IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

Identifying Ransomware-Specific Properties using Static Analysis of Executables

Deepti Vidyarthi[1], CRS Kumar[2], Subrata Rakshit[3], Shailesh Chansarkar[4]

Department of Computer Science & Engineering, Defence Institute of Advanced Technology, Pune, India[1, 2]

Center of Artificial Intelligence & Robotics, Bangalore, Karnataka, India[3, 4]

Abstract: Ransomware attacks have risen exponentially over the past decade with increasing severity, potency to cause damage, and ease of carrying out attack. The conventional anti-malware techniques are compelled to include advanced ransomware detection mechanisms. This paper presents the results of the study and analysis of ransomware executable files in order to identify the characteristic properties that distinguish ransomware from other malware and benign executable files. The program binaries are analyzed statically and dynamically to observe the typical behaviour and structure of the ransomware. Using the dynamic and static analysis technique, ransomware-specific properties are extracted from the executable files. The experiments show that higher accuracy of classification, using machine learning algorithms, is achieved by combining these properties with the set of generic malware properties for malware detection.

Keywords: Ransomware, Malware Detection, Static Analysis, Dynamic Analysis

I. INTRODUCTION

Ransomware is being widely used to target the computers recently, and new types are being constantly added to the family. These attacks are observed growing over the last decade, however, for the last 3-4 years, they have been increasing at an alarming rate. As predicted by Barkly Endpoint Security, ransomware continued to experience record growth in 2017-18. A new target organization is facing a ransomware attack at every 40 seconds [1]. According to Sophos labs report 2018, ransomware is being used as a service i. e. Ransomware-as-a-service (Raas) [2]. Recently, Android devices are likely to be target by ransomware authors [3], which is an indication that the threat is going to increase further in the coming years. The ransomware reaches the target computer using social engineering techniques and activates after the victim downloads the email attachments or clicks on insecure links from the email. As it spreads over the network easily, the number of victims exposed to this threat is very high. The ransomware can be classified into two types, 'Crypto ransomware' that can encrypt the whole data on the victim's machine, making it inaccessible to the victim and 'Locker ransomware' that can lock the whole system so that the victim is unable to use it [4]. Both the types of ransomware ask for the ransom demand for releasing the assets of the victim. The attackers prefer Bitcoin as a means for payment, as it is an anonymous payment mechanism and conceals their identity and location. These peculiarities encourage the attackers to choose ransomware in their malicious attacks out of the other malware.

Existing signature-based malware detection and generic malware analysis do not focus on the specific characteristics of ransomware. Detection and prevention techniques for ransomware have been proposed through the literature, which explore the ransomware working in detail. However it is required to evolve an advanced ransomware detection approach to complement the contemporary anti-malware techniques. For any detection mechanism, the core component is to identify the characteristics that would serve as a unique reference. This paper focuses on the identification of characteristics of the Portable Executable (PE) file that can be used effectively in ransomware detection. Such distinguishing characteristics termed as the ransomware-specific properties are proposed to improve the generic malware classification based on the combination of ransomware-specific and other generic malware properties.

II. RELATED WORK

Malware analysis is the study of malware by dissecting its different components and studying its behavior and effects on the computer system. Multiple malware analysis techniques have been proposed for malware detection and classification in general by various researchers over the years. Broadly, malware analysis methods can be classified into static analysis and dynamic analysis. The detection techniques can be characterized in the terms of executable's features used for detection and their classification approach. In our previous work [5], Random Forest, Decision Trees (J-48), Naive Bayes and Support Vector Machine (SVM) classifiers were trained towards malware classification based on the features extracted through static and dynamic analysis of the malware. In this paper, we focus on the analysis of ransomware. The approach we take here is based on the anticipation that the static and dynamic analysis of ransomware

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

specifically would result in features unique to ransomware. It proposed to study the effect of these newly extracted features being appended to the generic features used for malware classification.

