread through the system.

  Viruses may also perform other harmful actions other than just replicating, such as creating a backdoor for later use, damaging files, stealing information, creating botnets, render advertisements or even damaging equipment.

- ***Worm***. On the other hand, a worm is a self-replicating, active malicious program that exploits various system vulnerabilities to spread over the network. Particularly, it relies on vulnerabilities present in the target's operating system or installed software. Worms usually consume a lot of bandwidth and processing resources due to continuous scanning and may render the host unstable, sometimes causing the system to crash. Computer worms may also contain "payloads" to damage the target systems. Payloads are pieces of code written to perform various nefarious actions on the affected computers among which stealing data, deleting files or creating bots - which can lead the infected systems to become part of a botnet [17].

  ***Example**: **Stuxnet** was probably developed by the US and Israeli intelligence forces with the intent of setting back Iran's nuclear program. It was introduced into Iran's environment through a flash drive. Because the environment was air-gapped, its creators never thought Stuxnet would escape its target's network - but it did. Once in the wild, Stuxnet spread aggressively but did little damage, since its only function was to interfere with industrial controllers that managed the uranium enrichment process.*

These definitions lead to the observation that both viruses and trojans require *user intervention* to spread, whereas a worm spreads itself automatically. A virus, however, cannot execute or reproduce unless the app it has infected is running. This dependence on a host application makes viruses different from trojans, which require users to download them, and worms, which do not use applications to execute.

Furthermore, attackers can also install malware "manually" on a computer, either by gaining physical access to the target system or by using privilege escalation methods to obtain remote administrator access [6].

By what they do

There are a wide range of potential attack techniques used by malware, here are some of them:

- ***Adware***. *Adware*, or "Advertising supported software", is any software package which automatically plays, displays, or downloads advertisements to a computer. Some adware may also re-direct the user's browser to dubious websites. These advertisements can be in the form of a pop-up ads or ad banners that lure the user into making a purchase. The objective of the Adware is to generate revenue for its author. Adware, by itself, is harmless; however, some adware may come with integrated spyware, such as keyloggers, and other privacy-invasive software. This type of malware usually gets into users' computers from infected websites or unreliable download portals. It also may gain access by appearing to be an innocent ad or by attaching itself to another app, gaining access to the system when installing the apparently benevolent program.

Moreover, advertising functions are often integrated into or bundled with other software, which is often designed to note what Internet sites the user visits and to present advertising pertinent to the types of goods or services featured there. For example, an advertiser might use cookies to track the webpages a user visits to better target advertising. Adware is usually seen by the developers as a way to recover development costs, and in some cases it may allow the software to be provided to the user free of charge or at a reduced price. The income derived from presenting advertisements to the user may allow to motivate the developer to continue to develop, maintain and upgrade the software product. Conversely the advertisements may be seen by the user as interruptions or annoyances, or as distractions from the task at hand. Users sometimes unknowingly infect themselves with adware installed by default when they download and install other applications.

Some adware is also shareware, and so the word may be used as a term of distinction to differentiate between types of shareware software. What differentiates adware from other shareware is that it is primarily advertising-supported, like many free smartphone apps. Users may also be given the option to pay for a "registered" or "licensed" copy to do away with the advertisements.

Finally, there is a group of software (Alexa toolbar, Google toolbar, Eclipse data usage collector, etc.) that send data to a central server about which pages have been visited or which features of the software have been used. However, differently from "classic" malware, these tools document activities and only send data with the user's approval. The user may opt in to share the data in exchange to the additional features and services, or (in the case of Eclipse) as the form of voluntary support for the project. Some security tools report such loggers as malware while others do not. The status of the group is questionable. Some tools like PDF Creator are more on the boundary than others because opting out has been made more complex than it could be. However, PDF Creator is only sometimes mentioned as malware and is still subject of discussion.

***Example**: While there are hundreds of adware versions, some ot the most common examples include **Fireball**, **Appearch**, **DollarRevenue**, **Gator** and **DeskAd**. These adware strains often present themselves as a video, banner, full screen, or otherwise pop-up nuisance.*
*The adware called **Fireball** infected 250 million computers and devices in 2017, hijacking browsers to change default search engines and track web activity. However, the malware had the potential to become more than a mere nuisance. Three-quarters of it was able to run code remotely and download malicious files.*
Adware dominated the consumer threat detection category in 2019. Malwarebytes reports in its 2020 State of Malware Report that there were 54 million adware detections on Windows (24 million) and Apple (30 million) devices. Adware also captured the top business malware detection as well, increasing a incredible 463% from its 2018 detections.

- ***Backdoor***. A *backdoor*, also called Remote Access Trojan (RAT), is vulnerability deliberately buried in software's code that allows to bypass typical protection mechanisms, like credentials-based login authentication. In other words, it is a method of circumventing normal authentication procedures. Once a system has been compromised (by one of the mentioned methods, or in some other way), one or more backdoors may be installed in order to allow easier access in the future without alerting the user or the system's security programs. Backdoors may also be installed prior to malicious software, to allow attackers entry.

  Most device or software manufacturers place backdoors in their products intentionally and for a good reason. If needed, company personnel or law enforcement can use the backdoor to access the system when needed. Moreover, backdoors are sometimes also hidden in programs by intelligence services. For example, Cisco network routers, which process large volumes of global internet traffic, in the past were equipped with backdoors intended for US Secret Service use [7].

  ***Example**: Because backdoors are often intentionally built into products, the number of instances they've been used maliciously is numerous. in 2005, Sony BMG delivered millions of CDs with a rootkit that monitored listening habits and unintentionally left a backdoor to*

the device for cybercriminals. In 2017, more than 300,000 WordPress websites were affected by a malicious plug-in that allowed an attacker to place embedded hidden link on victim websites.

- ***Browser Hijacker***. A *Browser Hijacker*, also called "hijackware", noticeably changes the behaviour of the victim's web browser. This change could be sending the user to a new search page, slow-loading, changing the victim's homepage, installing unwanted toolbars, redirecting the user to sites he did not intend to visit, and displaying unwanted ads. Attackers can make money off advertising fees, steal information from users, spy, or redirect users to websites or apps that download more malware.

