Bitlis Eren Üniversitesi Fen Bilimleri Dergisi
BİTLİS EREN UNIVERSITY JOURNAL OF SCIENCE
ISSN: 2147-3129/e-ISSN: 2147-3188
VOLUME: 11 NO: 2 PAGE: 508-518 YEAR: 2022
DOI: 10.17798/bitlisfen.1038966

Ransomware Detection in Cyber Security Domain

Ömer ASLAN[1*]

[1]*Bandırma Onyedi Eylül University, Department of Software Engineering, Balıkesir-Turkey (ORCID: 0000-0003-0737-1966)*

Keywords: Cyber security, Ransomware detection, Behavior-based detection, Machine learning

Abstract

In recent years, ransomware has become highly profitable cyber attacks. This is because, everyday there are several new devices attending to computer networks before testing their security strength. In addition, it is easy to launch ransomware attacks by using Ransomware-as-a-Service. This paper proposed a new method that creates the ransomware specific features by using ransomware behaviors which are performed on file, registry, and network resources. The weights are assigned to the behaviors based upon where the actions are performed. The most feasible features are selected based on the assigned weights as well as Information Gain. The selected features are classified by using ML classifiers including J48 (C4.5), RF (Random Forest), AdaBoost (Adaptive Boosting), SLR (Simple Logistic Regression), KNN (K-Nearest Neighbors), BN (Bayesian Network), and SMO (Sequential Minimal Optimization). The experiments are performed on several ransomware variants as well as benign samples. The test results show that our proposed method is feasible and effective. The *DR*, *FPR*, *f*-measure, and accuracy are measured as 100%, 1.4%, 99.4%, 99.38%, respectively.

1. Introduction

Ransomware is a type of malware (malicious software) in the cyber security domain which is designed to prevent or limit access to a computer system until some amount of ransom is paid as a cryptocurrency. There are three ways that ransomware can affect a victim machine [1]: locker, crypto, and combination of locker-crypto ransomware. In locker ransomware, accessing a computer is blocked while in crypto ransomware, files are encrypted in the computer system. In combination with locker-crypto ransomware user access to computers is blocked as well as data being encrypted. Crypto ransomware is more destructive than locker ransomware because it is difficult to decrypt data without paying the money as a cryptocurrency.

Recently, ransomware attacks have increased in both frequency and severity. Covid-19 pandemic accelerates and enlarges the attack surface because of the remote working. Cyber criminals see the pandemic as a chance to increase the number of attacks against employees who work remotely. For instance, malicious emails were increased up to 600% because of Covid-19 [2]. In addition, 37% of organizations were affected by ransomware attacks in 2020 [3]. According to the Ransomware attack statistics report [4], in the first half of 2021 the number of ransomware attacks almost doubled when compared with the previous year 2020. The destructive effect of ransomware attacks is increasing. According to the National Security Institute, ransom fees are requested by about $5000 to $200.000 in the year between 2018 to 2020 [4]. Morgan estimated that in every 11 seconds, a ransomware attack occur in 2021 [5]. It can be seen from the above statistics that ransomware is becoming dangerous for individuals as well as organizations day by day.

To decrease the disruptive consequences of the ransomware, the detection and prevention system needs to be built. There are mainly two kinds of ransomware detection systems that can be deployed including static and dynamic analysis. Static analysis examines the source code of the executable without running the actual code. In static analysis, generally

*Corresponding author: omer.aslan.bisoft@gmail.com
Received: 20.12.2021, Accepted: 28.03.2022

signature is used to define [6] and separate ransomware from cleanware. The advantages of static analysis is that it detects the ransomware before running the actual code. The cons of static analysis is that it cannot detect new ransomware which is quite different from the existing ones. This is because intelligent ransomware is using various code obfuscation techniques to prevent being analyzed correctly. On the other hand, dynamic analysis examines the behaviors of ransomware while the code of the ransomware is being executed. With dynamic analysis, known ransomware as well as zero-day ransomware can be detected. Furthermore, it is resistant to code obfuscation techniques. However, some ransomware variants are not presenting their true behaviors when running under virtual machine and Sandbox environments. In this paper we proposed a method to eliminate the shortcomings of the dynamic analysis listed above and to better detect newly created ransomware. Besides, this research aims to decrease the disruptive consequences of the ransomware attacks by providing further investigation of victim machines.

In this study, a behavioral based ransomware detection method is proposed which uses data mining and machine learning (ML) techniques in the cyber security domain. Several ransomware variants are analyzed, and behaviors are collected by using dynamic analysis tools. While behaviors are created, one or a group of system calls are converted into higher-level operations which is called behavior. The behaviors are grouped to generate features. When behaviors are converted into features, system paths and activities that are performed in the system are taken into consideration. The behaviors are divided into three categories including file, registry, and network operations. Those operations are used to generate ransomware features. Then, most effective features are selected by using feature selection algorithms as well as known Information Gain algorithms. Finally, well known machine learning classifiers are used to separate ransomware from cleanware.