A. Ransomware Analysis

The majority of literature and work related to ransomware focuses on the encryption and deletion mechanisms and communication techniques. Analysis of the file system activities of ransomware samples suggested that it is possible to detect ransomware by looking at I/O request and Master File table in NFTS file system [6]. Another approach for detection of ransomware using C & C server DNS logs is discussed [7]. Here, the ransomware uses DGA algorithm to generate random fake domain name by detecting the encryption key while communication with command and control server, so that the detection of valid or invalid domain name is difficult. Cryptolocker, WannaCry, TeslaCrypt use DGA for generating domain name. Kramer and Bradfield [8] suggest a continuous monitoring approach for ransomware detection that includes - maintaining the ransomware signature and Indicators of Compromise (IOC), looking for file execution from %APPDATA% folder and %TEMP% folder, monitoring back-up files, checking file extensions, observing the anomalous network behaviour during key exchange and looking at I/O requests and Master File Table (MFT) in NTFS file Detection. Brewer [9] proposes an automated approach to track the changes to the system's desktop that indicate ransomware-like behavior. The above mentioned work is observed to be focused on analyzing the typical working mechanism of ransomware. K. P. Subedi et. al. [10], performed static and dynamic analysis and concentrated on developing signatures by reverse engineering. Their signature database is specific to crypto ransomware. A study proposed by Zimba et.al. [11] discusses the attack model of ransomware and its techniques are extracted through the static analysis. The study of ransomware variants, BitPaymer and KeyPass, details the behaviour of these ransomware and identifies the presence through code analysis [12]. Zavarsky et.al. [13] present the experimental analysis of ransomware on windows and android platform and recognize the main behavioral properties.
The ransomware behavior discussed through the literature is studied and the properties of ransomware executable which are responsible for the typical behavior are observed by static and run-time analysis of ransomware. Such ransomware-specific properties are identified in following sections and proposed to improve the general malware classification by training the classifier based on the combination of ransomware-specific and other generic malware properties.

III. ANALYSIS FOR RANSOMWARE IDENTIFICATION

The approach proposed is to identify the typical characteristics of ransomware through dynamic and static analysis of the ransomware PE (portable executable) file, which is a standard format for Windows executables.

A. Identification of properties through dynamic analysis

Dynamic analysis techniques are used to observe the run-time behavior of the executable file and its effect on the system [17]. The flow of monitoring the ransomware execution is as given in Fig. 1.

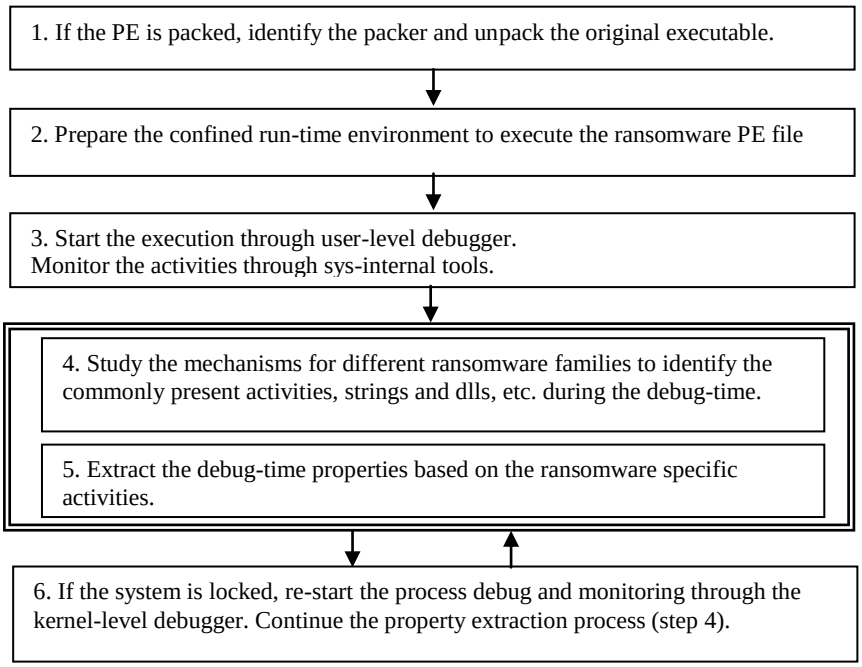


Fig. 1. Dynamic analysis workflow for property extraction

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

It is proposed to monitor the run-time behavior using user-level and kernel-level debugging. While execution, the system may get locked (in case of locker ransomware) or the execution may proceed (if the executable has sensed debugger and used evasive technique). In the case, when system is locked, the monitoring of process on the same system is not possible. In such case, the kernel-level debugging using another system machine is required.