  Example*: A handful of notable browser hijackers are* ***Ask Toolbar****,* ***Conduit****,* ***CoolWebSearch****,* ***Coupon Saver****,* ***GoSave****, and* ***RockTab****. These browser hijackers typically come in the form of an added toolbar, and because they are often included in other software downloads, users rarely recognize their potential harm.*

- ***Bots/Botnet***. In general, *bots* (short for 'robots') are software programs designed to automatically perform specific operations. Bots were originally developed to automatically manage chat IRC channels - Internet Relay Chat: a text-based communication protocol appeared in 1989.
  Some bots are still being used for legitimate and harmless purposes such as video programming, video gaming, internet auctions and online contest, among other functions. It is however becoming increasingly common to see bots being used maliciously. Malicious bots can be (and usually are) used to form botnets. A botnet is defined as a network of host computers (zombies/bots) that is controlled by an attacker - the *bot-master* [17]. Botnets are frequently used for DDoS (Distributed Denial of Service) attacks, but there are other ways that botnets can be useful to cybercriminals:

  – **Brute force & credential stuffing attacks** - Bots can be used to carry out different types of brute force attacks on websites. They'll use a pre-configured list of usernames and passwords combinations on website login pages. The hope is that with enough tries, they'll get lucky and find a winning combination.
  – **Data and content scraping** - Scraping uses botnets as web spiders to comb through websites and databases to cull useful information they can use to undercut their competition. This could be site content, pricing sheets, or other useful information.
  – **Botnet-as-a-service opportunities** - Cybercriminals sometimes rent their botnets to all types of malicious users - including those who are less tech savvy. This service is also sometimes called malware-as-a-service, according to the International Botnet and IoT Security Guide 2020 by the Council to Secure the Digital Economy (CSDE). These bot armies essentially serve as mercenaries-for-hire to take down a target's servers and networks.
  – **Spambot** - A botnet can also be used to act as a spambot and render advertisements on websites.
  – **Malware distributor** - Finally Botnets can even be used for distributing malware disguised as popular search items on download sites.

  Websites usually guard themselves from bots using CAPTCHA tests that verify users as humans.

  Example*: In 2008, the* ***Kraken*** *botnet with 495,000 bots infected 10% of the Fortune 500 companies. This instance of a botnet attack was also the first where malware went undetected by anti-malware software. In 2016, one of the biggest (and worse) botnets in existence,* ***Mirai****, left most of the eastern U.S. with no internet. This massive botnet of compromised IoT devices, with over 600,000 of them, is responsible for some of the biggest DDoS attacks in recent years and exposed just how vulnerable IoT devices could be and led to the "IoT Cybersecurity Improvement Act" of 2020.*

- ***Crypto-miner***. Crypto-miners are a relatively new family of malware. Cybercriminals employ this type of malicious tools to mine Bitcoin and/or other bitcoin-alike digital currencies

on the target machine. The victim system's computing power is used for this, without the owner realising it. The mined coins end up in the attackers' digital crypto wallets.

In some cases, the use of crypto-miners may be deemed legal. For example they could be used to monetize websites, granted that the site operator clearly informed visitors of the use of such tools [7].

Most crypto-mining apps are usually categorized as PUAs - potentially unwanted apps - or, in rarer cases, as trojans. However, there is a more modern method of crypto-mining - crypto-jacking - that also works in browsers.

Moreover, according to ESET, most crypto-miners focus mostly on *Monero* as target crypto-currency because it offers anonymous transactions and can be mined with regular CPUs and GPUs instead of expensive, specialized hardware.

- ***File-less malware***. File-less malware is a type of memory-resident malware that uses legitimate code that already exists within the target computer or device to infect a computer. As the term suggests, it is malware that operates from a victim's computer memory, not from files on the hard drive, taking advantage of legitimate tools and software (known as "LOLBins") that exist within the system. File-less malware registry attacks leave no malware files to scan and no malicious processes to detect. Since there are no files to scan, it is harder to detect and remove than traditional malware; this makes them up to ten times more successful than traditional malware attacks. Furthermore, it also renders forensics more difficult because when the victim's computer is rebooted the malware disappears.

  ***Example**: **Astaroth** is a file-less malware campaign that spammed users with links to a .LNK shortcut file. When users downloaded the file, a WMIC tool was launched, along with a number of other legitimate Windows tools. These tools downloaded additional code that was executed only in memory, leaving no evidence that could be detected by vulnerability scanners. Then the attacker downloaded and ran a Trojan that stole credentials and uploaded them to a remote server.*

- ***Keylogger***. Keystroke logging (often called *keylogging*) is the action of tracking (or logging) the keys struck on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored. The collected information is stored and then sent to the attacker who can then use the data to figure out passwords, usernames and payment details, for example. There are numerous keylogging methods, ranging from hardware and software-based approaches to electromagnetic and acoustic analysis. Key loggers can be inserted into a system through phishing, social engineering or malicious downloads.

  Key logging have also legitimate uses, in fact it is often used by law enforcement, parents, and jealous or suspicious spouses. The most common use, however, is in the workplace, where employers monitor the employees' use of the company computers.

  ***Example**: a keylogger called **Olympic Vision** has been used to target US, Middle Eastern and Asian businessmen for business email compromise (BEC) attacks. Olympic Vision uses spear-phishing and social engineering techniques to infect its targets' systems in order to steal sensitive data and spy on business transactions. The keylogger is not sophisticated, but it is available on the black market for $25 so it is highly accessible to malicious actors.*
  *A strain of keylogger malware dubbed **LokiBot** notably increased in 2020. CISA reported that LokiBot "employs Trojan malware to steal sensitive information such as usernames, passwords, cryptocurrency wallets, and other credentials".*

- ***RAM Scraper***. *RAM scraper* malware, also known as *Point-of-Sale (POS)* malware, harvests data temporarily stored in a system's memory, also known as Random Access Memory (RAM). This type of malware targets POS systems like cash registers or vendor portals where an attacker can access unencrypted credit card numbers. While this sensitive payment data is only available for milliseconds before passing the encrypted numbers to back-end systems, attackers can still access millions of records.

