## Streszczenie

Celem niniejszej pracy jest przegląd i analiza ataków malware na systemy informatyczne, w szczególności zwracając uwagę na ataki powiązane z ransomware, metody ich rozprowadzania i sposób działania, ich potencjalne skutki dla ofiar ataków i ogólnie dla społeczeństwa, a także to, jak ataki te zmieniały się na przestrzeni lat. Głównym celem badawczym tej pracy jest porównanie bieżących rozwiązań dedykowanych do ochrony przed tym typem złośliwego oprogramowania, a także ukazanie obecnych trendów w procesie tworzenia takich rozwiązań jako próba powstrzymania cyberprzestępczości w tej niezwykle powszechnej dziedzinie.



**słowa kluczowe:** informatyka, ransomware, malware, wirusy komputerowe, ataki sieciowe, cyberbezpieczeństwo



## Abstract

Main subject of this work is overview and analysis of ransomware attacks on IT systems, focusing mostly on attacks related to ransomware, methods of its distribution and operation, its potential effects on victims and, in general, on society, but also how these attacks were changing over the years. Main research goal of this work is a comparison of common security measures dedicated to defending against this type of malicious software, as well as showing current trends in a process of creating such solutions as an attempt to prevent operations of cybercriminals using this special kind of malware.



**keywords:** computer science, ransomware, malware, computer viruses, network attacks, cybersecurity

# Table of Contents

# Introduction

Looking at the evolution of ransomware from a perspective of about 40 years, it is quite astonishing to compare the size, the speed of spreading and complexity of attacks taking place today, when compared to how innocuously it all started. Before 2000's, similar to ransomware techniques were used in a form of a harmless joke, to show one's hacking abilities and gain respect – in general almost merely entertainment purposes. Ransomware, which does not need to be introduced to anyone these days, is a type of malware that aims to encrypt victim's data and demand ransom for it. In the early days of the Internet (or even before), ransomware was in fact very simple. Most often, it was just encrypting all possible user's data and deleting any private keys so that that data would not be possible to recover. [1] Sometimes there were demands, very small if any, ransomware was using poor cryptography and usually with hardcoded keys. Attacks were not sophisticated; their difficulty was high only because there were relatively few sources to gain knowledge about hacking and therefore the process of creating such malicious program was undoubtedly painful and time-consuming.

These days, ransomware is counted in hundreds of families, thousands of LOC's (Lines of Code) [1], hundreds of thousands attack each year and billions of dollars lost by attacked companies all over the world. Despite growing awareness in cybersecurity, countless security solutions designed to protect end users from all possible threats, no one is safe now - not even companies that are responsible for securing the Internet. [2] Especially in the face of global Covid-19 situation, which has created a host of new opportunities for cyber criminals. At the end of 2019, ransomware attack struck, statistically, every 14 seconds [3], and most of the victims chose to pay hefty ransoms to prevent losing access to their intellectual property.

This year is another milestone for ransomware business. There has been a shift from encrypting files, and demanding ransom for it, to extracting victim's data and selling it on the darknet. It is yet another big trend in this industry that is now observed, which has one plain reason – money. Similarly, in years 2016/17, there was a shift from attacking individuals and demanding ransom ranging from about 50$-500$, to a completely new model of attacking biggest companies, that were expected to afford ransom of millions of dollars. Due to everyone being online these days, and most often working from home on a daily basis, cybercriminals have gained an opportunity to attack with thousands of entities simultaneously using a variety of attack vehicles. Because of that, more infections are possible to deliver, email gateways are overwhelmed with threats from all sides i.e., massive botnet-driven email campaigns, polymorphic malware that outpaces security vendors' ability to build new signatures,

malicious URLs [1] and malvertising campaigns – and this is just about one most popular way of delivering malware to victims.
The biggest problem that most of the companies are facing at the moment, is that they are still working on a base of "*IoCs of last campaign*[2]" after attacks, instead of hardening and using generic rules, such as solid backup strategy etc. to prevent these attacks in the first place. [4] Because of that, attackers have the advantage and certainty that their efforts will not come in vain.
Main focus of this dissertation is to review and analyze in detail current state and common characteristics of ransomware attacks in recent years, as well as determine which security product, which belongs to categories such as antivirus software, or specialized anti-ransomware tools, has most suitable characteristics and most efficiency fighting against ransomware. It is expected that, while testing known ransomware samples from public repositories, most of tested products will going to present high detection rate, although based on publicized reviews it is highly probable that specialized tools will demonstrate higher quality of detection and be more informative to the user.
This research is divided into three parts that correlate with each other. At the very beginning, attention is drawn to common malware types, their characteristics, and how do they relate to ransomware and its families. User is also familiarized with most widespread attack vectors, internal structure of ransomware, or the ways it communicates with attackers. In general, *Chapter 1. Ransomware characteristics and history* explores wide field of cyberattacks and its consequences, focusing on ransomware specifically.
*Chapter 2. Countermeasures* discusses most important security tools, algorithms, guidelines and in general, solutions, both from network and local perspective, that are supposed to protect users from attacks mentioned in Chapter 1. With that perspective, last chapter presents scenarios and results from testing chosen set of security products against arbitrarily chosen group of ransomware samples that were accessed via public repositories. Utmost effort is put here on giving user the knowledge on how ransomware does operate, how does it communicate with attackers, how did it become one of most serious threats in cyberspace, up to date, and what is most important – how to stop it from spreading.

[1] URL (Uniform Resource Locator) – unique identifier to access web resources. Subset of URI (Uniform Resource Identifier), however, both terms are often used interchangeably
[2] IoCS (Indicators of Compromise) of last campaign – indicates often used strategy/approach of many companies towards information security, which is best described as taking no efforts towards better security, right just until serious security incident takes place – after that observer trend is reversed

# 1. Ransomware characteristics and history

## 1.1. Familiarizing with malware

Malware, or malicious software, is a term used to describe any malicious program that was installed without the consent of the user and designed to undertake actions harmful to that user i.e., steal passwords, trick user into executing something that discloses his sensitive data, or his privacy. Most common effects of installing such software are [5]:

- *Crippling computer performance* – which might be indicated by unusually high consumption of resources such as CPU, memory, or other components. Much higher network traffic, which comes with unusual lags, might as well be a good indicator. These might be caused by a program used to mine cryptocurrencies without the user knowledge, participating in a larger botnet used to carry other attacks or just serving as a proxy for criminal groups to communicate with each other;
- *Leaking user personal data* – very difficult to observe, there might be very few indicators, which require some computer knowledge to inspect, and therefore it is much more dangerous for potential victim. Once attacker has access to the victim's machine, he can find passwords, private photos etc. leading to possible other attacks on that user;
- *Encrypting user personal data* – performing actions that might seemingly lock access to user personal data or completely encrypt it so that user is not able to access it without proper decryption key;

Each of mentioned effects are associated with certain types of malware, or mostly associated with one specific type, but sometimes attackers deploy programs that exhibit characteristics related to many different types, therefore conducting more complex attacks. In the following chapters, the reader will be familiarized with most common malware types shortly, and after that the main subject of analysis will be ransomware – describing its structure, characteristics, common attack vectors used to deploy it and all mechanisms that are relevant to the subject. This will accommodate the reader with basic understanding of how ransomware works, hopefully preventing from being victimized by such attacks.

## 1.2. Types of malware

It is important to think of mentioned types of malware more as a certain set of qualities that may or may not be used together to create a specific tool, rather than separate programs that are used independently.
Most notorious attacks these days are a complex combination of various mechanisms delivered in a sequence of stages that result in compromised device. These stages were presented on Figure 1. Delivery and foothold, as the first stages, are undertaken by several unrelated measures. It might be for instance, malicious link, e-mail attachment, message from somebody impersonating our peers on social media or any other way that malicious software reaches targeted system.

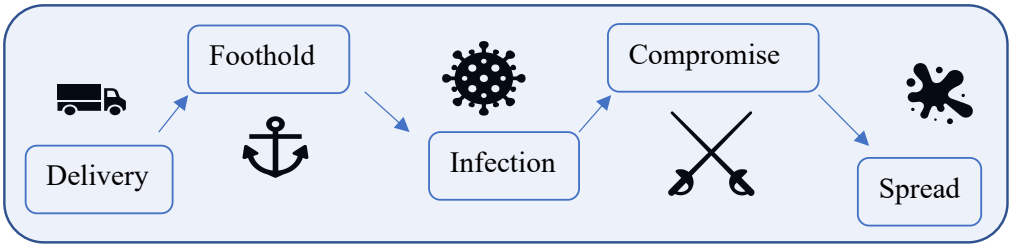


*Figure 1. Stages of cyberattack [6]*

After that, user must somehow interact with it and unknowingly execute malicious code after being tricked using social engineering techniques. Sometimes there is a case that no direct or any interaction is needed at all, because of unpatched system that contains exploitable vulnerabilities, or existing 0-day exploits[3] that even though are rare, are also possible. If all goes well up to this stage, and malicious activity is able to bypass system security mechanisms, such as antivirus software, sandboxing[4] if present etc. then the infection stage takes place. Once the system is compromised and attacker has a control over it, most common action is to replicate and spread malicious software over the local network looking for other targets, and the cycle repeats. As mentioned, specific characteristics of malicious programs are associated with specific categories of malware, of which most common are [6]:

- *Virus* – main characteristic of this type of malware is an infection mechanism, which is used to spread its presence in compromised system and other connected systems through the network, connected drives etc. This way it is much more difficult to recover the system to the pre-compromise state. In order to activate the infection mechanism, human interaction is usually necessary. First known virus was probably "Brain", developed in 1986. [5] It was designed as an aggressive prevention of pirating software, infecting boot sector of floppy disks used to distribute illegal copies of certain legitimate software;

[3] 0-day exploits – vulnerabilities in software that exist and are being actively exploited just before appropriate patch is released by company responsible for given vulnerable product
[4] More about this subject in second part of Chapter 3, discussing common countermeasures and protections

- *Worm* – this type of malware is designed to self-replicate and propagate through the system or systems connected through the network, originally consuming system resources and thus reducing performance. Presently, more common actions are stealing and deleting user data. Unlike viruses, it does not require human interaction in order to infect;
- *Adware* – as the name implies, this category is broadly used for advertisement purposes and is extremely common part of the interface of *freeware* software. Advertising is not the only effect of such infection – nowadays adware is more used to collect sensitive data about users through various tracking tools, having probably more and more common with *Spyware* category. Collected data is later being sold to third parties. Adware cannot be strictly categorized as malware, as in most cases it is part of legitimate software and is presented with small letters in very lengthy "Terms of service" displayed during installing programs. As nobody ever reads these and simply clicks "Agree" button, it cannot be technically deemed as malware, but rather as a *PUP* – Potentially Unwanted Software;
- *Spyware/trackware* – malware category co-existing with *Adware*, and used simply to collect data about compromised users;
- *Ransomware* – used in encrypting user's data and demanding ransom for their release, otherwise deleting decryption keys without the possibility of recovery. Some ransomware variants use techniques to lock out all access to victim's computer. They are less common since it sometimes makes it more difficult to pay the ransom. There are also cases where ransomware impersonates legitimate law enforcement agencies and suggests that victim has been caught on illegal activities; ransomware is referred to as cryptorworm;
- *Bots* – these are simply programs being commanded by attackers, that are usually connected in a larger network – called botnet (short of robot network), and used to carry out various attacks such as DDoS (Distributed Denial of Service), cryptomining, etc.;
- *Rootkits* – not categorized as malware per se, these are simply programs that allow remote control of a system by a third party. As one can imagine, they serve legitimate purposes, but when used together with other types of malware they can be an invaluable asset to the attacker;
- *Backdoors* – software/method used to facilitate unauthorized remote access to victim's system, without enabling normal access routines i.e., entering credentials; it can be installed by other types of malware;
- *Exploit kits* – collection of malicious programs (more precisely, exploits), that are used by attackers to compromise system and distribute other malware categories; such software was reportedly used with association to ransomware in numerous attacks;
- *Rootkits* – malicious program that is designed to enable unauthorized access to the system or parts of it, at the same time hiding its malicious activity; this process exploits known vulnerabilities of the system to gain control over it, often in order to install backdoors that allow to preserve access;

- *Trojan Horses* – or simply *Trojans,* this type of malware is designed to hide in plain sight by masquerading its malicious actions;
- *Cryptojacking malware* – programs abusing victim system's resources, in order to mine cryptocurrencies in unauthorized manner; often found on compromised websites of everyday use, such as public information portals, social services etc. Also delivered using spam campaigns; this form of malware is very popular, just as cryptocurrencies recently are, and is especially difficult to detect since it's disguised as a common, known process that might be run by authorized user. Today personal systems rarely reach 100% usage, making it almost impossible to detect by unsuspecting user if suddenly such malware starts executing – since it will usually claim very little computing power to remain undetected;
- *Downloaders* – used to download other malicious software, usually trojans, to victim's system;
- *Stalkerware, scareware, bossware, proctoring software* – broad category of software associated with different forms of electronic abuse [7] [8]; last mentioned category is in fact a legitimate service delivered by many companies around the world, which is developed to track students by their teachers in order to make sure they are present during lessons, or do not cheat during exams; however, such form of control has been meet with many protests, not only by students, because of many privacy threats that surveillance software is exposing students to [9];
- *APT (Advanced Persistent Threat)* – complex combination of different malicious categories mentioned above aiming at gaining unauthorized access to, or resulting in, complete compromise of victim's internal network, for extended period of time; such attacks are mainly focused on persistence and remaining undetectable, and because of that they are very well coordinated, most often state-sponsored [10];

## 1.3. Ransomware

Heart of this work is a profound analysis of ransomware attacks. Following sections combine general perspective from top-level point of view, including the evolution of this type of malware over the years, notorious ransomware families that caused a lot of harm, major differences between them and detailed perspective on structure of such attacks, covering techniques that are used by ransomware specifically to deploy and infect the system.
There is a one particular reason why malware in general was discussed in the first place. Ransomware as we know it rarely exists as a separate product that is used to encrypt victim's filesystem. In majority of cases, ransomware is in fact a complex composition of malicious software, which interacts with victim in a sequence of events – starting from infection, then system compromise and leading to encryption as a final stage. Ransomware payload, which purpose is to block access to victim's system or underlying information, is only a part of a bigger picture.

Ransomware has some distinct characteristics distinguishing it amongst other types. As opposed to other malware it does not need escalated privileges to execute, meaning

it does not interact with any special settings, does not try to exploit vulnerability. It is simply an operation on data that user has given it access to. The access can also be "granted" through malicious software such as exploit kits that exploit vulnerabilities of attacked systems. Finally, ransomware is probably the most interactive type of malware, as it demands certain actions to be taken by attacked user in return for access to their data or system. On the other hand, however, it still needs other types of malware in order to infect the system and propagate to other machines on local network. Ransomware can be categorized as [6]:

- *Locker ransomware* – locking out of the system, not encrypting any files;
- *Crypto ransomware* – its ultimate purpose is to encrypt user files and display a timer which, when expires, deletes master decryption key for that user, so that all the files are irreversibly lost. This category is especially dangerous to any public or government organizations, which are usually less prepared for such attack and hold a lot of sensitive data. In majority of such cases the only way out is to pay the ransom. Some cybercriminals also tend to threaten the victim that all their files will be published, unless the ransom demand is met (data extortion); Other common practice is deleting random files after small periods of time, in order to put time-pressure on the victim; the system cannot be turned off for any reason, simply because threat actors will delete victim's master decryption key in case ransomware installed on the system loses connection with attacker's servers;

Ransomware can be broadly categorized as a form of digital blackmail. It is intuitive association, as ransomware is all about demanding certain action from victim and threatening the victim if they have not met the demands in certain time. Interestingly, other most common types of digital blackmail include scareware, such as fake antivirus-like applications[5]. Another reported type of blackmail is called DDoS extortion. It's a category of blackmailing in which criminals threaten larger organizations with DDoS[6] attacks, and they promise to stop attacks if victim answers their demands. As for most companies that offer their service online, even a few moments of losing their availability usually costs a lot, so they often agree to attackers' demands in order to have their services left uninterrupted.

Although ransomware is a serious subject, a very common threat, which prevalence and advancements in recent years caused billions of dollars, there have been many mistakes made by ransomware authors leading to successfully recovering files without cybercriminals' cooperation. Just as any other software, ransomware is susceptible to any kinds of bugs, inconsistencies in design, that happen to any software in general. One of presentations on RSA Conference in 2017 was discussing that subject in detail. [11] According to the author of presentation, one of most common mistakes by ransomware creators is using either own cryptographic functions, weak cryptographic functions, or sometimes even system's cryptographic libraries, which may have its

[5] This type of malware is usually claiming that machine has been infected with some type of malware, which such antivirus can help to disinfect, for free.
[6] DDoS attacks – Distributed Denial of Service attacks, which aim to temporarily prevent legitimate users from accessing certain online service, by measures of putting high load on company's servers, usually by sending a number of requests from distributed set of nodes that belong to malicious network

own flaws. Many years ago, decryption keys were very often hardcoded into ransomware executables, and therefore were easy to recover by security analysts. Not only encryption process is proven to be troublesome for attackers to be developed properly, but sometimes the weak point is also a communication channel between deployed ransomware and C&C server. If, for example, ransomware was using unencrypted connection to inform attackers whether victim has issued the payment, the response from bitcoin wallet could be changed in order to trick ransomware into thinking, that the victim has already paid the ransom.

It is an interesting perspective to show that not only defenders make mistakes that are often very expensive. In any case, the war between attacking and defending sides is the main subject of this work, and proactive defense is proven to be the sometimes the best solution against the ever-changing malicious activities.

## 1.4. History of attacks

> *"Ransomware attacks have morphed from spray-and-pray phishing blasts to highly targeted and extremely damaging network-wide infections that can cause days or weeks of downtime for a whole organization"* [127]

Ransomware attacks originated in 1989 and have been the one of the most pervasive cyber threat since early 2000's. Since about year of 2016, share of attacks using ransomware has dramatically surged, leaving billions of dollars wasted because of only this category of cybercrime. According to Verizon [12], ransomware has moved from the 22$^{nd}$ most common variety of malware in 2014 to the most common variety in 2018. There have been more ransomware variants reported in merely last year, than from the first reported attack in 90's.

First ransomware ever was 1989 AIDS Trojan, which was distributed by Joseph L. Popp, Harvard University biologist, who targeted around 20 000 attendees of World Health Organization's International AIDS Conference. The ransomware was delivered via floppy disks, which masqueraded as informational bulletin for participants of that conference. After certain time the program revealed its malicious nature, encrypting all user's data on hard drive, demanding about 200$ paid to specific account in return for decryption key. [13]
According to [14], global market share is dominated by Windows and Android systems (in joined category of stationary and mobile systems), and therefore these systems were targeted specifically amongst all others. In 2013, there was the milestone year for ransomware, because there was reported first attack demanding payment in Bitcoin, and also there was a first mobile ransomware (dubbed "*Android Defender*") reported in the wild.

With growing popularity of ransomware in recent years, it was inevitable that loads of services assisting in automation of such attacks were created and became increasingly popular. With the use of such services, potential attacker is able to configure ransomware from scratch, with potentially zero knowledge of how it works.

[15] [16] Such novice is being instructed, step by step, what type of ransomware is the best choice for particular size of campaign, what size of ransom is recommended for particular targets, and how to deploy created ransomware and start getting profit from it. [17] It is also possible to configure own "ransom note" – entire process is smooth, elementary, and fully automatic. Such help is obviously not free – creators of such services earn much more money than their clients from attacks they launch. Such business has grown to the level of RaaS business model (Ransomware as a Service) – extremely lucrative branch of criminal industry. [18]

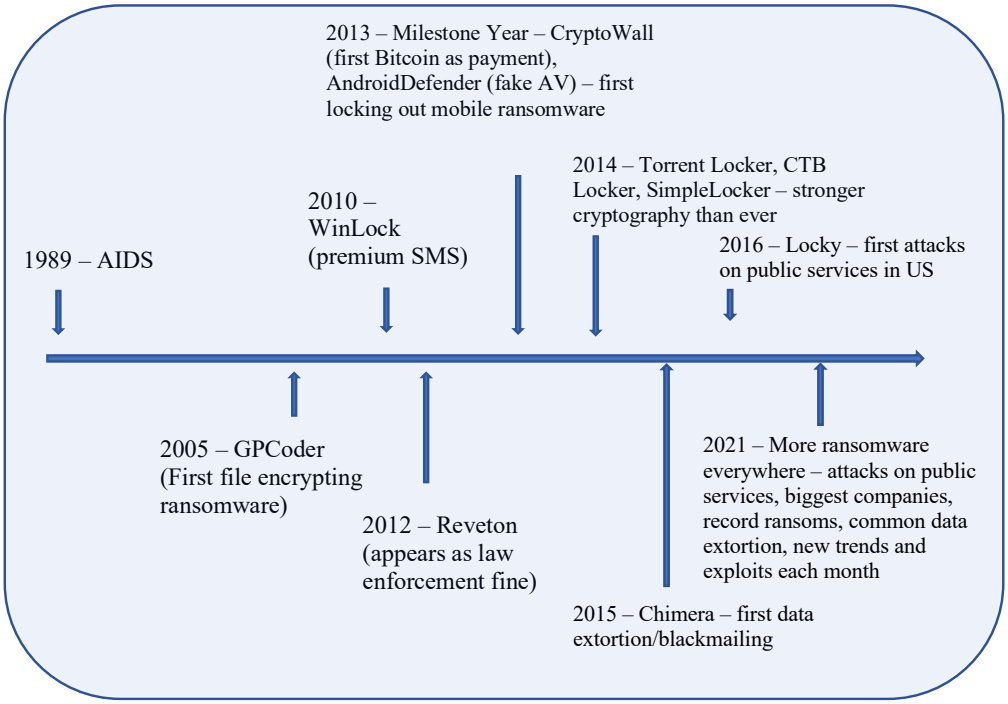


*Figure 2. History of ransomware attacks [3]*

Ransomware has changed in many ways, making it in general a lot harder to detect in initial stages of attack, or prevent infection and damage of the data on the system. It is highly adaptable model, which changed from simple encryption of data, to deleting metadata and all kinds of backups i.e., Microsoft Volume Shadow Copy and Restore Points (on Windows machines) as well, making it substantially harder to recover lost data by victims (without cooperating with the attacker. [19]
Finally, very recently another trend emerged in ransomware attacks. There has been observed a shift in modus operandi – from encrypting to extortion of data [20], as well as victimology – the attacks are now targeting bigger companies, demanding larger ransoms, whereas a few years ago most of the attacks were focused on private users. [21] In fact, in 2021 there was a record-breaking ransom targeted at Acer – about 50 million dollars. [22] Another important case that represents abovementioned shift in

cybercriminals interests' is attack on Apple. [23] Most important milestones in history of ransomware were visualized summarized on Figure 2.

## 1.5. Ransomware families

Most common classification method for ransomware is using their code signature, or in other words, a distinctive chain of actions represented by program, giving specific result (signature is based on multiple factors, both static – what characteristic strings particular executable contains or calculating hash sum of that executable and dynamic, i.e., how does it interact with the system). [6] Based on this, ransomware is called into different ransomware families. This section is a discussion of arbitrarily chosen ransomware families that represent most prominent characteristics often followed by their countless modifications.
These days, ransomware has the properties of a virus, worm, Trojan and rootkit to escape detection and defeat various countermeasures (such as sandboxes that help separating execution of files that were downloaded from the Internet from affecting the rest of the system).
Ransomware family is in fact rather a group of ransomware samples that exhibit similar function when they reach target system, use similar methods to compromise target, or simply have similar behavior.