Ransomware of different types i.e. Crypto-ransomware, Locker-ransomware, a combination of crypto-locker, etc. are executed to observe their behavior in form of suspicious commands, registry changes, suspicious strings, file encryption, etc. Fig. 2 shows execution of Locker-ransomware monitored using Microsoft sysinternals tool for process monitor. It is observed that the Lockyransomware is running by changing its name to rundll32.exe. Initially, user-level debugging is done where the process level break-points are set to analyze the code and different modules of the code independently. Fig. 3 shows the debugging process using Ollydbg with breakpoints at function call instructions. In the next step, or in the cases, where the system is locked immediately, kernel-level debugger is used for debugging to detect various 'crypt' engines used by the ransomware. Fig. 4 shows the dump from Windbg, during kernel-level debugging, the 'CRPTBASE' engine is used by ransomware.

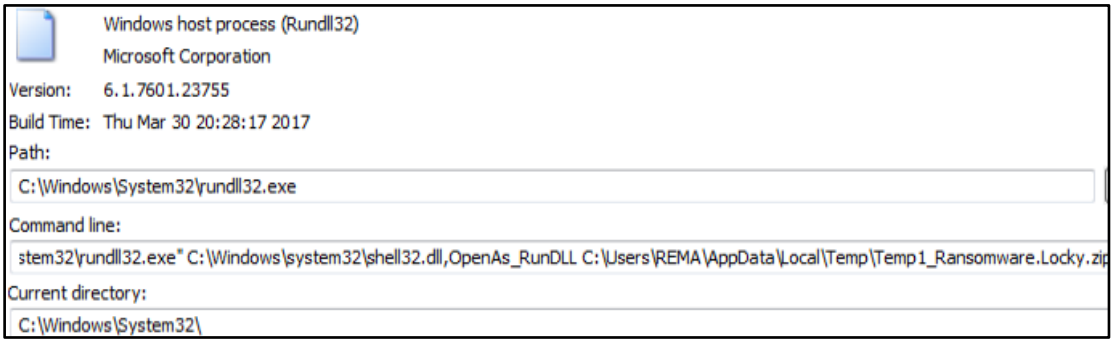


Fig. 2. LockyRansomware running as rundll32.exe

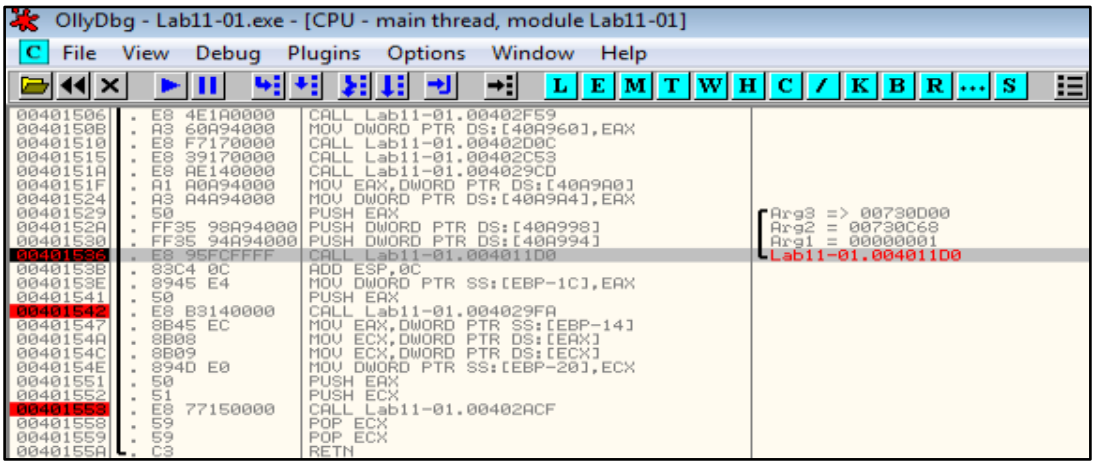


Fig. 3.Ollydbg debugging

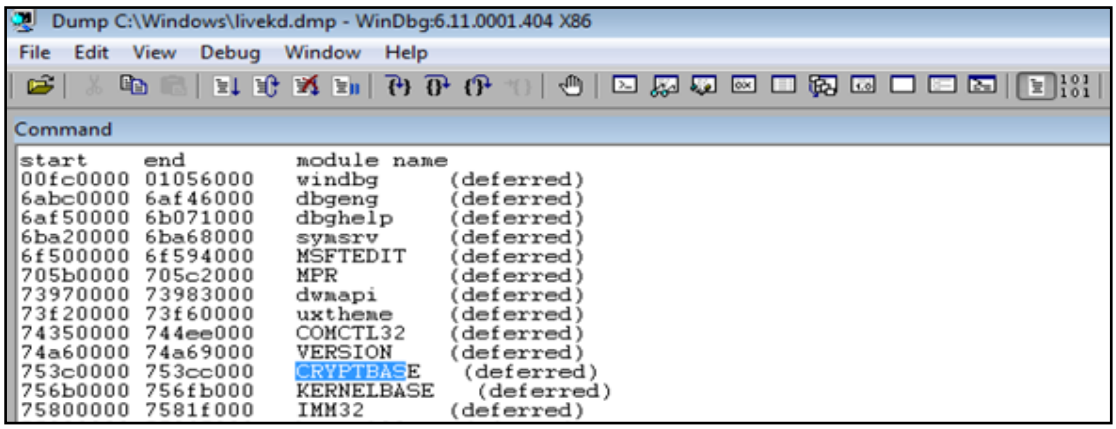


Fig. 4. Windbg debug-time analysis

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

Dynamic analysis is useful to extract the behavior ransomware for detection as given above. However, in cases of locker type ransomware as well as evasive malware, the true behaviour of process cannot be monitored. Thus dynamic analysis is combined with the static analysis for increasing the accuracy of detection.

B. Identification of properties through static analysis

Static analysis consists of examining the executable file without viewing the actual instructions. The PE file is dumped and disassembled to learn the structure and components of the file. The method of extracting the generic and ransomware-specific properties from a PE through static analysis is as shown in Fig. 5.

