Expert Systems With Applications 190 (2022) 116198

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

The rise of ransomware: Forensic analysis for windows based ransomware attacks

Ilker Kara [a,*], Murat Aydos [b]

[a] Department of Medical Services and Techniques, Eldivan Medical Services Vocational School Çankırı Karatekin University, Turkey
[b] Department of Computer Engineering, Hacettepe University, Turkey

ARTICLE INFO

Keywords:
Cybersecurity
Digital forensic
Malware attacks
Ransomware detection
Onion ransomware
Analysis techniques

ABSTRACT

While information technologies grow and propagate worldwide, malwares have modified and risen their efficiency towards information system. Recently, the attackers have started to use ransom software (ransomware) as an effective method of cyberattack because of their profitability. Ransomware infiltrate victim systems in various ways, usually encrypt files in the system, and demand a ransom to allow user access to the encrypted files again. Although security mechanisms such as firewalls, anti-virus programs, and automated analysis programs have been developed to combat this threat, these mechanisms have little success and fail to protect the valuable assets stored in local or cloud storage resources. In this study, an effective detection and analysis method against ransomware was proposed, and the proposed method was discussed in detail with a case study. As a result of the study, potential information about the attacker were found to be accessible through characteristic behavior analysis of the onion ransomware, which was analyzed in accordance with the proposed method. This paper also presents an insight to the ransomware threat and provides a basic review of the methods and techniques used in the detection and analysis of ransomware attacks.

1. Introduction

The development of malware detection and analysis techniques is of importance in order to reduce potential security vulnerabilities in information systems (Li et al., 2017; Chen, Alalfi, Tam et al., 2017). Attackers infect information systems by exploiting vulnerabilities in operating systems, vulnerabilities in programs, or the carelessness of users (Souri & Hosseini, 2018; Hou & Abdulhayoglu, 2017; Tam et al., 2017). Malware can slow down information systems gradually, encrypt files partially or completely, and make the system unusable (Raff et al., 2017; Ki, Kim & Kim, 2015). Among the numerous types of malware, Trojan horses, ransomware, backdoors and viruses are the most common (Fan & Chen, 2016; Ki & Kim, 2015; Spreitzenbarth et al., 2015). Ransomware attacks has significantly increased in the last five years due to their higher financial returns (Reddy et al., 2020; Qamar, Karim & Chang, 2019). Ransomware is a type of malware that encrypts important files, and demands a ransom from the victim for allowing access to the files, or locks the target system completely and makes it unusable (Gómez-Hernández & García-Teodoro, 2018; O'Kane, Sezer & Carlin, 2018; Kirda, 2017; Ferrante et al., 2017). Basically, there are two types of ransomware: crypto ransomware and locker ransomware (Akbanov, Vassilakis & Logothetis, 2019; Baldwin & Dehghantanha, 2018).

- Crypto-ransomware encrypt all data files in the victim system (operating system, PDF, Word, Excel, game files, photos, etc.). After the encryption process, the attacker sends a message to the victim about the process and regaining access to the files. In this message, the attacker states that access to the file contents can only be possible with the private key, and that it is mandatory to pay the ransom in order to obtain this key. The ransom is asked to be paid with bitcoin in a specified time frame.
- Locker ransomware, unlike the other type, make the system unusable by locking the entire system, not only the specific files. However, it also encrypts the files in the system (Kharraz et al., 2015). After the system becomes unusable, the attacker's message is displayed on the main screen.

Today, two techniques are widely used in malware detection and analysis (Yunus & Ngah, 2020; Kolosnjaji et al., 2016; Ki, Kim & Kim, 2015). These are automatic analysis and manual analysis techniques (Yerima et al., 2013; Schmidt et al., 2009). In the automatic analysis approach (Machine Learning), signature-based and behavioral detection

* Corresponding author.
E-mail addresses: karaikab@gmail.com (I. Kara), maydos@hacettepe.edu.tr (M. Aydos).

https://doi.org/10.1016/j.eswa.2021.116198
Received 10 December 2020; Received in revised form 3 November 2021; Accepted 5 November 2021
Available online 19 November 2021

techniques are used. The signature-based detection technique works on the basis of comparison with the current database of known and previously identified malware. Behavior based detection and analysis is a technique to find out the malware's file-directory movements and memory and network activities by executing the malware in an isolated environment. There are some disadvantages of automated analysis approaches (such as time constraints, and lack of intelligent decision making). Attackers who exploit these disadvantages can easily hack into the system. In order to overcome the disadvantages, manual methods are developed using intuitive detection techniques in addition to the above-mentioned techniques. The manual analysis approach is slower and more costly than the automatic analysis approach. In addition, there is no universally accepted method for manual analysis.

The main contributions of this paper are summarized as follows:

- In the study, a basic review of the methods and techniques used in the detection and analysis of ransomware is presented.
- We recommend using an applicable method for ransomware detection and analysis. This proposed method offers useful insight for the experts working in this field.
- The analysis programs used in the proposed method in the study are all free software. Applicability of the proposed method was shown on a sample study. In order to repeat the analyses performed in the case study, an accessible sample was used (http://ilkerkara.karatekin.edu.tr/).

The rest of this paper is organized as follows. Similar studies in the literature were reviewed in Section 2. Section 3 proposes a viable method for ransomware detection and analysis, Section 4 and Section 5 discuss how it can be implemented step by step on a real case, and finally concluding remarks are presented in Section 6.

2. Ransomware timeline

The history of malware almost dates back to the development of the computer. While it is not known when malware attacks were first carried out, the first known malicious malware attack was the one performed using the AIDS Trojan, developed by Joseph Popp in 1989 (O'Kane, Sezer & Carlin, 2018). AIDS enabled malware to infiltrate target systems via a floppy disk. The malware was encrypting the file extensions and filenames found on the system. However, the contents of the files were not encrypted.

After the AIDS, there was a calm period that lasted more than two decades until the Xorist ransomware emerged in 2011. The Xorist ransomware was intended to encrypt Windows operating system files. After encrypting the system files, it was displaying a ransom note with instructions for the victim.

In 2021 that followed, the number of attacks has reached to a frequency of one in every 11 s on average (Syed, 2021). In 2012, Reveton ransomware began to spread. It differs from Xorist in that it uses intimidation tactics to pressure its victims to pay ransoms. For example, there were threats that the user was logged into an illegal area and law enforcement would take action unless the victim pays the ransom. In addition, victims were forced to pay a ransom by claiming that their IP address has been identified and that the actual footage of the webcams has been taken. The ransom payment was required to be paid using the MoneyPak card.

Cryptolocker ransomware attacks began to appear in 2013. Its main features were propagation through infected e-mail attachments and encrypting victim files using the RSA encryption method. The ransom payment was asked to be paid using digital currencies, such as Bitcoin, to get the decryption key. In addition, the attacker was threatening to delete the private encryption key unless the payment was made before the deadline.

Although the first ransomware attacks targeted Windows operating systems, Fusob targeted mobile devices in 2015, and these attacks continued until March 2016. Similarly, the Reveton ransomware attacks were using intimidation tactics to force victims to pay the ransom. TeslaCrypt Mukesh was attacking via email in 2015–2016. Such ransomware targets the libraries ole32dir.dll, kernel32.dll, and apphelp.dll in the victimized system. In addition, some game files and special files (.doc, .pdf, .py, .ptx, .jpeg, etc.) were also found to be encrypted. In May 2016, this threat was eliminated with the release of the main decryption key by the attackers. The WannaCry ransomware attacks had a wider audience. Not only the end users, but also the large-scale agencies, such as FedEx, Renault in addition to departments of the British National Health Service (NHS) have been targeted. Wannacry ransomware was exploiting MS010-2 vulnerabilities to infiltrate the target system. After the infiltration, it was querying the kill switch domain, and starting the encryption process unless a successful connection was established. The attackers were using hybrid encryption. The original files on the target system were either encrypted or deleted. Using C&C (Command Control), it attempts to contact the attacker using TCP on port 445.

Onion ransomware is a new generation ransomware that demands ransom from the victim for decryption key, which encrypts target system files and user data with the AES encryption. It uses intimidation methods to get a ransom from the victim. The attacker demands the ransom in Bitcoin. As one of the most sophisticated ransomware nowadays, it has become a potential successor to Cryptolocker, which is a truly dangerous threat. This review of ransomware offers two facts: ransomware attacks are specifically targeted at Windows operating systems, one of the most widely used and ergonomic operating systems around the world. However, there has been a recent increase in attacks towards IOS, SCADA and Android mobile devices in particular.

2.1. Ransomware lifecycle

One of the most important steps in ransomware analysis is the need to know the ransomware attack stages, life cycle and attack strategies. Knowing this factor or factors well directly affects the success of defense mechanisms.

Before discussing a variety of studies dealing with ransomware analysis, it is important to know the processes in the ransomware life cycle. The life cycle of ransomware is defined as beginning from the moment the code with malicious purposes begins to spread and ending when the ransom is demanded from the victim (Fig. 1).

Developing a variety of actions to prevent the process before the ransomware attack occurs by knowing the ransomware life cycle may provide successful protection for devices, files and resources on the target system. The life cycle of ransomware software may be investigated in four phases.

2.1.1. Preparation phase

The ransomware life cycle begins after the content designed by the ransomware developer is prepared and is served on the Tor browser network by a distributor (Sgandurra et al., 2016). In this stage, the attacker makes some small changes to use the ransomware in line with their purposes or sometimes uses it as it is and this stage defines the process up until it is ready for use.

2.1.2. Distribution phase

In this stage, ransomware is packeted in a mail attachment or loaded to an infiltrated website and is defined as the distribution process (Mbol, Robert & Sadighian, 2016; Paik, Shin & Cho, 2016).

2.1.3. Phishing phase

The attacker begins to search for targets for the ransomware attack. Known as the exploration stage, this stage is defined as the stage when the ransomware benefits from possible security gaps or user weaknesses to infiltrate the target with infiltration methods (Scaife, Carter, Traynor & Butler, 2016).