According to Kaspersky [24] only in 2016 there were 44,287 new ransomware modifications. It would not be impossible to list all even most generic families, as it would take thousands of pages. On the other hand, there were ransomware types that became more notorious than others, so in order to build a general overview of differences between (subjectively) one of most important ransomware actors, a few of them will be listed here together with main characteristics and key differences.
Again, encrypting, or locking user's data, is in fact merely a small part of capabilities that such malware exhibits. Those families are in fact a complex set of programs used to infect, persist on victims' machines and propagate. In most cases, those general traits are being used to categorize as a specific ransomware family – hence, even though it uses exactly same piece of code for encrypting files as another family, they will not be categorized as one because other characteristics differ. A few examples of such families will follow below.

It is also usually the case that previously known ransomware comes as a brand-new family even though only a few minor characteristics were changed. Below are listed some of the most notorious ransomware families.

- *WannaCry* – used a vulnerability in SMB (Server Message Block) protocol to infect. Exploit[7] for this vulnerability, called Eternal Blue (CVE-2017-0144), was reportedly discovered by NSA, and leaked by unknown hacking group. First attacks were reported in 2017, and Microsoft had already released patches for this vulnerability. However, because updated were not always the most important thing for everybody, within a few days of the first attack WannaCry affected more than 200, 000 computers across 150 countries, with total damage reaching billions

[7] Piece of code in particular programming language, that exploits certain vulnerability in the system

of dollars. WannaCry, also known as WannaDecrypt0r or WannaCrypt was partially neutralized by a researcher that discovered a kill switch domain, which, registered in a DNS sinkhole [8] server, stopped the containment of attack. WannaCry used Windows encryption API in order to generate encryption keys. In some cases, Windows was not clearing prime numbers used to generate those keys, therefore allowing their recovery;

- *CryptoLocker* – used a general trojan targeting Windows (namely Zeus, or Zbot trojan), and first infections were reported back in 2014. It was propagating via an e-mail attachment in a seemingly legitimate e-mail. A ZIP file was hiding executable file with icon disguised as a PDF file. After executing, it was registering itself on a startup, and after successful computer restart was generating 2048-bit RSA keys and encrypting files with popular extensions. It is believed that CryptoLocker extorted approximately 3 million dollars from its victims;
- *Chimera* – ransomware family that did not exploit any nefarious vulnerability and was rather simple in structure. It was using spear phishing as the attack vector and propagated similarly to abovementioned CryptoLocker. Attacks originated in 2015. Two things were unique in this ransomware. Namely, it was utilizing a doxing/blackmailing techniques to put a time pressure on the victim in order to pay the ransom. These techniques were used by many different ransomwares since then, however Chimera was reportedly one of the first using P2P protocol in order to communicate with C&C server, instead of using *traditional* client-server architecture. This way, even if some of the nodes were taken by law enforcement, it was extremely difficult to take down all communication nodes;
- *CTB-Locker* – it stands for Curve-Tor-Bitcoin Locker. First part of this long name is taken from its use of ECC (Elliptic Curve Cryptography), in order to use much smaller key sizes and achieve sufficient level of security. It is actually one of the first ransomware which was using such strong cryptography. Another notorious characteristic of CTB-Locker was that it was the first multi-lingual ransomware – targeting users all around the world. It was using affiliate model, also known as RaaS as a form of campaign, meaning everyone was able to order a sample and infect whomever they please. First attacks were discovered in 2014;
- *TeslaCrypt* – ransomware family that was targeting files of online games specifically and was first discovered in 2015. It was resembling CryptoLocker in structure, spreading with the use of Angler (CVE-2015-3090) Adobe flash exploit. There were a few iterations that utilized more and more advanced cryptography. In 2016 developers of this ransomware decided to shut down the campaign and shared master private key, bringing end to this ransomware;
- *Petya* – ransomware that could be categorized as somewhat locking more than encrypting. It was delivered via common phishing campaigns and also offered as part of RaaS business. The demands were usually around 850$. When installed, it was encrypting MBR sector in NTFS, thus rendering Windows unusable. First attacks were seen in 2016. A year later, a variation of Petya was discovered, that

[8] DNS sinkhole is a server that gives false results (IP address) for given domain query. It might be used by attackers to trick users into visiting malicious website, but could also be used against attackers in the same way, as was in the case of WannaCry

was using Eternal Blue (*CVE-2017-0144*) and Eternal Romance (*CVE-2017-0147*) exploits for propagation, just like WannaCry ransomware. Several other improvements caused this new version of Petya being categorized as new ransomware family – called NotPetya.

NotPetya was developed to intentionally destroy all data in the system, not demanding any ransom for it. Because of that fact it was classified rather as "cyberweapon", not ransomware. Another similar ransomware that appeared at the time was PetrWrap. It used much of Petya code, and added some new functionality to the base; The second variant employs high quality cryptography functions, so that data encrypted by this ransomware was no different than after being cleaned by hard drive wiping tools with the exception that it could be recovered only if ransom was paid to attackers;

- *Ryuk* – first appeared in 2018, known for targeting large, public-entity networks with Windows servers. It was typically encrypting chosen data on the system. Ryuk was reportedly using Trickbot computer malware to install itself and gain control over infected systems, and Emotet trojan horse as an initial loader; Mostly used for tailored attacks, criminals using this ransomware usually gathered technical information about targets prior to attack and employed highly sophisticated social engineering tactics. Attacks used both mass spam campaigns and targeted vectors such as spear phishing and exploiting vulnerabilities in RDP software that victims were using. Ryuk is mostly associated with Lazarus APT group, and first appeared in August 2018. Demanding relatively high ransoms, it gathered about 4 million dollars in BTC just in the first half of active operations. After the infection, it needs to somehow escalate privileges and operate under administrator account, as opposed to most ransomware, which usually does not need any escalated privileges in order to work. Ryuk is written for both 64 and 32-bit architectures, and targets only stationary devices. It uses AES-256 and RSA-4096 for encrypting and sharing keys with attacker's server. Ryuk also deletes all shadow copies residing on Windows locally, making it impossible for user to recover without any external backup copies.
- *SimpleLocker* – one of the first ransomware targeting mobile devices, that was actually encrypting user's data residing on SD card of the smartphone. Discovered in 2013, and main attack vector was just disguising as legitimate application to be downloaded from Google's Play Store;
- *Locky* – Appeared in 2016, it is notorious for advanced anti-analysis and sandbox-escaping techniques. It was using an extensive file type list that was targeting during infection, which was causing havoc when hit enterprise systems. *Locky* was using RSA-2048 and AES-128 cryptographic functions, and encrypting all files on local drivers, removable drives and network shares. This ransomware was reportedly using phishing techniques, masquerading as ISP complaint notices, account verifications etc. document attachment, which once opened, prompted user to turn on macros. Those macros were downloading a malicious downloader (used to download *Locky*) and storing it in Windows `%localappdata%\Temp` folder. Another variant of *Locky* was spreading using malicious SVG file, which is known to allow dynamic content (JavaScript code that could download Locky). Once encrypted user's data, Locky was also deleting Shadow Copies. It is

believed that Locky originated from Russia, since it won't execute on machines located in Russia or having Russian language pack installed.

- *DearCry* – brand new ransomware, discovered in March 2021, targeting a notorious vulnerability for Microsoft Exchange servers that was discovered a year before (CVE-2021-26855) in order to compromise machines.
- *Cerber* - First appeared in 2016, widely known to be distributed via the RaaS business model. In that case, authors would make 40% of acquired ransom profit as a fee from their customer. Cerber was mostly distributed as part of phishing campaigns, usually bundled with some seemingly "free software" that contained exploit kits. *Cerber* was using RC4 and RSA algorithms for encrypting and sharing keys with C&C server. The extensions appended to encrypted files were usually `.cerber`, 4 random characters, etc. Cerber displays ransom note which gives the victim about 7 days to pay about $500 ransom in Bitcoin, which doubles after 7 days. Cerber was also able to fully operate offline, so disconnecting computer from its C&C server did not stop the encryption process. There were six variants of Cerber ransomware. At the end of 2016 it had already harvested around 2.3 million dollars. Typically to RaaS business, Cerber was very often updated with new features helping to evade new sorts of countermeasures.
- *SamSam* - First recorded attacks in 2016, it was leveraging legitimate tools from *Windows Sysinternals* [25] to analyze system for vulnerabilities. SamSam was deployed in massive attacks at the beginning, later moving to a more tailored attacks targeting high-profile organizations. Attacks with SamSam did not use social engineering tactics. Instead, this ransomware was targeting vulnerabilities in server applications (JBoos, FTP servers and trying to brute force weak password through RDP accounts in order to gain access to corporate networks). Such attacks were launched usually in the early morning or after midnight of local time, when IT administrators were off the guard. Once installed, it searches for all backups, volume shadow copies and deleting all of it in order to prevent recovery.

As mentioned, this is just a beginning of a very long list. Ransomware is constantly evolving, old families use new vulnerabilities and attack vectors, better cryptography, and use more targeted campaigns, instead of wide distribution model. [26] Other ransomware families that might have been listed here are RunExeMemory, Pysa, Liz Dharma, Rapid, Xorist, LockBit, Hakbit, SFile, PewPew, Stop, Sodinokibi or Maze, DMA Locker, CRyptXXX, CryptoWall. [27], [28] And the list goes and goes into a never-ending story. Other systems, such as MacOS were also targeted quite many times i.e., [29]. One thing that should be concluded in this section, is that all these families are relatively similar to each other and use similar techniques to spread and compromise machines. Because of that, and without considering special ransomware cases that were extremely unique, it is theoretically possible to develop such countermeasures that will be able to block most of such attacks. [30] However, as new ransomware emerges each hour, day, month and so on, it is essential to understand internal structure of this kind of malware in order to protect against it. This will be the purpose of the following chapters.

## 1.6. Ransomware structure

In general, there is no ready recipe for ransomware. There are great differences between known families in terms of how they proceed with the infection once system is compromised. In general, they can either encrypt data or block access to entire system (preventing user from logging into the machine) In the former case, attacker has also a few choices:

- *Encrypt entire file's contents* – this is a slow process, as it has a huge impact on the victim system's resources. It also takes time and could be interrupted by a suspecting user if he notices unusual load and turns off the machine. In this case, one must consider that turning off system while ransomware is encrypting data might result in decryption key lost forever and therefore some data might be unrecoverable. Usually, to speed up the process ransomware lists common file types and encrypts only most important ones;
- *Encrypting metadata directly connected to files* – this strategy is way faster because it encrypts only parts of the system, disorganizing the system's structure and therefore preventing access to files, without encrypting them. For instance, ransomware may encrypt MFT (Master File Table) on NTFS filesystem – a special data structure residing at early offset of each partition, and is responsible for specifying various attributes of files on that partition, such as path to the file, address of data on the disk etc. Without MFT, partition is just a bunch of data. Victim might just try to manually recover files based on their structure – each file has a header containing information regarding its type (Magic number), its size etc., and also underlying data. To manually recover these files, one must harvest drive contents byte by byte and save appropriate files on disk again. This might be very lengthy and painful process, however in some cases it might be successful and sufficient – take for instance, a user that has only lots of videos and photos on their system. It will disorganize meticulously organized photos into folders, subfolders etc. that were collected over the years – however all these data will be recovered and bunched without names into one folder with recovered data. Take, on the other hand a corporate user that has a lot of projects, configuration files, notes, data associated with projects etc. Organizing these data again after manual recovery is practically impossible;

In general, there is a specific time window between creating symmetric key and starting encrypting process up to finishing that process, sending encrypted symmetric key to C&C server and deleting symmetric key from the system. In that window, it is theoretically possible to find symmetric key by inspecting system's memory. This could be done by a similar technique to an attack called *Cold-boot attack* in which attacker freezes RAM chips**,** pulls them out of machine and immediately puts in a device that dumps their contents to a file. In principle, it might be possible to find symmetric key by scanning that file, and then recovering encrypted files. However, there is a huge risk that by doing so ransomware process gets interrupted, and symmetric key is lost forever before being sent to attacker's C&C.

Very interesting term associated with ransomware is *Kill Switch*. It is basically a simulated response to the ransomware that stops the system infection, tricking ransomware into *thinking* that the system is being encrypted. In some cases, it is achieved by creating a specific script in specific directory and putting a `readonly` flag in attributes of that script – so that when ransomware tries to create that script and start infection by itself, it will not be able to do so. In other cases, *kill switch* is usually achieved by creating crafted responses that look like real commands that ransomware would have received from C&C server.

### 1.6.1. Ransomware toolbelt

Even though every ransomware family is a unique piece of malware having own signature and specific behavior, there are certain functions and characteristics that are generally common amongst them. There are a few projects, which aim at creating ransomware that will serve educational purposes. One of first attempts [31] was ransomware called Hidden Tear. [32] According to its author, it is a simple *"ransomware-like file encrypter sample which can be modified for specific purposes"*. It is written in C# and targets Microsoft Windows systems. Probably due to its simplicity and efficiency, it has been used for illegal purposes [33] despite legal note issued by the author on the page of the project. Key elements of Hidden Tear are as follows:

- Encryption of files by specific directory, and according to the list of desired file extensions;
- Decryption of files accordingly;
- Key generation and exchange with C&C server;
- Creating ransom note with customizable message;
- Tiny size – increases portability;

Encryption is done using AES-256 with CBC mode. Author also claims that (as of 2015) this crypter is undetectable by most AV engines. As of 2021, however, it is not even possible to download repository as ZIP, as it's being blocked by antivirus (tested on ESET Antivirus, version 14.0.22.0 – see Figure 3).

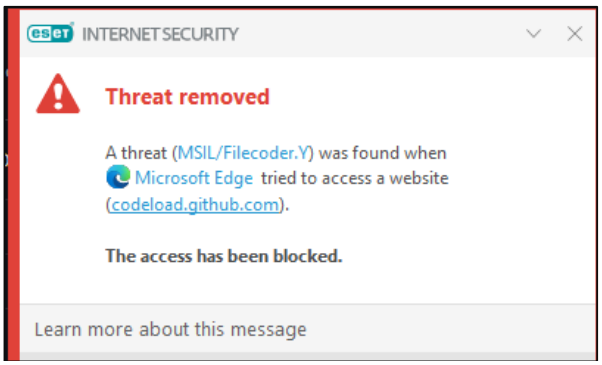


*Figure 3. ESET antivirus warns about Hidden Tear*

Hidden Tear project [32] is divided into two subprojects: `hidden-tear` and `hidden-tear-decrypter`. Focus here will be on the former, although the latter has symmetrical structure. Main source file is called `Form1.cs`, which is quite short, about less than 200 Lines of Code (LOC), mostly being *boilerplate code*. High level structure of its functions is as follows:

- `AES_Encrypt(bytesToBeEnrypted, passwordBytes)` – as the name suggests, it is encrypting received bytes using AES-256 CBC mode. Reader could also notice that PBKDF2 function is used before encrypting process, which decreases susceptibility to brute-force attacks by using salt and certain level of uniqueness (i.e., 1000) [34];
- `CreatePassword(length)` – simply building encryption key, based on random choice of certain set of ASCII characters;
- `SendPassword(password)` – basic version of communication with C&C server – simply sends generated encryption password to a preconfigured URL;
- `EncryptFile(file, password)` – given path to specific file, reads its contents and encrypts them using generated password. After that it overwrites original contents with encrypted data and changes the extension to `.locked`;
- `encryptDirectory(location, password)` – given specific path, it compares file's extensions against specific list, and if they match, that particular file is sent to the function `EncryptFile()`. Also traverses child directories;
- `startAction()` – executes entire flow, from generating password, through listing and encrypting files in specific, preconfigured directory, to creating ransom note;
- `messageCreator()` – creates ransom note as a text file in preconfigured directory (Desktop by default). Ransom message can be configured here;

In a summary, code is very concise and simple to use. Straightforwardness here is probably the strongest, and weakest at the same time, side of this project. It is probably also the reason of popularity of Hidden Tear amongst cybercriminals. However, one should notice that this ransomware does not find the most efficient way to lock victim out of its personal data. As already mentioned in chapters before, certainly more efficient way is encrypting specific metadata associated with environment files are residing it, such as MFT tables, or encrypting just parts of file. That being said, techniques presented in this section are probably the most concise definition of what ransomware does – and it is certainly sufficient for educational purposes.
Probably the best way to understand processes taking place inside ransomware, as well as other types of malware, is to conduct instrumentation, both static and dynamic analysis. Most informative technique, in terms of malware, is to reverse-engineer and decompile given sample and walk it through step-by-step. One such analysis of Petya ransomware was provided by Bitdefender AV company. [35] Petya has a complex structure and was designed with distinctive level of understanding filesystem that is attacking. It does not encrypt a single file, but rather a few underlying filesystem'

critical structures that result in system being entirely unusable. Specifically, according to the report by Bitdefender [35], Petya performs the following actions:

- Detects type of partition table – MBR (Master Boot Record), which is an older standard (using BIOS to boot up), but most compatible. It stores all information about partitions in a structure located at a specific offset. This is the only such data structure on the filesystem. If it gets damaged – system is unusable. Other type is GPT (GUID Partition Table – using UEFI), which is newer standard, supports larger disk capacities and the metadata here has a structure different than MBR. There are a few copies of GPT table in different offsets of the disk, just in case one of them gets damaged;
- Encrypts each byte in first sector of a disk with byte `0x07`, generates encryption key, overwrites first sector with code that is extracted from Petya's executable;
- First encryption stage takes place – all sectors from sector2 up to sector35 are being encrypted, then encryption key is saved to sector54, sector55 containing MBR is moved to sector56 and encrypted, and original sector of MBR get XORed with byte `0x07`;
- Next up it overwrites the sector where original MBR was residing with Petya's bootloader. After that it is time to reboot system, however, as power-related operations require certain privileges, Petya tries to acquire them and simulate crash by calling appropriate kernel function – as a result system displays BSOD (Blue Screen of Death);
- When the system boots up after BSOD, second stage of encryption takes place. Disguised as a fake CHKDSK scan that Windows usually runs after a crash, Petya is encrypting MFT structures. They contain information about files residing in particular partition. Without MFT, all data belonging to a partition is just a bunch of scattered data and is extremely time consuming, in some cases even recoverable;
- Last stage is generating ransom note for the user. The system is practically unusable;

Two completely different approaches were presented in this chapter. A simple to use ransomware that is not a very sophisticated in terms of internal structure. Nevertheless, it does what is designed to. On the other hand, there is highly complicated ransomware that employs vast knowledge about attacked filesystem and encrypts what it does a way smarter. To summarize, any of these approaches is equally common and reported in the wild, as it only depends on specific attack scenario, which exact techniques are employed by attackers.

## 1.6.2. Cryptographic functions

Cryptography is extremely important part of ransomware business. Unwisely chosen cryptographic functions might allow victim of attack to easily recover encrypted data, which defeats the purpose of the attack. One example of meticulously implemented cryptography in ransomware is the case of Petya. [36] [35] The process of key generation there might seem overcomplicated. On the other hand, it is a fantastic

example of how important this part of ransomware is important to cybercriminals. The procedure is as follows:

- The process starts with generating random seed. In case of Petya, it uses a number of different functions, both from Windows Cryptographic API and own functions just to generate single character at the time. Entire process is somehow unnecessarily redundant and is specific to this family;
- Based on generated bytes it concatenates character by character from range of numbers, lower and uppercase ASCII letters;
- When such buffer gets created, it is sent to Petya's internal function that generates sha512 value from that buffer;
- Then it generates second hash, based on first hash concatenated with number of the used structure (default is value *0*) and size of first, temporary hash;
- Then it generates final sha512 value based on previous hash as an input;
- First stage of decryption key is represented by first 48 bytes of final hash value, and is sent to symmetric algorithms;
- Received buffer is feed into AES-256 in CBC mode. Initial IV is a zero bytes buffer and initial key is a buffer containing values from 0 to 31;
- Resulted ciphertext is again fed into AES-256, this time in ECB mode. Plaintext is in this case the second 16 bytes block from the previously received ciphertext.
- Result of last encryption is used indirectly to update main decryption key through a XOR operation. Another buffer is extracted from the last 16 bytes of updated main key;
- Previous operation is repeated for each DWORD used to obtain individual characters for main decryption key. As a last stage, main AES context that was constantly updated through XOR-ing results of previous iterations, is feed into AES in CTR mode and appended to main decryption key in base56-encoded form. The resulting main decryption salsa key is 32-bytes long and stored on one of earlier mentioned sectors of hard drive;
- Another key, called "Personal Decryption Code", which is actually the decryption key, but encrypted with secp192k1 elliptic curve is stored on different offset;

Encryption algorithms used by i.e., Chimera ransomware are similar to Petya [35], but with some differences. Chimera also relies on CBC and CTR variations of AES, and it also uses elliptic curve for public key encryption. However, it uses secp256k1 instead of secp192k1. But it does not use Salsa20 algorithm. Both families implement encryption algorithms and don't rely on Windows API functions. Rokku ransomware also uses ECC for public key encryption, and same algorithm for encryption as Petya. In general, the flow of key generation was presented on Figure 4 below. Looking at technical advancements in quality of used cryptography by ransomware attackers, it is quite astonishing how fast, and adaptive is this model of cybercrime.

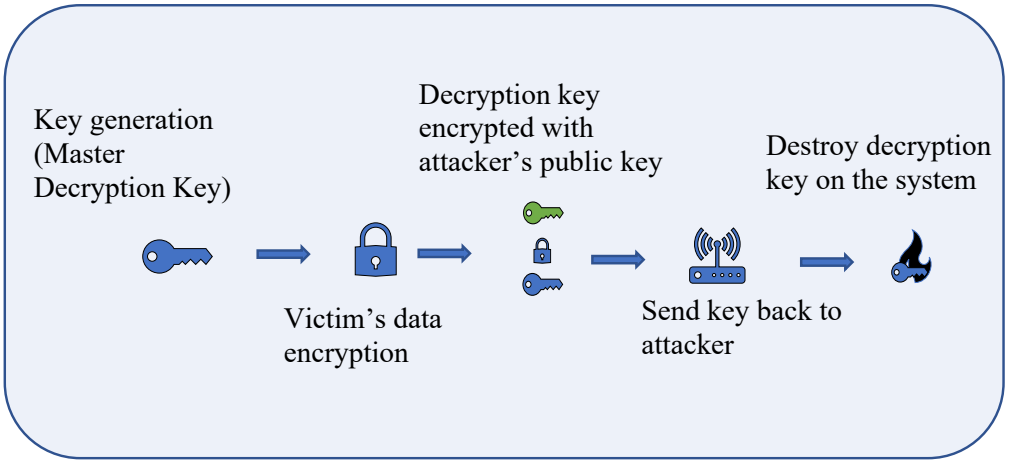


*Figure 4. Key generation flow in ransomware*

From early times, where attackers were reportedly using "broken crypto" – cryptography that is easy to crack, like XOR with hardcoded password, RC4 with hardcoded password etc., we now experience times where ransomware, in most cases, makes use of cryptography that is more up to date than most legal industry standards, government organizations etc. As mentioned before, since there is a lot of money at stake, cybercriminals have very high motivations to keep their malicious software uncrackable.