Step 1: The PE (binary executable file format for Windows OS) is read without execution for extracting its static properties. The executable may be packed by malware writer to making reversing difficult. It is unpacked to read the original file headers and sections.

Step 2: The malware executable will have distinctive values for various header fields. For example, the number of sections in a malicious executable is not as in a regular PE file. We extract such properties of PE through the analysis of three types of headers - DOS header, file header and optional headers, which contain metadata at different levels. Totally 60 generic properties are identified for static malware classification. The major properties for general malware detection are given in Table 1.

Step 3: The ransomware mechanism is studied to learn specific characteristics of ransomware executable. The general operational architecture of ransomware is as follows. It mainly consists of a 'dropper' that contains the encrypter [14]. The encrypter component contains a decryption engine and a password protected compressed component (zip), containing a copy of Tor [15] and several individual files with other information and the encryption key.

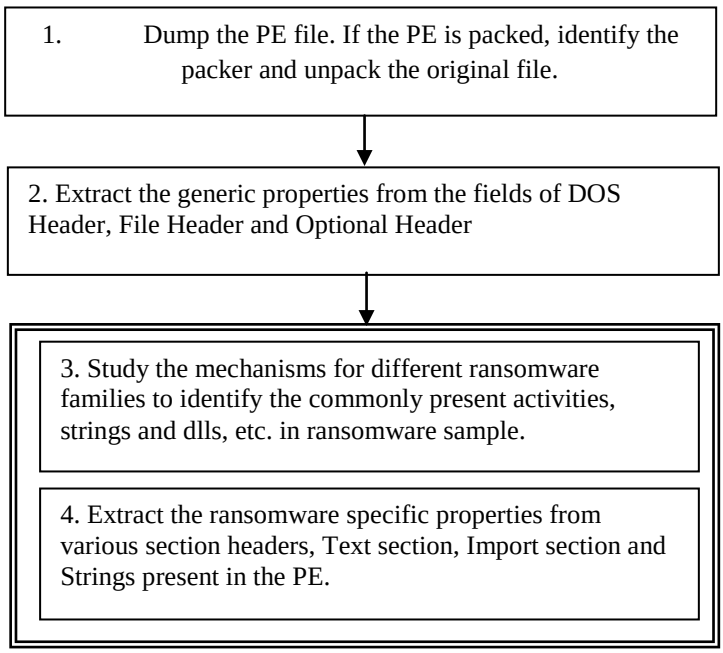


Fig. 5. Static analysis workflow for property extraction

First, the dropper tries to connect to a remote website say "xyz.etc" and exit after connection. If the connection fails in the first step, it then creates a service for example "mssecsvc2" with another name [15]. After creating a service the dropper extracts the encrypter binary from its resource and then executes it. The encrypter also checks for the presence of the component called mutex. The encrypter creates the "mutex" if not present, before execution. In another approach, the ransomware shares its encryption key using command and control server (C & C Server) [7]. In the case where the C & C server is not present, it uses hard-coded keys and uses the same key for all encryption [7]. The general mechanism can be described in the following four stages:

Infection: The first step is execution of the malicious ransomware file on a computer. It can be through a phishing email or an exploit kit. E.g. Angler exploit kit, (CryptoLocker) exploits the bugs in Adobe Flash and IE.

Destroying Back-up: It is a unique ransomware feature. The back-up files and folders on the system are targeted and removed to prevent system recovery on its own. E.g. vssadmintool to remove the volume shadow copies from the system.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

Table I Properties Extracted From PE Headers

Header	Field	Description
DOS header properties	e_cblp	Bytes on last page of file
	e_crlc	Pages
	e_cparhdr	Relocations
	e_lfanew	4-byte offset into the file where the PE file header is located
File header properties	Number Of Sections	Number of section headers and section bodies in the file
	Number Of Symbols	Symbols in the header
	Size Of Optional Header	Size of optional header (optional header contains initial stack size, program entry point location, preferred base address, operating system version, etc.)
	Characteristic	Specific characteristics about the file
Optional header properties	Size Of Code	Size of code section, in bytes, or sum of all such sections if multiple code sections.
	Size Of Initialized Data	Size of the initialized data section, in bytes, or the sum of all such sections if multiple initialized data sections
	Size Of Uninitialized Data	Size of the uninitialized data section, in bytes, or the sum of all such sections if multiple uninitialized data sections.
	Address Of Entry Point	Location of the entry point for the application

Encryption: Strong encryption algorithms like AES 256 are used for encrypting the targeted files/directories, etc. Secure key exchange is performed with the C & C server. The scope of encryption is different for different variants. E.g. CryptoWallv3 doesn't encrypt filename, CryptoWallv4 randomizes filename.
Ransom Demand: At the final stage, instructions for extortion and payment are saved onto the hard drive. The victim is given a few days to pay and after that time the ransom increases. Malware removes its traces from the victimized system.
Based on the mechanism, we analyzed the ransomware executables to identify the main characteristics such as packed code, network connections specific DLLs, mutex, and encryption related strings etc. The summary of analysis carried out is given in Table 2.

Table II PE Components Analyses for Ransomware-Specific Properties

Structural Analysis	Section properties
	Structural entropy
	Packed section detection
Import Analysis	Imported DLLs, API function names
String Analysis	Domain names
	Mutex detection
	Crypt specific strings

As depicted in Table 2, the structural analysis is carried out at various sections of the executable image to get gives information about packer's identity, entropy of file, entropy of sections like data and text section, etc. The import section of an executable file contains names of the imported DLLs and API functions. Import directory of ransomware PE is analyzed to note the particular DLLs used by ransomware that are uncommon in other software. As the commands, dialogues, function names etc are present in executable as string. The strings are analyzed to identify suspicious commands and function used by ransomware to perform various malicious activities. The identified

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

ransomware-specific 9 properties are given in Table 3. These ransomware-specific properties are validated through the classification of ransomware malware and comparing the performance with classification based on general malware properties. The classification is done using three algorithms, Random Forest, Decision Trees (J-48), and Naive Bayes. These algorithms are selected based upon their suitability and better performance for general malware classification.

Table III Ransomware-Specific Properties

Identified Property	Description
Ispacker	Presence of packer
Packer type	Identity of type of packer
Open mutex Create mutex	Checking and creating mutex for isolation
Entropy (e_file, e_text, e_data)	Entropy of file, text and data sections
Strings	Presence of common strings extracted through ransomware string analysis; E.g. crypt/decrypt/%amount%)
Ws2_32.dll	DLL used for network connections and communication
Get_news	Command used to modify the registry
Add_entry	Store information from the client
Get_add	Get access to the file from the given path

Static analysis is useful to extract the characteristics of PE file for ransomware detection as given above. However, in cases where complex obfuscation and packing techniques are used, simple static analysis may not be able to extract all the required features [16]. It is required to apply dynamic analysis technique as well to detect malicious functionalities of ransomware [17].
The static and debug-time analysis of ransomware has resulted into ransomware-specific PE file properties like high entropy, suspicious strings, typical DLL/API functions, and code constructs, etc. Along with the specific to ransomware properties, other distinguishing features of the general malware are present in ransomware as well. Hence, such extracted properties separately and combined with specific properties are used to prepare the data set for classification.