Phishing attack is used to learn the victim's social media or e-mail

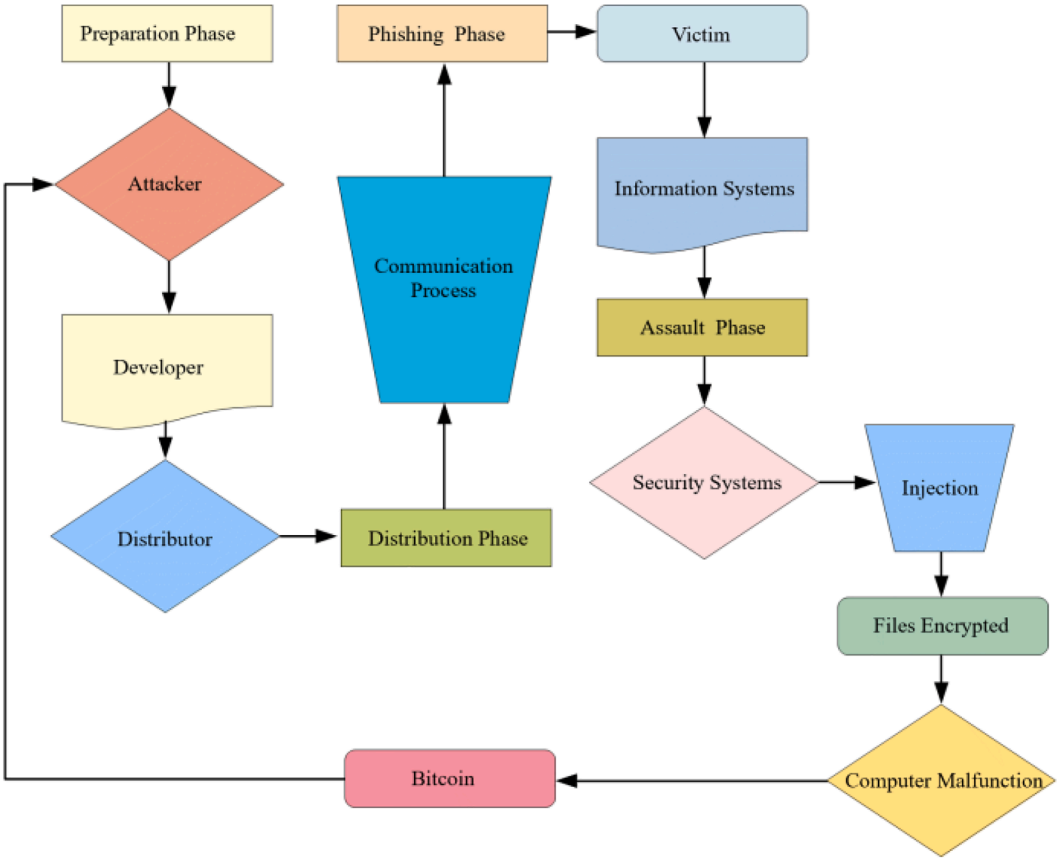

Fig. 1. Ransomware lifecycle.

password and also credit card information. The victim is directed to fake or malicious web sites with the help of an e-mail prepared as if it is from a bank or an official institution. Attackers create fake web pages that look like banking websites, social networking websites, e-mail services, online games and etc. for their phishing attacks, where, the victim's identity information, credit card number, password, etc. are requested. The goal is to steal the victim's information, by taking into account the requests in the e-mail message and on the fake site.

2.1.4. Assault phase

In the attack stage of ransomware, it begins to infiltrate the victim's machine by code dropper, mail attachment or download from an infiltrated website (Mbol, Robert & Sadighian, 2016; Ray, Hicks & Moyle, 2016; Kim & Kim 2015). After the ransomware has entered the victim's system, a range of actions begin. The ransomware in the victim's system takes actions like creating a unique computer identity, disabling some programs, loading the program to operate at start up and taking the internal IP address (Al-rimy et al., 2018). Later, ransomware creates a connection to the command and control (C&C) server to receive an encryption key (Zimba, 2017). Then, in the fourth phase, the malicious process searches the user's files for certain extensions like pdf, doc, xls, pptx and jpeg. These files are transferred to another location and then they are completely encrypted. The encrypted files are renamed and the original files are deleted (Al-rimy et al., 2018). Finally, the malicious system sends a text file or displays the ransom demands on the desktop screen of the victim (Prakash, Nafis, & Biswas, 2017). The ransomware life cycle contains all these steps.

Studies in the literature generally display small differences according to the type of ransomware attack stages, though the general mechanisms appear very similar. Kim et al. (2015) showed that ransomware infiltrates the victim's system using phishing methods. According to Wang et al. (2015) and Mbol et al. (2016), traditional ransomware uses gaps in the software system to infiltrate the target system and showed that ransomware can infiltrate the victim's system involuntarily through e-mail attachments or from destroyed websites with the same approach.

Analysis of a variety of ransomware by Gazet (2010) defined the ransomware attack in three stages. These are; target search, seizure of the files on the system and ransom demands from the victim. In the target search stage, information gathering and environmental exploration is performed. Based on the gathered information, attack is arranged according to the target user's resources. When the attack is implemented successfully, the process performed on the victim's system and the ransom demand message is reached. Similarly, Ahmadian et al. (2015) stated that a ransomware attack comprises several stages. These are the search to determine a victim, process implementation, general key change by infiltrating the victim's system, encryption, message display and decryption stages. Similarly, Kumar et al. (2013) discussed ransomware attacks as having three phases. These are the encryption, extortion and decryption phases. In short, in spite of fine differences observed between attack approaches in different ransomware families, ransomware is implemented in several common attack stages.

2.2. Methods for ransomware analysis

In this section, we focus on ransomware analysis approaches. In the literature, analysis approaches are divided into three types, these are; static analysis, dynamic analysis and hybrid analysis (Ganesh et al., 2016; Shijo & Salim, 2015; Wagner et al., 2015).

2.3. Static analysis

Static analysis is a passive approach without operating the ransomware (Galal, Mahdy & Atiea, 2016). It investigates structural properties without extracting the source code forming the ransomware (Wang &

Wang, 2015; Zhang & Tan, 2015). With this technique, ransomware is securely identified and it assists in defining a variety of features forming the ransomware (Galal, Mahdy & Atiea, 2016; Zhang & Tan, 2015). Andronio et al. (2015) recommended using static analysis for detection and analysis of crypto-ransomware. Similarly, Ferrante et al. (2017) adopted a model control technique investigating ransomware and crypto-ransomware including the bite code of the ransomware.

This approach focuses on monitoring functions related to the file encryption processes and catching certain commands in the code implemented for encryption. Static analysis includes the stages of investigating the subprocesses of the suspect files, investigating the file directory movements and detecting characteristic properties belonging to the ransomware family. Scaife et al. (2016), recommended the use of this approach as an early detection system.

The early detection system using static analysis properties for suspect files of CryptoDrop identifies features like content similarity and entropy measurements in order to identify the family to which crypto-ransomware the files belong.

With the static analysis approach, in spite of rapid, reliable and accurate identification of ransomware, the technique involves some flaws. It is very difficult to identify ransomware hidden in packaged content with this approach, (Banescu et al., 2015). They found out that it was vulnerable to the "obfuscation" and "polymorphic" techniques (Choudhary & Vidyarthi, 2015).

The methods commonly used within the scope of Static analysis of ransomware as given as follows (Fig. 3):

- Opcode (Operational codes) investigation: Code-based techniques are among the oldest methods in effective fight against malware. This method is based on the idea of detecting the type of malware before any action, by analyzing suspicious files at the code level.
- Open source research: It uses open source malware analysis tools where analysts can find examples of malware attacks in the past and share information with each other. Analysts can test, characterize and compare various malware activation options while learning the life cycle of malware attacks through open source research. Virus Total, a subsidiary of Google most widely used in open source research, provides free services to users.
- Control flow graph (CFG): It is widely used in the static analysis method in identifying and classifying the variants of malware. Unfortunately, this method can become ineffective by using a code packaging method while preparing malware.
- Lexical analysis: Lexical analysis: It is used to detect the source code or URL lexical features of malicious software. It may be possible to avoid detection if the attacker manipulates code packaging or URLs to circumvent lexical analysis.

Fig. 3 shows the workflow steps for the static analysis approach.

2.4. Dynamic analysis

Dynamic analysis is known as analysis while the ransomware is operating (Fig. 2). This approach operates the ransomware in a safe environment (like virtual machine (VM) or sandbox) and analyzes the activity of the files (like file-directory, Windows registry, IP movements). Dynamic analysis ensures observation of the ransomware's true behavior with the target operating system and analysis of available program files and data flow in a safe environment (Kaur & Singh, 2014). As a result, it is a very effective approach to understand the detailed interaction of ransomware codes and operating logic (Nauman, Azam & Yao, 2016). The dynamic analysis approach allows the opportunity to perform memory analysis of the victim's system.

In the literature, there are a variety of studies using dynamic analysis approaches for ransomware (Kharraz et al., 2015; Cabaj & Mazurczyk, 2016; Song, Kim & Lee, 2016). Kharraz et al. identified crypto-ransomware types with a dynamic analysis system they proposed

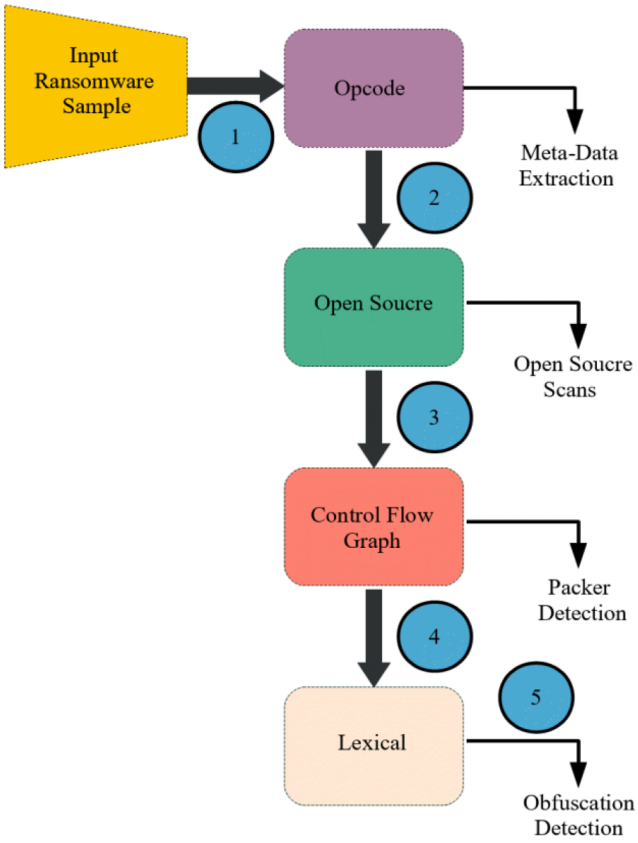

Fig.3. Static malware analysis workflow.