## 1.6.3. Communication between ransomware and attackers

In case of Petya [35], where the encryption process takes place in an environment without internet connection, it is not possible to transmit main decryption key to the attacker. To solve this issue, decryption key is encrypted using attacker's public key and stored as "Personal Decryption Code". Once transmitted to an address pointed by attacker, and once the payment was issued, attacker sends back that key, decrypted with their private asymmetric key. The procedure is as follows:

- Following compromise, ransomware generates a pair of asymmetric keys for the infected system, using secp192k1 elliptic curve for this purpose (`PubKV`, `PrivKV`)
- Attacker also has a pair of keys, namely (`PubKA`, `PrivKA`)
- `PubKA` is hardcoded into executable
- Shared secret is computed by multiplying `PubKA` and `PrivKV`, generating sha512 hash value based on the result and using first 32 bytes as encryption key for AES256

- Now, main decryption key is XORed with first 16 bytes of `PubKV` and encrypted with encryption key generated in previous step.
- Next step is to create a `PersonalBuffer` that stores concatenated `PubKV` and encrypted main decryption key, encoded it using base58 encoding (which is used instead of base64 probably because in base58 it will be more difficult for victim to confuse certain characters), and send it to sha256 function. First byte of the hash is prepended to `PersonalBuffer` and is used as a checksum.
- The result is known as Personal Decryption Code. When it's received by attacker, it is simple to reverse above steps. The only difference is that in the first step, the attacker will multiply `PubKV` with own `PrivKA`. Due to characteristics of public key cryptography, shared secrets are equal, and the key is recovered.

Communication between ransomware and its C&C server (*Command & Control*) is yet another angle that is extremely important to attackers. If not done right, the victim, or security analysts, might be able to sniff out decryption keys that are sent back to attackers. The quality of encryption is one thing, surely. Another issue of interest is non-traceability. Cybercriminals want to ensure that any communication with their malicious software will not be traced back to them, thus revealing people responsible for the attack, their locations etc. One solution to this problem is using TOR network. This communication uses multiple layers/intermediate nodes for exchanging data, relaying traffic from random node to a random node in an unpredictable manner, and TOR network nodes are scattered around the world. Because of TOR structure, it is very difficult to trace the network traffic. And this type of network serves well to all kinds of cybercriminals. [37] [38]

Communication via TOR network is not an ideal solution, as it does not provider 100% guarantee of non-traceability. There are several other approaches to the problem. One example could be Chimera case. [39] Its authors decided to use P2P message application (which works in fact similar to TOR), in this case *Bitmessage*. There are many more examples, and the stranger, the better for hiding illegal communication.

## 1.6.4. Deployment techniques (attack vectors)

About 51% mid-to-large enterprises around the world were hit by ransomware in 2021 – only 3% less than in 2017. [40] One of causes of this slight decrease is the shift in types of targets. In the past, attacks were more common amongst private users and micro-sized enterprises, and the ransoms were not high. Cybercriminals were spreading ransomware through various attack channels in order to infect as many victims as possible. It changed apparently and nowadays the attacks are more targeted at larger companies, the demands are substantially higher and, what is most important – this particular branch of criminal industry has not diminished over the years (in fact, quite the opposite). As targets have changed over the years, so did the attack vectors. What remains unchanged is, however, that e-mail campaigns are still the biggest ransomware distribution method (as well as malware in general). [40], [6] This category includes spam messages, phishing scams, spear phishing, whale phishing,

and the list goes on and on. These techniques will be discussed in detail in later stages of this chapter.

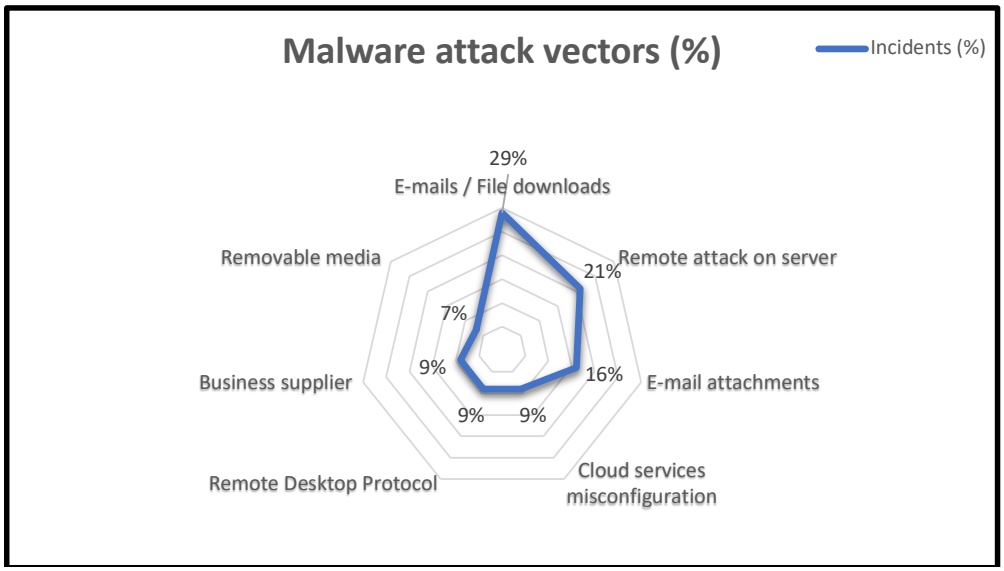


*Figure 5. Common malware attack vectors in 2020 [40]*

Amongst the new trends, cloud-related attacks are taking precedence, not surprisingly. It was a fact that everything, everywhere was moving to the cloud since a few years back, today it is already in the cloud. The problem is that cybersecurity-awareness and general knowledge did not follow this trend.

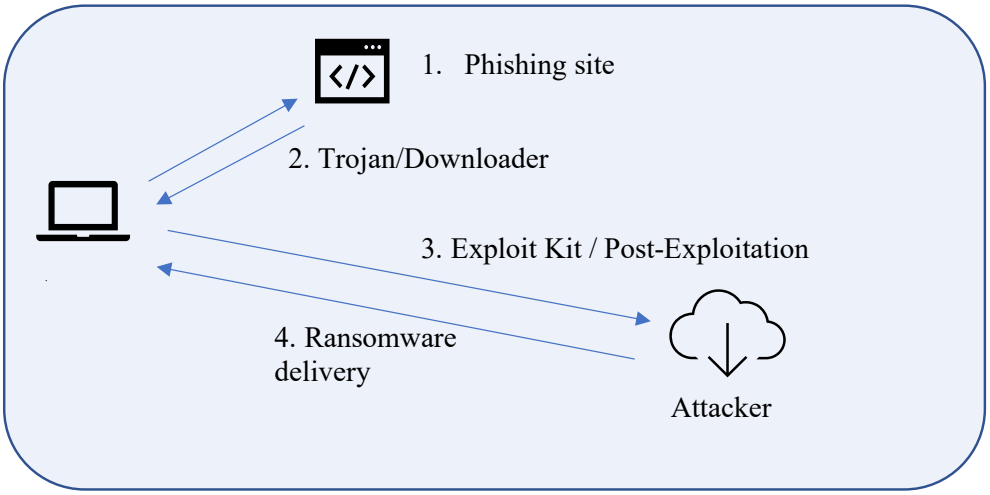


*Figure 6. Typical chain of actions in ransomware attack*

This is the reason why cloud-related attacks, including basic misconfiguration, but also more advanced attacks are being carried away successfully. Similarly, there has been a huge shift to work-from-home business model, mostly due to Covid-19 pandemic situation, and it is the reason why RDP (Remote Desktop Protocol), VPN etc. related attacks and development of exploits have skyrocketed. Lastly, although the vast majority of attacks are network-related, physical attack vectors are also commonly reported. Figure 5 provides general overview of trends in ransomware attack vectors. [41] As to how the train of events typically looks like in ransomware attack, see Figure 6. Now, with this perspective in mind, let us discuss specific techniques, with their impact and probable remediations. Those techniques are listed as the following subheadings of this section.

### Spam e-mail, phishing

Most common delivery method for ransomware, as well as other types of malware, is, and always has been, via e-mail. [42] This method for exchanging information is surprisingly still very common in our everyday lives, and most common corporate communication since its invention in 1971. As of 2020, it is estimated that around 300 billions of e-mails were sent and received only that year. [43] On the other hand, only in 2019 about 55% of e-mails worldwide were accounted as unsolicited e-mails, or spam. [44] Before going further, it is worth mentioning that e-mail has a huge impact on our environment, in terms of carbon footprint. Scientists warn about power consumption of today's technology and its possible consequences in the future. One of the top shares in this category belongs to e-mail. [45]

The definition of spam messages is in general, *"commonly used in advertising campaigns for business promotions; however, it can also be used for more dangerous purposes such as spreading malware or acquiring confidential information, such as login credentials and financial information from the victims"*. [6] Spreading malware as such, requires spam messages to draw user's attention and initiate interaction. Various social engineering techniques are employed for this purpose. According to this [44] report, spam is most often disguised as one of these categories:

- *Online dating* – advertising various (fake) websites for online dates or sending invitations from specific users that are interested in developing closer relationship with spam victim. Such messages are often visual for obvious reasons. Another commonly reported type of spam in this category is related to medicaments aiming at general betterment of sexual life;
- *New Apple products* – aiming at users interested with a specific set of products, typically of higher quality and designed to be unique. Users in this category stereotypically have more financial resources to spend and therefore are more lucrative targets. Raise in such campaigns often correlates with dates of conferences delivered by such companies. After such events, number of malicious attempts to redirect users to scam websites imitating official services of these companies grows in number significantly;

- *Fake technical support* – offering assistance in matters related to specific products, recommending help of highly qualified staff being ready to help;
- *New features* – promoting newly developed features in online services that are most popular, also providing an instruction on how to make most of them, and link to log in to that service to access them faster; Similar technique applies to luring victims into discounts, one-time limited offers and similar, typically requiring from victim a quick interaction to claim those promotions; Interestingly, exactly the same techniques are often used for legitimate marketing purposes, which makes it substantially harder to tell apart legitimate offers from scam;
- *Mailshot* – this category has a form of automatic notifications sent on behalf of certain services, i.e., banking services, social media, on-line shopping etc. Such e-mail usually prompts for account confirmation, taking immediate action on one's account which can be accessed by provided link, or informing that the payment has been declined (and the reason of unsuccessful transaction can be accessed by link provided via e-mail); Again, sometimes these e-mails are impossible to tell apart from legitimate sources, which is also complicated by the fact that e-mail header (containing sender's address and information) can be easily spoofed. One difference is that, in general, legitimate e-mail will not ask user, under any circumstance, to enter any URL provided in e-mail or ask to log in anywhere;
- *Financial spam* – related to banking offers, loans etc.;
- *Job offers* – e-mails containing very interesting/demanded work offers. In other words, exclusive offers from most desired employers in particular branch of businesses. Such e-mails often make users into installing a "special application" required to take part in recruitment process; such application is later downloading other trojans and infecting victim's system.

Above general categories are certainly not the complete set of spam capabilities in these days, and yet this list seems overwhelming – spam is present in almost every, if not every, part of our life; Most of this traffic ends up caught by spam filters or left unread. It's only the minority of such messages that are noticed, but as discussed in before, they still make the largest attack vector amongst all others. Problem is, even in spite of many security measures, security awareness trainings and general acquaintance with various marketing techniques which are used everywhere, people are still vulnerable to those techniques used in social engineering. Not much can be changed here unfortunately, because even experts in the field who use such influencing techniques on a daily basis are proven to be vulnerable. [46]

In order to deliver ransomware into victim's machine, some form of interaction is required. In case of spam messages, two ways of triggering such delivery mechanisms are distinct [6]:

- *E-mail attachments* – script hidden inside document, or executable disguised as innocuous file type. The former needs to overcome security mechanisms in software used to read such documents; for instance, Microsoft Word has scripts disabled by default, so that user must be tricked somehow to turn them on; this is

mostly done using social engineering techniques. The latter might look like any media type, even the icon will be changed so that this file would appear legit. The only thing that might protect user from executing such file is noticing that extension will be ended by ".exe" (for example *"Readme.pdf.exe")* but it requires Windows user to explicitly uncheck option *"Hide extensions for known file types"* in settings, which is turned on by default and hard to reach for most users;

- *Malicious URLs* – emails might also contain links to malicious websites which, once accessed, download malware either directly or through certain elements on such websites, such as fake advertisements. It should be noted here that e-mails often use HTML for text formatting, which allows attacker to disguise malicious URL;

It is also usually the case that ransomware uses legitimate executables to hide itself, such as java updater, google crash handler, or various installers for PDF readers, media editors etc. Code is then inserted manually and is present at different sections of executable file. While the entry point remains unchanged, some branches are being changed so that while executing legitimate actions the flow is directed towards malicious encryption process. Sometimes, the icon is also replaced with an icon of a document of a certain type, such as PDF, DOCX, XLSX, PNG etc. Malicious sections are usually encrypted and packed within the executable file (i.e., using self-modified UPX packers [9]) to make it substantially harder to analyze the malware by security analysts, and after that executable is loaded by the dropper malware in user mode [10], these malicious sections are being recovered on the fly during execution process. [35] Spam is known to focus on targeting financial sector. Banks and credit organizations received the greatest number of attacks in the past year, according to Kaspersky lab. [44] On the top of this list are also global internet portals / social networks and payment systems. Only these three categories take about 60% of attacks, globally. For detailed comparison between targets of biggest spam campaigns, see Figure 7. Spam campaigns can also be divided into the following categories, according to their contents [6]:

- *Bulk messages* – general advertisements being sent to a large number of e-mail addresses, typically taken from leaks from popular online services. Such messages do not target any particular type of user, and are not very sophisticated as such;
- *Phishing* – more targeted, aimed at specific type of users, but also sent to a larger group of recipients. Often disguise as e-mails from genuine sources, and use various techniques in attempt to collect user-sensitive information;
- *Spear phishing* – much more personalized form of above, usually targeting specific individual, organization, or enterprises;
- *Whale phishing* – similar to *spear phishing* in terms of personalization, form etc. however, as the name suggests, this type of spam is targeting only high-profile

[9] UPX packer – executable packers, mostly custom versions of this standard, were used in the past as a form of compression, sometimes used by legitimate companies to make cracking their software harder, but also by cybercriminals for basically the same purpose

[10] User mode refers here to code executed by non-administrative user of a system

employees within organizations (having a high position) in order to steal sensitive information that other people may not have the access to;

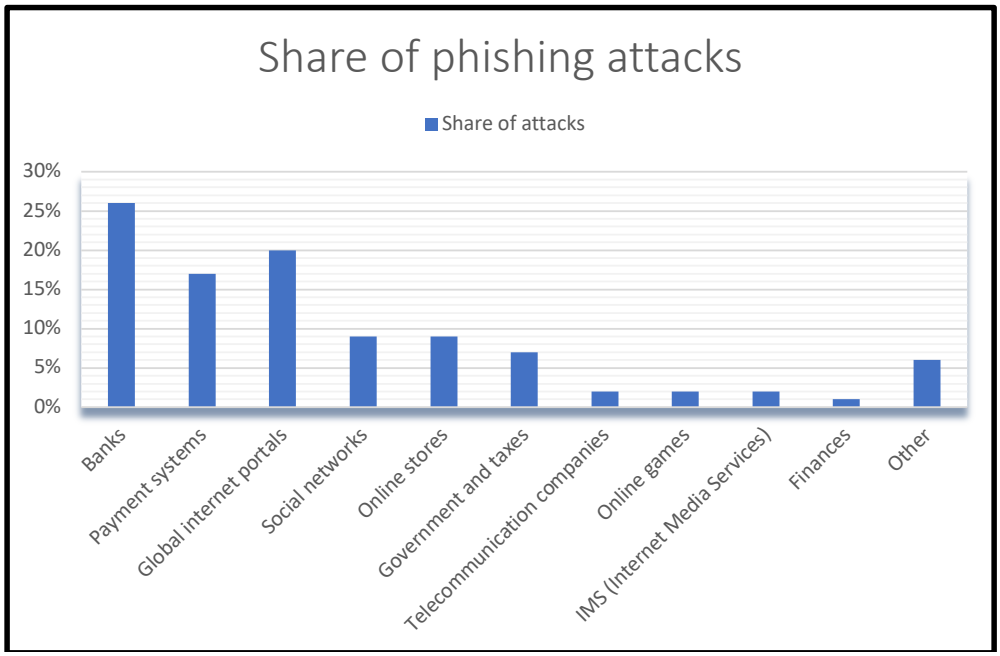


*Figure 7. Share of phishing attacks*

Even though most of portrayed attacks do not end up victim having proper ransomware installed, having their credentials is usually sufficient as initial attack vector to perpetrate another attack. Another case is that installing malicious software through either e-mail attachments or malicious URLs in e-mails might not instantly install and run ransomware, but rather try to infect other machines on local network silently, and postpone alarming actions, such as encrypting user's data with ransomware, to the later stages of attack.

### Watering Hole Attacks

In this scenario, attacker gathers intelligence on specific individual/company, monitoring their behavior in order to determine what webpages are being visited most often his victim. This information is used to compromise websites that are being considered fertile ground to other potential attacks. User is visiting such website, unaware of potential threat, only doing its routine tasks, and that leads to malware being downloaded to his computer, which gives the attacker an initial point of access to the victim's machine. [47]

### Malvertising

This term applies to any form of abusing legal advertisement channels (i.e., Google Adsense, Monumetric, Adversal, Media.net etc.) in a form of injecting code inside legitimate ads. Advertisement vendors provide certain services that are attractive from a perspective of a company, for example selling space for advertisements on some popular website. Just like billboards in real life, and other physical types of space for ads. Usually, companies prefer to outsource their advertisement campaigns to a third-party. And so do cybercriminals. They buy space for commercials from a third party, only to inject malicious code into ads, so that next time user visits the website, his computer is being infected by some form of malware, usually a rootkit, which finds specific vulnerabilities on victim's machine and as a result delivers initial stage for other attacks.

### Removable media

There is one particular category that does not relate no network directly, and that is abusing physical access to targeted machines. Probably the most common technique to force victim into plugging an USB drive into their computer is dropping such piece of memory in a public space, so that victim will grab and use it probably just out of curiosity. This method is surprisingly effective, at least according to studies conducted in 2016. [48] [49] It is not the only way to plug an USB drive into user's machine. If victim leaves their computer unattended, either in a public space or corporate environment, this fact could be exploited by another ill-intentioned person. These are only a few examples, which might be also extended using more complex social engineering techniques. In general, this form of delivery was used in one of the most famous attacks in history, namely the Stuxnet worm in 2010, which resulted in installing malware on Iranian nuclear facility network.

### Pirated Content from the Internet

It is known phenomenon that illegal media, such as movies, games, programs, downloaded from the internet almost always contain viruses. There is also one notorious ransomware family that was targeting gamers specifically – TeslaCrypt. [50] One of a first malware in history was from a company, which planted malicious code into cracked versions of their product and released them into the web However, even though people are usually aware of such techniques and other risks, piracy is still one of the greatest problems, huge source of income for cybercriminals, and this entire situation does not look promising for the future. As of today, still most of the users blindly believe that *"if it is on the internet, it means it is free"* [51].

### Remote desktop connection

There might be many reasons for facilitating remote access to machines, from client support to outsourcing company's support to third parties. By exposing direct access to internal resources, it makes company critically vulnerable, unless properly secured. Unfortunately, this is often not the case. [6] Apart from vulnerabilities in software used for RDP (Remote Desktop Protocol) connections, instances exposed to the Internet are often misconfigured and are secured by weak passwords that are easily bruteforced. Moreover, there are many publicly accessible search engines that allow finding machines listening for RDP connections. [52] This leads to such services being actively exploited in the wild, and access to them being sold on the Dark Web. And since the beginning of Covid-19 pandemic situation, use of such misconfigured services and response from cybercriminals has grown exponentially. [53]
RDP is not the only protocol that should be mentioned as method for remote access, which inadequately configured, yields companies vulnerable to attacks. There is SSH[11] protocol for server administration, FTP[12] for sharing files and a broader category – VPN[13] for accessing otherwise private resources on company's private network. All these protocols suffer from attacks like stealing credentials, hijacking legitimate sessions, forging requests in the name of server (i.e., SSRF[14]) to more common information leakage – the list of issues that were reported in last year is countless just for VPN services. [54] [55] [56] [57] [58] Therefore, any tools that allow for either remote management or access to certain resources should be used with greatest care, and constantly updated, as it is very widely exploited field that is extremely popular in these days, both for legitimate businesses, and attackers.

### Cloud services misconfiguration

Since many years ago cloud computing has become an important part of our everyday lives, and almost every company uses many types of services related to cloud. For instance, SaaS (Software-as-a-Service), IaaS (Infrastructure-as-a-Service) or PaaS (Platform-as-a-Service). It was inevitable that cybercriminals have also shifted their focus onto making use of such services. Problems that pose security risks do not necessarily come always from client's side. There were many cases in recent years, where, because of some vulnerabilities in cloud services itself [59] [60] [61], privacy and security of end-users were compromised. However, such cases are rare, simply because cloud services are usually very concerned about their security as they service numerous clients that are all accessing their infrastructure and putting information in the cloud that is vital to life of their businesses. What is more, cloud service providers usually provide huge bug bounties for finding bugs in their systems, so that potential

[11] SSH – Secure Shell – terminal protocol for remote management of servers
[12] FTP – File Transfer Protocol – allows to transfer files/data between servers
[13] VPN – Virtual Private Network – protocol that allows to establish connection between remote sites, so that they both seem as one private, local network to each other.
[14] SSRF – Server-Side Request Forgery – web attack that tricks remote server into executing request to its internal resources inaccessible to attacker directly, allowing attacker to get hold of those resources

attacker, even if does not have good intentions, has much more profit for reporting such vulnerability to the vendor than selling it to the cybercriminals. [62]
Much more common issue with cloud services is that clients usually have a lot less security knowledge and awareness. They often use default configurations, weak credentials for access to their sensitive information, and also rendering an initial point of compromise for attacks on other users of such cloud services. Thus, in case of cloud services breach in one company could cause problems for others too. [63] [64]

### Drive-by downloads

Last discussed category of attack vectors is probably most terrifying, as it does not require any user interaction in order to infect his machine. In this attack scenario, victim is visiting a compromised web server that contains malicious scripts. Because of existing vulnerabilities in victim's system – it could be the web browser, other application that is connecting to that specific web server or the system itself is flawed, and just after accessing the web server and loading scripts, malicious software is downloaded automatically, without user interacting with the elements displayed on the webpage. [65] [66] Consequently, it is relatively uncommon and difficult to perpetrate attack scenario, as it requires synchronization of two independent factors [67]:

- Vulnerable web server, that will store malicious software
- Vulnerable user, that uses unpatched versions of browser, operating system or other underlying software and is accessing the compromised web server

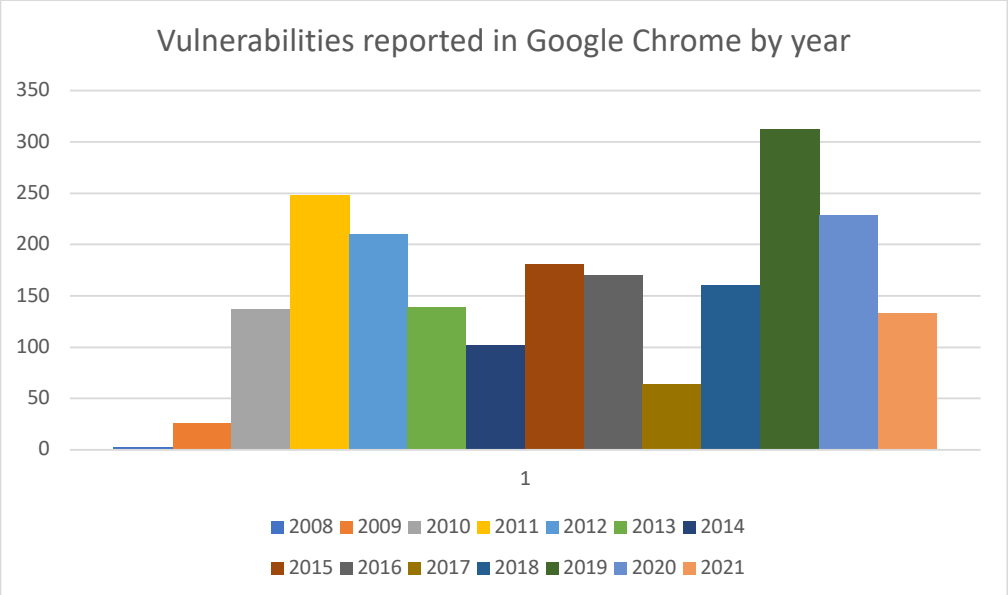


*Figure 8. Vulnerabilities reported in Google Chrome browser in recent years [74]*

If these two factors come into play at the same time, victim is out of luck. It is worth mentioning that weaknesses in web browsers are not that uncommon.
Zero-days or exploits for specific vulnerabilities that are being used in the wild before vendor of vulnerable software releases patch for that vulnerability were constantly reported for most common web browsers, such as Chrome, Firefox, Opera, Safari or Edge. [68] [69] [70] [71] [72] Moreover, it does not need necessarily has to be a zero-day vulnerability, in order to issue drive-by download on victim's machine. There are hundreds of vulnerabilities each year reported for each of most popular web browsers. One of most popular online databases for such reports is CVE Details [41]. For instance, vulnerabilities for Google Chrome are presented on Figure 8.