IV. RESULT AND ANALYSIS

The ransomware detection is based upon the behaviour and structure of executables extracted using the dynamic and static analysis techniques as discussed above.

A. Dynamic Behavior for Ransomware Identification*:* The dynamic analysis is performed over different types of ransomware. The results of the analysis for Crypto-Locker, Locky, WannaCry are discussed in following section. Crypto-locker locked the system during the run-time analysis, Fig. 6 shows the names of the DLLs and functions specific to ransomware.

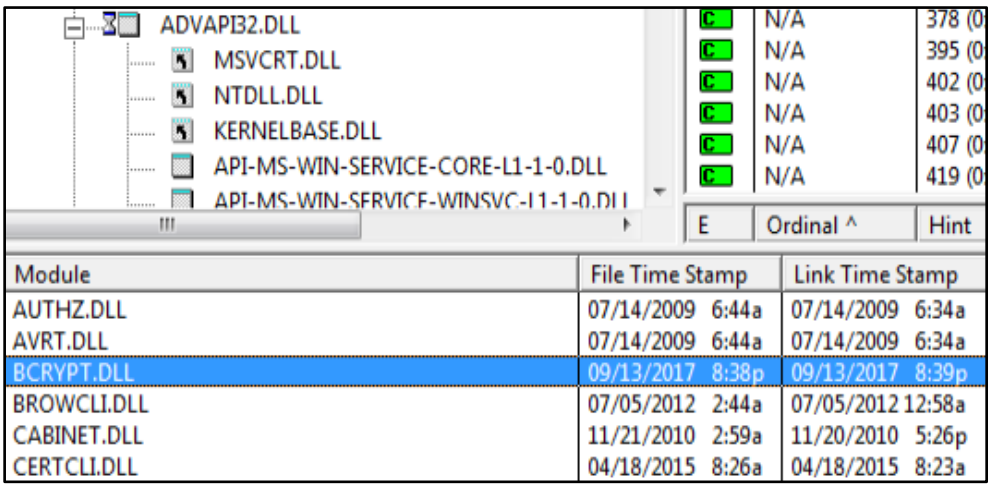


Fig. 6. Crypto-Locker

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

Locky did not lock the system immediately; instead it showed evasive behaviour and used another name for execution. The suspicious strings identified during analysis are shown in Fig. 7.

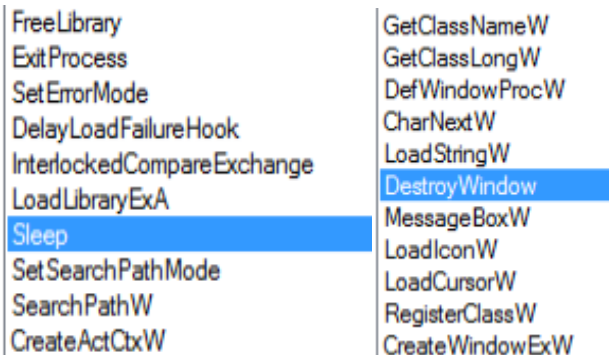
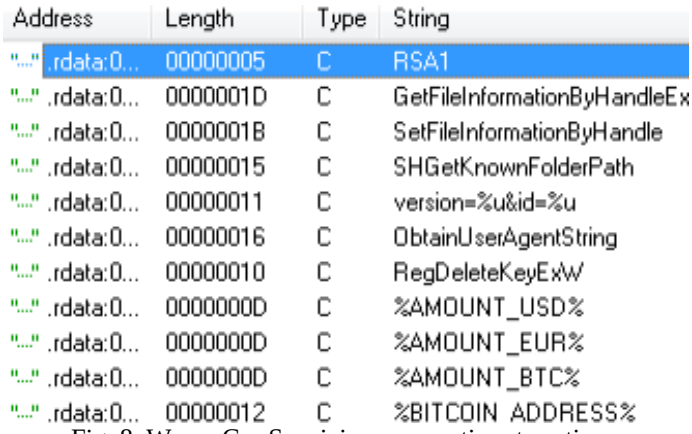


Fig. 7. String found in 'Locky-Ransomware'

WannaCry encrypted the files on the system and changed the registry values. The suspicious strings and RSA1 encryption used by this ransomware are shown in Fig. 8.