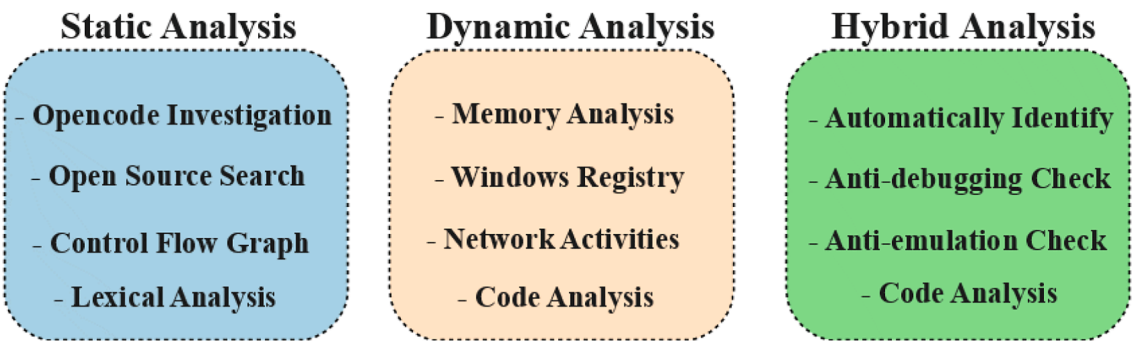

Fig. 2. Commonly employed malware detection methods in the literature.

called UNVEIL (Kharraz et al., 2015). This dynamic analysis method focused on investigation of three elements. These are; a) memory entropy, b) system access points and, c) file system activities. Similarly, Song et al. proposed a dynamic analysis detection model (Song, Kim & Lee, 2016). In this proposed approach, a dynamic analysis method is used to observe the CPU, memory and file movements in the target system.

Though the dynamic analysis method includes many advantages like "performing live analysis", there are some limitations. Some of these include factors like when ransomware notices the analysis it acts outside its true purposes, acting harmlessly or becomes active at a time other than the initial infiltration into the system (24 or 72 h later) to organize the attack (Choudhary & Vidyarthi, 2015; Shijo & Salim, 2015). The ransomware produced by attackers taking advantage of these limits can easily overcome dynamic analysis methods. Additionally, it is important to ensure the security of the analysis environment in order to prevent harm to the analysis environment when ransomware is operated in dynamic analysis (Egele, Scholte, Kirda & Kruegel, 2008).

Data obtained with the dynamic analysis method can be said to be more significant compared to static analysis because it allows the opportunity to fully see the movement capability, activity and capacity when the ransomware is operated. Additionally, dynamic analysis offers limited appearance. Ransomware may display different behavior according to the status of the analysis environment and time. Additionally, performing analyses in virtual environments limits determination of all behavior of some malicious software.

The methods commonly used within the scope of dynamic analysis of ransomware can be defined as follows (Fig. 2):

- Memory analysis: Although malware develops a number of circumvention tactics to avoid detection, malware leave a trace in the memory of the system they are running on. Memory analysis describes the analysis made by taking the traces left by the malware on the memory while it is running. The advantage of this method is that the data kept in memory is volatile.
- Windows registry: A database that stores low-level system settings for the Windows operating system. There is a lot of information in this area, especially retrieved from security, services and user account settings. This information has a very important place in malware analysis.
- Network activity: Includes activities to detect suspicious network traffic and packet analysis on the computer where the malware is running using various analysis tools. Attackers may try to contact the attacker to report the malicious software they produce when the victim becomes active in the system or to send data from the victim system. If the anomalies in the network traffic can be analyzed, it can allow the attackers to be traced.
- Code analysis: It covers the analysis to reveal what malware does and what kind of function it has. Malware can be produced packaged or obfuscated to circumvent code analysis. For this reason, malicious code analysis is very difficult and complicated.

2.5. Hybrid analysis

Hybrid analysis is the analysis technique where both static and dynamic analyses are performed together (Fig. 2). In this method, the features obtained from static and dynamic analysis methods are used together in a certain methodology (Alkhateeb, 2017). The basic target of the hybrid analysis method is to ensure elimination of the limitations encountered in static and dynamic analyses by using these two methods jointly.

Additionally, there are systems called sandboxes which automatize malware analysis (Hull, John, & Arief, 2019). Sandboxes provide a complete static and dynamic analysis of ransomware in an automatized way and present reports of all processes and outcomes. Unfortunately, apart from several sandbox softwares, when most sandbox software is operated within the operating system, anti-analysis techniques may be developed easily by a variety of agents (these include event/network monitoring agents, virtualization services, etc.). Additionally, some ransomware notice when they are operated in sandbox environments and hide their true behavior and appear like harmless software.

The methods commonly used within the scope of hybrid analysis of ransomware are given as follows (Fig. 2):

- Automatic identification: It defines the automatic analysis of malware running in a controlled environment using tools such as sandbox. The disadvantage of this method is that when malicious software realizes that it is running in a sandbox environment, it can change its activities.
- Anti-debugging: Anti-Debugging is one of the popular anti-analysis techniques. It is a technique against malicious code being debugged and used in analysis. However, attackers can also develop ways to circumvent Anti Debug with the methods they have developed.
- Anti-emulation: Anti-Emulation checks are made to determine whether malware has anti-emulator (Anti-VM) features. Attackers commonly use this method to prevent malware detection in the analysis environment.

2.6. Techniques and tools for detecting and analyzing ransomware

This section provides a brief summary of the techniques and analysis tools used in the detection and analysis of ransomware (Fig. 4).

Fig. 4 shows the most common tools used in detection and analysis of ransomware. These tools used in the analysis of ransomware are given in Table 1 in detail.

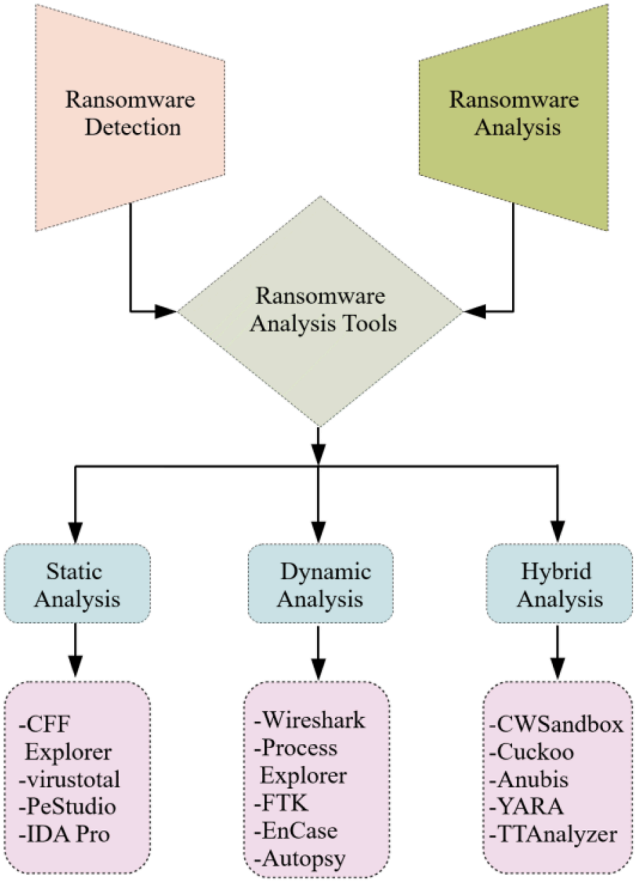

Fig. 4. Ransomware detection and analysis tools.

Table 1
Tools available for ransomware analysis (Talukder and Talukder, 2020).

Static	Dynamic	Hybrid
CFF Explorer: Analyse suspicious PE files.	Wireshark: Suspect network and packet protocol analyzer.	CWsandbox: Automatic sandboxed analysis of ransomware.
Virustotal: Online analysis of ransomware samples.	Process Exploler: Analysis of applications and application extensions (DLLs).	Cuckoo: Open source sandbox for Analyzing ransomware.
PEStudio: Analyse spot artifacts of executable files.	FTK (Acsees data): Dynamic analysis of ransomware.	YARA: Ransomware research and detection.
IDA Pro: Analyse of ransomware binary instructions.	EnCase: Forensic data investigation.	TTAnalyzer: Automated detection and analysis of Ransomware.
	Autopsy: Free dynamic analysis of ransomware.	

2.7. Related work

To intervene against these attacks, in 1996 Young and Yung (1996) found a solution to save encrypted files. From then until the present day, ransomware analysis has developed a lot and there are many studies about analysis methods for ransomware. In this section, we focus on the most commonly used methods and review some in brief. Our focal point in these attacks with broad range of targeted systems will be our area of expertise: Windows OS.

Shijo and Salim (2015) proposed an analysis technique integrating detection and classification for unknown files. In this approach, static analysis was completed using the Printable Strings Information (PSI) feature and they used dynamic analysis approaches to pick out Window API system calls. In the experiments, static analysis results had detection rate of 95.8%, while dynamic implementation and hybrid approaches had detection rate of 98.7%.