The rest of the paper is organized as follows. In section 2, literature review is discussed. In this section, various types of ransomware, and ransomware spread methods are examined. In addition, leading methods, which are designed to stop or detect ransomware in the literature, are discussed. In section 3, the proposed method is explained. In this section, data collection, feature creation and selection as well as classification techniques are presented. In section 4, results and discussion are presented. Finally, in section 5 conclusion and future research direction is given.

2. Literature Review

This section is divided into three main subsections. In the first subsection, background information about ransomware including ransomware types, propagation techniques, and evolving of ransomware over the years are explained. In the second subsection, the leading methods in the literature which detect or prevent ransomware attacks are given. Finally, the evaluation, pros and cons of each study has been discussed.

2.1. Evolving of Ransomware over the Years

The first ransomware example was the AIDS Trojan which was seen in 1989 [7]. At that time, it was not as dangerous as today's ransomware. Ransomware is written for revenue generation. There are 4 common revenue generation ways including fake antivirus scams, misleading applications, crypto, and locker ransomware. At first, misleading applications as well as fake antivirus tools appeared and got attention between 2005 to 2010. Timely, locker and crypto ransomware are created. Locker ransomware got popular between 2011 to 2012. Crypto ransomware got popular from 2013 up to these days. Between 2005 to 2021, ransomware attacks evolved from a malicious floppy disk which was demanding 189 dollars to a billion dollars' businesses with sophisticated tools over the years. These days, it is easy to launch ransomware attacks because new markets offer Ransomware-as-a-Service (RaSS). Recently, ransomware related attacks are targeting critical systems including finance, oil, gas, transportation and healthcare. It is also targeting IoT (Internet of Things) and mobile devices as well as cloud computing environments.

There are mainly two types of ransomware: Locker and crypto ransomware. Locker ransomware which can be defined as a computer locker denies requests to computers or other devices. On the other hand crypto ransomware, which can be defined as a data locker, prevents accessing files and data. Both locker and crypto ransomware prevents users from accessing something important unless paying requesting money as a cryptocurrency. Locker ransomware is only blocking access to the computer interface, it does not make changes on files and data on the computer system. However, crypto ransomware generally encrypts the important files and data on the computer system which makes the crypto ransomware more destructive.

There are four main stages of ransomware: Infect the system, locking the system or data, demand ransom, and release the files. In order to infect the

victim system, the ransomware needs to spread the target machine by spam email, phishing, and other techniques. After the victim system is infected, the payload of the ransomware is executed to generate public-private key pairs to encrypt the files. Then, a ransom message pops up which shows the amount of money needed to be paid. Finally, when the requested money has been paid, attackers send required keys to the victim to decrypt the files.

There are several techniques to spread ransomware from one system to another. Traffic distribution systems, social engineering techniques, spam email, downloader, and exploit kids are well-known ransomware spreading techniques. The well-known ransomware attacks, spreading methods, and consequences from 2013 to 2021 can be seen in table 1. Email attachment, exploiting software vulnerabilities, exploiting users' trust, and credentials theft have been seen as spreading methods over the years (Table 1). Consequences can be to encrypt all the files in the victim system and can affect many countries along the globe.

After the ransomware infected the victim system, the message appeared to demand money from the victim system. Examples of typical ransomware (WannaCry) messages can be seen in figure 1. It can be seen from figure 1, the ransom should be paid as a cryptocurrency in this example Bitcoin with a specific time period. After the demanded ransom has been paid, the attackers send required keys to decrypt the files.

Figure 1. Shows WannaCry ransomware message when victim files are encrypted

Table 1. List of well-known ransomware attacks over the years

Ransomware Attack	Year	Spread Method	Consequences
CryptoLocker	2013-2014	It spread by email attachments as well as by Gameover ZeuS botnet	It encrypted files on desktops as well as network shares and demand for ransom
TeslaCrypt	2015	It lured users to click phishing email	It encrypted the files and pop up a message for asking $500 ransom as a bitcoin to decrypt the files
WannaCry	2017	It exploited a Windows vulnerability	It affected 150 countries and encrypted computer hard drive
NotPetya	2017	It exploited a vulnerability CVE-2017-0144 on Windows Server Message Block protocol	It was one of the most destructive ransomware attacks in the history and affected many industries such as banks, power companies, and airports
LockerGoga	2019	Malicious emails, credentials theft, and phishing scams	It blocked the victims' accessing to the system and lost millions of dollars
CovidLock	2020	It exploited users' trust which claims to provide statistical data about COVID-19	It affected Android devices that encrypted data and denied the accessing data
REvil (Ransomware Evil)	2021	It exploited Microsoft exchange server vulnerability	The attackers demanded 50 million dollars and also leaked some data which included bank communications, balances, and images of financial spreadsheets

2.2. State-of-the-art Studies on Ransomware

There were only a few studies which specifically detect ransomware among malware or cleanware. Different studies used different methods to separate ransomware from cleanware. The methods that have been used in the literature were examined based upon the main idea, proposed method, and obtained performances.