  ***Example**: Since 2008, RAM scraping has been a boon for retailers. A handful of years later, the now infamous spyware dubbed **BlackPOS** led to the compromise of 40 million*

Target customers and 56 million Home Depot customers. Heading into the 2020s, a few notable RAM scraping malware families are ***FrameworkPOS****,* ***PoSeidon/FindStr****,* ***FighterPOS****, and* ***Canabak/Anunak****.*

- ***Ransomware****. Ransomware*, also known as "encryption" or "crypto" Trojan, is a malicious program that, after having infected a host or network, holds the system captive and requests a ransom from the host/network users. In particular it encrypts data on the infected system (or anyway locks down the system so that the users have no access) and only unblocks it when the correct password - decryption key - is entered. The latter is not given to the victims until after they have paid the ransom to the attacker. Messages informing the system user of the attack and demanding a ransom are usually displayed. Without the correct decryption key, it's mathematically impossible for victims to decrypt and regain access to their files.

  Digital currencies such as Bitcoin and Ether are the most common means of payment, making it difficult to track the cybercriminals. Moreover, there is no guarantee that payment will result in the necessary decryption key being handled back or that the decryption key provided will function properly. Additionally, some forms of ransomware threaten victims to publicize sensitive information within the encrypted data.

  Ransomware is one of the most profitable, and therefore one of the most popular, and dangerous kinds of malware programs of the past few years: Verizon's 2020 Data Breach Investigation Report shares that 27% if all malware-related incidents they tracked in 2019 involved ransomware. Companies, in particular, have recently received demands to pay millions to unblock critical services. The most well-known ransomware variants include WannaCry and Petya.

  The "Five Uneasy E's" of ransomware, according to Tim Femister - vice president of digital infrastructure at ConvergeOne - are:

  - **Exfiltrate**: Capture and send data to a remote attacker server for later leverage.
  - **Eliminate**: Identify and delete enterprise backups to improve odds of payment.
  - **Encrypt**: Use leading encryption protocols to fully encrypt data.
  - **Expose**: Provide proof of data and threaten public exposure and a data auction if payment is not made.
  - **Extort**: Demand an exorbitant payment paid via cryptocurrency.

  Example*: With vendors and organizations increasingly moving online, more data is at risk of exposure. Attackers know this and often take advantage of small to mid-sized oranizations with weaker network security, requesting an amount they know the organization can afford. Notable examples from the 2010s included* ***CryptoLocker****,* ***Locky****,* ***WannaCry****,* ***Hermes****,* ***GrandCrab*** *and* ***Ryuk****.*
  In year 2019, the city of Baltimore was hit by a type of ransomware named ***RobbinHood****, which halted all city activities, including tax collection, property transfers, and government email for weeks. This attack has cost the city more than $18 million. The same type of malware was used against the city of Atlanta in 2018, resulting in costs of $17 million.*
  In year 2020, a new ***CryCryptor*** *ransomware masquerading as COVID alert - the official COVID-19 contact-tracing app for Canada - was detected. Thankfully, ESET researchers were able to create a decryption tool to help those whose files were encrypted by the ransomware.*

- ***Rogue Security Software****.* Rogue Security Software is a from of ransomware or scareware. An attacker enabling this method tricks users into thinking their system or device is at risk. The malware program will present itself as a fake security tool to remove the problem at a cost. In actuality, the user pays up and the artificial security software installs more malware onto their system.

  Example*: Some of the most common rogue security software have come in spam campaigns and adware. however, a different infection vector for this malware is the technique known as* ***Black Hat SEO****. By following the most popular keywords on the internet through public records like Google Trends, attackers use malicious scripts to generate websites that appear legitimate.*

- ***Rootkit***. A *rootkit* is generally considered as a type of malicious software, or a collection of software tools, designed to remotely access or control a computer without being detected by users or security programs. Once a rootkit has been installed it is possible for the malicious party behind the rootkit to remotely execute files, record user activities, access/steal information, modify system configurations, alter software (especially any security software that could detect the rootkit), install concealed malware, mount attacks on other systems or control the computer as part of a botnet. Rootkit prevention, detection, and removal can be difficult due to their stealthy operation. Because a rootkit continually hides its presence, typical security products are not effective in detecting and removing rootkits. As a result, rootkit detection relies on manual methods such as monitoring computer behaviour for irregular activities, system file signature scanning, and storage dump analysis. Rootkits can be injected into applications, kernels, hypervisors, or firmware. They spread through phishing, malicious attachments, malicious downloads, and compromised shared drives.

  More recently, the term 'rootkit' has been frequently used also for denoting concealment routines in a malicious program. These routines are very advanced and complex and are written to hide the malware within the legitimate processes on the infected computer. In fact, once a malicious program is installed on a system, it is essential that it stays concealed, to avoid detection and disinfection. The same is true when a human attacker breaks into a computer directly. Techniques known as rootkits allow this concealment, by modifying the host's operating system so that the malware is hidden from the user. They can prevent a malicious process from being visible in the system's list of processes, or keep its files from being read.

  In an attempt to keep the user form stopping a malicious process, another is sometimes installed to monitor it. When the process is stopped (killed), another is immediately created. Modern malware starts a number of processes that monitor and restore one another as needed. In the event that a user running Microsoft Windows is infected with such malware (if they wish to manually stop it), they could use Task Manager's 'processes' tab to find the main process (the one that spawned the "resurrector" process(es)"), and use the 'end process tree' function, which would kill not only the main process, but the "resurrector(s)" as well, since they were started by the main process. Some malware programs use other techniques, such as naming the infected file similarly to a legitimate or trustworthy file to avoid detection in the process list.

  Traditionally, rootkits can install themselves in kernel level, although some sources state that they can install themselves all the way up to user level. This means that they can get as much (or as little) access as necessary.