However, Shaid and Maarof recommended taking images of malware. In this technique, the Windows API system calls of the malware are caught and they are transformed to visual clues or images and used for detection of malware (Al-rimy et al., 2018).

In another study, Singh et al. (2017) detected ransomware using behavior-based multiple API system calls. In this method, multiple API sequences are created using priority n-grams. Dice coefficient, Cosine Coefficient and Tversky index were used to determine similarities between ransomware while determining multiple API sequences. The created sequences were classified using algorithms of malicious software. In this method, the desired performance values could not be obtained according to the manual analysis due to the difficulties of defining the correct behavior of the ransomware, the high number of extracted features, and the fact that the ransomware hides its real behavior in the virtual environment.

In other studies, both Salehi et al. (2014) and Han et al. (Han, Kim & Im, 2012) classified malware based on API system calls. Salehi et al. used Windows API frequencies to classify Windows API calls. They classified malware with 98.4% accuracy in their results. Additionally, Ma et al. (2016) proposed a method combining static and dynamic classifiers to reduce false positivity in classification of ransomware. This method increased accuracy of static import functions and dynamic search functions while reducing false positivity.

Kharraz et al. (2015) researched several different ransomware types encrypting operating system files and stealing personal information from the system in their study. They identified that this type of ransomware was very dangerous in reality and just as it might be used to delete and encrypt all files on the system, at the same time it may extract important information and send it elsewhere without the user's knowledge.

In the detection and analysis literature related to ransomware, the content of very unique approaches should be noted. In the study recommended by Galal et al. (2016), both static and dynamic features are used together. In this Windows-based new hybrid method, they used the static analysis features and opcodes, then calculated the term frequency (tf) and formation frequency of each opcode. The PE files operated in a sandbox environment are transformed to system calls and are identified in DLL loading and function diaries. Then, error records created by the system with these results are vectorized. They showed that malware classification with machine learning methods and the hybrid approach provided higher accuracy compared to both static and dynamic methods.

Das et al. (2016) proposed a hardware-based real-time ransomware detection system. In this proposed system, about 30% of ransomware is detected while the program is running. The detection system is hardware-based and provides some advantages over automatic-based detection systems. The proposed approach first extracts the system calls and creates high-level semantic features by modeling malicious behavior. The real-time analysis method used in the study has a relatively higher success rate in detecting ransomware, as the analysis takes more time than the automatic analysis method. In our study, similar to Das et al.'s study, we preferred real-time analysis of ransomware.

Cabaj et al. (2016) allowed infiltration of a computer in order to understand how ransomware called CryptoWall worked in their case study. After the ransomware infiltrated the victim's system, they identified that it attempted to create a successful connection with the attacker through the C&C server to send an encryption key and to encrypt files. Moving from this point, they proposed a methodology to identify the suspect proxy servers with dynamic analysis based on behavior related to the desire of the software to contact the attacker and to prevent this type of attack by blacklisting these proxy servers. Similarly, in the method proposed in our study, we used dynamic analysis methods to check network traffic to reach information belonging to the attacker.

Another study by Hasan and Rahman (Hasan and Rahman., 2017) performed static and dynamic analysis to detect ransomware. Similarly, in our study, we used a method consisting of static and dynamic analyzes in the detection and analysis of ransomware. According to the authors, they argued that due to the complexity of the code structure of ransomware, its behavior could be understood easier and better with a manual approach.

Another study used program tools (such as FTK, Autopsy, Wireshark) to detect and analyze ransomware (Talib, 2018). In this approach, by creating a controlled and clean analysis environment, suspicious files were analyzed using static and dynamic analysis. Similarly, in this study, they used analysis software tools to detect and analyze ransomware. At this point, we paid particular attention to using free software tools that are accessible to everyone for analysis, rather than program tools that work with a dongle.

In 2019, Kara (2019) recommended a methodological algorithm about how to perform malware analysis. In a similar study, Kara et al. (2019) discussed how they would use a detection and analysis approach in a real cyberattack example. There are two significant deficiencies in the method used in this approach. The first is that the majority of programs used in this proposed analysis approach operated with an equipment lock (dongle). Secondly, there was no dataset or online website reference in the paper in order to be able to assess the useability of the recommended method. In this context, it is clear an online website (http://ilkerkara.karatekin.edu.tr/) containing this dataset and where analyses can be performed without cost will be beneficial. To be more specific, we aimed to create a model where analysis can be performed by determining whether suspect exe files have malicious purposes to contribute to the ransomware detection and analysis problem without any negative example.

As a result, we added an online website with an appropriate dataset where ransomware detection and analysis can be performed both for use in real attack case studies investigated in this study and for use in advanced studies. We cooperated with an information security company

(Comodo Group, Inc. USA) to create this dataset for current malware. This company has a special team and system to collect malware examples. In our study, real malware families, acquired from Comodo, for Windows operating systems were used. The proposed dataset is publicly available for non-commercial purposes and can be accessed via the link of http://ilkerkara.karatekin.edu.tr/

3. Materials and methods

In this section, we describe the recommended method in the study. The recommended method generally comprises four steps (Fig. 5). The application steps for the method are explained in the relevant sections.

3.1. Image copy process

In order for no changes to occur in the victim's system during ransomware detection and analysis investigations (in situations where there is the possibility that ransomware may harm the victim's system during analyses and for preservation of the integrity of evidence in forensic investigations), investigation should be made on a copy of the evidence (image). Apart from some special situations (like being unable to take an image of the victim's system or to prevent loss of instantaneous volatile evidence), it is necessary to abide by this rule for ransomware detection and analysis. The image-taking process involves taking a copy of every sector (available data, deleted data, hidden sections, other data found in data storage units) in a system (computer, laptop, tablet, smartphone,

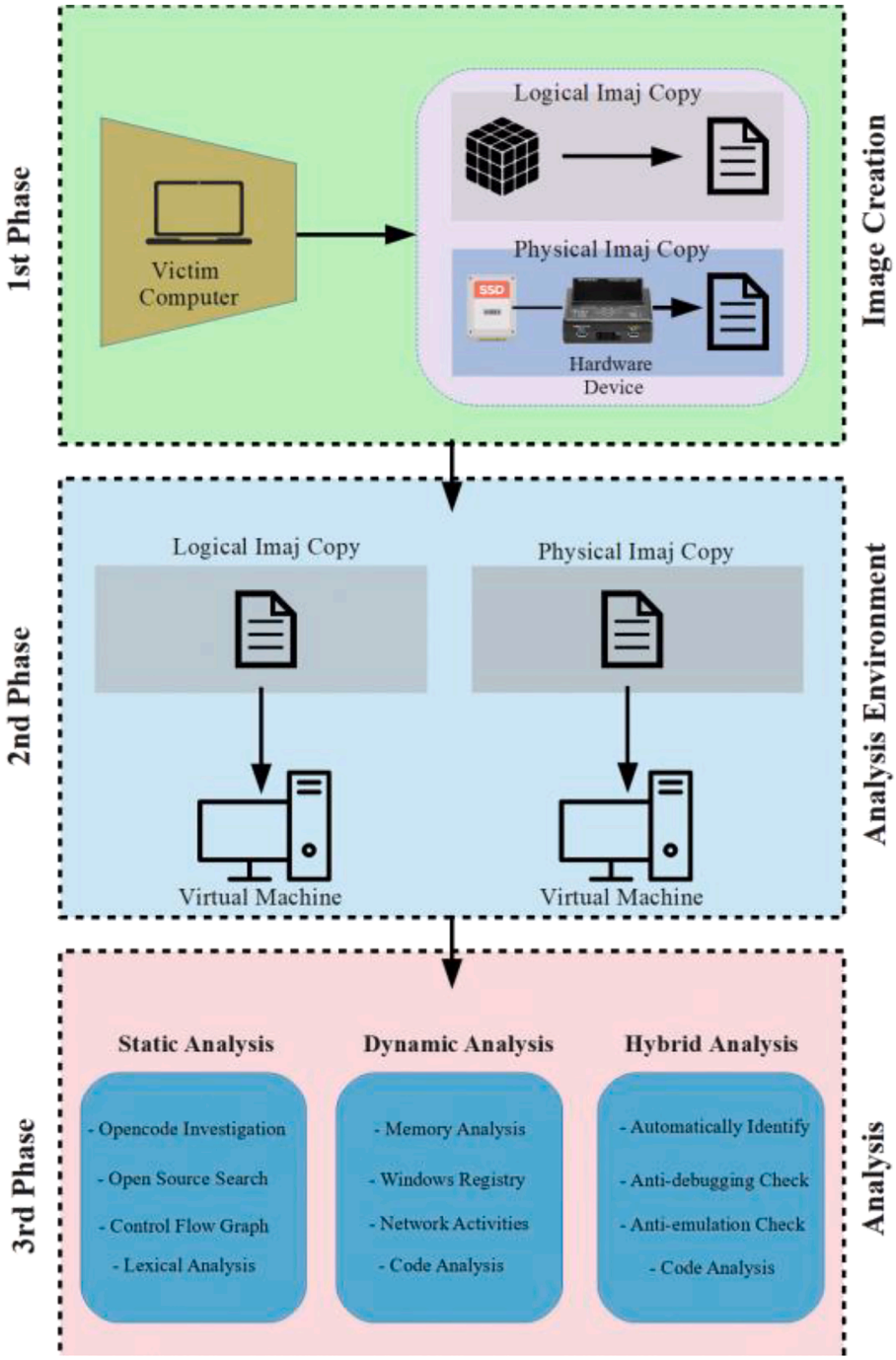

Fig.5. The workflow of the approach for ransomware analysis algorithm.

mobile hard disk, etc.) by using a device or program (Kara, 2019). It is necessary to ensure accuracy of the image taken according to international standards (write blocker). Image confirmation is defined as the use of a range of letter and number combinations to check whether the integrity of any file is complete or not. With this aim, it is necessary to be able to produce MD5 or SHA-1, SHA-256 hash values to confirm the original system and the image when image-taking procedures are completed.