Detection, prevention, and cure of ransomware attacks was presented by Brewer [8]. According to the author, targeted attacks were increased which were related to ransomware. Besides, some of the mass distribution of the attacks were automated which accelerate the infection process as well as demanding more ransom. In the first place, organizations and big companies need to get ready before attacks take place. For instance, companies need to eliminate the vulnerabilities before getting infected by ransomware and take regular system back-up in the safe place. Because most of the ransomware exploited system vulnerabilities to

propagate, and delete back-up files when running ransomware payloads. When companies are affected by ransomware, they are not only losing money, but also loss of business, possibly the permanent loss of important files, and suffer the effects of lost productivity.

Sgandurra *et al.* proposed an EldeRan which used dynamic analysis and machine learning techniques to classify ransomware [9]. The proposed approach monitored the activities that were performed by ransomware during the first installation. Paper stated that these sets of characteristic features were common across families and assisted the early detection of novel variants. After the feature generation process finished, the Mutual Information criterion was used to select most significant features. Then, Regularized Logistic Regression was used as a classifier. Experimental results presented that EldeRan performance based on area under the ROC curve measured as 0.995.

Nieuwenhuizen discussed a behavioral approach to detect ransomware [1]. As stated in the paper that static analysis which relies on signatures was not resistant to code obfuscation techniques. Thus, could not detect unknown ransomware. On the other hand, combining ransomware behavioral traits with machine learning algorithms increased the detection performances and also could detect zero-day ransomware. This is because core behavioral traits are not changing among the different variants. In other words, even though the code order of the ransomware changes, most of the behaviors remain the same [10].

Vinayakumar *et al.* evaluated the deep and shallow networks to distinguish ransomware from cleanware [11]. Cuckoo Sandbox was used to collect API calls and their frequency. Extracted APIs were given to the MLP (Multi-Layer Perceptron) as well as DNN (Deep Neural Network) to gather optimal feature sets. Then, machine learning classifiers were applied to selected features to detect and classify ransomware families. Test results showed that MLP gained highest accuracy with 1.0 and classified the ransomware families with accuracy of 0.98.

Crypto ransomware detection method, which was using http traffic, on a software defined network (SDN) was proposed by Cabaj *et al.* [12]. Authors assumed that http message sequences and their content sizes were good indicators of features when detecting new CryptoWall and Locky ransomware families. At first, ransomware network traffic was gathered to generate characteristic features from the outgoing http messages and its size. CryptoWall communicated with the command and control server by using domain names instead of direct IP addresses, and it also used HTTP POST messages. Besides, it directed traffic to hacked proxy servers and used the RC4 algorithm to encrypt the data. Locky ransomware communication patterns were similar to the CryptoWall family. Then, in the second step, for each ransomware family, feature vectors were prepared and the centroid vector. Finally, data obtained from two previous steps were used for detection in SDN based solutions. The proposed method was tested on CryptoWall and Locky ransomware families traffic. The test results indicated that performance is feasible with detection rates from 97% to 98% with 4 to 5% false positives when relaying on POST triples and domains.

Almashhadani *et al.* presented a behavioral analysis of network activities for crypto ransomware specifically on locky ransomware families [13]. Locky's PCAP execution traces of the MCFP (Malware Capture Facility Project) dataset were collected. Paper emphasized that locky has many network actions which might be used in order to extract behavioral features. From the TCP, HTTP, NBNS, and DNS traffic, 18 features were extracted. These properties are common in the locky ransomware family which can distinguish this ransomware from the benign ones. After features were generated, BN, RF, and LibSVM were used for classification. As stated in the paper, the proposed method could track the ransomware network activities, specified the valid extracted features, and achieved high detection accuracy as 97.08%, while decreasing the *FPR* (False Positive Rate).

Bae *et al.* presented machine learning-based ransomware detection [14]. The proposed approach first, extracted the API sequences by using *n*-gram techniques. API sequences were used to generate features, then features were represented as a vector. Finally, six machine learning classifiers including RF, Logistic Regression (LR), NB, Support Vector Machine (SVM), KNN, and Stochastic Gradient Descent (SGD) were performed for classification. The suggested method could separate different types of ransomware, benign files, as well as other malware variants. According to experiments, the presented method detected known and unknown ransomware among other malware types.

In our previous study, we have examined the detection of ransomware as well as other malware types such as virus, worm, Trojan horse, rootkit, etc. [15]. We found that building an effective and feasible approach to recognize all malware is a very difficult task, and more novel academic studies are needed to effectively detect ransomware as well as other

malware types. Trends in malware creation techniques are changing dramatically over time while the success of malware detectors' performances are decreasing timely. Hence, combining several methods and technology together may create a more feasible detector. For instance, combining the behavioral features with deep learning in the cloud environment can build more efficient detectors in ransomware recognition.