  There are different types of rootkits, which are typically categorized by the reach of the system they affect:

  - User-level/application level rootkits - These rootkits can alter security settings, allowing the attacker to replace executables and system libraries and modify interface behaviour.
  - Kernel-level rootkits - These rootkits alter the very core of the system, the kernel. Resembling device drivers or loadable modules, they operate at the same security level as the OS, giving the appearance of credibility.
  - Hardware/firmware rootkits - Firmware is often use by organizations, however, their persistent presence in the router, network card, hard drive, or BIOS makes detecting it difficult if used maliciously.
  - Bootkit rootkits - A type of kernel-mode rootkit infecting boot functionality during computer start-up, subverting the kernel upon powering on.
  - Virtualization rootkits - Also known as a hypervisor, the rootkit hosts the target OS as a virtual machine (VM). It can forgo modifying the kernel and subvert the OS.

***Example**: **Zacinlo** infects systems when users download a fake VPN app. Once installed, Zacinlo conducts a security sweep for competing malware and tries to remove it. Then it opens invisible browsers and interacts with content like a human would - by scrolling, highlighting and clicking. This activity is meant to fool behavioural analysis software. Zacinlo's*

payload occurs when the malware clicks on ads in the invisible browsers. This advertising click fraud provides malicious actors with a cut of the commission.

- ***Scareware***. Scareware is a generic term for malware that uses social engineering to frighten a user, inducing him into thinking their system is vulnerable or has been attacked. The objective is to convince the user to install a specific software. However, in reality no danger has actually been detected: it is a scam. The attack succeeds when the user purchases unwanted - and potentially dangerous - software in an attempt to eliminate the "threat". Generally, the suggested software is additional malware or allegedly protective software with no value whatsoever [7].

  Some versions of scareware act as a sort of shadow version of ransomware; they claim to have taken control of the victim's system and demand a ransom. However they are actually just using tricks - such as browser redirect loops - to fool the victim into thinking they have done more damage than they really have [6].

- ***Spyware***. *Spyware* is a type of malicious software that uses functions in the infected host's operating system with the aim of spying on the user activity. More specifically it collects small pieces of information about users, like for example credit card details and passwords, without their knowledge. The information gathered is then sent back to the cybercriminal(s) responsible for it. The presence of spyware is typically hidden from the user, and can be difficult to detect.

  While the term spyware suggests software that secretly monitors the user's computing, the functions of spyware extend well beyond simple monitoring. Spyware programs can collect various types of personal information, such as Internet surfing habits and sites that have been visited, but can also interfere with user control of the computer in other ways, such as installing additional software and redirecting Web browser activity. Spyware is known to change computer settings, resulting in slow connection speeds, different home pages, and/or loss of Internet connection or functionality of other programs. They spread by attaching themselves to legitimate software, Trojan horse or even taking advantage of known software vulnerabilities. In an attempt to increase the understanding of spyware, a more formal classification of its included software types is provided by the term *privacy-invasive software*.

  One of the most common ways that *spyware* is distributed is a Trojan horse, bundled with a piece of desirable software that the user downloads from the internet. When the user installs the software, the spyware is installed alongside. Spyware authors who attempt to act in a legal fashion may include an end-user licence agreement that states the behaviour of the spyware in loose terms, which the users are unlikely to read or understand.

  Classification of code as spyware (or sometimes browser cookies as "tracking" cookies) can be controversial. Often the software is installed by the user knowing that some amount of monitoring will take place (Users generally agree to this activity to get free software and it is often associated with music and video sharing). Some such software allows the user to turn off the monitoring, assuming they are aware of it and can find instructions for disabling it. Anti-spyware is usually part of anti-virus programs.

  Law enforcement, government agencies and information security organizations often use spyware to monitor communications in a sensitive environment or during an investigation. Spyware is however also available to private consumers, allowing them to spy on their employees, spouse and children [8].

  Example*: Spyware often comes in the form of adware, trojans, keyloggers, and rootkits. Some of the best-known spyware strains include* ***CoolWebSearch****,* ***Gator****,* ***Internet Optimizer****,* ***TIBS Dialer****, and* ***Zlob****. For example,* ***CoolWebSearch*** *used Internet Explorer vulnerabilities to direct traffic to advertisements, infect host files, and rewrite search engine results.*
  One recent example of spyware is ***DarkHotel****, which targeted business and government leaders using hotel Wi-Fi. It used several types of malware in order to gain access to the systems belonging to specific powerful people. Once that access was gained, the attackers installed keyloggers to capture their targets passwords and other sensitive information.*

Other cyber-threats

Other cyber threats which are not strictly malware are, for example:

- ***Bug***. In the context of software, a bug is a flaw in a segment of code which produces an undesired outcome. These flaws are usually the result of human error and typically exist in the source code or compilers of a program. Minor bugs only slightly affect a program's behaviour and as a result can go for long periods of time before being discovered. More significant bugs can cause crashing or freezing. All software has bugs, and most go unnoticed or are mildly impactful to the user. Security bugs, however, are the most severe type of bugs and can allow attackers to bypass user authentication, override access privileges, or steal data. Bugs can be prevented with developer education, quality control, and code analysis tools.

- ***Malvertising***. Malvertising is the use of legitimate ads or ad networks to covertly deliver malware to unsuspecting users' computers. For example, a cybercriminal might pay to place an ad on a legitimate website. When a user clicks on the ad, code in the ad either redirects them to a malicious website or installs malware on their computer. In some cases, the malware embedded in an ad might execute automatically without any action from the user, a technique referred to as a "drive-by-download".

- ***Phishing***. While not being really a malware type, Phishing is a type of social engineering attack commonly used to perform cyber attacks. Phishing, and social engineering in general, is a type of email attack that attempts to trick users into divulging passwords (or anyway personal and financial information), downloading a malicious attachment or visiting a website that installs malware on their system. Phishing is successful since the emails sent, text messages and web links created, look like the are from trusted sources.