There are many methods for obtaining images in the literature, but they are generally divided into two categories. These are a) images taken using special software and b) physical images taken with image kits.

3.2. Image animation process

Ransomware detection and analysis investigations should be performed in a safe environment prepared for the investigation and this analysis environment should be distant from the possible effects of the ransomware. As a result, analyses require special equipment (workstation). After preparing the safe analysis environment, the image animation process is performed. The image animation process means full visualization of the image content with appropriate investigation software. After the image is animated, the detection and analysis stages begin.

3.3. Detection and analysis

Before beginning the analysis stage, it is necessary to identify the presence of possible ransomware. As a result, the most important stage in this step is 'detection'. The detection stage may be simple or complicated according to the aims of the attacker. In simple situations, if the attacker wants to obtain financial gains from the attack on the victim's system, they leave a trace behind (like a warning message). This trace may be assessed to obtain information about the type and outcomes of the attack. In complicated situations, if the attack is performed to observe the system, steal information or harm the victim's system, the attacker may not expend special efforts to leave a trace in the victim's system. In this situation, firstly investigations are performed beginning with the last operations on the victim's system and working backwards through older operations (like internet history, network movements or user commands) to detect the attack.

In the analysis stage, procedures generally follow a path from easy to difficult and complicated. With this aim, the relatively easier static analysis research into open source information is performed and information is obtained about internet history, text statements contained in the ransomware, functions used, structural content in Windows file directory and whether it is packaged or not, hash confirmation values (like MD5, SHA1) and dates when the victim's system was infiltrated. Later dynamic analysis observes the behavior of the ransomware involving memory analysis of which machines and programs are affected in the victim's system, Windows file directory, record books, IP traffic and network activity and movement capability. Dynamic analysis allows the opportunity to fully observe the features, activity and capacity of the malware.

4. The proposed method

In this section, we show detection and analysis in detail with our ransomware analysis method for a case study.

4.1. Experimental setup

All analyses were performed on a Dell Precision T5820 brand work station with Xeon W-2133/16 GB/256 GB SSD running Windows 10 Pro software. In order to prevent being affected by possible attacks from the ransomware example on the image copy taken from the victim's computer, the workstation was operated in Virtual Machine mode. Firstly "VirtualBox 6.1.16 (Free Version)", "FTK Imager 4.2.0 (Free Version)", "Autopsy 4.16.0 (Free Version)" and "Wireshark 3.2.7 (Free Version)" programs for detection and analysis were loaded onto the workstation and used for analyses. Analysis was started on 2021.01.02 10:41:35 UTC (see Fig. 9. Last analysis time). All analyzes took approximately 2 weeks to complete.

4.2. Analysis of case study

It is a well-known fact that a well-chosen and real case sample is vital for data-driven studies. The most important criterion for case selection is the selection of the most appropriate case for the purpose of the research. For this purpose, attention has been paid to the fact that the case sample selected in the study provides in-depth information about the subject, which studies a current phenomenon within its real-life framework (content), and provides practical benefits in practice. Similar to other types of ransomware, the "Onion" is an encrypting ransomware. User data is encrypted and a countdown mechanism is employed. This countdown is a compelling threat for users to pay up in Bitcoins. Usually a strict 72-hours deadline is forced on users by cybercriminals, otherwise it is almost certain that the files will be lost forever. The name "Onion" is given by Kaspersky Lab due to the fact that this malware uses the anonymous network Tor (the Onion Router). This router enables malicious code hide its bad nature. It is also very hard to track the criminals behind. (Kaspersky, 2021).

Technical improvements to the ransomware have made it a potential successor to Cryptolocker, a truly dangerous threat as one of the most sophisticated encryptors today (Zeng et al., 2020; Kaspersky, 2021).

For this reason, "Onion Ransomware", which is one of the most common real cyber attacks recently, was preferred in the study.

The ransomware analysis method recommended in the study was applied step-by-step to a real ransomware attack. With this aim, a computer with suspected cyberattack was chosen. The virtualization software called VirtualBox was used on the workstation to complete the animation processes for the image copy of the victim's computer taken in E01 format with the FTK Image program. When the VirtualBox virtualization software is installed, it uses the VBoxManage tool in the directory to transform the image file to ImageFile.raw format with VBoxManage.exe converter. This .raw file is imported into the VirtualBox virtualization software and the image is operated live and the specifications of the victim's computer are identified first (Table 2).

To perform analyses, the process starts with the observation of encrypted files found on the victim's computer desktop (Fig. 6).

Investigations encountered a message from the attacker in the "Read_ME.html" file on the desktop, and when file content was examined, it was seen the attack was a ransomware attack (Fig. 7).

Fig. 7 shows the message left by the attacker to the victim system. It was understood that the type of attack made from the message content and the data was encrypted as a result of the attack. Moreover, it is stated that the "Decrytor" application must be purchased in order to regain access to the encrypted files. When the "Buy Descriptor" button in this message is clicked, it is seen that the application runs (Fig. 8).

After determining the type of attack, the static analysis step begins

Table 2
Image information of victim system.

Product Name	Microsoft Windows 10 Home
Install Date	03.12.2019–18:51:36 UTC
Last Shotdown Time	17.12.2020–13:27:09 UTC
Physical Disk	985.713.218 Sectors 485,1 GB
Total Size	500.107.862.016 Bytes (485,1 GB)
Total Sectors	985.713.218
Acquisition MD5	e56bbd39ff5a523f8250fd43ca12ya
Verification MD5	e56bbd39ff5a523f8250fd43ca12ya
AcquisitionSHA1	ckd453498543a063cd160u745b7sa34deac2b2
VerificationSHA1	ckd453498543a063cd160u745b7sa34deac2b2

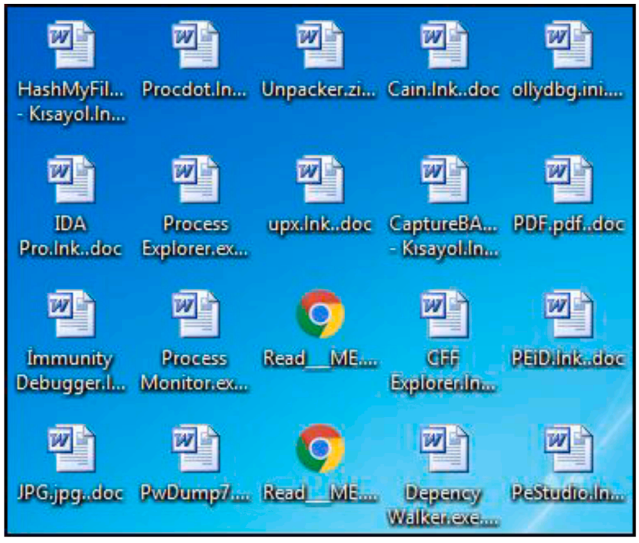

Fig. 6. Screenshot of file encryption operations performed on the system by the ransomware.

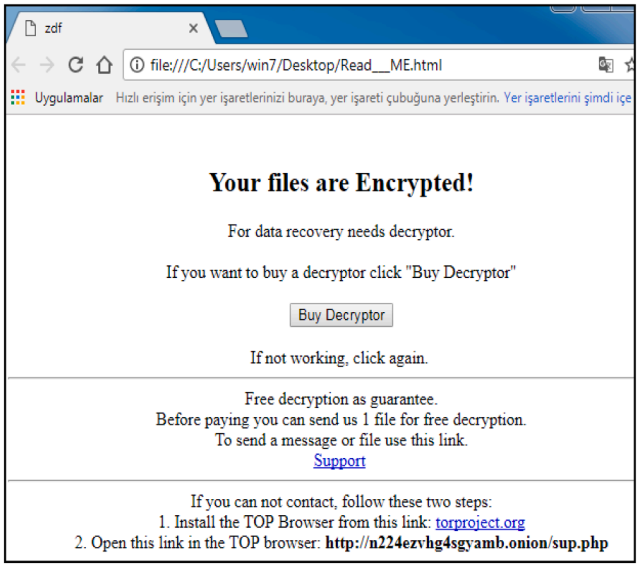

Fig. 7. Screenshot of the content of the file "Read_Me.html".

(Fig. 8). Firstly, the internet history of the victim's computer was investigated to identify the ransomware completing the attack. It appeared a suspect file called "JH67RdfgD.exe" was downloaded in the final operation, so our analyses were focused on this file. With this aim, open source research interrogated the "JH67RdfgD.exe" file on the www.virustotal.com website (Fig. 9).

As a result of the query made on the virustotal website, it was seen that the suspicious "JH67RdfgD.exe" file was identified as malicious software (crypto-ransomware) by different anti-virus programs. After it was understood that the suspicious file was of the type of crypto-ransomware, the dynamic analysis phase was started. Within the scope of dynamic analysis, "JH67RdfgD.exe" was run in a secure environment and file-directory and Windows-registry movements were examined with the Autopsy program.

When the malicious code in Fig. 10 is examined, it is seen that Onion Ransomware first creates its copy in the Users\Admin\AppData \Roaming\JH67RdfgD.exe file. Then, it creates "Read_Me.html" file on the desktop. After completing the infiltration to the victim system, it becomes active under the C:\Windows\Prefetch directory. After

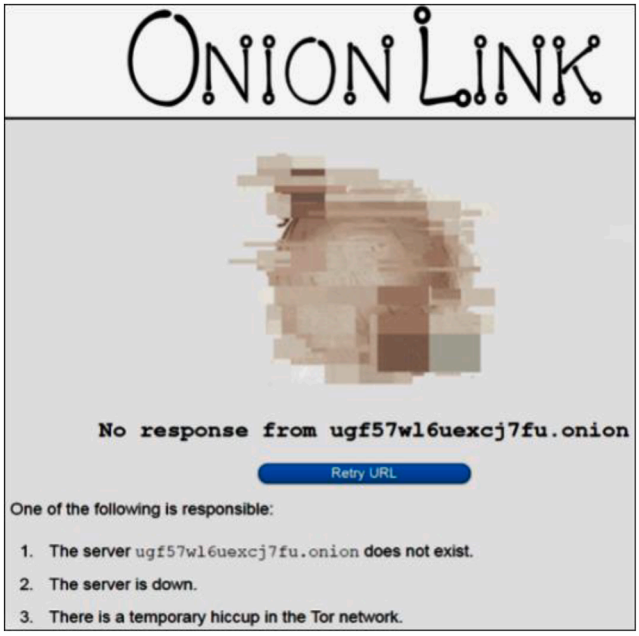

Fig. 8. Screenshot of the inactive page to be used for purchasing the decryptor application.