On the other hand, there are certain countermeasures that can help with mitigating effects of such attack. Antivirus software might prevent from proceeding with infection by either blocking execution downloaded malware or blocking connection to attacker's servers in order to download other malicious software. Most of web browsers implement some form of sandboxing[15], thus providing another level of security. There are also specific web browser extensions, which might block executing some, or all scripts on chosen webpage. One example of such extension is NoScript [75], which disables by default all scripts on chosen webpage, so that user must manually whitelist scripts that are relevant to that particular website's logic. Such tools provide high customizability, probably preventing not only drive-by attacks, but also all kinds of adware, spyware – their absence might also make the website less cluttered and work substantially faster. The disadvantage of such solution is a certain level of knowledge about web development that a user of such extension must represent in order to make it work.

## 1.7. Pay or not to pay

> *"Ransomware incidents inflicted an estimated $24.1 million in financial losses in 2015, according to 2,453 reported hackings. With 13 million U.S. users encountering some form of ransomware, our own studies have shown that 50 percent2 of them would actually pay to regain access to their data."* [35]

It is an endless doubt and ever-lasting existential question whether the victim should pay the ransom. And there is unfortunately no simple answer. Some sources, such as researchers at Kaspersky [76] do not recommend paying the ransom, as it does not guarantee that you can get your data back, but also it is a form of financing, and thus encouraging, cybercriminals to continue carrying their ruthless business. Despite that, the same company has found that more than 56% of victim has paid their ransom, and only a 17% of those who paid did see their data again. [77] The rest includes cases where, despite attacker's help and cooperation, the data could not be decrypted since there were some problems with ransomware design; some of encrypted data, or ransomware itself got damaged during the attack or keys were sent back to the attackers in bad condition. All those situations are certainly in the vast majority of

[15] Sandboxing is discussed in detail in chapter 3.2 – Local Protections

cases where data was not recovered after ransom was paid. And the best thing here is, ransomware authors do not lose anything if such process gets interrupted. They get their money and disappear. But the company that paid the ransom, followed all instructions sent by attackers, is left in very uncomfortable situation. As to those who paid, there have been cases where victim has paid the ransom, the data was recovered, but they did not take any action to patch the vulnerabilities that were used by attackers to start attack in the first place, and because of that they were encrypted the second time, and had to pay the ransom again. [78] Another new trend that emerged very recently is double encrypting the data by two different strains of ransomware. That way, even if victim has paid the ransom, or successfully decrypted their data without paying the ransom somehow, the data is secured by another ransomware. [79]
Attackers usually do not make it easier to decide whether paying the ransom is the best option. There have been cases very recently, where cybercriminals were using various social engineering techniques, including cold calling, or threatening, as well as taking physical actions such as deleting random files from filesystem after some time if the victim is still hesitating with paying the ransom. [80] There have also been opposite cases, where ransomware authors were releasing keys to the victims, reasoning that after certain time they understand the harm they did and offering sincere apologies for their evil actions. [81] All of these stories certainly do not state whether it is best to pay the ransom or not. It certainly depends on the situation and particular cybercriminal group that started the attack, their intentions, and the money at stake. In any case, there are, and always will be doubts and different outgoings of both decisions.

## 1.8. Recovery after attack

Once ransomware is finally able to execute and encrypt user's data, there is no way back. Although such action does take some time to finish – around 18 seconds to 16 minutes, statistically speaking, for 1000 media files (around 70 MB) to be encrypted. [6] Apart from high resources usage, there are known Indicators of Compromise (IoCs) that strongly suggest ransomware being active on the system:

- *File extensions* – monitor for known ransomware file extensions
- *Bulk file renames* – monitor for large volumes of files being renamed
- *Security Tools* – Endpoint, network, and behavior analysis tools report strange behavior
- *Ransom Notes* – Explicit ransom notices left on the system
- *User reports* – Cannot open or find files, PC running slow

One of this first steps that are usually taken after one of such IoC is detected, is containing the infection – in order to stop ransomware from spreading through internal network, which it usually tries to do. There are guidelines on how to implement IR (Incident Response) Lifecycle, which is a set of actions that should be taken to approximate the loses, stop attack from spreading, understand how it got into company's network in the first place, and warn customers about the threat. Such steps allow company to deal with threat efficiently, and help preventing similar attacks in

the nearest future. [82] [83] There are two approaches to visualize incident response process [84] – SANS, as shown on the Figure 9 below, presents it in 6 basic steps, whereas NIST in 4. Both of these representations have similar sense, although the latter has a more concise verbiage.

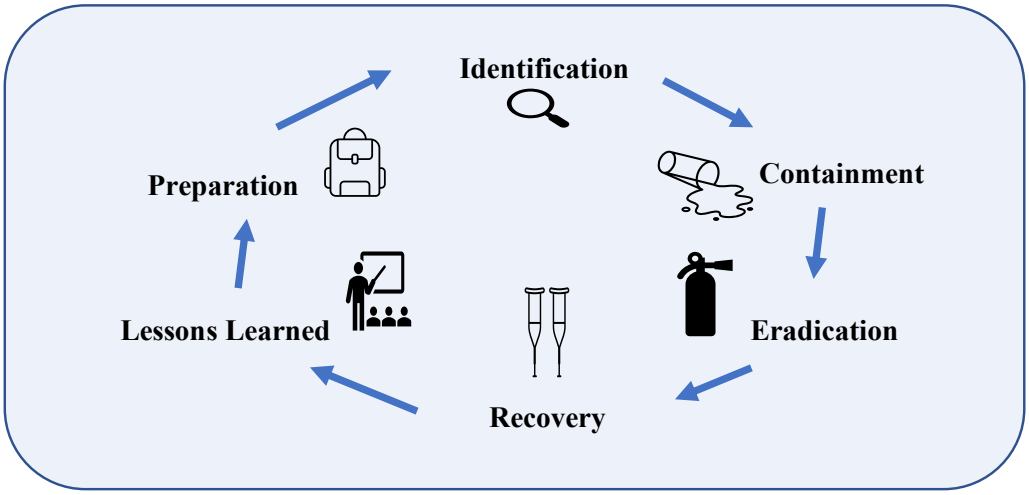


*Figure 9. SANS approach to Incident Response Lifecycle [84]*

Several projects that help victims of ransomware attacks exist. They usually focus on providing decryption tools for known ransomware families, providing help during and after attacks, and raising awareness of society on such threats. One example of such projects is NoMoreRansom. [85]

## 1.9. Prevention

There is an invisible, seventh step in the Incident Response Lifecycle, that is miraculously able to become the only one, if implemented carefully. Prevention of attack, that is. Apart from being able to quickly act in case of attack, gather all information and learn lessons and all these actions taken after the act, prevention might actually help in strengthening defenses, addressing weaknesses in one's system, raising awareness of attack and its consequences. It is not only about hardening company perimeter to make it impenetrable, or close to impenetrable; it is even more on making sure that when the attack happens, and, statistically speaking, it will happen, the harm that malicious software causes is limited, restricted, blocked by many policies saying what resources have access to what other resources and so on. In a few words, prevention is making attacker's task harder, not easier.

From organization perspective, another interesting point is being able to identify the attackers quickly. There are multiple ways to do that, amongst them is setting up

honeypots [86] – which are dummy servers that simulate internal network traffic that might seem valuable to the attacker. It serves two purposes, really – for once, it stops attacker from penetrating real network, and doing real harm. It is a win-win situation – both parties, attacked company and attacker are satisfied. Second purpose is learning what techniques, exploits attackers currently employ in their work. Similar technique to honeypots is honey accounts – which work on the same base, and honey documents – or canary tokens – which are a normal media i.e., PDF, Word, movies, music etc. that are programmed connect to specific server upon opening – thus revealing attacker's sensitive data (IP address, location, name etc.). Yet another important point is constant network analysis, both manual and automated – specifically for the purpose of anomaly detection. This prevents situation where attackers will be able to linger in the internal network for too long. All that and other important prevention, deception advice was covered in [4]. But to stay in the subject of prevention specifically, there is a short checklist that every company should consider implementing as a strong part of strengthening defenses:

- Backup regularly and test backups
- Disable office macros by default
- Use firewalls to block C&C callback
- Scan email for malicious content
- Network segmentation – segregate networks where possible
- Use anti-virus/anti-malware protection
- No administrator rights by default
- Enforce access control permissions
- Educate users about ransomware
- Block ads and unnecessary web content

And advice directed to network communication specifically:

- Cloud and offline backups
- Patch internet facing systems
- Implement 2FA (Multi-factor Authentication)

As to local protections, IT administrators might also want to look at prevention of automatic execution of applications from unknown sources. On Windows, such options include disabling AutoPlay either via system settings, or via Group Policy. Another idea might be whitelisting/blacklisting certain applications, and AppLocker might help in setting such lists for specific user, and Software Restriction Policy settings for entire system.

Finally, there are online checklists that gather information regarding recognition of specific ransomware families, prevention, general advice etc. Two examples of such information sources are [87] and [88], and these already cover hundreds of known ransomware variations. Lastly, there are various guidelines and checklists in regard to

specific type of threat i.e., web exploitation. [89] Spending some time to review and build own defenses based on insight of such projects might greatly decrease the risk of successful attack.

## 1.10. Data exfiltration

Ransomware has recently shifted its focus from data encryption to leaking valuable information. It is a relatively new trend that gives even more profit to cybercriminals. Business data is still going to be encrypted after successful ransomware attack, but at the time of encryption this type of malware will often try to exfiltrate business data outside company's local network – so that attackers gain income both from ransom, and data sold on the darknet (see Figure 10 below). Although data exfiltration is not a new subject, and has been only recently associated with ransomware attacks, techniques are similar to those that have been in use for years. In general, entire subject comes to various ways in which the communication with attackers is obfuscated by use of legitimate protocols. [90] [91]

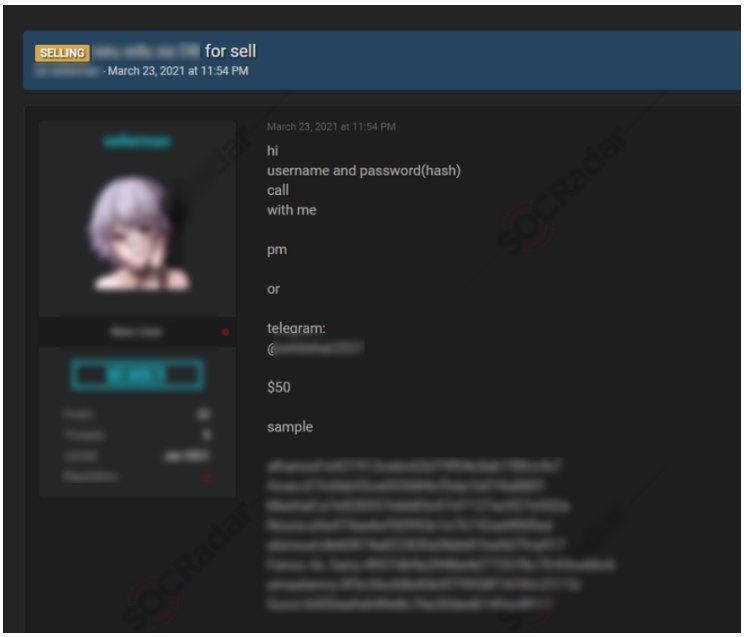


*Figure 10. Leaked data selling auctions on the darknet [92]*

One of most common examples of exfiltration is using ICMP packets (ping requests). Structure of this protocol is shown on the Figure 11. Last part of this type of packet is ICMP data and has variable length.

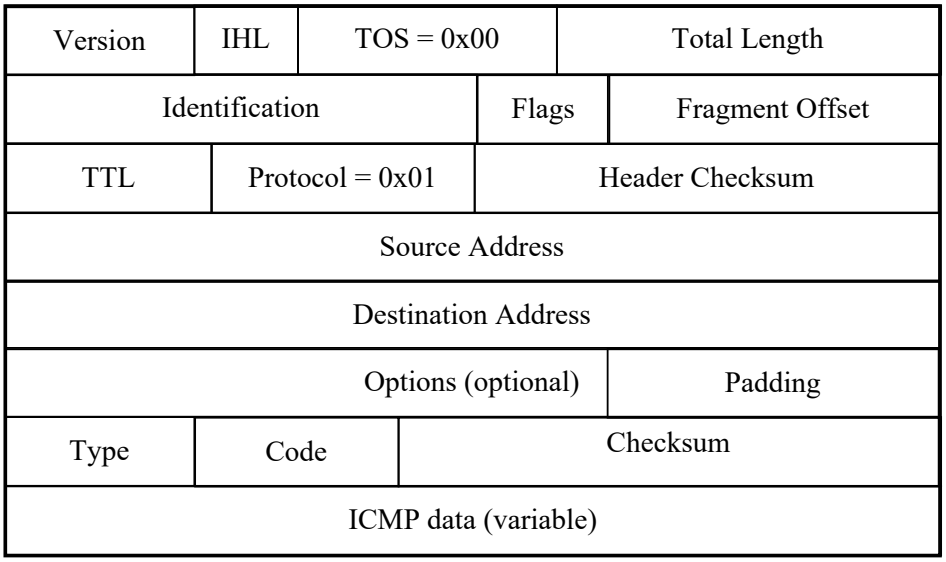


*Figure 11. ICMP packet structure*

That means, if attackers are able to issue ping requests from attacked machine, and this field is not controlled/inspected by security measures (i.e., firewall), attackers will be able to exfiltrate data out of attacked network without raising any suspicion and what is more important – leak sensitive data without being stopped. [93]
Other creative ways of tunneling data through covert channels include DNS requests (i.e., through Fully Qualified Domain Name – FQDN records) [94] [95], HTTP requests (HTTP errors, less known but sometimes left unblocked debug methods etc.) and others. The list could go on and on, and if there is really no way to exfiltrate data using covert channels, attacker might also try to just use traditional, encrypted channel – this way they might get caught much faster and not be able to leak as much data, but it is still another way for exfiltration anyway.
To fight with this type of cybercrime, companies need to create strong access policies, and configure firewalls and other protections so that there is little or no flexibility as to what protocols, and what options might be exploited by attackers. However, as it is often the case with systems connected to the Internet, there will always be a way to leak some data, so that sometimes the best solutions are to just keep most sensitive data inaccessible to anyone and enforce strong physical security measures.

## 1.11. Summary

This chapter evaluated current state of history of ransomware, its structure, common techniques used to deliver and infect victim with malicious software and covered most important trends in development and attacks associated or caused by ransomware actors.

In order to better protect against the threats, one has to understand most common malicious software types (and how they cooperate with each other), ransomware families and their characteristics, how does ransomware communicate with attackers, and what is the most often way in – all of that and more was covered in this chapter, to bring a better understanding of what actually ransomware is. With this knowledge, next chapters will look into current state of countermeasures and security products, solutions that are dedicated to fight with this kind of malicious software, so that the reader will gain a complete perspective from both defending and attacking sides – which will hopefully complement each other and bring more awareness, and thus security, to the world.

# 2. Countermeasures

One of the major challenges that company might face, in terms of securing its data, is a choice and collection of proper tools and their configuration. It has been more than 30 years since the concept of simple packet-filtering firewalls was created and many more advancements have followed, that make it crucial company asset today. There are several different approaches for securing network communication, from simplest packet and rule-based filtering, ACLs (Access Control Lists), to advanced signature-based detection.
Most of these approaches will be briefly discussed here, presenting their most important features in case of defending against ransomware attacks. However, most of them are not 100% proof against ransomware attacks, and therefore they must cooperate with endpoint local protections.

There are many types of systems in the use today that are being attacked by ransomware, mobile or stationary end-user devices. The biggest impact on security of company assets has endpoints with Windows system onboard, and that makes this work mostly focus on this system.
Following chapters will cover various security mechanisms and solutions that are implemented to protect system on both network and local level. This will be another important element in picturing this war with ransomware, and good foundation before reaching the practical part of this work that will cover effectiveness in preventing ransomware attacks under various circumstances.

## 2.1. Network protections

Network security can be implemented in various ways that can be also combined for a better effect. Such solutions are usually implemented as a software running on company servers, or as a separate device that is connected to the same network and is inspecting traffic. In any case, such protection usually works as a proxy that intercepts all traffic that is coming from outside the network, either closely inspecting it on the fly, therefore rendering higher latency but also providing higher security, or passing the traffic and inspecting its copy, which makes accessing company servers substantially faster but provides slower response of security protections. Network solutions are also often implemented on hosts in form of antivirus software' specialized modules. Either way, there are some general principles by each of these work. They will be briefly discussed in the following chapters.

## 2.2. Evolution of firewalls

One of the oldest network protections, and most elementary at the same time, is packet-filtering firewall. It is designed to operate at inline junction points, such as switches or routers. It works by comparing each IP packet in the stream against known ruleset: allowed/blocked IP address, port number, protocol, flags and other important packet headers. Packets can be either dropped or allowed to pass based on their match to specific criteria. Such protection is deemed to have a minimal impact on the performance of network; however, it does not offer any security against web-based attacks (attacks based on protocols that are from highest layers in ISO/OSI model).

Stateful firewall brought some improvements to the initial idea, as it not only does compare each packet against defined set of criteria, but it also keeps track of state of TCP sessions. As the network connection is initiated by TCP protocol, first action at the network layer level is TCP handshake – in general it is a request from client to the server to open connection, and server's answer that it is ready to connect. Stateful firewall is inspecting each packet similarly to packet-filtering firewall but is also able to block/allow traffic based on its origin. Imagine HTTP server running on port 80. It makes sense to block all outgoing connections in most cases, as it makes sense that somebody will initiate request to the server, but on the other hand it should be highly suspicious that the server wants to initiate connection elsewhere. Stateful firewall also takes into account the payload of intercepted packet, allowing to inspect traffic based on its content.
This form of protection is more precise than the basic packet-filtering firewall and helps to prevent certain cases that packet-filtering firewall is not able to detect. However, it has more impact on performance of the network. One simplification of the stateful firewall is circuit-level gateway – its main task is to inspect TCP handshakes or other session initiation messages specific to other network protocols.
This allows to determine whether session established between the local and remote hosts (in any order) is legitimate i.e., whether the remote system is considered trusted. This type of inspection puts a lot less weight on network performance than stateful firewall. Multilayer inspection firewall, which is yet another variation of stateful firewall, delivers protection across multiple protocol layers in the ISO/OSI model, thus providing more insightful, more customizable protection, which is also more resource-intensive to the network.

Next advancement in the firewall design that will be discussed here is Zone Policy Firewall (ZPF). It works similarly to stateful firewall but introduces the concept of zones. Zone is usually associated with specific physical port in switch/router – most common example would be Outside Zone (the Internet) connected to one port, and Inside Zone (internal network) connected to another.

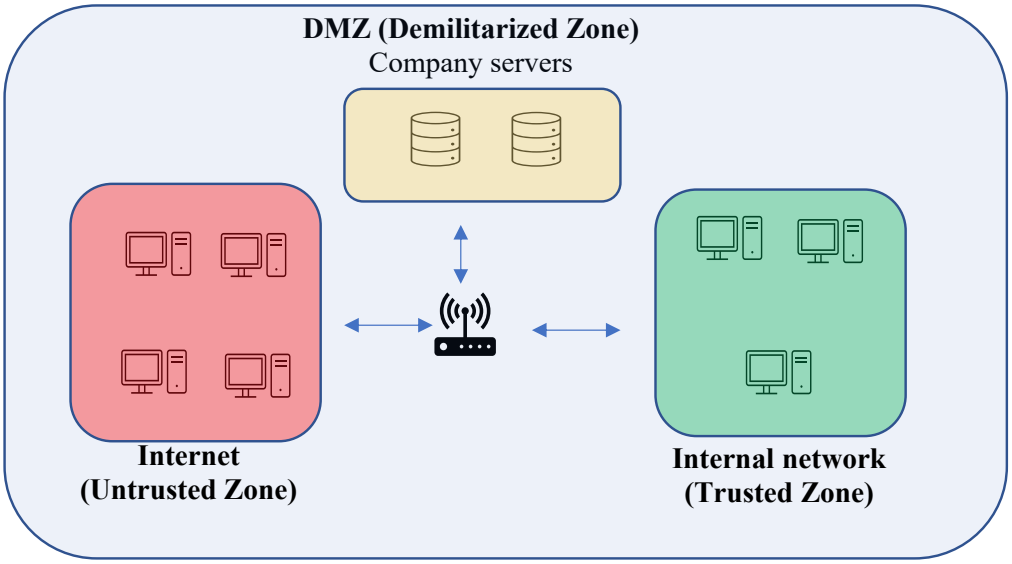


*Figure 12. Example of Zone Policy Firewall structure*

It allows for further limitations for the sake of security, i.e., by defining what kind of traffic is able to reach which zone, from where. Often there are specific protocols, ports and ACLs associated with policy set for traffic from zone to zone. Example of zones that are usually defined in ZPF are visualized on the Figure 12.

Going to the top layers of ISO/OSI model, Web Application Firewall was developed to prevent HTTP-specific attacks, as well as other attacks related to protocols of Application Layer. This type of firewall provides fine-grained security controls that prevent sensitive information leakage, block access to malicious websites based on their content or even block specific pages on that website – all that because Web Application Firewall is able to inspect traffic on the highest level of abstraction, which is rendered directly to the user. Another important point is that this type of firewall is able to prevent certain aspects of web exploitation, thus protecting HTTP server by blocking certain attacks such as XSS, Path Traversal, requests forgery, template injections or even DDoS attacks or enumeration of the server. All this protection comes unfortunately with the greater load put on network performance.

Many of the latest released firewalls are usually defined as NGFW (Next-Generation Firewalls). There is no specific definition for this type of firewall, and usually it's defined as a security device/service combining the features and functionalities of many different types of firewalls. These firewalls include deep-packet inspection (DPI), surface-level packet inspection, tracking state of connection etc. Next-generation firewalls may include other technologies as well, such as Intrusion Prevention Systems (IPS – See next section) to automatically stop attacks against your network.