  Some are highly sophisticated and can fool even the most savvy users. Especially in cases where a known contact's email account has been compromised and it is then used to spread phishing attacks or malware such as worms. Others are less sophisticated and simply spam as many emails as they can with a message about 'checking your bank account details', for example.

  More targeted efforts at specific users or organizations are known as *spear phishing*. Because the goal is to trick the user, attackers will research the victim to maximise trick potential.

  There are different types of Phishing. Here are mentioned some of them:

  - *Deceptive Phishing* - The most common type. It uses an email headline with a sense of urgency from a know contact. This attack blends legitimate links with malicious code, modifies brand logos, and evades detection with minimal content.
  - *Spear Phishing* - As noted, spear phishing targets specific users or organizations by exploring social media, recording out-of-office notifications, compromising API tokens, and housing malicious data in the cloud.
  - *Whaling* - Even more targeted than spear phishing, whaling targets chief officers of an organization by infiltrating the network, exposing the supply chain, and following up the malicious email with a phone call to give legitimacy.
  - *Vishing* - Targeting victims over the phone, vishing is the use of Voice over Internet Protocol (VoIP), technical jargon, and ID spoofing to trick a caller into revealing sensitive information.
  - *Smishing* - Smishing also targets phone users, but this one comes in the form of malicious text messages. Smishing attacks often include triggering the download of a malicious app, link to data-stealing forms, and faux tech support.
  - *Pharming* - Moving away from trying to trick users, pharming leverages cache poisoning against the DNS, using malicious email code to target the server and compromise web users' URL requests.

- ***Spam***. In IT security, spam is unwanted email. Usually, it includes unsolicited advertisements, but it can also have attempted fraud or links or attachments that would install malware on the victim's system. Most spam emails contain one or more of the following:
  - Poor spelling and grammar
  - An unusual sender address
  - Unrealistic claims
  - Links that look mighty risky

  Spam might be one of the most universally understood forms of malicious attacks. As billions of users enable email for their everyday lives, it makes sense that malicious actors try to sneak into their inbox. Some of the most common types of spam emails include fake responses, PayPal, returned mail, and social media. All of which are disguised as legitimate but contain malware.

General considerations on malware types

Malware samples are usually categorised both by a means of infection and a behavioural category: for instance, WannaCry is a ransomware worm.
Moreover, a particular piece of malware may have various forms with different attack vectors: e.g., the banking malware called *Emotet* has been spotted in the wild as both a trojan and a worm [6].
Finally, many instances of malware fit into multiple categories: for example Stuxnet is both a worm, a virus and a rootkit.

Additionally, in recent years, also ***mobile devices-targeted attacks*** have grown popularity more and more. In fact, among the huge amount of available apps, an increasing amount are not desirable; and the problem is even more acute with third-party app stores. While app store vendors try to prevent malicious apps from becoming available, some inevitably slip through. These mobile malware threats are as various as those targeting desktops and include Trojans, Ransomware, Advertising click fraud and more. They are mostly distributed through phishing and malicious downloads and are a particular problem for jail-broken phones, which tend to lack the default protections that were part of those devices' original operating systems.

Mobile-malware example: Triada *is a rooting Trojan that was injected into the supply chain when millions of Android devices shipped with the malware pre-installed. Triada gains access to sensitive areas in the operating system and installs spam apps. The spam apps display ads, sometimes replacing legitimate ads. When a user clicks on one of the unauthorized ads, the revenue from that click goes to Triada's developers.*

2.1.3 Malware History

Malware history began in the 1960s. Then, hackers used to design computer viruses mainly for fun, as an exciting prank/experiment; their creations would generally display harmless messages and then spread to other computers [9]. There are numerous examples of malware created at that time within a laboratory setting: for example the ***Darwing game*** in 1962, ***Creeper*** in 1971, ***Rabbit Virus*** in 1974 and ***Pervading Animal*** in 1975.

In particular, the malware called ***Creeper*** was designed to infect mainframes on ARPANET. The program did not alter the machines' functions, nor it stole or deleted data. It only displayed the message "I'm the creeper: Catch me if you can" while illegitimately spreading from one mainframe to another. This malware was later upgraded with the ability to self-replicate and became the first known computer worm [10].

In the early 1980s, the concept of malware caught on in the technology industry, and numerous examples of viruses and worms appeared both on Apple and IBM personal computers. With the introduction of the World Wide Web and the commercial internet in the 1990s it eventually became widely popularized, so much that Yisreal Radai coined the term **malware** in 1990.

The previously mentioned 1960s and 1970s malware were all kept within a laboratory environment and never managed to escape to the wild. ***Elk Cloner*** (1981) was the first known virus to have been able to escape its creation environment. Then, following the success of that prank gone wild, the first Microsoft PC virus, called ***Brain***, was created in 1986. Again, like *Elk Cloner*, Brain was mostly annoying rather than harmful, but it was also the first known virus capable of concealing its presence on the disk thus evading detection. In 1988 the first worm, called ***Morris*** worm, an experimental, self-propagating, self-replicating program was released on the internet [17]. In 1988 made its appearance also the first example of intentionally harmful virus, the ***Vienna*** virus, which encrypted data and destroyed files. This led to the creation of the first antivirus tool ever [9].

In the following decades malware has evolved both regarding its complexity and malware sample numbers.

The growth in malware complexity can be divided 5 different malware generations [17]:

- *First generation*: (DOS Viruses) malware mainly replicate with the assistance of human activity
- *Second generation*: malware self-replicate without help and share the functionality characteristics of the first generation. They propagate through files and media.
- *Third generation*: malware utilise the capabilities of the internet in their propagation vectors leading to big impact viruses.
- *Fourth generation*: malware are more organization-specific and use multiple vectors to attack mainly anti-virus software or systems due to the commercialisation of malware.
- *Fifth generation*: malware is used in cyberwarfare and the now popular malware as-a-service makes its appearance.

Each jump in generation is linked to an increase in malware complexity and more propagation vectors being available. Newer generations of malware always re-utilise older techniques while introducing newer ones. Finally newer generations are more and more evasive due to the commercial value in having access to exploited systems.