Fig. 9. Screenshot of the query of the "JH67RdfgD" file on the www.virustotal.com website.

creating the files, it deletes the running process by the C:\exception.log command.

When the malicious code in Fig. 11 is examined, it is seen that Onion Ransomware creates its entry under the HKCU\Software\Microsoft \Windows\currentversion key. It then creates a control panel on the desktop with the "control panel\desktop\" command. And, it encrypts the files in the target system with the HKLM\Software\policies

Fig. 10. File-directory logs of Onion Ransomware, called "JH67RdfgD.exe".

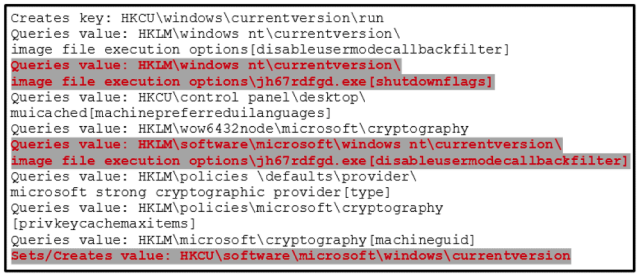

Fig. 11. Registry logs of the Onion Ransomware named "JH67RdfgD.exe".

\Microsoft\cryptography command.

As a result of the static analysis, "JH67RdfgD" file was detected as a malicious software of the kind that performs file encryption (crypto-ransomware), and then the dynamic analysis phase was started to examine the behaviour of the malware.

After examining the file-directory and registry movements of the malware named "JH67RdfgD", behavior analysis was carried out with the Wireshark program to determine the domain name and IP address, in which the malware can communicate with the attacker. As a result of the analysis, the malware was observed to create "Read_Me.html" file in the victim computer, without communicating directly with any domain name and IP address after being activated in the system (Fig. 7). The examination of "Read_Me.html" revealed that the message content tells user to connect to the domain name "ugf57wl6uexcj7fu.onion.link" on the Tor network to buy the decryptor application, which will decode the files for a ransom (Fig. 8).

After detecting that the malware communicates with the "ugf57wl6uexcj7fu.onion.link" Tor address through the "Read_Me.html" file it creates, Wireshark software was used to analyze its network activities in order to detect its IP service provider address (hosting). In the analysis, domain name and IP address were found to be accessible (Fig. 12).

IP address detection is used in investigations to reach the attacker. By using the IP address, the name, surname, contact information of the domain owner belonging to the attacker and the information of the domain registrar can be accessed.

The attacker's Whois information was tried to be obtained from the identified IP address, and according to the results of the query made via the www.domaintools.com website, attacker's Whois information was found to be accessible (Fig. 13).

Password decryption attempts were unsuccessful due to the 256-bit algorithm AES electronic data encryption feature of the selected cyberattack example, Onion Ransomware v5.0.5.

5. Discussion

There are many methods used for ransomware detection and analysis in the literature. These methods generally use the technical characteristics of ransomware to classify them, and focus on the detection and prevention of similar ransomware attacks, by making use of the properties of these classifications. Attackers, on the other hand, try to circumvent the security measures by using different methods (such as packaging) to avoid detection of ransomware they have programmed. This reduces the success rate of detection and prevention of ransomware. Another problem is the high cost of software used in the security and protection of personal and institutional private data. It is not enough to purchase these tools on a one-time basis since they need to be updated on a paid basis at specific intervals to combat current threats.

The biggest disadvantage of Automated analysis is that it is determined as normal due to the possibility that malware may not show all its behaviors while running in virtual machine and sandbox environments (Alosefer, 2012). In addition, while the malware is being designed, with the ability to detect that it is running in virtual environment (anti-sandbox techniques), it can make smart decisions (Amro, 2017).

However, in the manual analysis approach, the activities of malicious software can be monitored in real time by using various analysis tools (Egele et al., 2008). In addition to this, attackers can add time constraints for malware to run in order to bypass the automated analysis method. With its general structure, this control of malware that performs debugger control stands out as a technique that complicates the automatic analysis method. It is possible to perform this control at the time of first boot, before it performs the malicious behavior, or at any time during the runtime (Linn and Debray, 2003).

Similar to the work of Cabaj (Cabaj et al., 2016), the proposed ransomware case analysis method bases ransomware detection and analysis on the use of appropriate analysis tools. The approach used in this study is called a case analysis. Case analyses are required for intervention and status analysis after cyberattacks occur both for detection of the source of the attack and for correct operation of legal processes. These processes involve some difficulties.

The detection and analysis processes for ransomware attacks are difficult and time-consuming. It is necessary to work under threat of possible attack in the analysis environment used for investigation. As a result, experts choose to perform analyses in virtual machine

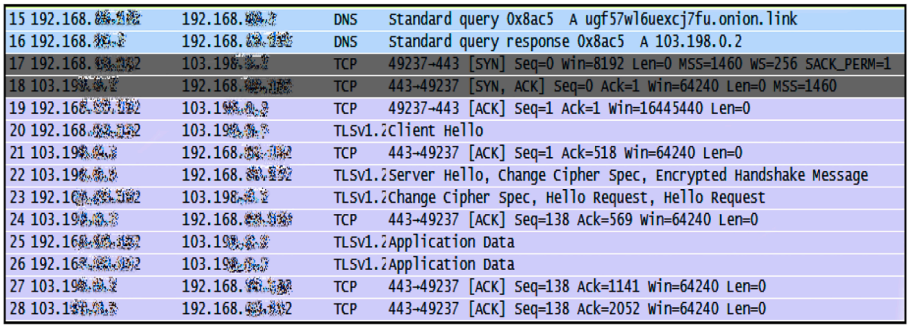

Fig. 12. Screenshot of the data recorded through the Wireshark software.

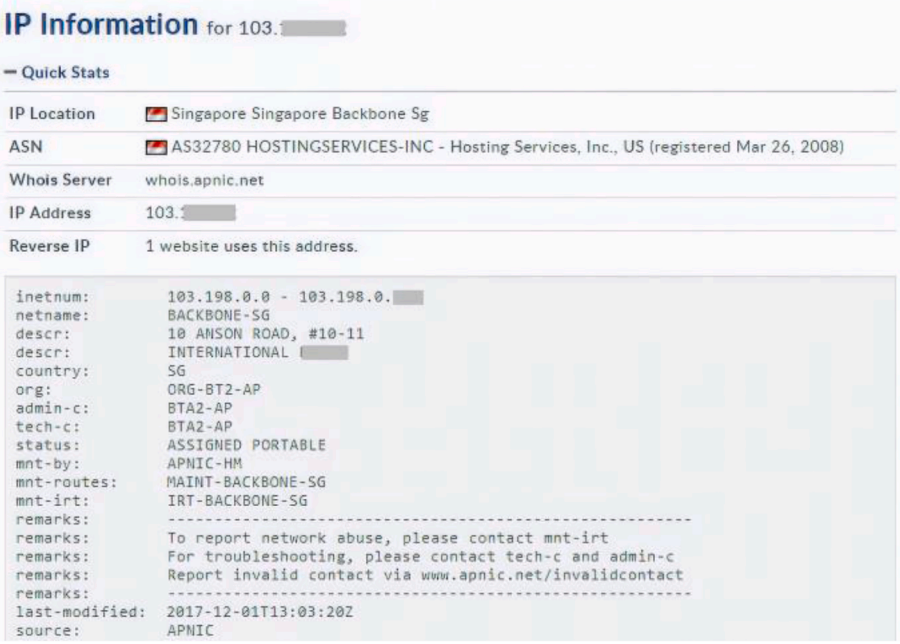

Fig. 13. Screenshot of the query of the Whois information file on the www.domaintools.com website.

environments to minimize threats. This leads to uncertainty about how the ransomware threat occurs in a real environment. Some types of ransomware have different operating capability than their target when they operate in analysis environments. This situation puts the reliability of the processes in doubt and leads experts into error.

Many ransomware detection and analysis methods have been developed and continue to be developed to overcome these problems. Especially we believe automatic detection and analysis mechanisms strategies for newly-developed artificial intelligence, machine learning-based ransomware will contribute to the struggle against these large-scale attacks.

5.1. Preventive measures and matters needing attention by all stakeholders: Best practice

As a result of this study and common knowledge, critical actions and considerations for ransomware detection, analysis and prevention can be listed as follows for all stakeholders:

For Company Owners/Managers;

- Since ransomware threats can reach users from different network points, comprehensive protection should be provided with security technologies.
- Ransomware attacks target the entire system from the point where it infiltrated the system. For this reason, authority limitations should be made for each personnel. Do not share the admin profile (username/password). Most software cannot be installed without administrator rights, which provides protection from many potential problems.
- It is easier to take action against ransomware than to repair damage after it has been hacked. Therefore, all personnel should be aware of this threat through Trainings and Practices.

For IT Employees;

- Up-to-date anti-virus software should be installed for each user to prevent ransomware attacks. Unwanted codes should be prevented from running in browsers, anti-virus and firewall software should be available and updated on each user's computer, and every downloaded file should be checked by anti-virus software.
- At the moment of the attack, the electricity must be disconnected without interfering with the victim computer. Then the victim computer should be booted by running it in safe mode.
- System Restore on the victim computer should be reviewed. The System Restore point option snapped before the attack can be used to get rid of some ransomware attacks. However, in the new generation attacks, it is seen that the attackers delete the System Restore files.
- Formatting or scanning the hacked files by anti-virus software could be employed, but it should be noted that the purpose here is not to get rid of the ransomware, but to remove the damage it causes.
- Keeping backups of encrypted files may be important due to the possibility that they will be decrypted in the future.
- Take regular backups and keep the recent backup in a different location. Businesses with up-to-date backups of files need not fear ransomware.