Address	Length	Type	String
".rdata:0...	00000005	C	RSA1
".rdata:0...	0000001D	C	GetFileInformationByHandleEx
".rdata:0...	0000001B	C	SetFileInformationByHandle
".rdata:0...	00000015	C	SHGetKnownFolderPath
".rdata:0...	00000011	C	version=%u&id=%u
".rdata:0...	00000016	C	ObtainUserAgentString
".rdata:0...	00000010	C	RegDeleteKeyExW
".rdata:0...	0000000D	C	%AMOUNT_USD%
".rdata:0...	0000000D	C	%AMOUNT_EUR%
".rdata:0...	0000000D	C	%AMOUNT_BTC%
".rdata:0...	00000012	C	%BITCOIN_ADDRESS%

Fig. 8. WannaCry Suspicious properties at runtime

The observations, during the run/debug-time analysis of the ransomware samples, are summarized in the Table 4.

Table IV Run-time observations of Ransomware

Ransomware Family	**Observations**	**Effects on System**
Crypto-Locker	Sample is Packed	Encryption of files
	Suspicious DLLs	System is locked
	Write/Edit commands	Relocations
Locky	Sample is Packed	Not locked
	Sleep/Destroy windows commands	Relocations
	Changed the process name to rundll32.exe	
Wanna-Cry	Sample is Packed	Not locked
	RSA1 encryption technique	Encryption of files
	strings (%bitcoins%/ %Amount% etc)	Changes in Registries

The most commonly observed behaviours are listed below:

1. Presence of suspicious DLLs in the import section at run-time. Most common functions are to write /edit etc.
2. Change in windows registry.
3. Deletion and modification of windows directories.
4. Hiding the traces by using 'DestroyWindow' command and changing the process name.
5. Catching the encryption key used for encryption of the file during the ransomware attack.
6. Encryption techniques, such as RSA1.
7. Presence of suspicious strings found at run-time.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

B. Static Properties based Classification for Ransomware Identification

To validate the significance of the ransomware-specific properties identified in Table 4, malware classification is performed first with the general malware properties and then with the collection of all the general and specific properties. The data from 2488 benign samples and 2722 malware samples (combination of ransomware and other malware) is used for training and classification of malware. The static properties for general malware classification are used for malware classification with three algorithms, Random Forest, Decision Trees (J-48), and Naive Bayes as shown in Table 5. The random forest algorithm has shown best performance with the accuracy of 98.34%.

Table V Classification based on general malware static properties

Classification Algorithm	TPR	FPR	Precision	Recall	Accuracy (%)
Naïve Bayes	0.620	0.349	0.744	0.62	62.789
J48	0.973	0.028	0.973	0.973	97.28
Random Forest	0.983	0.017	0.983	0.983	98.34

The details of the classification using random forest classifier for general properties are discussed below. The classification is estimated using following evaluation metrics:

$$\text{True Positive Rate (TPR)} = \frac{TP}{(TP+FN)} \quad (1)$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{(FP+TN)} \quad (2)$$

$$\text{Precision} = \frac{TP}{(TP+FN)} \quad (3)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (4)$$

$$F-\text{measure} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (5)$$

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

Where,
TP: True Positive, i.e. malware classified as malware
FP: False Positive, i.e. benign classified as malware
TN: True Negative, i.e. benign classified as benign
FN: False Negative, i.e. malware classified as benign

We have used random forest classification algorithm as implemented in Weka with 10-fold cross validation. The classification summary based on the correctly and incorrectly classified instances is given in the Weka output screen shot, Fig. 9. It shows the summarized and detailed performance of the classification using the general malware properties. To check the significance of identified ransomware-specific properties in the classification, the specific properties are added to the set of general properties for classifying the same set of malware including ransomware. The classification using random forest classifier based on the ransomware-specific properties is shown through Weka output screenshot in Fig. 10. The figure shows classification summary and detailed performance of the classification using the ransomware specific and general malware properties.