Here is an overview of the most famous malware or malware-related events in recent history:

- ***Melissa*** (1999) - This was a mass-mailing macro virus released in 1999. As it was not a stand-alone program, it was not classified as a worm. It targeted Microsoft Word and Outlook-based systems, and created considerable network traffic. The virus would infect computers via Email, the email being titled "Important Message From", followed by the current username. Upon clicking the message, the body would read: "Here's that document you asked for. Don't show anyone else ;)." Attached was a Word document titled "list.doc" containing a list of pornographic sites and accompanying logins for each. It would then mass mail itself to the first 50 people in the user's contact list and then disable multiple safeguard features on Microsoft Word and Microsoft Outlook.
- ***ILOVEYOU*** (2000) - Sometimes referred to as *Love Bug* or *Love Letter for you*, spread like wildfire in year 2000. It is a computer worm that infected over ten million Windows personal computers when it started spreading as an email message with the subject line "ILOVEYOU" and the attachment "LOVE-LETTER-FOR-YOU.txt.vbs". That file extension ('vbs') was most often hidden by default on Windows computers of the time, leading users to think it was a normal text file. Opening the attachment activated the Visual Basic script. The worm inflicted damage on the local machine, overwriting random types of files, and sent a copy of itself to all addresses in the Windows Address Book used by Microsoft Outlook. This made it spread much faster than any other previous email worm.
- ***SQL Slammer*** (2003) - This malware, exploiting a buffer overflow bug in Microsoft's SQL Server, caused a denial of service on some Internet hosts and dramatically slowed general Internet traffic. It spread rapidly, infecting most of its 75,000 victims within ten minutes.

- ***MyDoom*** (2004) - Also known as *W32.MyDoom@mm*, *Novarg*, *Mimail.R* and *Shimgapi*, it is a computer worm affecting Microsoft Windows. It became the fastest-spreading e-mail worm ever, exceeding previous records set by the Sobig worm and ILOVEYOU, a record which as of 2021 has yet to be surpassed.

- ***Storm botnet*** (2007) - Also known as *Storm worm botnet*, *Dorf botnet* and *Ecard malware*, it is a remotely controlled network of "zombie" computers (or "botnet") that have been linked by the Storm Worm, a Trojan horse spread through e-mail spam. At its height in September 2007, the Storm botnet was running on anywhere from 1 million to 50 million computer systems, and accounted for 8% of all malware on Microsoft Windows computers. Finally it infected about 10 million computers in 9 months in 2007.

- ***Koobface*** (2008) - It was a network worm that attacked Microsoft Windows, Mac OS X, and Linux platforms. This worm originally targeted users of networking websites like Facebook, Skype, Yahoo Messenger, and email websites such as GMail and Yahoo Mail. It also targeted other networking websites, such as MySpace, Twitter, and it could infect other devices on the same local network.

- ***Conficker*** (2008) - Also known as *Downup*, *Downadup* and *Kido*, it was a computer worm targeting the Microsoft Windows operating system. It used flaws in Windows OS software and dictionary attacks on administrator passwords to propagate while forming a botnet. The Conficker worm infected over 15 million Windows systems including government, business and home computers in over 190 countries, making it the largest known computer worm infection since the 2003 *Welchia*.

- ***Zeus*** (2007-2009) - Also known as *ZeuS*, or *Zbot*, it was a Trojan horse malware package that ran on versions of Microsoft Windows. While it could be used to carry out many malicious and criminal tasks, it was often used to steal banking information by man-in-the-browser keystroke logging and form grabbing. It was also used to install the *CryptoLocker* ransomware. Zeus spread mainly through drive-by downloads and phishing scams. First identified in July 2007 when it was used to steal information from the United States Department of Transportation, it became more widespread in March 2009.

- ***Stuxnet Worm*** (2010) - It was an extremely sophisticated worm that infected computers worldwide. However, in reality, it mostly harmed Iranian nuclear facility at Natanz, where it damaged uranium-enrichment centrifuges. Supposedly it was built with that objective by U.S. and Israeli intelligence agencies [6].

- ***CryptoLocker*** (2013) - It is considered as the first widespread ransomware attack. It targeted computers running Microsoft Windows and it propagated via infected email attachments, and via an existing *Gameover ZeuS* botnet. When activated, the malware encrypted certain types of files stored on local and mounted network drives using RSA public-key cryptography, with the private key stored only on the malware's control servers. The malware then displayed a message which offered to decrypt the data if a payment (through either bitcoin or a pre-paid cash voucher) was made by a stated deadline, and it threatened to delete the private key if the deadline passes. Its code now keeps getting repurposed in similar malware projects.

- ***Mirai*** (2016) - First malware to scan the Internet of Things (IoT) - such as IP cameras and home routers - vulnerable devices and used them to perform DDoS attacks on various sites. The Mirai botnet was first found in August 2016 and has been used in some of the largest and most disruptive distributed denial of service (DDoS) attacks, including an attack on computer security journalist Brian Krebs' web site. The source code for Mirai was published on Hack Forums as open-source. Since the source code was published, the techniques have been adapted in other malware projects.

- ***Petya*** *and* ***NotPetya*** (2016-2017) - These malware attacks spread globally, however their damages particularly targeted Ukraine, where the national bank was hit. The Petya ransomware family caused an estimated $10 billion in damages worldwide [9]. *Petya* targeted Microsoft Windows-based systems, infecting the master boot record to execute a payload

that encrypted a hard drive's file system table and prevented Windows from booting. It subsequently demanded that the user make a payment in Bitcoin in order to regain access to the system. Variants of Petya were first seen in March 2016, which propagated via infected e-mail attachments. In June 2017, a new variant of Petya was used for a global cyberattack, primarily targeting Ukraine. The new variant propagated via the EternalBlue exploit, which was generally believed to have been developed by the U.S. National Security Agency (NSA), and was used earlier in the year by the WannaCry ransomware. This new version was called *NotPetya* to distinguish it from the 2016 variants, due to these differences in operation.