For End Users;

- Care should be taken when opening spam e-mail attachments or links of unknown origin. Opening attachments from undesirable sources, clicking on impressive or surprising images, or clicking on an innocent-looking advertisement can redirect the user to a harmful website and install malware on the computer. This method is often used by attackers in ransomware attacks.
- Unused remote accesses should be closed. In ransomware attacks, malicious content usually tries to communicate with the victim computer to transmit critical information to the attacker.
- As a result of a ransomware attack, the attacker should not pay for the encryption key promised to be sent for decryption. Because usually the files are deleted at the time of the attack, and even if the encryption key is reached, the files cannot be accessed.

6. Conclusion

When the detection approaches of ransomware in the literature and the analysis methods developed using these approaches are examined, some deficiencies that reduce the performance have been identified.

These deficiencies can be listed as follows:

- Automated analysis approaches fail to detect ransomware signatures;
- Unclarity of ransomware behavior;
- Ransomware has a lot of features and these features are not related to each other;
- Inadequate detection of next-generation ransomware;
- The analysis made in the virtual environment is easily circumvented by ransomware and is not resistant to attacks and cloaking techniques.

These deficiencies were taken into account in the manual analysis method proposed in this study and necessary improvements were made at each step. Contributions can be listed as follows:

Suspicious files can be analyzed manually using analysis tools. By using static analysis method, ransomware behaviours can be determined by examining their activities, whose characteristics are determined in real time. While evaluating these interactions, the behaviors and the severity level for each behavior are calculated by looking at the interactions made and in which file path these interactions are made. Suspicious files should be analysed within the framework of a certain method using several different analysis tools. For this purpose, a useful and easily applicaple analysis method for ransomware was proposed and analyzes were performed on a real case sample. Since the formal method steps and the dataset are available for researchers, and it is publicly available, all analyses can be carried out by anyone in need.

As a result of the analyzes made on the sample case in the study, different behaviors were obtained by looking at the type of ransomware and the family it belongs to, and by examining these behaviors, ransomware-specific findings were obtained. As a result of the findings, it was seen that the attacker was traceable.

In this context, practical suggestions for the detection and analysis of ransomware can be listed as follows:

- Strong analysis methods should be recommended against ransomwares cloaking techniques. Because cloaking techniques hide the actual behavior of ransomware, they make ransomware analysis difficult and cause false flags.
- In order to determine the right behavior of ransomware, the analyzed software needs to be run in different environments.
- By combining the good aspects of the ransomware analysis approaches (manual-automatic analysis), more effective and holistic approaches should be proposed.

This study presents a basic review of the methods consisting of free tools that can be used in the detection and analysis of ransomware. The proposed method was applied to a selected ransomware sample. In order for this method to be repeatable by experts, an easily accessible sample was selected.

To support this study, we published the selected ransomware example for public interest (http://ilkerkara.karatekin.edu.tr/). Furthermore, we added an online website to our study containing a dataset. Dataset contains 11 different malware families, which can be used for different studies.

As a result, the recommended approach is promising for use in detection and analysis of ransomware used for similar attacks. Due to the increase in ransomware threats we believe forensic investigation and analyses will gain more important in the near future.

This manuscript presents first phase of our research and proves that the proposed approach is applicable in real life cases. The second phase of our research is going to focus on the capabilities of Machine/Deep Learning based detections methods and the compatibility of our proposed analysis method with the ML techniques.

CRediT authorship contribution statement

Ilker Kara: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft. **Murat Aydos:** Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank Comodo Group, Inc. USA for providing the malware dataset. Thus, we added an online website with an appropriate dataset where ransomware detection and analysis can be performed and investigated. This link may also be used for justification of our results in this study and for future use in advanced studies.

Funding/support

The research was supported by the Cankiri Karatekin Universty. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Cankiri Karatekin Universty.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2021.116198.

References

Akbanov, M., Vassilakis, V. G., & Logothetis, M. D. (2019). Ransomware detection and mitigation using software-defined networking: The case of WannaCry. *Computers & Electrical Engineering, 76*, 111–121. https://doi.org/10.1016/j.compeleceng.2019.03.012

Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security, 74*, 144–166. https://doi.org/10.1016/j.cose.2018.01.001

Ahmadian, M. M., Shahriari, H. R., & Ghaffarian, S. M. (2015, September). Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)* (pp. 79-84). IEEE. https://doi.org/10.1109/ISCISC.2015.7387902.

Alkhateeb, E. M. S. (2017, August). Dynamic malware detection using api similarity. In *2017 IEEE international conference on computer and information technology (CIT)* (pp. 297-301). IEEE. https://doi.org/ 10.1109/CIT.2017.14.

Andronio, N., Zanero, S., & Maggi, F. (2015, November). Heldroid: Dissecting and detecting mobile ransomware. In international symposium on recent advances in intrusion detection (pp. 382-404). Springer, Cham. https://doi.org/10.1007/978-3-319-26362-5_18.

Alosefer, Y. (2012). Analysing web-based malware behaviourthrough client honeypots. Doctoral dissertation, Cardiff University School of Computer Science & Informatics. http://orca.cardiff.ac.uk/id/eprint/29469.

Amro, B. (2017). Malware detection techniques for mobile devices. *International Journal of Mobile Network Communications & Telematics (IJMNCT), 7*(4). https://doi.org/10.2139/ssrn.3430317

Baldwin, J., & Dehghantanha, A. (2018). Leveraging support vector machine for opcode density based detection of crypto-ransomware. In *Cyber threat intelligence* (pp. 107–136). Cham: Springer. https://doi.org/10.1007/978-3-319-73951-9_6.

Banescu, S., Wuchner, T., Salem, A., Guggenmos, M., Ochoa, M., & Pretschner, A. (2015, October). A framework for empirical evaluation of malware detection resilience against behavior obfuscation. In *2015 10th international conference on malicious and unwanted software (MALWARE)* (pp. 40-47). IEEE. https://doi.org/ 10.1109/MALWARE.2015.7413683.

Cabaj, K., & Mazurczyk, W. (2016). Using software-defined networking for ransomware mitigation: The case of cryptowall. *IEEE Network, 30*(6), 14–20. https://doi.org/10.1109/MNET.2016.1600110NM

Chen, Z. G., Kang, H. S., Yin, S. N., & Kim, S. R. (2017). September. Automatic ransomware detection and analysis based on dynamic API calls flow graph. In *Proceedings of the international conference on research in adaptive and convergent systems* (pp. 196–201). https://doi.org/10.1145/3129676.3129704

Choudhary, S. P., & Vidyarthi, M. D. (2015). A simple method for detection of metamorphic malware using dynamic analysis and text mining. *Procedia Computer Science, 54*, 265–270. https://doi.org/10.1016/j.procs.2015.06.031

Das, S., Liu, Y., Zhang, W., & Chandramohan, M. (2016). Semantics-based online malware detection: Towards efficient real-time protection against malware. *IEEE Transactions on Information Forensics and Security, 11*(2), 289–302. https://doi.org/10.1109/TIFS.2015.2491300

Egele, M., Scholte, T., Kirda, E., & Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys (CSUR), 44*(2), 1–42. https://doi.org/10.1145/2089125.2089126

Fan, Y., Ye, Y., & Chen, L. (2016). Malicious sequential pattern mining for automatic malware detection. *Expert Systems with Applications, 52*, 16–25. https://doi.org/10.1016/j.eswa.2016.01.002

Ferrante, A., Malek, M., Martinelli, F., Mercaldo, F., & Milosevic, J. (2017, October). Extinguishing ransomware-a hybrid approach to android ransomware detection. In *International symposium on foundations and practice of security* (pp. 242-258). Springer, Cham. https://doi.org/10.1007/978-3-319-75650-9_16.

Galal, H. S., Mahdy, Y. B., & Atiea, M. A. (2016). Behavior-based features model for malware detection. *Journal of Computer Virology and Hacking Techniques, 12*(2), 59–67. https://doi.org/10.1007/s11416-015-0244-0

Ganesh, N., Di Troia, F., Corrado, V. A., Austin, T. H., & Stamp, M. (2016). March. Static analysis of malicious Java applets. In *Proceedings of the 2016 ACM on international workshop on security and privacy analytics* (pp. 58–63). https://doi.org/10.1145/2875475.2875477

Gazet, A. (2010). Comparative analysis of various ransomware virii. *Journal in Computer Virology, 6*(1), 77–90. https://doi.org/10.1007/s11416-008-0092-2

Gómez-Hernández, J. A., Álvarez-González, L., & García-Teodoro, P. (2018). R-Locker: Thwarting ransomware action through a honeyfile- based approach. *Computers & Security, 73*, 389–398. https://doi.org/10.1016/j.cose.2017.11.019

Han, K. S., Kim, I. K., & Im, E. G. (2012). Malware classification methods using API sequence characteristics. In *Proceedings of the international conference on IT convergence and security 2011* (pp. 613-626). Springer, Dordrecht. https://doi.org/10.1007/978-94-007-2911-7_60.

Hasan, M. M., and Rahman, M. M. (2017). RansHunt a support vector machines based ransomware analysis framework with integrated feature set. Paper presented at the 20th international conference of computer and information technology (ICCIT), Dhaka, Bangladesh. https://doi.org/10.1109/ICCITECHN.2017.8281835.

Hull, G., John, H., & Arief, B. (2019). Ransomware deployment methods and analysis: views from a predictive model and human responses. *Crime Science*, *8*(1), 2. https://link.springer.com/article/10.1186/s40163- 019-0097-9.

Hou, S., Ye, Y., Song, Y., & Abdulhayoglu, M. (2017, August). Hindroid: An intelligent android malware detection system based on structured heterogeneous information network. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1507-1515). https://doi.org/10.1145/3097983.3098026.