## 2.2.1. IDS (Intrusion Detection) / IPS (Prevention Systems)

While firewalls remain fundamental passive protection to the perimeter of the network, they do not provide much flexibility to offer. The advantage of IPS/IDS systems in this case proactive protection, that is constantly changing its definition of threat, and also level of customization. These systems base their protection on different kinds of signatures and are designed to actively monitor the network and learn about typical network traffic versus its deviances, in order to either stop it, preventing possible attacks, or help security analysts take informative decisions in an ever-changing, challenging environment.
The power in IPS/IDS systems, which for the moment will not be told apart yet, comes from defining signatures, or definitions of malicious traffic. These could be divided into following categories [96]:

- *Conventional signatures* – most basic type, matching specific patterns (i.e., strings) in decoded traffic, or specific flags or fields in packet headers, to more advanced i.e., YARA rules, which are kind of extended grep for binary content; such signatures could be either atomic (i.e., matching single packet), or compound (i.e., used to analyze traffic from specific service/port, match strings, multi-strings (divided into multiple packets) etc.);
- *Policy-based signatures* – certain rules, or conditions, that can be applied to specific traffic in given network; such rules could concern types of traffic (based on protocol), IP addresses for source/destination, time of day where specific traffic is allowed or not and more – the main take here is that these rules are usually applied by admins, they are known and cannot be circumvented;
- *Anomaly-based signatures* – such detection requires a baseline of known (considered "normal") traffic to be established, and after such process it is able to react to traffic that stands out, based on what IPS/IDS has learned earlier. This is strong, advanced protection to be considered, although its effectiveness, as one can imagine, strongly depends on the learning period – if network has witnessed some form of malicious activity during learning process, the protection will be biased as well;
- *Reputation-based signatures* – there are known services that gather information about reputation of specific IP-addresses, registered domains, and other names related to privacy, security issues or malicious traffic; IPS/IDS systems are able to consider such rankings in their analysis of traffic. Such information does not guarantee whether specific traffic belongs to the rule/pattern that reputation is based on, but it could be a strong indicator of what can be expected in that specific case;

These categories can be used interchangeably to strengthen the quality of protection. Each of them has its strengths and weaknesses, both in terms of possible network load and provided security, or detection evasion techniques. Conventional signatures might seem identical to what firewalls do, however IPS/IDS systems employ several customizations in this case, making it much more comprehensive tool for detection of malicious traffic based on its payload. Obviously, IPS/IDS systems should be used together with firewalls to maximize the quality of protection. Other signature types

simply extend fundamental protection such as firewall. Again, for most security, these categories of signatures should be combined together. [97]
It is worth to mention that very similar concept to IPS/IDS can be applied locally, by system security measures, to react to any malicious changes by constantly analyzing system resources. [98] In case of ransomware, system resources can be analyzed in terms of CPU load, as encryption is highly intensive task for processor, disk usage, for obvious reasons, and if that correlates with bulk-replacing all files in known/common directories in filesystem, such behavior is being detected and stopped. Extensive usage of system cryptographic libraries might also be useful indicator, which added to other abovementioned factors might pose a very strong case of malicious activity.

Getting back to main subject of this section, it is time to differentiate between IPS and IDS. In general, IDS is known for being passive, and thus less resource-intensive (bringing less load to the network traffic). Its main task is to analyze traffic and create informative reports that help SOC (Security Operation Center) specialists to make informative decision as to whether malicious activity is present. On the other hand, IPS was designed to actively monitor network and prevent malicious traffic that matches specific signatures. [97] Thus, IPS does not require operator to function properly (to secure network). The distinction is now clear, and choice of system depends on specific situation. For some networks, better solution in general could be IDS, adding no latency to the network but leaving decision to human factor, whereas IPS might prove useful for other use-cases, where automatic prevention is more recommended. Another consideration is the placement of such solution.

There are two terms mostly associated with IPS, but could be applied to IDS as well, and they are – HIPS (Host-based IPS) – describes situation where IPS is implemented as part of the system. It is not concerned about general network state, but rather about security of system it is installed on. It is most effective against all threats that are directed to specific endpoint but lacks general perspective of attack that NIPS has (Network-based IPS). NIPS is a separate device connected to the network that is most effective against attacks that spread through the network. NIPS it could be implemented as in-line device – so that all traffic must first pass through its analysis before it reaches the network. Traffic that does not match signatures is usually dropped. In this case, availability of all network nodes depends on this bottleneck – which also introduces bigger latency to the network. Other mode of implementation for IPS is promiscuous mode – it is a separate endpoint that receives copy of all network traffic and makes decisions based on it – in this case there is less lag, but those decisions might not be in time if there is a lot of traffic to analyze.

### 2.2.2. Data Leak Prevention Systems

DLP is more known as a strategy to prevent data breaches or unauthorized access to sensitive information, than a specific set of tools created for that purpose. There are of course dedicated solutions, i.e., Imperva File Firewall, which indeed has a very accurate name, considering what was discussed about firewalls in above sections, but apart from that, tools like IDS/IPS, firewalls, or SIEM (Security Information and

Event Management) can cooperate to prevent extraction of sensitive data beyond company's network perimeter. [99]

Considering recent shift in ransomware operators' MO (Modus Operandi), sufficient protection against extraction of data, data breaches, disgruntled employees and various unintentional data exposure might be more important than ever to all companies around the world, no matter what size they are – in many cases serious data leaks are sufficient reason for a company to bankrupt. Few important strategies that should be applied in regard to DLP systems to prevent this situation [100] [101]:

- *Securing data in motion* – securing network perimeter so that no data can be carried beyond it without the knowledge of administrators, or without leaving a trace;
- *Securing endpoints* – keeping track of data transferred between processes locally in the system, or between groups of users; apart from logging transactions, also ensuring strong access control policies – depending on data identification/categorization, to protect information on the system;
- *Data retention* – ensuring strong encryption, solid backup policy of company's archives;
- *Monitoring breach-related sites* – good intelligence can prove useful in case of future incidents, as it might help contain current ones; although this strategy can rather be only used to warn users before more harm is done anyway;

It is impossible to avoid leaks in today's connected world. Especially because information is the key to everything and is possibly most valuable asset these days. Given the level of competition everywhere, the pace, all advancements in every branch of every business, it is natural that every company that wants to stay above the surface, must learn to gather information about rivals, as well as learn to protect own data. While this section's purpose was to familiarize the reader with concept of DLP and provide general strategies, in order to use them effectively they must be adjusted to situation of specific company. Only best choice of tailored strategy components can ensure lowest probability of possible information leakage.

## 2.3. Local protections

This section discusses common local security measures, mostly focusing on solutions for user behavioral analysis, monitoring usage of system resources, virtualization concepts that help in automated dynamic analysis, as well as policies and guidelines that might help prevent ransomware infection. First of all, it should be emphasized that although network protections are as important as endpoint protections, they are often recommended to be used together. [102] This is recommended, because many attacks combine together network exploitation with infecting systems, physical access, escalating privileges etc. When network defenses were successfully bypassed, local protections might often be the only border between malicious software and valuable information and because of that, they are often referred to as the last line of defense.

### 2.3.1. Monitoring system resources

Ransomware is notoriously known for high usage of system resources, as it tries to encrypt and exfiltrate data that often counts in gigabytes, if not terabytes. Not only a high consumption of CPU time for one process or small group of processes (and other hardware resources i.e., hard drive), but also bulk encrypting huge number of files in popular directories, changing their extensions might all be strong indicators that ransomware is undergoing its work in the system. [102]
These characteristics of ransomware make their authors look for new and smarter ways to lock users out of their data, out of their systems, all that because the simplest, most primitive idea to just encrypt all data on the system is a slow, painful process, that might easily be detected by security products. Therefore, cryptoworms try to encrypt only small parts of files, which usually makes them unusable anyway, files metadata, which holds information about location of file, size, name etc. and without that information files are often extremely difficult to recover. If that is the case, detecting ransomware behavior might be more challenging. There were experiments however [98] that aimed to prevent malicious encryption based on monitoring low-level system resources, such as CPU temperature, fan speed, memory load, CPU load etc. Effectiveness of such approach was remarkably high – for about 6000 files on the servers, only 100 were encrypted before malicious action was stopped.

### 2.3.2. Behavioral analysis

There are two subjects for behavioral analysis in the system, that might help detecting malicious actions: monitoring user's behavior (User Behavioral Analysis – UBA) i.e., through learning baseline of everyday usage, or monitoring the system itself. The latter is more focused on tracking background changes in the system, that user is not aware of, whereas the former prevents both against malicious users (insider threats) and non-malicious users that undertake dangerous actions unknowingly (i.e., being influenced by social engineers) – it does not matter really, because behavior of either type of mentioned user is deviating from the norm. [103]

User Behavioral Analysis tools [104] are essentially trackers of all kinds of activity that users can exhibit and information they leave behind when interacting with the system, i.e., authentication logs, access to resources, speed of typing [105], mouse movements (and other patterns), activity flows (commonly used applications, together with timestamps), location etc. All that data is collected and compared to historical logs, which might be later presented to security operators via SIEM software reports or used by those tools to automatically stop users or warn them about possible consequences of their actions. [106]

UBA is using AI and unsupervised learning instead of signature-based detection which might even help prevent 0-day threats. The main strength of this solution is based on determining baseline behavior of user or system – it is not easily achieved, as there are many factors that have to be taken into account. It takes time for such tool to learn what behavior is expected and if that process is contaminated by malicious

software already present on the system, or user undertaking malicious actions for whatever reason, the quality of such protection will be substantially worse.

### 2.3.3. Dynamic analysis and sandboxing

This section is about solutions for automated behavioral (dynamic) analysis of malicious samples. Using these techniques, security products are able to determine key characteristics of suspected malicious code, which executes in a separate environment (called *sandbox*), so that in case malware tries anything harmful for the system, no changes will be respected in reality (see Figure 13 below). [107] [108]

However, dynamic analysis does not need to be executed in a separate environment, the term just describes a process of running certain code, examining what data attempts to access, what changes to system make, what other processes it calls, what functions from which libraries, in general – dynamic analysis is about defining software behavior and understanding its relationship with the system based on data collected at runtime. [109] Most fundamental tools that exist for this purpose are i.e., `strace` (for analyzing system calls) [110] or `ltrace` [111] for Linux systems; Process Explorer, Process Monitor and other Microsoft Sysinternals for Windows system [112], etc. Apart from analyzing local resources, tools like Wireshark [113], Telerik Fiddler [114] and others might help establish baseline of communication with resources outside the local system, which might also give an interesting account (for example detect suspicious behaviors such as printer trying to connect to server's database).

Finally, some kind of automated analysis are black-box debuggers, usually with scripting capabilities. Those tools help determining malware behavior under the supervision of malware analyst.

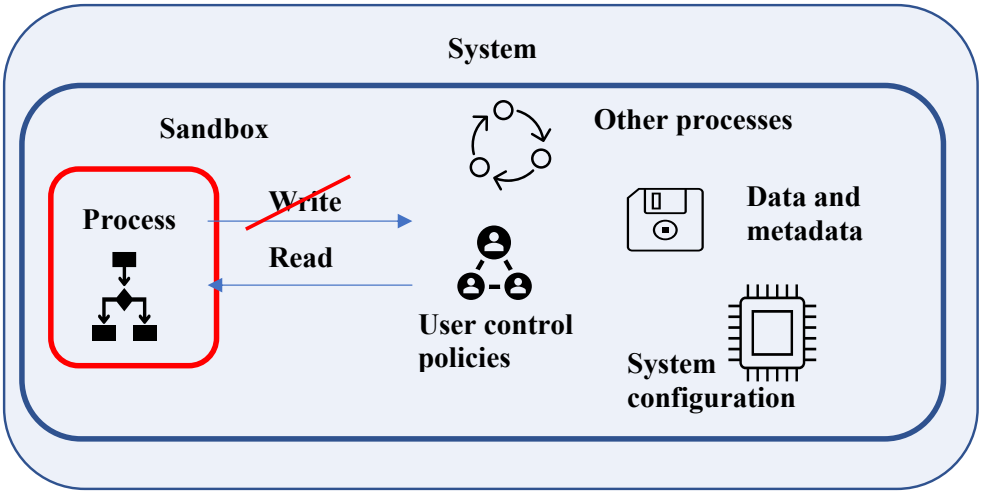


*Figure 13. Sandboxing as a separation between execution environment and the system*

In some cases, dynamic analysis is difficult to conduct, because malware creators have learned various techniques over the years, which hide certain malicious actions, if some kind of automated analysis was detected by malware. [115] There are, for instance, ways to determine whether a debugger is running in the system, either by comparing list of running processes against known debugger names, inspecting contents of certain CPU registers, or simply checking the integrity of malware code executing in the PC memory. Because of that, in some cases, antivirus software will not be able to determine whether given sample is malicious or not, so that the sample needs to be inspected manually by malware analyst, and such process is known as static analysis. [116]

Sandboxing serves as a guard between malware and filesystem and is particularly useful for dynamic analysis. Using separate environment, by either process instrumentation (i.e., docker containers) or virtualization (virtual machines), given malware can be safely executed and its baseline behavior determined. On the other hand, however, there are common ways to detect whether process is running inside some sort of separate environment, and that is the case, again, malware hides its malicious actions to remain undetected. [117] [118] One of the tools that helps integrate dynamic analysis and sandboxing is Cuckoo – dedicated Linux distribution for this purpose.

### 2.3.4. Guidelines, common policies

Possibly the most affordable way of protecting system from infecting with ransomware is keeping to the general guidelines based on common sense and vast experience of security researchers.

First thing that should be implemented in the system is strong access control policy, meaning it will allow access to certain resources for other resources/users that should have it, and deny access for those who should not. Simplest case of such policy is Access Control List (ACL), which allows for either blacklisting or whitelisting certain resources. In certain cases, there should be implemented more sophisticated methods for access control, defining specific roles in the system, privileges/permissions, clearance levels etc. It all depends on specific use case, and if such policy is in place, it could prevent unauthorized access to valuable data from outside, even if other protections failed. [119] Group Policy Object (GPO) restrictions are Windows method for access control to system resources. It provides granular control over execution of files, so that for instance, anything trying to execute inside `%AppData%` or `%LocalAppData%` is automatically blocked, which is extremely common scenario in ransomware attacks. Another example is disabling ability to run executables from e-mail attachments. [120] In general, deep understanding of ransomware structure brings even the most elementary solutions that might help with infection prevention. [121]

As to other Microsoft Windows security measures, enforcing UAC might stop users from executing malicious file, even though this protection is commonly considered an impediment to everyday work, it gives an additional proverbial second for a user, to

consider risk and whether it is worth to execute certain program. Raising user awareness is a key concept in preventing certain types of attacks. Forcing system to display full file extensions for all types of files might prevent situation where user accidentally executes malicious file that was masking as i.e., PDF document. As to malicious scripts, some kind of protection gives changing default action from executing script with interpreter for given language, to opening file in text editor. Once user is able to contents of script first, it might frighten them out from trying to execute it, even if that person is not proficient with IT. Disabling Windows Host Script (WSH) [122] might also successfully prevent infection by decreasing possibilities by which system might be infected with malicious macros/scripts. Finally, there are common browser extensions, i.e., adblockers, dynamic content blockers, that might make browsing Internet less comfortable for the user, but on the other hand they might bring additional security layer that will help preventing system infection. Although in this case even more important is appropriate awareness training, that will explain to user why such protections are extremely valuable in certain situations from security perspective.

Some of abovementioned protections might work as "killswitch" of a kind, to ransomware attacks. Apart from preventing execution from certain folders on system, for instance renaming default names of system services that are responsible for backups, shadow copies, might block ransomware from executing. Again, the key to protecting system is constantly updated knowledge of common ransomware attacks, so that certain rules and protections will be constantly adjusted to this ever-changing environment, but also strong access-control policies and above all – exceptional value usually bring security awareness trainings that help explain users what threats they might face and help them get prepared for those threats.

## 2.4. Summary

In this chapter there were two categories of protections discussed – network security measures as a broader perspective, as well as local protections, which work more efficiently as endpoint protections, often last line of defense. Both sides are critical to protect company's internal network from all sorts of threats. They often complement each other, as summarized below:

Network protections:

- Signature based detection
- Signatures updated regularly
- No protection from zero-day attacks
- Resource intensive to deploy and run
- Broader perspective of attack

Endpoint protections:

- User Behavioral Analysis (UBA)

- Algorithmic approach
- No signatures required, zero-day protection
- Low impact on end points
- Local perspective

Again, these two categories provide a more complete picture of attack, allowing victim to act fast, with comprehensive intelligence based on multiple sources and minimalize possible negative outcomes of an attack.

# 3. Testing efficiency of dedicated protections against ransomware

With today's rate of advancements in every branch of information technology it is quite natural that all sorts of bugs, mistakes and misconceptions appear at every stage of development, and vulnerabilities are being constantly discovered and patched. There is also the legendary human error that will always be the case in any system, which leads to countless social engineering tactics being used for almost all attacks in different forms.

Having such complex reality, it is rational to undertake any security measures to make computer system more secure. From basic firewall and local access control to advanced network protections, employing all sorts of algorithms, AI solutions, all to protect network perimeter and internal systems.

It is, unfortunately, hardly enough to protect from zero-day attacks, or social engineering attacks. Thus, in certain cases local protections should be considered as well. Those include automated behavioral analysis in separate environment (i.e., sandboxing) of any new (or from unknown source) executable files; resource management protection, preventing a situation where, for instance, all files in certain directory are being replaced by their encrypted copies. Apart from analyzing usage of system resources, establishing user behavior baseline in order to detect conspicuous events, there are also solutions that aim to make user aware of threats and even stop them from undertaking certain actions. These solutions can be in a form of AV module, browser extension or external program that is dedicated to preventing ransomware. Such products might be helpful in mitigation of social engineering attacks, as an addition to profound cybersecurity-awareness trainings.

This part of thesis is dedicated to testing various products that offer protection against malware attacks, or ransomware in particular, on many different levels. They are placed in different scenarios against real malicious samples, in order to determine which of them are most efficient, what functionality they offer, and when each of solutions can be used to prevent infection. This research is devoted to testing response of popular products against arbitrary set of ransomware samples. Similar approaches to test security products against malware in general have been conducted in the past. To access frameworks that were used for such tests and their results, the reader can refer to [123], [124].

## 3.1. Test environment structure

Environment for the tests is a small internal network consisting of two virtual machines. One of them is running Windows 10 (version: 10.0.19043.985), and this machine serves as a victim in conducted experiment scenarios. The other one is Ubuntu (20.04.1 LTS focal) and the role of this machine is to serve (as simple HTTP server) malicious files that are being downloaded by the victim. These machines are connected by VirtualBox (version: 6.1.22). Complete set up of that network is shown in section 4.1.1. below.
In these experiments, malicious server is hosting a compromised website called *bestnewserver.com*, which is serving their RSS Reader application in two forms:

- plain installer;
- archived, encrypted installer;

Victim is downloading RSS Reader, which, in reality, is an arbitrarily chosen ransomware sample from a prepared set of samples. The response of selected security software is the subject of test here. The response is measured against downloaded executable, downloaded encrypted executable, as well malicious script in invoice for this RSS Reader, which downloads one of ransomware samples and tries to execute it. That gives about three different scenarios, tested for each of selected samples, and each of selected protections.

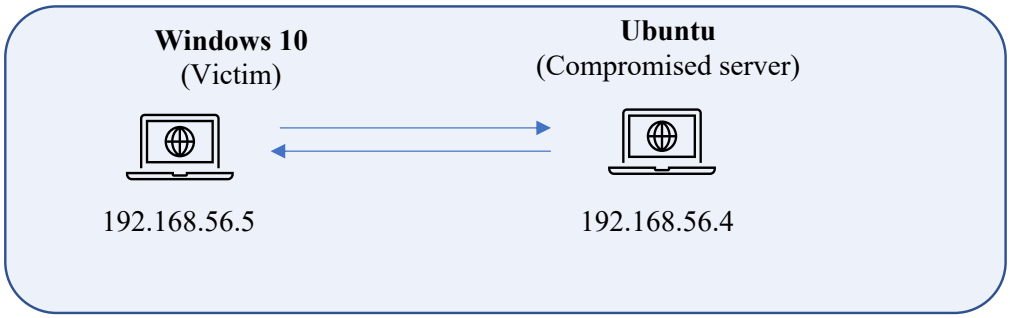


*Figure 14. Example network used in tests*

Test are measuring response of about 11 different security products, from two categories: general Antivirus software, and specialized anti-ransomware tools. All these products are being hit by 20 different ransomware samples, plus one legitimate installer for some normal, non-malicious software (for the sake of sanity).
First things first, each of those scenarios will be shortly presented below. At the end of this section there are results of the tests, discussion of the outcome, and final remarks.

### 3.1.1. Lab preparation

In order to set up internal network in VirtualBox, both machines must be attached to *Host-only Adapter*, and have the same number assigned (i.e., *VirtualBox Host-Only Ethernet Adapter #2*).

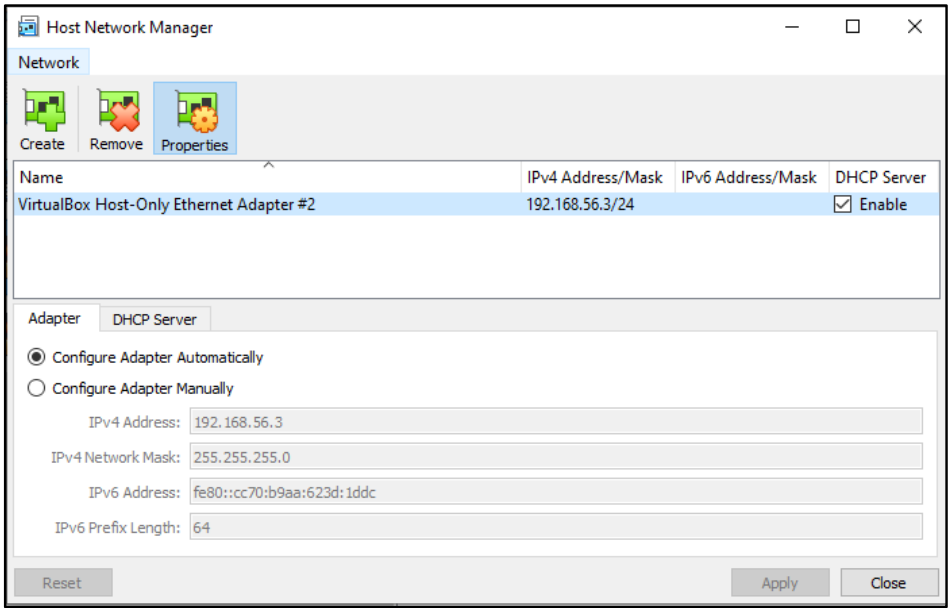


*Figure 15. Example Host Network Manager configuration in VirtualBox*

This setting can be changed in Network settings of that particular VM (Virtual Machine). First step, however, is to access Host Network Manager settings under File, and set up internal network in the first place. Firstly, click button *Create*, then check the box under DHCP server in order have automatic IP address assignment. Option *Configure Adapter Automatically* at the bottom part of window should also be checked. Figure 15 represents example configuration.

Under the tab for DHCP server, it is also best to leave everything as default, only making sure that enable DHCP server is checked there as well (see Figure 16 below).