- ***WannaCry*** (2017) - WannaCry is considered one of largest ransomware attack in history. It targeted computers running the Microsoft Windows operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency. It propagated through EternalBlue, an exploit developed by the United States National Security Agency (NSA) for older Windows systems. EternalBlue was stolen and leaked at least a year prior to the attack. While Microsoft had released patches previously to close the exploit, much of WannaCry's spread was from organizations that had not applied these, or were using older Windows systems that were past their end-of-life. The attack was halted within a few days of its discovery due to emergency patches released by Microsoft and the discovery of a kill switch that prevented infected computers from spreading WannaCry further. It spread infecting systems at a terrifying rate of 10,000 PCs per hour [9]. The attack was estimated to have affected more than 200,000 computers across 150 countries, with total damages ranging from hundreds of millions to billions of dollars.

- ***Equifax data breach*** (2017) - In 2017, a very difficult year for cybersecurity, Hackers managed to crack Equifax - one of the four major credit reporting bureaus. This event is considered one of the most devastating data breaches in history. Hackers were able to access the personal data - such as credit card numbers, load and debt info, bank account details, birthdays, among others - of 143 million people [9].

- ***Emotet*** (2018) - This malware, also known as *Heodo*, was first detected in 2014 and deemed one of the most prevalent threats of the decade. First versions of the Emotet malware functioned as a banking trojan aimed at stealing banking credentials from infected hosts. Throughout 2016 and 2017, Emotet operators, updated the trojan and reconfigured it to work primarily as a "loader," a type of malware that gains access to a system, and then allows its operators to download additional payloads. Second-stage payloads can be any type of executable code, from Emotet's own modules to malware developed by other cybercrime gangs. Initial infection of target systems often proceeds through a macro virus in an email attachment. The infected email is a legitimate-appearing reply to an earlier message that was sent by the victim. It has been widely documented that the Emotet authors have used the malware to create a botnet of infected computers to which they sell access in Malware-as-a-Service. Emotet is known for renting access to infected computers to ransomware operations, such as the Ryuk gang. In 2020, Emotet campaigns were detected globally, infecting its victims with TrickBot and Qbot, which are used to steal banking credentials and spread inside networks. In January 2021, international action coordinated by Europol and Eurojust allowed investigators to take control of and disrupt the Emotet infrastructure.

- ***LockerGoga*** (2019) - This is a new ransomware family that has been detected attacking industrial companies, severely compromising their operations. It has the ability to spawn different processes in order to accelerate the file encryption in the system. The file-encrypting malware's entrance to the scene began when it was allegedly involved in attacking an engineering consulting firm based in France.

- ***Ryuk*** (2019-2020) - This is a type of ransomware known for targeting large, public-entity Microsoft Windows cybersystems. It typically encrypts data on an infected system, rendering the data inaccessible until a ransom is paid in untraceable bitcoin. Ryuk is believed to be used by two or more criminal groups, most likely Russian, who target organizations rather than individual consumers.

- ***COVID-19 related attacks*** (2020) - In 2020, many cybercriminals shamelessly took advantage of the people's fear of coronavirus during the COVID-19 pandemic through COVID-19 related phishing scams. Using fake communications, for example spoofing the World Health Organization, attackers deployed malware and got access to targets' sensitive information among other nefarious actions [9].
  Another COVID-19 attack was that of a malicious Android app called *CovidLock*, which claimed to be a real-time coronavirus outbreak tracker but instead was a ransomware that attempted to trick the user into providing administrative access on their device and then locked it requesting a ransom.

2.2 Obfuscation and detection evasion

From the creation of the first malware in 1970 [11], there has been a strong competition between attackers and defenders. To defend from malware attacks, anti-malware groups have been developing increasingly complex (and clever) new techniques. On the other hand, malware developers have conceived and adopted new tactics/methods to avoid the malware detectors.

The first type of anti-malware tools were mostly based on the assumption that malware structures do not change appreciably during time. In fact, initially, the malware machine code was completely unprotected. This allowed analysts to exploit opcode sequences to recognise specific malware families. Recently, however, a big advancement led to the so-called "second generation" malware [12] which, to evade such opcode signatures, employs several obfuscation techniques and can create variants of itself. This posed a challenge to anti-malware developers.

Obfuscation is a technique that generally makes programs harder to understand [13], both for humans and automatic tools. To do so, it converts a software to a new, structurally different, version while retaining the same functionality as the original. Originally, this technology was conceived for legitimate purposes to protect the intellectual property of software developers; however it has been widely exploited by malware authors to evade detection [14]. Particularly, in order to elude anti-malware scanners, malware can, using obfuscation techniques, evolve their body into new generations [15], which eventually can be even harder to disassemble and analyse.

The first time a malware has been recognised to exhibit detection avoidance behaviour was in 1986 with the *Brain* virus [16]. In fact, such malware managed to conceal the infected disk section whenever the user attempted to read it, forcing the computer to display clean data instead of the infected part. From that moment on, the ever increasing popularity of detection evasion techniques among malware writers has shown that malware survival has become the number one priority: the longer the malware remains undetected, the more harm it can do and the more profitable it is to its writer [17].

2.2.1 Anti-disassembly techniques

Anti-disassembly techniques use specially crafted code and/or data in a program to cause disassembly analysis tools to generate an incorrect program listing [18]. The attackers' usage of these techniques thus implies a time-consuming analysis for malware analysts, ultimately preventing the retrieval of the source code in a reasonable time.

Any executable code can be reverse engineered, but by armouring their code with anti-disassembly and anti-debugging techniques, attackers increase the skill level required by analysts. Furthermore, anti-disassembly techniques may also inhibit various automated analysis tools and heuristic-based engines which take advantage of disassembly analysis to identify or classify malware.

These techniques exploit the inherent weaknesses present in disassembler algorithms. Moreover, disassemblers, in order to work properly, make certain assumptions on the code being analysed. However, when these assumptions are not met, there is an opportunity for malware authors to deceive the analyst.