Beama *et al*. discussed the analysis of ransomware attacks based on the challenges, recent advances as well as future research directions [16]. The paper stated that static analysis is mostly evaded by code obfuscation techniques. In addition, based on some academic papers, certain dynamic analysis methods could be eluded by obfuscation techniques. Access control and data backups could be used as a prevention techniques to reduce ransomware destructive consequences, but these techniques suffer from various deficiencies as well. This is because access control and backups can increase the overhead significantly. Furthermore, current ransomware detection systems generate high false alarms while decreasing the detection rate. Authors claim that machine learning techniques can be used for ransomware detection more efficiently. ML-based models can learn to identify the general behavior patterns by classifying suspicious behaviors. Thus, it can detect unknown malware which have not been in the wild before.

2.3. Evaluation of State-of-the-art Studies

In the literature studies, various ransomware detection techniques were examined based upon the proposed methods, the main idea, and gained performances. Few static analysis versus several dynamic analysis studies were used to create ransomware features. A summary of each ransomware detection method is given in table 2. Most of the studies were performed merely on a few ransomware files which cannot be generalized for all ransomware variants. Besides, in most studies in the literature, the feature space was pretty big and the number of features increased when more program samples were analyzed which leads to requiring more computational times for the learning process. Thus, we conclude that current ransomware detectors are not good enough to recognize and classify the unknown ransomware variants. On the other hand, our proposed method performed on different ransomware variants with high performance, decreased feature space drastically, while decreasing the computation time for learning and detection processes. In addition, our proposed method increases the DR and accuracy for known and unknown ransomware strains.

3. Materials and Methods

This section presents materials and proposed methods. We have changed our previous proposed method to detect ransomware files [10]. Figure 2 shows the extended version of our previous proposed method architecture [10]. The section is split into five parts including proposed method, data collection, feature creation and selection, detection, and performance evaluation. The ransomware samples were downloaded from online websites and analyzed under dynamic analysis tools. The execution traces of ransomware activities were obtained. Then, ransomware behaviors and features were generated from the ransomware activities. After that the most significant features were selected by using features' weights as well as Information Gain. Finally, we used well known machine learning classifiers to separate ransomware from benign samples.

3.1. Proposed Method

After the execution traces of ransomware collected, the ransomware behaviors and features were created. While creating ransomware behaviors, one or a group of system calls are converted into higher-level operations which is called behavior. To illustrate, if the order of the activities are CreateFile, WriteFile, and CloseFile; the associated ransomware behavior will be WriteFile. After ransomware behaviors are created, we perform a proposed algorithm to generate features. When behaviors are converted into features, system paths and activities that are performed in the system are taken into consideration. The behaviors are divided into three categories: file operations, registry operations, network operations. Those operations are used to generate ransomware features. When ransomware features are created, the most significant features are counted. As it can be seen in figure 3, we divided general program behaviors into 3 categories: M (Malware behaviors), B (Benign behaviors), and R (Ransomware behaviors)

$$M = x + a + b + z \quad (1)$$

$$B = \mathrm{b} + \mathrm{z} + \mathrm{y} \quad (2)$$

$$R = a + b \quad (3)$$

Table 2. Current ransomware detection methods which are represented in the literature

Paper	Year	Proposed Method	Goal/Success
Sgandurra *et al.* [9]	2016	Dynamic analysis and machine learning techniques to classify ransomware	Area under the ROC curve measured as 0.995
Vinayakumar *et al.* [11]	2017	Extracted API calls are evaluated by the MLP as well as DNN to select best feature set	Classified the ransomware families with accuracy of 0.98
Cabaj *et al.* [12]	2018	Http message sequences and their content sizes are used as features	Highest detection rate measured as 98% while FPR is measured as 5%
Almashhadani [13]	2019	Locky's PCAP execution traces of the MCFP dataset were collected	Achieved high detection accuracy as 97.08%, while decreasing the FPR
Bae *et al.* [14]	2020	API sequences by using n-gram techniques	Could separate different types of ransomware, benign files, as well as other malware variants efficiently
Proposed method	**2021**	**Ransomware specific behavioral patterns combining with ML techniques**	**Could effectively detect known and unknown ransomware with high accuracy**

We mostly consider the behaviors (*a*) that are only seen in ransomware samples. Plus, behaviors (*b*) that are seen mainly in ransomware with high frequency, while rarely seen in benign samples with less frequency. Behaviors *a* and *b* are determined by using weights assigned to where the actions are performed (Figure 2). Thus, when features are created from the behaviors, the behaviors which have smaller weights than threshold and the features that have smaller frequency are eliminated from the dataset. That way, ransomware specific behaviors are generated before classification takes place.

a) File related features: One of the most common ways for ransomware to interact with the system is through file operations. The ransomware tries to protect its existence in the system by creating files or making changes to the existing files. To create the features from the file related behaviors, the behavior itself (read, write, execute, etc.) , the location of the behaviors that are performed, and extension of the files are taken into consideration. For instance, ransomware performs more read and write behaviors, because it needs to read every file and encrypt those files or first copy the original files into different locations then encrypt the copied files. Thus, more weights are assigned to those behaviors when generating the features. Ransomware also executes behaviors generally in automatic startup file locations as well as temp locations. For those behaviors, more weights are assigned during the feature generation. The behaviors which are performed on different files are also considered, and assigned more weights. For example, sometimes ransomware injects itself into system processes such as svchost. exe, explorer. exe, etc. and inject itself into most used DLLs.