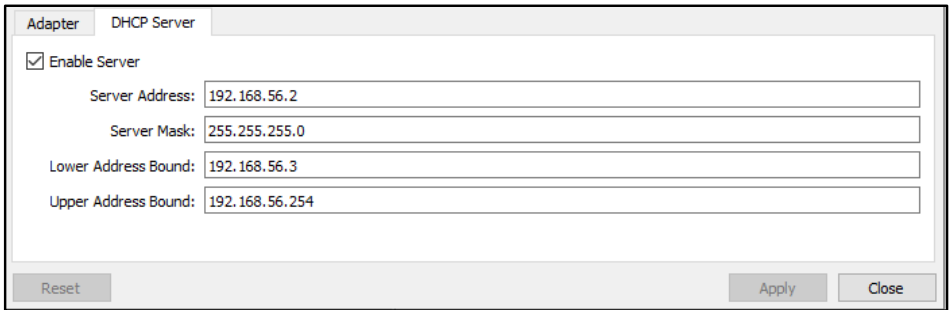


*Figure 16. DHCP configuration in Host Network Manager in VirtualBox*

Once that is done, machines will be able to ping each other. Next step is to spin up the malicious HTTP server, which can be done with Python using one line below.

```
$> hostname -I
192.168.56.4
$> sudo iptables -A INPUT -p tcp –dport 80 -j ACCEPT
$> sudo ufw reload
Firewall reloaded
$> sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) …
```

*Figure 17. Configuration and firewall and server set up*

One thing to remember is that firstly appropriate port must be opened in firewall, otherwise victim will not be able to reach the server. All relevant commands are visible on Figure 17.

In order to test the set-up, victim downloads News RSS application from *bestnewsever.com* (this domain pointing to malicious address can be set up in *hosts* file on Windows). In this test application that is being served in reality by malicious server is PolyRansom. No protections were installed, Windows Defender disabled as well. Despite that, Windows Edge is still able to figure out that downloaded software might contain viruses – this is due to Windows Smart Screen (see Figure 18 below).

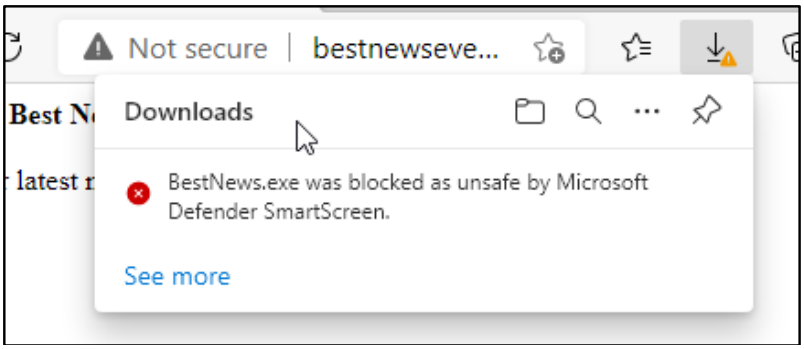


*Figure 18. Microsoft Smart Screen warning about threat*

Next steps include clicking *See more*, allow application, then downloading application and trying to execute it. If everything goes well, next warning should appear from Smart Screen (see Figure 19):

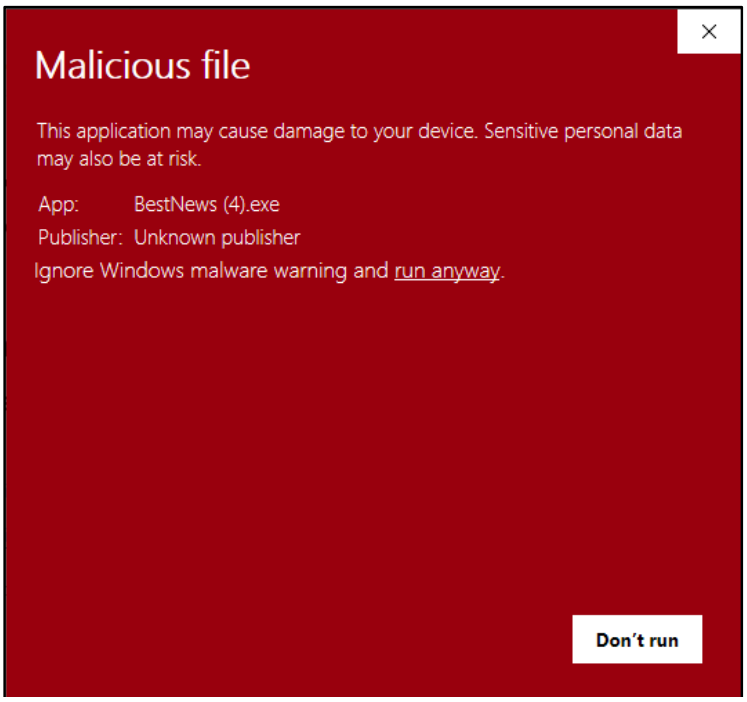


*Figure 19. Microsoft Smart Screen protecting against executing malicious fill*

If this warning will also be ignored, and application run, after a few seconds from run a PolyRansom ransom note will appear (see figure 20). In this case, ransomware was executed successfully and access to files probably lost forever.

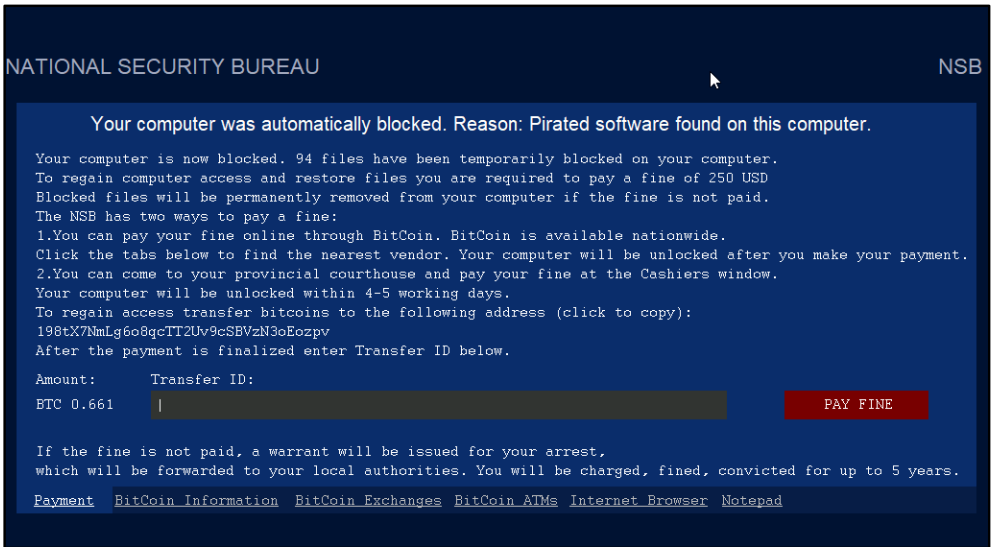


*Figure 20. PolyRansom ransom note, as executed in tests*

Please mind that some of the chosen samples might not have an immediate, visible effect on system or files. Most of these samples are already a few years old, they are

being executed in virtual environment after all, which might trigger their anti-debugging functionalities, and some of C&C servers for these ransomwares might not be available anymore, leaving those samples kill-switched.

In these experiments, even if system did not respond in visible way to executed ransomware, but there was no reaction from tested protection software – this is pointed as successful infection no matter whether at the low-level ransomware has actually started doing its malicious business, or not. With these assumptions, let us discuss test scenarios in detail and move on to test results.

### 3.1.2. First scenario – download executable from malicious server

It all starts with one compromised server. In this scenario, victim is trying to download brand new RSS Reader, called BestNewsEver, from the Internet. From the technical point for implementation of this scenario, the HTTP server on Ubuntu was set up with Python module, with firewall configured prior to launching it (see Figure 17**)**.
With this set-up, the victim (on Windows 10) is accessing the HTTP server on bestnewsever.com (Server IP was added to *hosts* file). Website is hosting installer to mentioned RSS Reader, which can be downloaded using link on the webpage (see Figure 21 below).

Victim is accessing the website using Microsoft Edge web browser. As the download begins, victim is warned by Microsoft SmartScreen that application might contain viruses. The subject of the test in this scenario is response from installed protections. It is being observed whether protection software has prevented infection after user ignores Smart Screen warnings, downloading and executing BestNews.exe. That reaction in form of threat signature, if got detected, will be later presented in results of research.

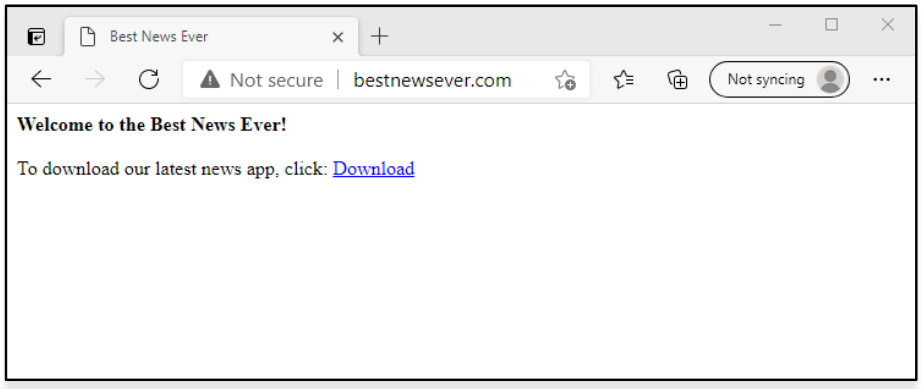


*Figure 21. Malicious web server rendered in victim's browser*

### 3.1.3. Second scenario – download encrypted archive with executable

Modified first scenario, where the only difference is that victim is downloading encrypted 7zip archive from the website mentioned in first scenario. Password is given on the website; victim decompresses the archive and tries to execute application. Encryption here serves two purposes: firstly, some users might want, or have to, download encrypted archives, as some of the programs are shared in this form on the Internet. Secondly, this allows to bypass some of the protections in tested solutions.

Similarly, in first scenario all victim actions were being monitored by chosen protection software, and the quality of protection is the subject of test. That is the case here as well.

### 3.1.4. Third scenario – e-mail attachment with malicious invoice

In this scenario victim receives an e-mail with invoice for downloaded application (Figure 22).

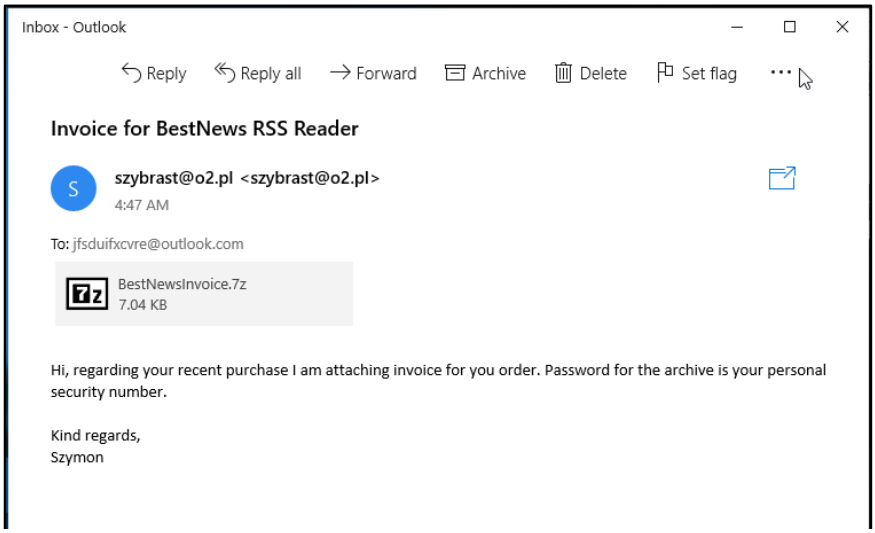


*Figure 22. Malicious mail that contains ransomware, as opened by victim*

The invoice is in fact Libre Office Writer document (`.odt`) encrypted in 7zip archive. That document contains malicious VBA macro that downloads and tries to execute ransomware.
Victim downloads and opens the attachment, not being aware of scripts executing under the hood. Similarly, reaction of installed protection software is the subject of test here. The macro is in fact modified version of [125] – see figure 23 for details.

```
Private Declare Function URLDownloadToFile Lib "urlmon" Alias "URLDownloadToFileA" _
                        (ByVal pCaller As Long, _
                         ByVal szURL As String, _
                         ByVal szFileName As String, _
                         ByVal dwReserved As Long, _
                         ByVal lpfnCB As Long) As Long

Sub DownloadToFile
    URLDownloadToFile(0,_
    "http://bestnewsever.com/BestNews.exe",_
    Environ$("USERPROFILE") & "/Desktop/BestNews.exe",_
    0, 0)
    Call Shell(Environ$("USERPROFILE") & "/Desktop/BestNews.exe", vbNormalFocus)
End Sub
```

*Figure 23. Malicious VBA script example*

This test assumes that executing macros protections in Libre Office have been disabled in order to make things simpler. This is not by default; however, these countermeasures can usually be circumvented if there is specific vulnerability for Libre Office known. For settings that are recommended to turn off for this scenario, see Figure 24.

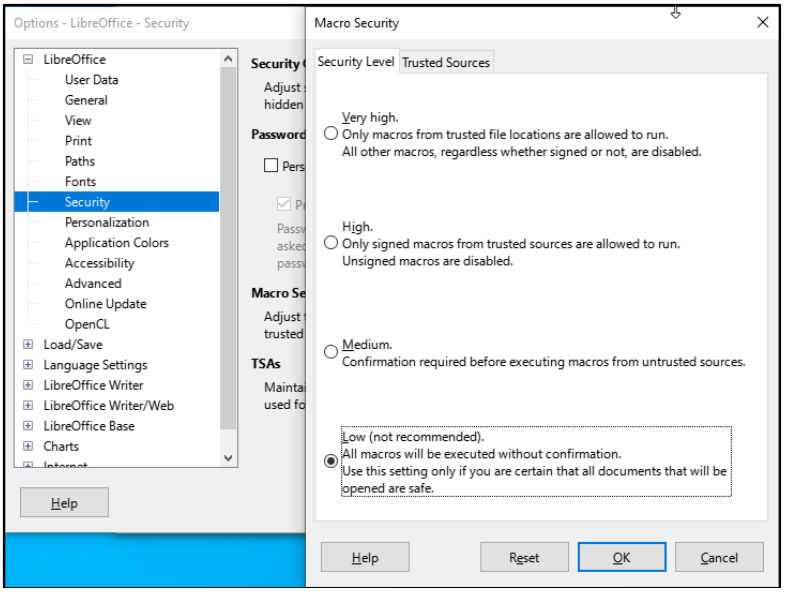


*Figure 24. Disabling dynamic content protections in LibreOffice - for the purpose of testing*

## 3.2. Test data

This section shortly describes chosen ransomware samples and chosen protections accordingly. Samples were taken from publicly available services, such as [126]. Table 1 gives details on each of the chosen samples.

| Nr. | Name | First appeared | Rating[16] | SHA2-256 |
|---|---|---|---|---|
| [1] | 7ev3n | 2016 | 84% | 7d373ccb96d1dbb1856ef31afa87c2112a0c1795a796ab01cb154700288afec5 |
| [2] | BadRabbit | 2017 | 88% | 630325cac09ac3fab908f903e3b00d0dadd5fdaa0875ed8496fcbb97a558d0da |
| [3] | Birele | 2011 | 85% | b2dcfdf9e7b09f2aa5004668370e77982963ace820e7285b2e264a294441da23 |
| [4] | Cryptowall | 2014 | 88% | 45317968759d3e37282ceb75149f627d648534c5b4685f6da3966d8f6fca662d |
| [5] | DeriaLock | 2016 | 72% | 4f5bff64160044d9a769ab277ff85ba954e2a2e182c6da4d0672790cf1d48309 |
| [6] | Fantom | 2016 | 82% | f4234a501edcd30d3bc15c983692c9450383b73bdd310059405c5e3a43cc730b |
| [7] | Krotten | 2011 | 92% | e79f164ccc75a5d5c032b4c5a96d6ad7604faffb28afe77bc29b9173fa3543e4 |
| [8] | Mamba | 2021 | 84% | 2ecc525177ed52c74ddaaacd47ad513450e85c01f2616bf179be5b576164bf63 |
| [9] | Matsnu | 2011 | 84% | 7634433f8fcf4d13fb46d680802e48eeb160e0f51e228cae058436845976381e |
| [10] | Non-Malicious Program[17] | | 20% | 546c5b6af3296c1eb1dc358f0aa7702189e7599feab537204b23ef52a94e1aca |
| [11] | NoMoreRansom | 2018 | 87% | 2aab13d49b60001de3aa47fb8f7251a973faa7f3c53a3840cdf5fd0b26e9a09f |
| [12] | Petrwrap | 2017 | 95% | 027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745 |
| [13] | PolyRansom | 2020 | 87% | 1898f2cae1e3824cb0f7fd5368171a33aba179e63501e480b4da9ea05ebf0423 |

[16] Rating is measured as a ratio taken from VirusTotal for particular sample, and multiplied by 100%
[17] Arbitrarily chosen installer for some legitimate media conversion software, used as a "sanity check"

| Nr. | Name | First appeared | Rating[18] | SHA2-256 |
|---|---|---|---|---|
| [14] | PowerPoint | 2020 | 91% | 9c3f8df80193c085912c9950c58051ae77c321975784cc069ceacd4f57d5861d |
| [15] | Radamant | 2020 | 84% | 2c4c8066a1a7dfdf42c57ff4f9016f1ba05bcb004ff8b0ffc0989165d2ad30e2 |
| [16] | Rex | 2016 | 60% | 762a4f2bf5ea4ff72fce674da1adf29f0b9357be18de4cd992d79198c56bb514 |
| [17] | Thanos | 2020 | 85% | 5d40615701c48a122e44f831e7c8643d07765629a83b15d090587f469c77693d |
| [18] | Vipasana | 2016 | 87% | e49778d20a2f9b1f8b00ddd24b6bcee81af381ed02cfe0a3c9ab3111cda5f573 |
| [19] | ViraLock | 2016 | 91% | 418395efd269bc6534e02c92cb2c568631ada6e54bc55ade4e4a5986605ff786 |
| [20] | Winlocker.VB6.Blacksod | 2017 | 63% | 68b0e1932f3b4439865be848c2d592d5174dbdbaab8f66104a0e5b28c928ee0c |
| [21] | Xyeta | 2019 | 84% | 3ab3833e31e4083026421c641304369acfd31b957b78af81f3c6ef4968ef0e15 |
| [22] | HiddenTear | 2015 | 78% | 615f56c71a00903eb4d8e802a531d56b90eeba3fde4a06fae1cc7c6ec030f98f |

*TABLE 1. TEST SUBJECTS – RANSOMWARE SAMPLES*

These samples have been uploaded to Virus Total for general comparison. These results can be found on virustotal.com/gui/file/{SHA2-256} where SHA2-256 is the value found in column SHA2-256 with corresponding name in Table 1.

| Test subject | Comments |
|---|---|
| Bitdefender Total Security | Does not even allow to save malicious files. It prevents downloading files it finds malicious, without the option to unblock and download on user's responsibility. |

[18] Rating is measured as a ratio taken from VirusTotal for particular sample, and multiplied by 100%

| Test subject | Comments |
|---|---|
| AVG Antivirus Free | Advanced protection against ransomware. Protects most important folders (Desktop, Documents, Downloads) against any unwanted changes (encryption). Has VirusChest for storing detected malware. After a few downloads from malicious website, it automatically sends RST packets to terminate TCP session. It seems that it learned to blacklist that particular URL. It did not apply to downloading encrypted ransomware (scenario 2), though – these could be downloaded, but not executed |
| McAfee Total Protection | Does not share a lot of information about identified threat, thus leaving user unaware of class/category of it. Prevents executing malicious downloaded files, no option to make exception in case of tested samples. |
| Norton 360 | It would not even allow to download, deleting temporary download files (before even entire file was able to be downloaded). |
| Windows Defender | The only from tested solutions, that leaves a possibility to run infected program on user's responsibility. However, even when allowed to run ransomware does not start encryption. |
| BullGuard Antivirus | Does not allow downloading tested samples at all, with no option to apply rule exception. Together with blocking download, "malicious server" was receiving 5-10 additional requests for ransomware, each time. It happened for every ransomware sample, only for this antivirus |

*Table 2. Test subjects – first group, antivirus software*

| Test subject | Comments |
|---|---|
| Kaspersky Anti-Ransomware Tool | Not very informative signatures, high efficiency of protection and usability |
| AppCheck - AntiRansomware | Analyzes system behavior, in some cases where ransomware started encrypting files on desktop, it managed to stop and reverse the process. In other cases, where ransomware had direct effect on the system / critical system settings, it did not react to malicious actions |
| MalwareBytes AntiRansomware | Works similarly to AV Software, has many different anti-malware modules. Gives clear signature names and information to the user |
| TrendMicro ransomBuster | Appears a very solid anti-ransomware tool that prevents executing ransomware based on system's behavior |
| GridinSoft Anti-Ransomware | Similarly, to other tools in this category, tries to analyze system behavior and rest to any unwanted changes in the filesystem. Achieved very low results in the tested scenarios. |

*Table 3. Test subjects – second group, specialized anti-ransomware tools*

Next part is the presentation of chosen protections. They are divided into two categories – multi-layered malware protection – AV software with comprehensive set of modules (Table 2) and specialized anti-ransomware tools (Table 3). A few more solutions that have been tested, but did not appear to provide any direct protection against ransomware attacks (despite claiming on their websites that they protect against ransomware) or were not updated anymore: RansomOff, Cybersight RansomStopper, NeuShield Data Sentinel, CryptoPrevent Anti-malware (this solution had actually a few advanced settings, such as stopping execution of files disguised as media i.e. PDF files, however it did not prevent execution of any of tested samples).