For example, while disassembling a program, sequences of executable code can have multiple disassembly representations, some of which may be invalid and obscure the real purpose of the program. Thus, the malware authors, in order to add anti-disassembly functionality to their creations, can produce sequences of code that deceive the disassembler into outputting a list of instructions that differs from those that would be executed [18].

There are two types of disassembler algorithms: linear and flow-oriented (recursive). The linear one is easier to implement, but it is also more simplistic and error-prone.

Linear Disassemblers

The *linear* disassembly strategy is based upon the basic assumption that the program's instructions are organized one after the other, linearly. In fact, this type of disassemblers iterates over a block of code, disassembling one instruction at a time, sequentially, without deviating. More specifically, the tool uses the size of the currently disassembled instruction to figure out what bytes to disassemble next, without accounting for control-flow instructions [18].

Linear disassemblers are easy to implement and work reasonably well when working with small sections of code. They introduce, however, occasional errors even with non-malicious binaries. The main drawback of this technique is that it blindly disassembles code until the end of the section, assuming the data is nothing but instructions packed together, without being able to distinguish between code, data and pointers.

In a PE-formatted executable file, for example, the executable code is typically contained inside a single ".text" section. However, for almost all binaries, this code section contains also data, such as pointer values. These pointers will be blindly disassembled and interpreted by the linear disassembler as instructions.

Malware authors can exploit this weakness of linear-disassembly algorithms implanting data bytes that form the opcodes of multi-byte instructions in the code section.

Flow-Oriented Disassemblers

The *flow-oriented* (or *recursive*) disassembly strategy is more advanced than the previous one and is, in fact, the one used by most commercial disassemblers like *IDA Pro* [18].

Differently form the linear strategy, the flow oriented one examines each instruction, builds a list of locations to disassemble (the ones reached by code) and keeps track of the code flow.

This implies that, if disassembling a code section we find a JMP instruction, this type of disassembler will not blindly parse the bytes immediately following the JMP instruction?s ones, but it will disassemble the bytes at the jump destination address.

This behaviour is more resilient and generally provides better results, but also implies a greater complexity.

In fact, while a linear disassembler has no choices to make about which instructions to disassemble at any given time, flow-oriented disassemblers have to make choices and assumptions, in particular when dealing with conditional branches and call instructions.
Particularly, in the case of conditional branches, the disassembler needs to follow both the false branch (most flow-oriented disassemblers will process the false branch of any conditional jump first) and the true one. In typical compiler-generated code there would be no difference in output if the disassembler processes first one branch or the other. However, in handwritten assembly code and anti-disassembly code, taking first one branch or the other can often produce different disassembly for the same block of code, leading to problems in analysis.

Chapter 3

Proposed Tool

Description of the proposed tool..

Chapter 4

Results

Results analysis..

Chapter 5

Conclusions

Qui si inseriscono brevi conclusioni sul lavoro svolto, senza ripetere inutilmente il sommario.

Si possono evidenziare i punti di forza e quelli di debolezza, nonché i possibili sviluppi futuri o attività da svolgere per migliorare i risultati.

Bibliography

[1] R. Sharp, "An introduction to malware." https://orbit.dtu.dk/en/publications/an-introduction-to-malware, 2017

[2] R. Moir, "Defining malware: Faq." https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10)?redirectedfrom=MSDN, 2009, Accessed: 2021-03-15

[3] NIST, "malware." https://csrc.nist.gov/glossary/term/malware, Accessed: 2021-03-15

[4] C. Crane, "What is malware? 10 types of malware and how they work." https://www.thesslstore.com/blog/what-is-malware-types-of-malware-how-they-work/, 2020, Accessed: 2021-03-15

[5] Symantec, "Difference between viruses, worms, and trojans." https://knowledge.broadcom.com/external/article?legacyId=TECH98539, 2019, Accessed: 2021-03-15

[6] J. Fruhlinger, "Malware explained: How to prevent, detect and recover from it." https://www.csoonline.com/article/3295877/what-is-malware-viruses-worms-trojans-and-beyond.html, 2019, Accessed: 2021-03-15

[7] MyraSecurity, "What is malware?." https://www.myrasecurity.com/en/what-is-malware/, Accessed: 2021-03-15

[8] McAfee, "What is malware?." https://www.mcafee.com/en-us/antivirus/malware.html, Accessed: 2021-03-15

[9] J. Regan, "What is malware? how malware works and how to prevent it." https://www.avg.com/en/signal/what-is-malware, 2019, Accessed: 2021-03-15

[10] B. Lutkevich, "malware." https://searchsecurity.techtarget.com/definition/malware, Accessed: 2021-03-15

[11] P. Szor, "The art of computer virus and defence", Symantec press, 1st ed., 2005, ISBN: 978-0-321-30454-4

[12] A. Sharma and S. K. Sahay, "Evolution and detection of polymorphic and metamorphic malwares: A survey", International Journal of Computer Applications, vol. 90, Mar 2014, pp. 7–11, DOI 10.5120/15544-4098

[13] A. Balakrishnan and C. Schulze, "Code obfuscation literature survey." http://pages.cs.wisc.edu/~arinib/writeup.pdf, 2005

[14] E. Konstantinou, "Metamorphic virus: Analysis and detection." https://www.ma.rhul.ac.uk/static/techrep/2008/RHUL-MA-2008-02.pdf, 2008, Technical Report of University of London

[15] I. You and K. Yim, "Malware obfuscation techniques: A brief survey", 2010 International Conference on Broadband, Wireless Computing, Communication and Applications, 2010, pp. 297–300, DOI 10.1109/BWCCA.2010.85

[16] E. Skoudis and L. Zeltser, "Malware: Fighting malicious code", Prentice Hall Professional, 2004, ISBN: 978-0-131-01405-3

[17] A. P. Namanya, A. Cullen, I. U. Awan, and J. P. Disso, "The world of malware: An overview", 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), 2018, pp. 420–427, DOI 10.1109/FiCloud.2018.00067

[18] M. Sikorski and A. Honig, "Practical malware analysis: The hands-on guide to dissecting malicious software", No Starch Press, 1st ed., 2012, ISBN: 978-1-59327-290-6