b) Registry related features: The registry is a database that hierarchically holds the operating system and application settings such as drive, startup, network, user account information, etc. Ransomware usually uses the registry to start automatically in the system. In other worlds, the ransomware can automatically run itself in the background every time the system is started. We assigned more weights on the behaviors related to registry autostart locations, access to registry keys and make changes on those keys, and some system specific registry locations.

c) Network related features: Network related behaviors are source and destination IP addresses, average packet size, port numbers that are used, the number of packets that are exchanged between the machines, etc. Ransomware mostly shows anomalous behavioral patterns. For example, generally ransomware uses the order of http packets to send required keys to the victim machine from the command and control center. Size of the packets are taken into consideration. Also, ransomware performs network operations to be able to spread in the network environment and gain unauthorized access to other systems, and there are many Windows API methods that can be used for this purpose. More weights are assigned to those network behaviors when generating network related features.

3.2. Data Collection

The ransomware samples are collected from different sources: Malshare, theZoo aka Malware DB, Tekdefense, and VirusShare [17-20]. To ensure the ransomware variants, Virustotal is used to label the samples. For the experiment, 346 ransomware samples and 304 benign are analyzed under Process Monitor to capture the activities that ransomware displayed. Instead of Process Monitor we could also use other dynamic analysis tools such as Capture BAT, Cuckoo Sandbox, API Monitor, Regshot, and Wireshark. However, we selected Process Monitor because it is an enhanced monitoring tool which shows real-time file, registry and network activities.

When ransomware make changes on important files and registry entries, Process Monitor shows these changes. Besides, filtering can be performed easily according to many categories. The dataset consists of various ransomware variants including locky, Jigsaw, ransomlock, cryptolocker, petya, Wannacry, and CTB-Locker. The benign samples are also collected from different categories: system tools, office documents, games, multimedia, and other programs.

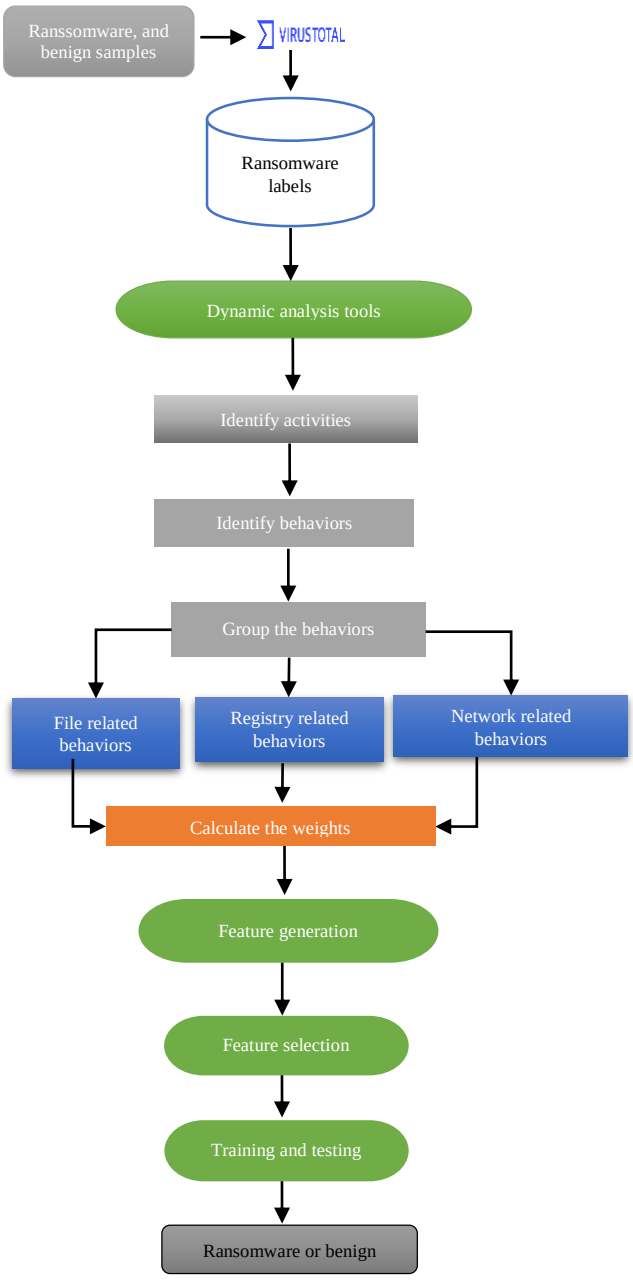

Figure 2. Proposed ransomware detection architecture

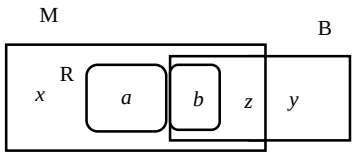

Figure 3. Shows the malware, benign and ransomware behaviors

3.3. Feature Creation and Selection

The Process Monitor is used to monitor ransomware behaviors. Collected ransomware and benign files are analyzed in virtual machines Windows 7, 8, and 10 (Figure 4). For each ransomware, the clean version of the virtual machine is used. During the feature generation, the implementation is carried out by using Python scripting language. When creating a property, following stages are followed:

Stage 1: Converting activities into behaviors

Stage 2: Separate the behaviors as file, registry, and network

Stage 3: Calculate the weights for each behavior where the action is performed and which action is performed

Stage 4: Group the behaviors based on the different system resources

Stage 4: Group the behaviors on different instances of the same resources

Stage 5: Extract the features from 1 to 5 consecutive order behaviors

Stage 7: Computing the weights for each feature

Stage 8: Calculate the frequency of each feature

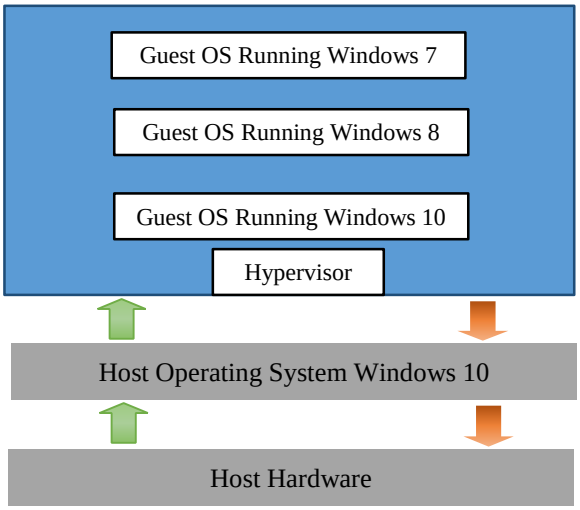

Figure 4. Ransomware analysis environment

After the features, their weights, and frequencies are obtained, the feature vector is created for each ransomware as well as benign samples. In other words, each analyzed file is represented as a feature vector. If the feature is repeated *x* times, *x* is written for frequency value. If the feature is repeated *0*, *0* is written for frequency value. For the feature selection phase, first more significant features are chosen based on the features' weights. During the weights assignment, the feature itself and the locations of features where performed is considered. In this process, the features are divided into 3 groups: file, registry and network related. Then each group is divided into subgroups. For example, if a feature about the file shows operations for the directory where the file is located, the weight will be low, if the feature creates another file and copies its own data to this file, it will be heavily weighted. Whether the feature is active or passive also affects the weight vector to be assigned. For example, if the feature is "read", the weight given is low, if it is "write", it is high. If the program frequently uses Ntdll.dll instead of Wininet.dll and kernel32.dll, or if it includes features that use high-risk methods such as NtReadProcessMemory and NtAdjustTokenPrivileges, the weight value to be assigned is high as well. After the weight assignments process is finished for each feature, the Information Gain algorithm is used to decrease the number of features further. In Information Gain, the property which has the maximum gain is chosen repeatedly when selecting the most significant features. The Information Gain can be calculated as the following for a given dataset.

Information $Gain\ (A) = Information(D) - Information_A(D)$ (4)

$Information(D) = -\sum_{j=1}^{v} pi\ log_2\ (pi)$ (5)

$Information_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} log_2\ (\frac{|D_j|}{|D|})$ (6)

$Gain(A)$ represents how much information will be obtained when splitting using the property *A*.

3.4. Detection

Although machine learning algorithms are used extensively in various areas for years, they are not used sufficiently in ransomware detection. Hence, several ML classifiers including J48, RF, AdaBoost, SLR, KNN, BN, and SMO are used in this study. We cannot say one classifier is more effective than the others since each classifier can perform better than others based on the numbers of features used, distributions of data, and association among properties. After the feature selection process is finished, the classification is performed. In this phase, selected features are given into the C4.5, RF, AdaBoost, SLR, KNN, BN, and SMO classifiers as an input and ransomware or benign is generated as an output.

3.5. Performance Evaluation

To assess the proposed method performance, detection rate (*DR*), false positive rate (*FPR*), *f*-measure, and accuracy are used. These measures are calculated by using the confusion matrix (Table 3). These measures are presented by the *TP* (The number of ransomware is marked as ransomware), *TN* (The number of benign is marked as benign), *FP* (The number of benign is mistakenly marked as ransomware), and *FN* (The number of ransomware accidentally marked as benign). By using these values, *DR*, *FPR*, f-measure, and accuracy are calculated as the following:

DR = Recall= TP/ (TP+FN) (7)

FPR = FP/ (FP+TN) (8)

Precision = TP/ (TP + FP) (9)

*F-Measure = (2 * precision * recall)/ (precision + recall)* (10)

Accuracy = TP+TN/ (TP+TN+FP+FN) (11)

Table 3. Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

When training and testing is applied, holdout (75% and 25% split) as well as 10-fold cross-validation procedures are performed. At first, when a few ransomware samples were used, the performance of the holdout was less than cross-validation. However, when more ransomware samples were analyzed, the holdout performances increased.