## 3.3. Test results

| RW AV | Kaspersky Anti-Ransomware Tool | AppCheck - AntiRansomware | MalwareBytes AntiRansomware | TrendMicro ransomBuster | GridinSoft Anti-Ransomware |
|---|---|---|---|---|---|
| [1] | PDM.Trojan.Win32.Bazon.a | X | Ransom.7ev3n | Ransomware | X |
| [2] | PDM.Trojan.Win32.Bazon.a | Ransomware | Ransom.BadRabbit | Ransomware | X |
| [3] | PDM.Trojan.Win32.Bazon.a | X | Malware.AI.4289006885 | Ransomware | X |
| [4] | PDM.Trojan.Win32.Bazon.a | X | Trojan.Zbot.KE | Ransomware | X |
| [5] | PDM.Trojan.Win32.Bazon.a | Ransomware | Ransom.DeriaLock | Ransomware | X |
| [6] | PDM.Trojan.Win32.Bazon.a | X | Ransom.Fantom | Ransomware | X |
| [7] | PDM.Trojan.Win32.Bazon.a | X | Trojan.Agent | Ransomware | X |
| [8] | PDM.Trojan.Win32.Bazon.a | X | Ransom.FileCryptor | Ransomware | X |
| [9] | PDM.Trojan.Win32.Bazon.a | X | Ransom.Agent.ED | Ransomware | X |
| [10] | X | X | X | X | X |
| [11] | PDM.Trojan.Win32.Bazon.a | X | Ransom.Cerber | Ransomware | X |

| RW AV | Kaspersky Anti-Ransomware Tool | AppCheck - AntiRansomware | MalwareBytes AntiRansomware | TrendMicro ransomBuster | GridinSoft Anti-Ransomware |
|---|---|---|---|---|---|
| [12] | Cannot execute | Cannot execute | Ransom.Petya.EB | Ransomware | Cannot execute |
| [13] | PDM.Trojan.Win32.Bazon.a | X | Trojan.Agent.RND1Gen | Threat | X |
| [14] | PDM.Trojan.Win32.Bazon.a | Ransomware | Ransom.FileCryptor | Ransomware | X |
| [15] | PDM.Trojan.Win32.Bazon.a | X | Ransom.Radamant | Ransomware | X |
| [16] | Cannot execute | Cannot execute | Cannot execute | Ransomware | Cannot execute |
| [17] | PDM.Trojan.Win32.Bazon.a | Ransomware | Trojan.Crypt.MSIL.Generic | Ransomware | Ransomware |
| [18] | PDM.Trojan.Win32.Bazon.a | Ransomware | Ransom.VawTrak | Ransomware | X |
| [19] | PDM.Trojan.Win32.Bazon.a | X | Trojan.Agent.RND1Gen | Threat | X |
| [20] | PDM.Trojan.Win32.Bazon.a | X | Malware.AI.4206510018 | Threat | X |
| [21] | PDM.Trojan.Win32.Bazon.a | Ransomware | Spyware.PasswordStealer.XGen | Ransomware | X |
| [22] | PDM.Trojan.Win32.Bazon.a | X | Ransom.HiddenTear.Generic | Ransomware | X |
| [23][19] | PDM.Trojan.Win32.Bazon.a | X | Exploit payload file | Ransomware | X |

*Table 4. Test results – specialized tools*

This section covers results of presented test scenarios. For rows numbered 1-22, these are total 22 test subjects, ransomware samples chosen at random from public repositories. Each of them has been tested in the first scenario, in the form of

[19] Scenario 3 – malicious VBA script trying to download and execute Birele (sample nr. 3)

executable downloaded directly, and second scenario, in the form of downloaded encrypted archive that contain the sample.

| RW AV | Bitdefender | Avast/AVG | McAfee | Norton | Windows Defender | BullGuard |
|---|---|---|---|---|---|---|
| [1] | Trojan.GenericKD.45257720 | Win32:Ransom-AXU | Suspect! 9f8bc96c96d4 | Ransom.Seven | Ransom:Win32/Empercrypt.A | HEUR/AGEN.1100584 |
| [2] | Trojan.GenericKD.6139887 | Win32:Malware-gen | Suspect! 9f8bc96c96d4 | Ransom.BadRabbit | Ransom:Win32/Tibbar.A | HEUR/AGEN.1123435 |
| [3] | Gen:Variant.Razy.445801 | Win32:Evo-gen | Suspect! 9f8bc96c96d4 | Heur.AdvML.B | Ransom:Win32/Genasom.FH | TR/BAS.Samca.fyzpg |
| [4] | Trojan.GenericKD.2080196 | Win32:Cryptowall-B | Suspect! 9f8bc96c96d4 | Ransom.Cryptodefense | Ransom:Win32/Crowti.A | TR/Crypt.XPACK.134743 |
| [5] | Gen:Variant.MSILPerseus.65145 | MSIL:Ransom-T | Suspect! 9f8bc96c96d4 | Heur.AdvML.M | Ransom:Win32/Dereilock | TR/Genasom.wzara |
| [6] | Trojan.Agent.BXUU | Win32:Malware-gen | Suspect! 9f8bc96c96d4 | Ransom.Fantom | Ransom:MSIL/Fantomcrypt.A | TR/Downloader.axzu |
| [7] | Gen:Trojan.RegistryDisabler.dqX@aOUJXbmi | Win32:GenMalicious-JTR | Suspect! 9f8bc96c96d4 | Trojan.StartPage | Trojan:Win32/Krotten.B | TR/Sirery.A |
| [8] | Gen:Variant.Ransom:HDDCrypt.1 | Win64:Malware-gen | Suspect! 9f8bc96c96d4 | Ransom.HDDCryptor | Ransom:Win32/Mambretor.A | TR/FileCoder.rrsau |
| [9] | Gen:Variant.Symmi.23512 | Win32:Crypt-PHP | Suspect! 9f8bc96c96d4 | Trojan.Betabot!gm | Trojan:Win32/Matsnu | HEUR/AGEN.1127899 |
| [10] | X | FileRepMalware [PUP] | PUP | X | PUA:Win32/ICBundler | X |

| RW AV | Bitdefender | Avast/AVG | McAfee | Norton | Windows Defender | BullGuard |
|---|---|---|---|---|---|---|
| [11] | Gen:Variant.Ransom.Shade.22 | Win32:Malware-gen | Suspect!9f8bc96c96d4 | Packed.Generic.459 | Ransom:Win32/Troldesh.A | HEUR/AGEN.1106830 |
| [12] | Trojan.Ransom.GoldenEye.B | MBR:Ransom-C | Real Protect-LS!71b6a493388e | Ransom.Petya | Ransom:Win32/Petya | TR/Ransom.ME.12 |
| [13] | Win32.Virlock.Gen.4 | Win32:VirLock | Suspect!63210f8f1dde | W32.Virlock | Virus:Win32/Nabucur.A | TR/Crypt.XPACK.Gen |
| [14] | Gen:Variant.Mikey.74604 | MBR:Ransom-A | Suspect!70108103a531 | Trojan.Bootlock.B | Trojan:MSIL/Cryptor | BOO/Ransom.AB |
| [15] | Generic.Malware.SF.F3BA6A8A | Win32:Malware-gen | Suspect!6152709e741c | Ransom.Radamant | Ransom:Win32/Radamcrypt!rfn | TR/Proxy.Gen |
| [16] | Gen:Variant.Trojan.Linux.DDos.1 | ELF:Rex-A | Suspect!6152709e741c | Trojan Horse | Ransom:Linux/Daco.A | LINUX/Rex.mjss |
| [17] | Gen:Variant.Ransom.Thanos.3 | Win32:RansomX-gen | Suspect!6152709e741c | Ransom.Cryptolocker | Ransom:MSIL/Thanos.AR!MSR | HEUR/AGEN.1140759 |
| [18] | Generic.Ransom.Cryak.CD721E02 | Win32:Malware-gen | Real Protect-LS!e01e11dca5e8 | Ransom.Cryakl | Ransom:Win32/Criakl.D | HEUR/AGEN.1121085 |
| [19] | Win32.Virlock.Gen4 | Win32:WirLock | Suspect!6152709e741c | W32.Virlock | Virus:Win32/Nabucur.A | TR/Crypt.XPACK.Gen |
| [20] | Gen:Variant.Strictor.112354 | FileRep Malware | Real Protect-LS!e01e11dca5e8 | Trojan.Gen.2 | Trojan:Win32/Skeeyah.A!rfn | TR/AD.Skeeyah.ixvmj |
| [21] | Trojan.Generic.5494985 | FileRep Malware | Suspect!9d15a3b31460 | Downloader.Lofog!gen4 | Ransom:Win32/Blocker | TR/Downloader.Gen3 |

| RW AV | Bitdefender | Avast/AVG | McAfee | Norton | Windows Defender | BullGuard |
|---|---|---|---|---|---|---|
| [22] | Trojan.GenericKD.9929187 | FileRep Malware | Ransomware-FTD!C1FACBE2847E | Ransom.HiddenTear!g1 | Ransom:MSIL/Ryzerlo.A | HEUR/AGEN.1100992 |
| [23] [20] | Gen:Variant.Razy.445801 | Win32:Evo-gen | Suspect!9f8bc96c96d4 | Heur.AdvML.B | Ransom:Win32/Genasom.FH | HEUR/AGEN.1123435 |

*Table 5. Test results – antivirus software products*

If security product has detected both scenarios, only one signature is written to particular cell. Otherwise, there is information which scenario was successfully prevented, and which was not. As to scenario 23, it is test of scenario 3, where victim downloads malicious Libre Office document (that tries to download and execute Birele ransomware). For results of generic AV programs, see Table 5, as to specialized anti-ransomware tools, see Table 4.

## 3.4. Test results analysis

In terms of AV software, there is 100% efficiency of detection of tested samples. Specialized anti-ransomware tools have a bit lower results – about 65% – so their credibility is therefore substantially lower. Most of specialized solutions have already been integrated into AV software, and a number of tools has not been updated for some time. What is more interesting here, however, is the diversity of reported threats. Each of the tools has their own system for identification of threats, different naming convention of signatures, and yet, there is one thing in common amongst all of them – type of threat that is being reported to the user. Figure 25 summarizes the findings in terms of categorization of reported signatures, focusing on ransomware sample, compared to other signatures (i.e., Trojan, malware, generic threat etc.) and – of course – threat not being reported at all. Figure 26 presents the same data, only categorized by tested security product.

Second comparison shows findings identified by tested products in a form of cartesian product, where X axis represents tested samples, and Y axis represents accuracy of signature naming convention, as compared to accuracy of security product that reached the best score.
Score assigned to particular type of signature was presented below. Visualization of this comparison is presented in the Figure 27 below. The formula defining results for the following diagrams:

*(Accuracy) % = (sum of findings) / (total sum per sample),*

[20] Scenario 3 – malicious VBA script trying to download and execute Birele (sample nr. 3)

where the sum of findings is a total of marks claimed to specific sample by, respectively:

6 – Ransomware/Cryptor
5 – Trojan
4 – malware / Virus
3 – Generic / Cryptor / FileCoder
2 – Downloader / Spyware
1 – Other
0 – Not identified

Total points that could be assigned in this test:

| | | |
|---|---|---|
| Max Points: | 66 | *(per sample)* |
| Max Points: | 126 | *(per security product)* |

*Table 6. Test results – max points in both categories*

More detailed results for Figures 26 and 27 above can be found in Table 7 and Table 8 below. In both cases results have been sorted by effectiveness.

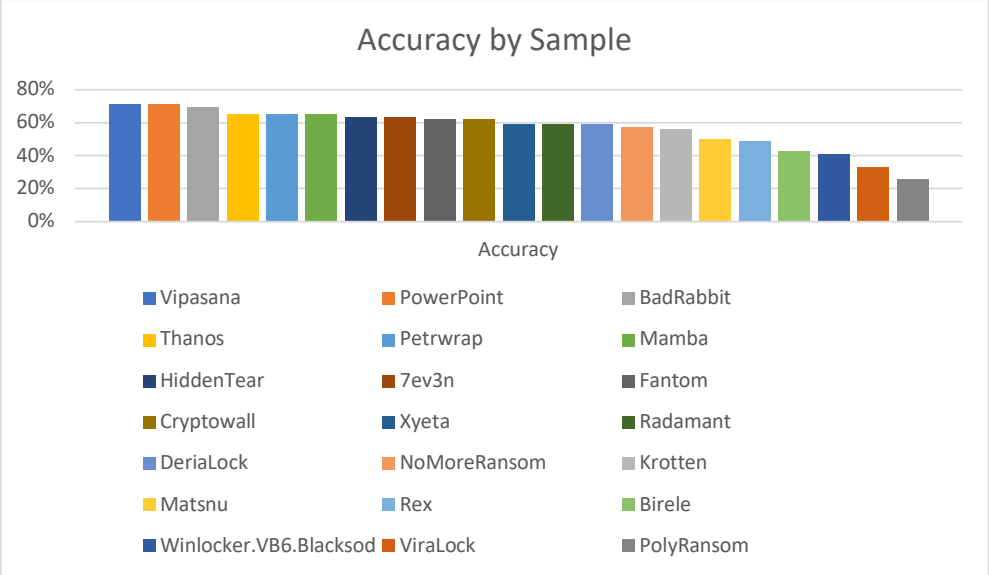


*Figure 25.Test results as categorized by ransomware sample*

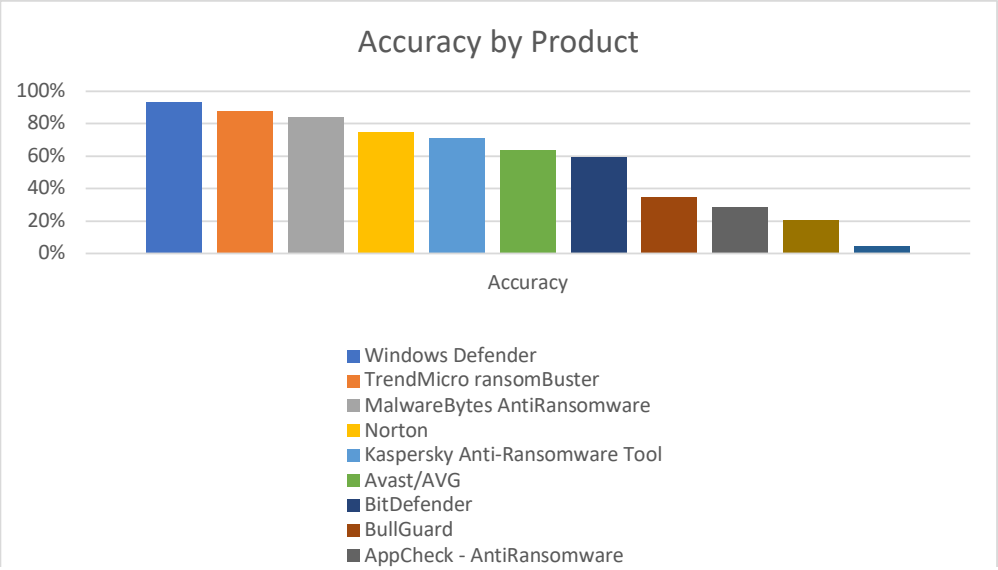


*Figure 26. Test results categorized by security solution*

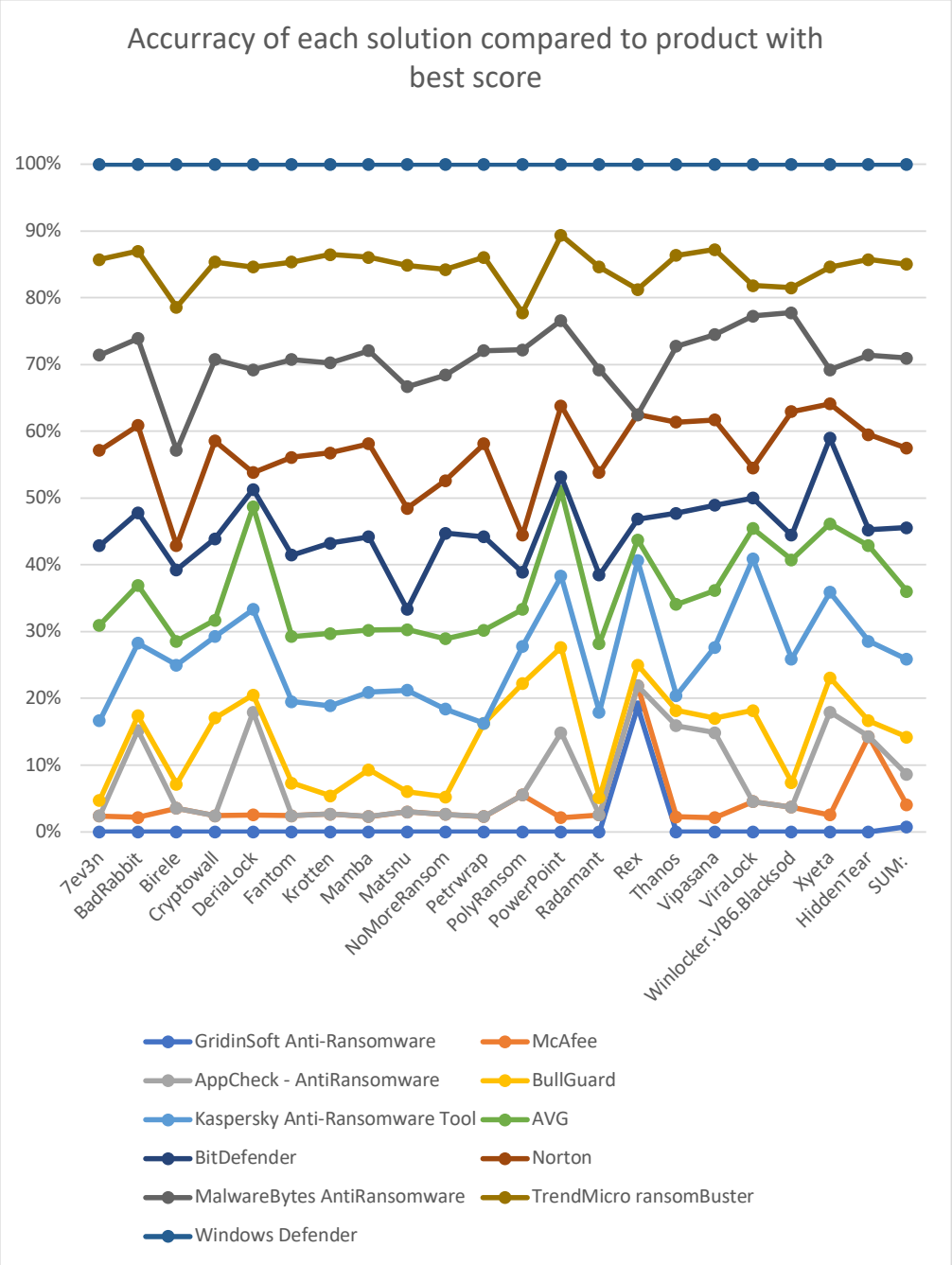


*Figure 27. Accuracy of each solution compared to product with best score*

| Security product: | Total Points: | Accuracy (%): |
|---|---|---|
| Windows Defender | 118 | 94% |
| TrendMicro ransomBuster | 111 | 88% |
| MalwareBytes AntiRansomware | 106 | 84% |
| Norton | 94 | 75% |
| Kaspersky Anti-Ransomware Tool | 90 | 71% |
| Avast/AVG | 80 | 63% |
| Bitdefender | 75 | 60% |
| BullGuard | 44 | 35% |
| AppCheck - AntiRansomware | 36 | 29% |
| McAfee | 26 | 21% |
| GridinSoft Anti-Ransomware | 6 | 5% |

*Table 7. Test results – categorized by security products*

| Sample: | Total points: | Accuracy (%) |
|---|---|---|
| Vipasana | 47 | 71% |
| PowerPoint | 47 | 71% |
| BadRabbit | 46 | 69,70% |
| Thanos | 43 | 65% |
| Petrwrap | 43 | 65% |
| Mamba | 43 | 65% |
| HiddenTear | 42 | 64% |
| 7ev3n | 42 | 64% |
| Fantom | 41 | 62% |
| Cryptowall | 41 | 62% |
| Xyeta | 39 | 59% |
| Radamant | 39 | 59% |
| DeriaLock | 39 | 59% |
| NoMoreRansom | 38 | 58% |
| Krotten | 37 | 56% |
| Matsnu | 33 | 50% |
| Rex | 32 | 48% |
| Birele | 28 | 42% |
| Winlocker.VB6.Blacksod | 27 | 41% |
| ViraLock | 22 | 33% |
| PolyRansom | 17 | 26% |

*Table 8. Test results – antivirus software products*

Additionally, as a form of comparison against VirusTotal findings, measured accuracy with extended levels/categories of detected threats was compared to table providing Virus Total accuracy in section 4.1.5, Table 1. Results cannot be compared directly – VirusTotal shows how many detections, divided by how many products were used to test particular sample. In this work, the tests were similar, but were more focused on the quality of signature of detected threat, and thus – information given to the user. In this case, the results of both VirusTotal and this work should correlate with each other to some extent. The results on Figure 28 below are shown as a percentage of accuracy per sample.

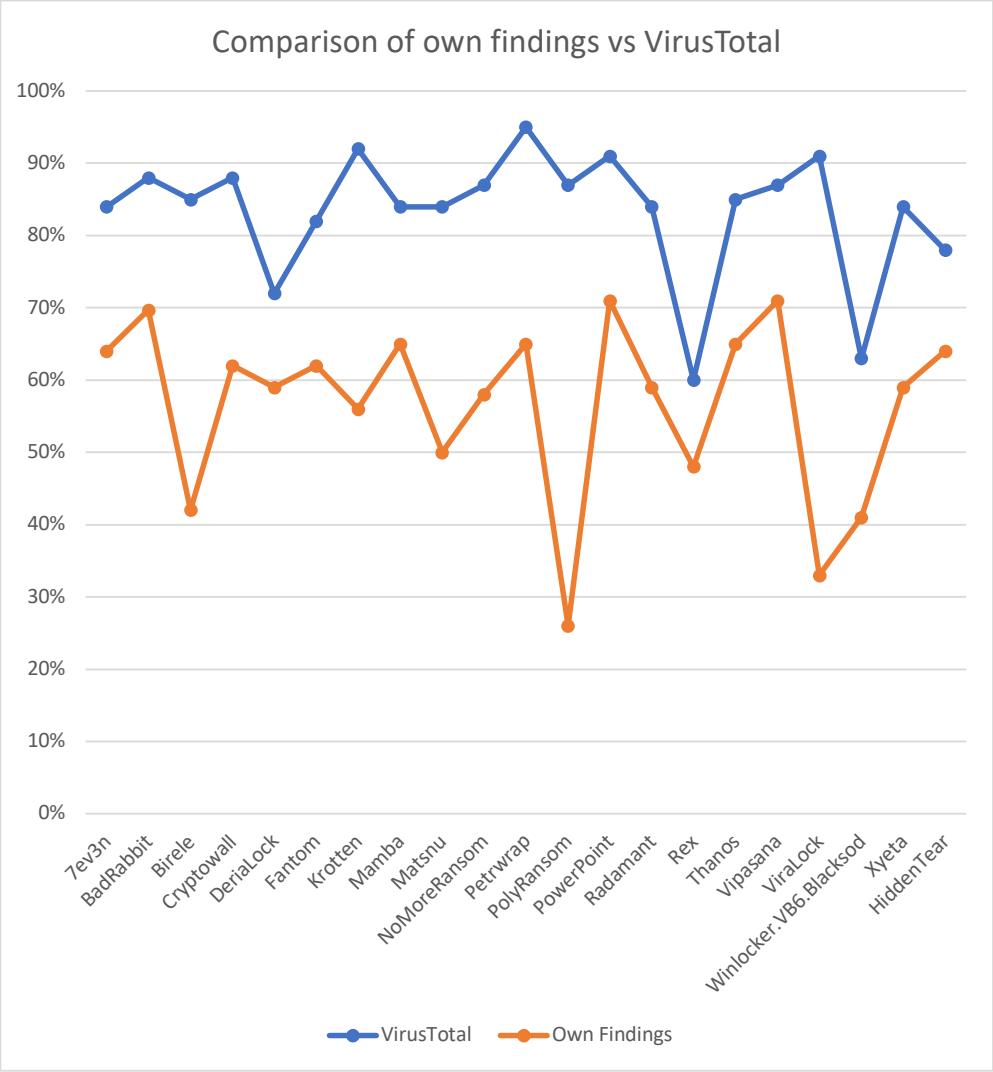


*Figure 28. Comparison of own findings against VirusTotal results*

## 3.5. Summary of findings

It would be difficult to measure effectiveness of chosen security products against a set of outdated ransomwares, or in other words, those that already have signature in most of security related products that are up to date. That being said, another useful approach that comes to mind is measuring the quality of detection and signature convention. From a binary question, whether ransomware is detected or not, emerges a useful perspective – if ransomware was detected, what signature was it assigned with? What information was presented to the victim of attack by security product? That is much more interesting standpoint, and the main question that this research was dedicated to answer.

One surprising outcome of this work is that most accurate software that was subject of the test is actually built into most popular system ever – Windows 10. Windows Defender, which is most basic countermeasure, that is enabled by default in every installation. It is not perfect, as it allowed to execute on user's responsibility some of tested ransomware samples, however none of it actually succeeded in encrypting user files, so it is a matter of discussion whether in fact it was executed at all. Windows Defender comes with many features dedicated to protecting specifically against ransomware, so it suffices to say that Microsoft takes seriously this category of threat. Second place in the comparison takes RansomBuster, which is in fact a specialized tool to protect against ransomware. 3rd place took another specialized tool, MalwareBytes AntiRansomware. Each of solutions in top 3 had very high accuracy compared to other products. All results are presented on Figure 25.