Kara, I. (2019). A basic malware analysis method. *Computer Fraud & Security, 2019*(6), 11–19. https://doi.org/10.1016/S1361-3723(19)30064- 8

Kara, İ., & Aydos, M. (2019). The ghost in the system: Technical analysis of remote access trojan. *International Journal on Information Technologies & Security, 11*(1). https://eds.a.ebscohost.com/eds/pdfviewer.

Kaur, R., & Singh, M. (2014). A survey on zero-day polymorphic worm detection techniques. *IEEE Communications Surveys & Tutorials, 16*(3), 1520–1549. https://doi.org/10.1109/SURV.2014.022714.00160

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). July. Cutting the gordian knot: A look under the hood of ransomware attacks. In *International conference on detection of intrusions and malware, and vulnerability assessment* (pp. 3–24). https://doi.org/10.1007/978-3-319-20550-2_1

Ki, Y., Kim, E., & Kim, H. K. (2015). A novel approach to detect malware based on API call sequence analysis. *International Journal of Distributed Sensor Networks, 11*(6), 659101. https://doi.org/10.1155/2015/659101

Kim, D., & Kim, S. (2015). Design of quantification model for ransom ware prevent. *World Journal of Engineering and Technology, 03*(03), 203–207. https://doi.org/10.4236/wjet.2015.33C030

Kirda, E. (2017, February). UNVEIL: a large-scale, automated approach to detecting ransomware (keynote). In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)* (pp. 1-1). IEEE. https://doi.org/10.4236/10.1109/SANER.2017.7884603.

Kumar, S. M., & Kumar, M. R. (2013). Cryptoviral extortion: A virus based approach. *International Journal of Computer Trends and Technology (IJCTT), 4*(5), 1149à.

Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). December. Deep learning for classification of malware system call sequences. In *Australasian Joint Conference on Artificial Intelligence* (pp. 137–149). Cham: Springer. https://doi.org/10.1007/978-3-319-50127-7_11.

Li, L., Li, D., Bissyandé, T. F., Klein, J., Cai, H., Lo, D., & Le Traon, Y. (2017). On locating malicious code in piggybacked android apps. *Journal of Computer Science and Technology, 32*(6), 1108–1124. https://doi.org/10.1007/s11390-017-1786-z

Linn, C., & Debray, S. (2003). October. Obfuscation of executable code to improve resistance to static disassembly. In *Proceedings of the 10th ACM conference on computer and communications security* (pp. 290–299). https://doi.org/10.1145/948109.948149

Ma, X., Biao, Q., Yang, W., & Jiang, J. (2016, May). Using multi-features to reduce false positive in malware classification. In *2016 IEEE information technology, networking, electronic and automation control conference* (pp. 361-365). IEEE. https://doi.org/10.1109/ITNEC.2016.7560382.

Mbol, F., Robert, J. M., & Sadighian, A. (2016, November). An efficient approach to detect torrentlocker ransomware in computer systems. In *International conference on cryptology and network security* (pp. 532-541). Springer, Cham. https://doi.org/10.1007/978-3-319-48965-0_32.

Nauman, M., Azam, N., & Yao, J. (2016). A three-way decision making approach to malware analysis using probabilistic rough sets. *Information Sciences, 374*, 193–209. https://doi.org/10.1016/j.ins.2016.09.037

O'Kane, P., Sezer, S., & Carlin, D. (2018). Countering cyber threats for industrial applications: An automated approach for malware evasion detection and analysis. *Journal of Network and Computer Applications, 7*(5), 321–327. https://doi.org/10.1049/ntw2.v7.510.1049/iet-net.2017.0207

Paik, J. Y., Shin, K., & Cho, E. S. (2016). May. Poster: Self-defensible storage devices based on flash memory against ransomware. *Proceedings of IEEE symposium on security and privacy*.

Prakash, K. P., Nafis, T., & Biswas, S. S. (2017). Preventive measures and incident response for locky ransomware. *International Journal of Advanced Research in Computer Science, 8*(5).

Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2017). Malware detection by eating a whole exe. *arXiv preprint arXiv:1710.09435*. https://www.aaai.org/.

Ray, O., Hicks, S., & Moyle, S. (2016). Using ILP to analyse ransomware attacks. In *ILP (Short Papers)* (pp. 54–59).

Talib, M. A. (2018). Testing closed source software: Computer forensic tool case study. *Journal of Computer Virology and Hacking Techniques, 14*(2), 167–179. https://doi.org/10.1007/s11416-017-0302-x

Tam, K., Feizollah, A., Anuar, N. B., Salleh, R., & Cavallaro, L. (2017). The evolution of android malware and android analysis techniques. *ACM Computing Surveys (CSUR), 49*(4), 1–41. https://doi.org/10.1145/3017427

Talukder, S., & Talukder, Z. (2020). A survey on malware detection and analysis tools. *International Journal of Network Security & Its Applications, 12*(2), 37–57. https://doi.org/10.5121/ijnsa10.5121/ijnsa:2020.1220010.5121/ijnsa:2020.12203

Souri, A., & Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. *Human-centric Computing and Information Sciences, 8*(1), 3. https://doi.org/10.1186/s13673-018-0125-x

Spreitzenbarth, M., Schreck, T., Echtler, F., Arp, D., & Hoffmann, J. (2015). Mobile-Sandbox: Combining static and dynamic analysis with machine- learning techniques. *International Journal of Information Security, 14*(2), 141–153. https://doi.org/10.1007/s10207-014-0250-0

Syed, S. A. (2021). Industry trends in computer software. In *Ethical hacking techniques and countermeasures for cybercrime prevention* (pp. 54–59). IGI Global.

Reddy, B. V., Krishna, G. J., Ravi, V., & Dasgupta, D. Machine Learning and Feature Selection Based Ransomware Detection Using Hexacodes (2020). In *Evolution in computational intelligence* (pp. 583-597). Springer, Singapore. https://doi.org/10.1007/978-981-15-5788-0_56.

Salehi, Z., Sami, A., & Ghiasi, M. (2014). Using feature generation from API calls for malware detection. *Computer Fraud & Security, 2014*(9), 9–18. https://doi.org/10.1016/S1361-3723(14)70531-7

Schmidt, A. D., Bye, R., Schmidt, H. G., Clausen, J., Kiraz, O., Yuksel, K. A., ... & Albayrak, S. (2009, June). Static analysis of executables for collaborative malware detection on android. In *2009 IEEE international conference on communications* (pp. 1-5). IEEE. https://doi.org/10.1109/ICC.2009.5199486.

Scaife, N., Carter, H., Traynor, P., & Butler, K. R. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In *2016 IEEE 36th international conference on distributed computing systems (ICDCS)* (pp. 303-312). IEEE. https://doi.org/10.1109/ICDCS.2016.46.

Shijo, P. V., & Salim, A. J. P. C. S. (2015). Integrated static and dynamic analysis for malware detection. *Procedia Computer Science, 46*, 804–811. https://doi.org/10.1016/j.procs.2015.02.149

Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *arXiv preprint arXiv:1609.03020*.

Singh, A., Arora, R. and Pareek, H. (2017). Malware analysis using multiple API sequence mining control flow graph. arXiv preprint arXiv:1707.02691.

Song, S., Kim, B., & Lee, S. (2016). The effective ransomware prevention technique using process monitoring on android platform. *Mobile Information Systems, 2016*, 1–9. https://doi.org/10.1155/2016/2946735

Qamar, A., Karim, A., & Chang, V. (2019). Mobile malware attacks: Review, taxonomy & future directions. *Future Generation Computer Systems, 97*, 887–909. https://doi.org/10.1016/j.future.2019.03.007

Wagner, M., Fischer, F., Luh, R., Haberson, A., Rind, A., Keim, D. A., & Aigner, W. (2015). A survey of visualization systems for malware analysis. In *Eurographics conference on visualization (EuroVis)* (pp. 105–125). https://doi.org/10.2312/eurovisstar.20151114

Wang, P., & Wang, Y. S. (2015). Malware behavioural detection and vaccine development by using a support vector model classifier. *Journal of Computer and System Sciences, 81*(6), 1012–1026. https://doi.org/10.1016/j.jcss.2014.12.014

Yerima, S. Y., Sezer, S., McWilliams, G., & Muttik, I. (2013, March). A new android malware detection approach using bayesian classification. In *2013 IEEE 27th international conference on advanced information networking and applications (AINA)* (pp. 121-128). IEEE. https:/doi.org/10.1109/AINA.2013.88.

Yunus, Y. K. B. M., & Ngah, S. B. (2020, February). Review of hybrid analysis technique for malware detection. In *IOP conference series: materials science and engineering* (Vol. 769, No. 1, p. 012075). IOP Publishing. https:/doi.org/10.1088/1757-899X/769/1/012075.

Young, A., & Yung, M. (1996, May). Cryptovirology: Extortion-based security threats and countermeasures. In *Proceedings 1996 IEEE symposium on security and privacy* (pp. 129-140). IEEE. https:/doi.org/10.1109/SECPRI.1996.502676.

Zimba, A. (2017). Malware-free intrusion: a novel approach to ransomware infection vectors. *International Journal of Computer Science and Information Security*, *15*(2), 317. https://sites.google.com/site/ijcsis/ ISSN 1947-5500.

Zhang, P., & Tan, Y. (2015, May). Hybrid concentration based feature extraction approach for malware detection. In *2015 IEEE 28th Canadian conference on electrical and computer engineering (CCECE)* (pp. 140-145). IEEE. https:/doi.org/10.1109/CCECE.2015.7129175.

Zeng, H., Liu, Z., & Cai, H. (2020, October). Research on the application of deep learning in computer network information security. IOP Publishing. In Journal of Physics: Conference Series,1650 (3), 3, (pp. 032117). https:/doi.org/doi:10.1088/1742-6596/1650/3/032117.

(Kaspersky., 2021). https://www.kaspersky.com/resource- enter/threats/onion-ransomware-virus-threat.