4. Results and Discussion

This section of the paper presents the test results and interprets the proposed method performances. The results are summarized in table 4, table 5, and figure 5.

Table 4 shows the performance of ML classifiers on created ransomware dataset. Various ML classifiers as well as metrics are used to evaluate the performance. As it can be seen from table 4. except SMO, other classifiers' performances are quite high. To illustrate, J48 *DR* measured as 100%, *FPR* measured as 1.4%, *f*-measure calculated as 99.4%, and accuracy measured as 99.38%. Similar results are obtained from the classifiers RF, AdaBoost, SLR. The performances of the KNN and BN classifiers are satisfactory, too. These performance results show that our proposed method, which used to create the ransomware features, is effective and feasible to separate ransom specific features from the benign samples.

Table 4. Classifiers performances on the created ransomware

Classifier	DR (%)	FPR (%)	F-Measure (%)	Accuracy (%)
J48	100	1.4	99.4	99.38
RF	99.1	0.7	99.3	99.23
AdaBoost	98.8	0.7	99.1	99.07
SLR	97.8	1.4	98.3	98.14
KNN	91	7.9	92	91.53
BN	91.3	11.2	90.8	90.15
SMO	78.3	6.3	85.2	85.53

Figure 5 presents the accuracy results on different ML classifiers before feature selection and after feature selection. When ransomware and benign features are created by using the proposed method, the most significant features are selected by using proposed weights measures. Feature selection is also applied by using Information Gain before ML classifiers are performed. As it can be seen from the figure 5, before the feature selection the second time, the performances were quite good, but selecting features by Information Gain one more time increased the model performances. This shows that a few most significant features can lead to the conclusion of more than several hundreds of features.

The list of file-registry related features which are mostly seen in ransomware samples rather than benign in our testbed can be seen in table 5. It is observed that ransomware performs more operations on the created files and reads files' contents from one file into another. In addition, it copies itself into automatic startup locations on the registry, so it performs many registry reading and changing operations. Even though some of the listed features can be seen rarely in few benign files, the frequency of the features are quite low when compared to the frequency of features counted in ransomware. ReadFile, CreateFileReadFile, ReadFileWriteFile, WriteFileCreateFileMapping,SetBasicInformationFile, RegOpenKey, RegQueryValue, RegSetInfoKey, RegSetValue, and RegDeleteValue features are seen in ransomware samples with high frequencies in important file directories and registry locations.

Most of the current ransomware detection methods are performed only on a few ransomware samples which cannot be generalized for all ransomware families and strains. Besides, they have difficulties to detect unknown ransomware variants, and are not resistant to code obfuscation techniques. On the other hand, the proposed method could effectively perform on ransomware and benign samples. The proposed method tested on different types of ransomware, and increased the detection and accuracy rate for known and newly created ransomware. Furthermore, the proposed method decreased the number of features which can lead to separate ransomware from the cleanware as well as it is resistant to code obfuscation techniques.

Table 5. The list of file-registry related features that frequently seen in ransomware rarely seen in benign files

List of Features
ProcessStart
ThreadCreate
LoadImageReadFile
ReadFile
CreateFileReadFile
RegOpenKey
RegQueryValue
RegSetInfoKey
RegEnumValue
RegCreateKey
RegSetValue
RegDeleteValue
ReadFileWriteFile
WriteFileCreateFileMapping
SetBasicInformationFile
RegSetInfoKeyRegQueryKey
RegSetInfoKeyRegOpenKey
RegSetInfoKeyRegEnumValue
RegOpenKeyRegEnumValue
WriteFileSetBasicInformationFile
CreateFileSetBasicInformationFile
ReadFileQueryBasicInformationFile