Figure 26 visualizes accuracy of detection per specific sample. It might not be 100% intuitive, but actually the highest score means that particular sample was detected with highest accuracy i.e., had signature with name related to ransomware. It does not indicate which of ransomware was most successful in attacks. And again, Vipasana, PowerPoint and BadRabbit were in the top three in this category. All of these threats were highly reported in last 5 years (2016-2021). More descriptive form of test results is presented in the Table 7 and Table 8.

Another approach to visualization of test results was presented in Figure 27. It has basically form of cartesian product of ransomware sample, and accuracy of detection particular security product. It basically shows, in a more descriptive way, relation of accuracy per sample, product, all compared to product that scored the highest result in these tests. In other words, it takes data from Figure 25 and Figure 26, and combines them into one.

Lastly, another point of view is shown on Figure 28. It illustrates comparison of results in this research, against results of automated tool (VirusTotal) that exists for similar purpose – namely measuring accuracy of security products against malicious programs. One has to admit that both trends share similar shape, even though exact results cannot be directly compared. Whereas this research was measuring quality of signature naming convention, including analysis of efficiency of tested products against ransomware, VirusTotal is taking into account only the latter aspect.

# Conclusions

Main subject of this work is a profound analysis of ransomware behavior, its structure, similarities to other malicious software, current trends in attacks and common techniques used to deliver this type of threat, as well as the present state of security measures used to defend against ransomware. The goal of practical research was to determine which security products, looking at merged two categories – antivirus products and specialized anti-ransomware tools, have the most efficiency against ransomware, and provide the most profound defense.

The results were not surprising, considering that ransomware samples used in these tests were commonly known, available on public repositories for security researchers, and for that reason (almost) all tested security products scored high detection rate. In order to somehow differentiate between them, the results were analyzed from slightly different angle – namely from perspective of quality of signatures that accompany detections. As it turns out, antivirus software provides better quality of protection against known ransomware samples, with known signatures, and is in general more informative to the user. Most of the specialized tools base their protection on behavioral analysis, rather than static signatures, and therefore are a bit slower in protecting against known threats, but on the other hand, are in fact most effective against new malicious software.

One important point made in this dissertation is that, unfortunately, even most advanced protections used with their signatures up to date are not sufficient in protecting against ransomware attacks, or malware attacks in general. Some of mentioned sources strongly claim that security is not a product that can be bought, marked as done, and giving 100% guarantee of protection. On the contrary – it is an ongoing, never ending and demanding constant attention process, that takes into account network security, physical security, local system security as well as susceptibility to social engineering attacks, just because human is the weakest part of any IT system security. The development of this research is certainly a stimulating perspective of currently used security products and recommendations in war against ransomware, and as such does deliver new outlooks in this particular branch of cybercriminal business, giving the reader a strong foundation for future research in this field.

# Bibliography


[1] A. H. Michael Sikorski, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software," 2012.

[2] N. P. By David E. Sanger, "FireEye, a Top Cybersecurity Firm, Says It Was Hacked by a Nation-State," New York Times, 2021.

[3] W. Carbon Black, "Advance Your Ransomware Defences," Carbon Black, 2020.

[4] B. H. B. Webcast, "OK, let's talk about ransomware," 2021.

[5] Kaspersky, Malware & Computer Virus Facts & FAQs.

[6] N. A. Hassan, Ransomware Revealed, 2019.

[7] E. Galperin, "#Spouseware and #Stalkerware: Where Do We Go from Here?," [Online]. Available: https://www.youtube.com/watch?v=QvorPIKXrYA.

[8] G. GEBHART, "Watch EFF Cybersecurity Director Eva Galperin's TED Talk About Stalkerware," 2020. [Online]. Available: https://www.eff.org/deeplinks/2020/05/watch-eff-cybersecurity-director-eva-galperins-ted-talk-about-stalkerware.

[9] J. KELLEY, "Students Are Pushing Back Against Proctoring Surveillance Apps," 2020. [Online]. Available: https://www.eff.org/deeplinks/2020/09/students-are-pushing-back-against-proctoring-surveillance-apps.

[10] kaspersky, "What Is an Advanced Persistent Threat (APT)?," [Online]. Available: https://www.kaspersky.com/resource-center/definitions/advanced-persistent-threats.

[11] J. Lyne, "The Anatomy Of Ransomware," in RSA Conference, 2017.

[12] Verizon, "Verizon Data breach Investigations Report," Verion, 2019.

[13] Varonis, "A Brief History of Ransomware," 2020. [Online]. Available: https://www.varonis.com/blog/a-brief-history-of-ransomware/.

[14] StatCounter, "Desktop Operating System Market Share Worldwide," [Online]. Available: According to https://gs.statcounter.com/os-market-share#monthly-201801-201901-bar .

[15] CrowdStrike, "RANSOMWARE AS A SERVICE (RAAS) EXPLAINED," 2021. [Online]. Available: https://www.crowdstrike.com/cybersecurity-101/ransomware/ransomware-as-a-service-raas/.

[16] Orpheus, "Makop RaaS Campaign targets South Korean Entities," 2021. [Online]. Available: https://orpheus-cyber.com/blog-makop-raas-campaign-targets-south-korean-entities/.

[17] TrendMicro, "Ransomware-as-a-Service: Ransomware Operators Find Ways to Bring in Business," 2016. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ransomware-as-a-service-ransomware-operators-find-ways-to-bring-in-business.

[18] T. Micro, "New Crypto-Ransomware JIGSAW Plays Nasty Games," 2016. [Online]. Available: https://www.trendmicro.com/en_us/research/16/d/jigsaw-ransomware-plays-games-victims.html?_ga=2.243469440.462195162.1615659143-1475313936.1613915437.

[19] TrendMicro, "THE STATE OF RANSOMWARE 2020's Catch-22," 2020. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-state-of-ransomware-2020-s-catch-22.

[20] Hasherezade, "Inside Chimera Ransomware – the first 'doxingware' in wild," 2015. [Online]. Available: https://blog.malwarebytes.com/threat-analysis/2015/12/inside-chimera-ransomware-the-first-doxingware-in-wild/.

[21] kaspersky, "The rise of ransomware 2.0: cybercriminals shift focus from encrypting data to publishing confidential information online," 2020. [Online]. Available: https://www.kaspersky.com/about/press-releases/2020_the-rise-of-ransomware-20-cybercriminals-shift-focus-from-encrypting-data-to-publishing-confidential-information-online.

[22] D. Todd, "Acer Hit with Highest Ransomware Demand Ever," 2021. [Online]. Available: https://www.secureworldexpo.com/industry-news/acer-hit-with-highest-ransomware-demand-ever.

[23] Wired, "Apple's Ransomware Mess Is the Future of Online Extortion," 2021. [Online]. Available: https://www.wired.com/story/apple-ransomware-attack-quanta-computer/.

[24] kaspersky, "STORY OF THE YEAR: THE RANSOMWARE REVOLUTION," 2016. [Online]. Available: https://media.kaspersky.com/pdf/b2b/kaspersky-story-of-the-year-ransomware-revolution.pdf.

[25] Microsoft, "Windows Sysinternals," [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/.

[26] J. Fruhlinger, "Recent ransomware attacks define the malware's new age," 2020. [Online]. Available: https://www.csoonline.com/article/3212260/recent-ransomware-attacks-define-the-malwares-new-age.html.

[27] K. Labs, "Top 11 Ransomware Attacks in 2020-2021," 2021. [Online]. Available: https://www.keepnetlabs.com/top-11-ransomware-attacks-in-2020-2021/.

[28] ETCISO, "The worst ransomware attacks in the last 5 years," 2019. [Online]. Available: https://ciso.economictimes.indiatimes.com/news/the-worst-ransomware-attacks-in-the-last-5-years/68919750.

[29] C. Xiao, "New OS X Ransomware KeRanger Infected Transmission BitTorrent Client Installer," 2016. [Online]. Available: New OS X Ransomware KeRanger Infected Transmission BitTorrent Client Installer (paloaltonetworks.com).

[30] Acronis, "Ransomware Attacks Someone Every 11 Seconds," [Online]. Available: https://www.acronis.com/en-us/solutions/ransomware-protection/.

[31] "Hidden Tear," [Online]. Available: https://en.wikipedia.org/wiki/Hidden_Tear.

[32] goliate, 2015. [Online]. Available: https://github.com/goliate/hidden-tear.

[33] R. ARMSTRONG, "Hidden Tear Ransomware Worth Crying Over," 2020. [Online]. Available: https://www.speartip.com/resources/hidden-tear-ransomware-worth-crying-over/.

[34] A. F. M. F. O. C. G. C. Fox, "Password-based encryption approach for securing sensitive data," 2020.

[35] BitDefender, "Petya Ransomware Goes Low Level - Petya Ransomware analysis and full details on Bitdefenders Vaccine Preventing File Encryption," 2016.

[36] Y. Hu, "A Brief Summary of Encryption Method Used in Widespread Ransomware," 2017. [Online]. Available: https://resources.infosecinstitute.com/topic/a-brief-summary-of-encryption-method-used-in-widespread-ransomware/.

[37] A. Hern, "New ransomware employs Tor to stay hidden from security," 2014. [Online]. Available: https://www.theguardian.com/technology/2014/jul/25/new-ransomware-employs-tor-onion-malware.

[38] B. J. C. Ibrahim Ghafir Masarykova univerzita, J. Svoboda and V. Prenosil, "Tor-based malware and Tor connection detection," in International Conference on Frontiers of Communications, Networks and Applications (ICFCNA 2014 - Malaysia), 2014.

[39] L. Abrams, "Chimera Ransomware uses a Peer-To-Peer Decryption Service," 2015. [Online]. Available: https://www.bleepingcomputer.com/news/security/chimera-ransomware-uses-a-peer-to-peer-decryption-service/.

[40] Sophos, "The state of ransomware 2020," 2020.

[41] K. &. V. CarbonBlack, "A Tale Of two ransoms".

[42] D. Slater, "7 new social engineering tactics threat actors are using now," 2021. [Online]. Available: https://www.csoonline.com/article/3613937/7-new-social-engineering-tactics-threat-actors-are-using-now.html.

[43] statista, "Number of sent and received e-mails per day worldwide from 2017 to 2025," 2021. [Online]. Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/.

[44] T. S. T. S. MARIA VERGELIS, "Spam and phishing in Q1 2019," Securelist, 2019.

[45] S. Griffiths, "Why your internet habits are not as clean as you think," 2020. [Online]. Available: https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think#:~:text=Perhaps%20unsurprisingly%2C%20the%20footprint%20of,University%20who%20researches%20carbon%20footprints..

[46] J. E. Douglas, Mindhunder: Inside the FBI's Elite Serial Crime Unit, 2017.

[47] Kaspersky, "Kaspersky uncovers a creative water hole attack discovered in the wild," 2020. [Online]. Available: https://usa.kaspersky.com/about/press-releases/2020_kaspersky-uncovers-a-creative-water-hole-attack-discovered-in-the-wild.

[48] M. Tischer, Z. Durumeric, S. Foster, S. Duan, A. Mori, E. Bursztein and M. Bailey, "Users Really Do Plug in USB Drives They Find," 2016.

[49] Sophos, "Almost half of dropped USB sticks will get plugged in," 2016. [Online]. Available: https://nakedsecurity.sophos.com/2016/04/08/almost-half-of-dropped-usb-sticks-will-get-plugged-in/.

[50] Sophos, "TeslaCrypt ransomware attacks gamers – "all your files are belong to us!"," 2015. [Online]. Available: https://nakedsecurity.sophos.com/2015/03/16/teslacrypt-ransomware-attacks-gamers-all-your-files-are-belong-to-us/.

[51] J. CHIOU, "If I Found It on the Internet, It's Free…right?," 2019. [Online]. Available: https://www.tablefortwoblog.com/i-found-it-on-the-internet/.

[52] "shodan.io," [Online]. Available: shodan.io.

[53] kaspersky, "Kaspersky Security Bulletin 2020," 2020.

[54] V. SARVEPALLI, "VPN - A Gateway for Vulnerabilities," 2019. [Online]. Available: https://insights.sei.cmu.edu/blog/vpn-a-gateway-for-vulnerabilities/.

[55] D. Palmer, "Ransomware crooks are targeting vulnerable VPN devices in their attacks," 2021. [Online]. Available: https://www.zdnet.com/article/ransomware-crooks-are-targeting-vulnerable-vpn-devices-in-their-attacks/.

[56] J. Cole, "Worst VPNs in cybersecurity history," 2021. [Online]. Available: https://vpnpro.com/blog/worst-vpns-in-cybersecurity-history/.

[57] E. Kent, "Capcom concludes ransomware investigation, details what happened," 2021. [Online]. Available: https://www.eurogamer.net/articles/2021-04-13-capcom-concludes-ransomware-investigation-details-what-happened.

[58] P. Arntz, "Take action! Multiple Pulse Secure VPN vulnerabilities exploited in the wild," 2021. [Online]. Available: https://blog.malwarebytes.com/malwarebytes-news/2021/04/take-action-multiple-pulse-secure-vpn-vulnerabilities-exploited-in-the-wild/.

[59] S. Curry, "We Hacked Apple for 3 Months: Here's What We Found," 2020. [Online]. Available: https://samcurry.net/hacking-apple/.

[60] B. C. Phipps, "A recent AWS information leakage vulnerability may put your digital information at risk of exposure.," [Online]. Available: https://uncomn.com/a-recent-aws-information-leakage-vulnerability-may-putting-your-digital-information-at-risk/.

[61] M. Kumar, "Microsoft Azure Flaws Could Have Let Hackers Take Over Cloud Servers," 2020. [Online].

[62] LiveOverflow, "Hacking into Google's Network for $133,337," [Online]. Available: https://www.youtube.com/watch?v=g-JgA1hvJzA.

[63] B. O'Connor, "Misconfiguration on the Cloud is as Common as it is Costly," 2020. [Online].

[64] TrendMicro, "Misconfigured Cloud Services Pose High Security Risks for Organizations," 2018. [Online].

[65] C. Grier, "Manufacturing compromise: The emergence of exploit-as-a-service," in Proceedings of the 2012 ACM conference on Computer and Communications Security, 2012.

[66] Kaspersky, "What Is a Drive by Download," [Online]. Available: https://www.kaspersky.com/resource-center/definitions/drive-by-download.

[67] I. W. X. G. P. K. Van Lam Le, "Anatomy of drive-by download attack," Proceedings of the Eleventh Australasian Information Security Conference, 2013.

[68] L. O'Donnell, "Google Warns Mac, Windows Users of Chrome Zero-Day Flaw," 2021. [Online]. Available: https://threatpost.com/google-mac-windows-chrome-zero-day/164759/.

[69] K. Team, "Chrome in the zero-day crosshairs," 2019. [Online]. Available: https://www.kaspersky.com/blog/google-chrome-zeroday-wizardopium/29126/.

[70] F. Crast, "Microsoft releases security update for Edge, zero-day exploited in the wild," 2020. [Online].

[71] C. Cimpanu, "Firefox gets fixes for two zero-days exploited in the wild," 2020. [Online]. Available: https://www.zdnet.com/article/firefox-gets-fixes-for-two-zero-days-exploited-in-the-wild/.

[72] A. Owaida, "Apple patches three iOS zero-days under attack," 2021. [Online].

[73] "cvedetails," [Online]. Available: cvedetails.com.

[74] "Google » Chrome : Vulnerability Statistics," [Online]. Available: Google Chrome : CVE security vulnerabilities, versions and detailed reports (cvedetails.com).

[75] M. Brinkmann, "The Firefox NoScript guide you have all been waiting for," [Online]. Available: https://www.ghacks.net/2014/02/10/firefox-noscript-guide-waiting/.

[76] A. Drozhzhin, "Why you should NOT pay ransom to malware creators," 2016. [Online]. Available: https://www.kaspersky.com/blog/no-no-ransom/13364/.

[77] kasperksy, "Over half of ransomware victims pay the ransom, but only a quarter see their full data returned," 2021. [Online]. Available: https://www.kaspersky.com/about/press-releases/2021_over-half-of-ransomware-victims-pay-the-ransom-but-only-a-quarter-see-their-full-data-returned.

[78] D. Palmer, "Ransomware: A company paid millions to get their data back, but forgot to do one thing. So the hackers came back again," 2021. [Online]. Available: https://www.zdnet.com/article/ransomware-this-is-the-first-thing-you-should-think-about-if-you-fall-victim-to-an-attack/.

[79] L. H. Newman, "Ransomware's dangerous new trick is double encrypting your data," 2021. [Online]. Available: https://www.wired.com/story/ransomware-double-encryption/.

[80] C. Cimpanu, "Ransomware gangs are now cold-calling victims if they restore from backups without paying," 2020.

[81] L. Abrams, "Ziggy ransomware shuts down and releases victims' decryption keys," 2021.

[82] D. ELLIS, "6 Phases in the Incident Response Plan," [Online]. Available: https://www.securitymetrics.com/blog/6-phases-incident-response-plan.

[83] Cynet, "NIST Incident Response," [Online]. Available: https://www.cynet.com/incident-response/nist-incident-response/.

[84] E. GIRKEN, "Incident Response Steps and Frameworks for SANS and NIST," 2020. [Online]. Available: https://cybersecurity.att.com/blogs/security-essentials/incident-response-steps-comparison-guide.

[85] "No More Ransom Project," [Online]. Available: nomoreransom.org.

[86] D. O. M. Thomas W. Edgar, Research Methods for Cyber Security, 2017.

[87] "Ransomware Overview," [Online]. Available: https://docs.google.com/spreadsheets/d/1TWS238xacAto-fLKh1n5uTsdijWdCEsGIM0Y0Hvmc5g/htmlview.

[88] A. C. Guardian, "Ransomware Encrypted File Extension List," [Online]. Available: https://avepointcdn.azureedge.net/assets/webhelp/compliance_guardian_installation_and_administration/index.htm#!Documents/ransomwareencryptedfileextensionlist.htm.

[89] OWASP, "Project Top Ten," [Online]. Available: https://owasp.org/www-project-top-ten/.

[90] A. G. a. V. H. B. a. G. V. Cybenko, "Data Exfiltration and Covert Channels," Thayer School of Engineering, Dartmouth College, Hanover.

[91] S. M. a. S. S. Lewis, "Embedding Covert Channels into TCP/IP," in Proceedings of the 7th Information Hiding Workshop , 2005.

[92] SOCRadar, "The Week in Dark Web – 26 March 2021 – Grand Theft Data," 2021. [Online]. Available: https://socradar.io/the-week-in-dark-web-26-march-2021-grand-theft-data/.

[93] K. Polley, "Detecting and Verifying ICMP Exfiltration with AI," 2019. [Online]. Available: https://www.patternex.com/threatex/detecting-and-verifying-icmp-exfiltration-with-ai-enabled-platform.

[94] G. Farnham, "Detecting DNS Tunneling," 2013.

[95] A. Nadler, "INTRODUCTION TO DNS DATA EXFILTRATION," 2017. [Online]. Available: https://blogs.akamai.com/2017/09/introduction-to-dns-data-exfiltration.html.

[96] Cisco, "CCNA Ceritifaction Community - Cisco IPS/IDS Fundamentals," [Online]. Available: https://learningnetwork.cisco.com/s/question/0D53i00000KsuxD/cisco-idsips-fundamentals.

[97] J. PETTERS, "IDS vs. IPS: What is the Difference?," 2020. [Online]. Available: https://www.varonis.com/blog/ids-vs-ips/.

[98] J.-M. R. S. S. a. S. B. Alireza Sadighian, "A Context-Aware Malware Detection Based on Low-Level Hardware Indicators as a Last Line of Defense," in SECURWARE, 2017.

[99] E. Ouellet, "Gartner Magic Quadrant for Content-Aware Data Loss Prevention," 2013.

[100] imperva, "Data Loss Prevention (DLP)," [Online]. Available: https://www.imperva.com/learn/data-security/data-loss-prevention-dlp/.

[101] J. D. Groot, "What is Data Loss Prevention (DLP)? A Definition of Data Loss Prevention," 2020. [Online].

[102] N. Lord, "Ransomware Protection & Removal: How Businesses Can Best Defend Against Ransomware Attacks," 2020. [Online]. Available: https://digitalguardian.com/blog/ransomware-protection-attacks.

[103] S. Activity, "MONITORING SOFTWARE BLOG," 2020. [Online]. Available: https://www.softactivity.com/ideas/what-are-user-behavior-analytics/.

[104] A. GREEN, "What is User Behavior Analytics?," 2020. [Online]. Available: https://www.varonis.com/blog/what-is-user-behavior-analytics/.

[105] M. Zalewski, Silence on the Wire: A Field Guide to Passive Reconnaissance and Indirect Attacks, 2005.

[106] P. Borkar, "The New Breed of “Fileless Malware” and How It Can Be Stopped with Behavioral Analytics and Machine Learning," 2019. [Online]. Available: https://www.exabeam.com/ueba/fileless-malware-behavioral-analytics-machine-learning/.

[107] M. F. Z. Jantan, "A Framework for Defining Malware Behavior Using Run Time Analysis and Resource Monitoring".

[108] K. Baker, "Malware Analysis," 2020. [Online]. Available: https://www.crowdstrike.com/cybersecurity-101/malware/malware-analysis/.

[109] " Basic Dynamic Analysis," 2017. [Online]. Available: https://samsclass.info/126/proj/pDC5.htm.

[110] "Strace man," [Online]. Available: https://man7.org/linux/man-pages/man1/strace.1.html.

[111] "ltrace man page," [Online]. Available: https://man7.org/linux/man-pages/man1/ltrace.1.html.

[112] "Microsoft Sysinternals," [Online]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/process-explorer.

[113] "WireShark," [Online]. Available: https://www.wireshark.org.

[114] "Telerik Fiddler - network debugging tool," [Online]. Available: https://www.telerik.com/fiddler.

[115] M. F. Zolkipli and A. Jantan, "Malware Behavior Analysis: Learning and Understanding Current Malware Threats," 2010.

[116] linuxtesting, "Static Analysis vs. Dynamic Analysis," [Online]. Available: http://linuxtesting.org/static-vs-dynamic.

[117] G. Wagener, "Malware behaviour analysis," 2008.

[118] S. S. Hansen, T. M. T. Larsen, M. Stevanovic and J. M. Pedersen, "An approach for detection and family classification of malware based on behavioral analysis," 2016.

[119] R. K. (. David Ferraiolo (NIST), "Role-Based Access Controls," 1992.

[120] Microsoft, "GPO documentation," [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/desktop/policy/about-group-policy.

[121] M. Weckstén, J. Frick, A. Sjöström and E. Järpe, "A novel method for recovery from Crypto Ransomware infections," 2016.

[122] Microsoft, "WSH documentation," [Online]. Available: https://www.computerhope.com/issues/ch001788.htm.

[123] O. M. Benchmark, "Malware Benchmark," 2015. [Online]. Available: https://github.com/openmalwarebenchmark/Spoon-Knife/blob/master/BENCHMARK%20REPORT.pdf.

[124] Acronis, "How to Find out If Your Computer Is Protected Against Ransomware," 2017. [Online]. Available: https://www.acronis.com/en-us/blog/posts/how-find-out-if-your-computer-protected-against-ransomware.

[125] "Libre Office Macros Tutorial," [Online]. Available: https://wiki.documentfoundation.org/Macros/General/007.

[126] Endermanch, "Malware Samples Databasse," [Online]. Available: github.com/Endermanch/MalwareDatabase.

[127] S. Sjouwerman, "https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-20-billion-usd-by-2021/," KnowBe4.

[128] G. Greenwald, "Edward Snowden: the whistleblower behind the NSA surveillance revelations," The Guardian, 2013.