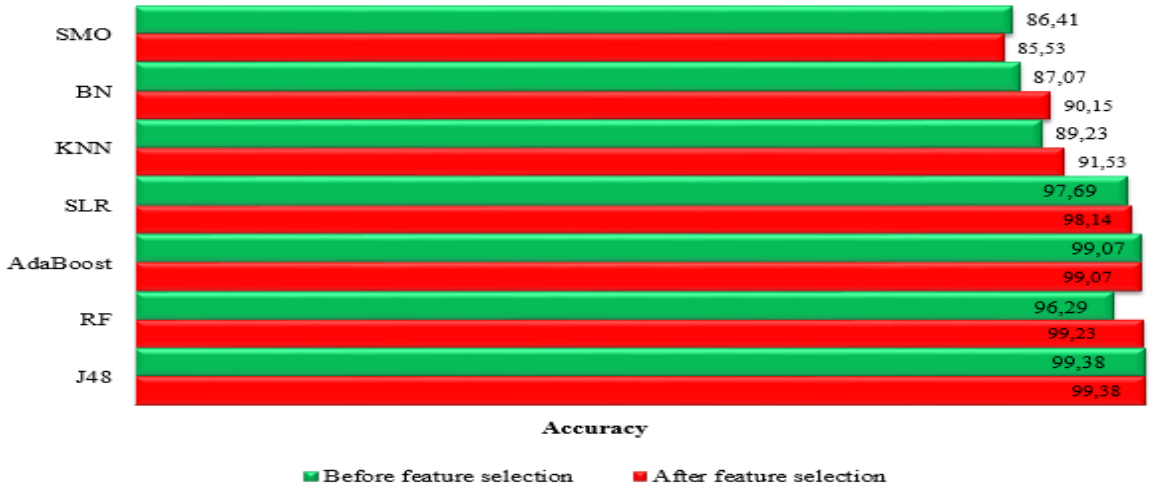

Figure 5. Proposed method performances before feature selection and after feature selection

Although the proposed method can effectively detect the several ransomware variants, there are still some limitations that need to be mentioned. The proposed method is tested on a few hundred ransomware and benign samples. The number of analyzed samples needs to be increased. Some behaviors are similar in ransomware and benign samples. Those behaviors are increased when more samples are analyzed. We need to modify our feature creation algorithm and update the threshold values for weight assignments to decrease those behaviors and associated features. In this study, the proposed method is tested on only Windows operating systems, the proposed algorithms can be modified and used for different operating systems.

5. Conclusion and Future Research Direction

Ransomware can be seen as one of the most destructive malware among the other cyber attacks. The ransomware attacks can deny access to the victim data by locker and crypto ransomware. While in crypto ransomware, files are encrypted in the computer system, in locker ransomware, accessing the computer is blocked. This paper proposed a behavioral based ransomware detection method which uses data mining and machine learning (ML) techniques. Several ransomware variants are analyzed, and behaviors are collected by using dynamic analysis tools. The behaviors are grouped into three categories including files, registries, and networks. Also, the behaviors' locations are considered when creating a feature's weights. The most effective features are selected by using our feature selection algorithm by using assigned weights as well as known Information Gain algorithm. The experiment's test results confirm that our proposed method is effective and feasible by measuring the various metrics including *DR*, *FPR*, *f*-measure, and accuracy. The best performance is measured as 100% for *DR*, 1.4% for *FPR*, 99.4% for *f*-measure, and 99.38% accuracy on the same classifier. Similar results are obtained for other used ML classifiers. As a future work, we aim to analyze more ransomware samples as well as classify the ransomware samples based on the different ransomware families. Furthermore, more dynamic analysis tools such as Wireshark, API Monitor, Regshot, and Sandboxes will be used for future studies.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics

References

[1] D. Nieuwenhuizen, "A behavioural-based approach to ransomware detection," *MWR Labs Whitepaper*, 2017.
[2] Associated Press, "The Latest: UN warns cybercrime on rise during pandemic," 2020.
[3] Sophos Report, "The State of Ransomware 2021," 2021.

[4] Cognyte CTI Research Group, "Ransomware Attack Statistics 2021 – Growth & Analysis," 2021.
[5] S. Morgan, "Global Ransomware Damage Costs Predicted To Reach $20 Billion (USD) By 2021," *Cybercrime Magazine*, 2019.
[6] Ö. Aslan and R. Samet, "Investigation of possibilities to detect malware using existing Tools," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1277-1284), 2017.
[7] J. P. Tailor and A.D. Patel, "A comprehensive survey: ransomware attacks prevention, monitoring and damage control," *Int. J. Res. Sci. Innov*, vol. 15, pp. 116-121, 2017.
[8] R. Brewer, "Ransomware attacks: detection, prevention and cure," *Network Security*, vol. 9, no. 5-9, 2016.
[9] D. Sgandurra, L. Muñoz-González, R. Mohsen and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv preprint arXiv:1609.03020*, 2016.
[10] Ö. Aslan, R. Samet and Ö. Ö. Tanrıöver, "Using a Subtractive Center Behavioral Model to Detect Malware," *Security and Communication Networks, 2020*.
[11] R. Vinayakumar, K.P. Soman, K.S. Velan and S. Ganorkar, "Evaluating shallow and deep networks for ransomware detection and classification," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 259-265, 2017.
[12] K. Cabaj, M. Gregorczyk and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Computers and Electrical Engineering*, vol. 66, pp. 353-368, 2018.
[13] A. O. Almashhadani, M. Kaiiali, S. Sezer and P. O'Kane, "A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware," *IEEE Access*, vol. 7, pp. 47053-47067, 2019.
[14] S. I. Bae, G. B. Lee and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, e5422, 2020.
[15] Ö. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access,*vol. 8, pp. 6249-6271, 2020.
[16] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Computers and Security*, vol. 111, pp. 102490, 2021.
[17] Malware downloading website, https://malshare.com/, accessible in 2021.
[18] Malware downloading website, https://thezoo.morirt.com/, accessible in 2021.
[19] Malware downloading website, http://www.tekdefense.com/, accessible in 2021.
[20] Malware downloading website, https://virusshare.com/, accessible in 2021.