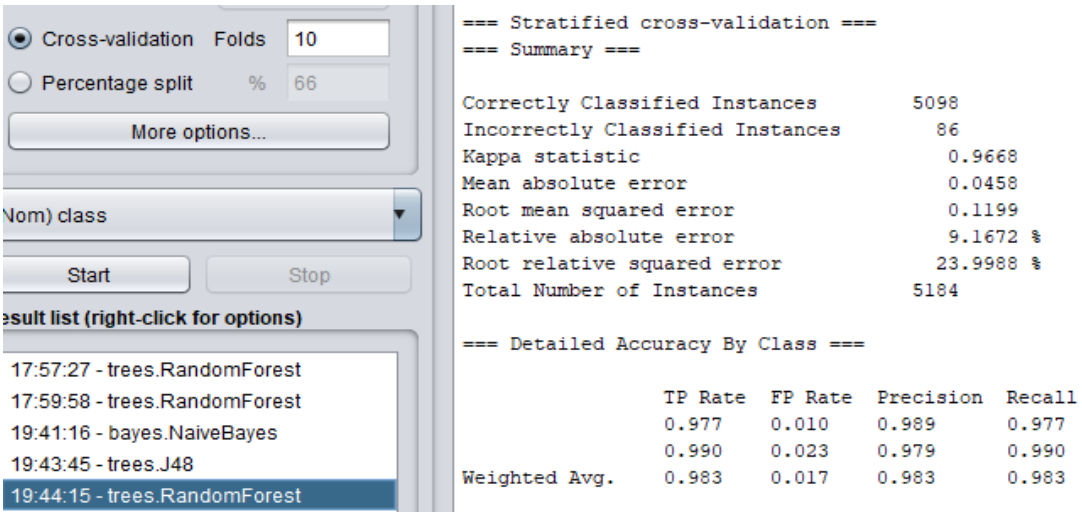


Fig. 9. Classification using Random forest based on generic malware properties

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

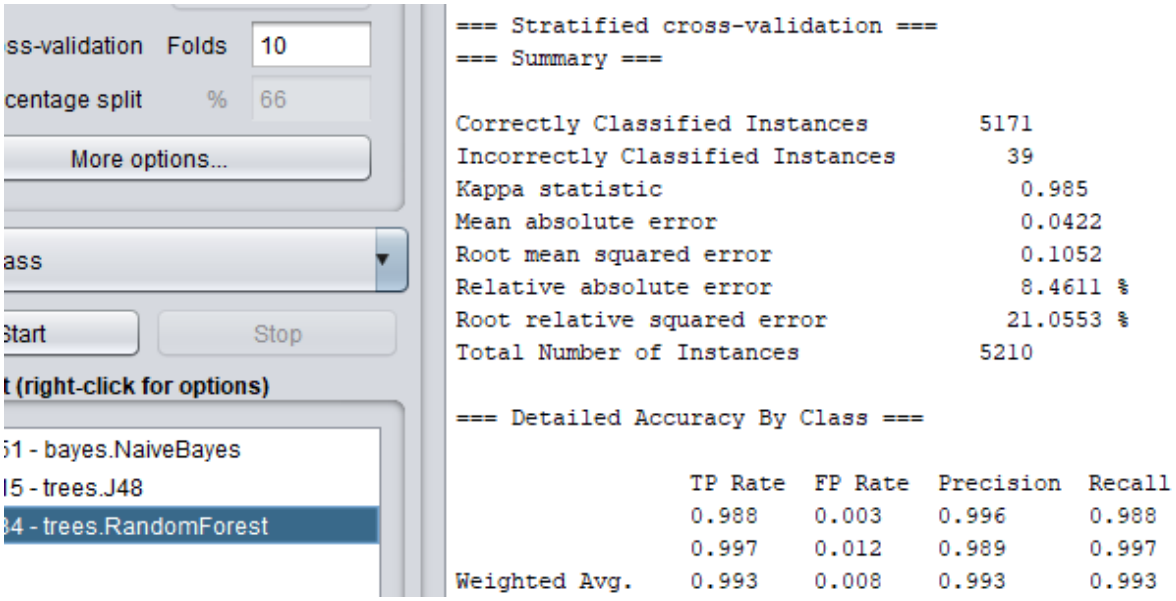
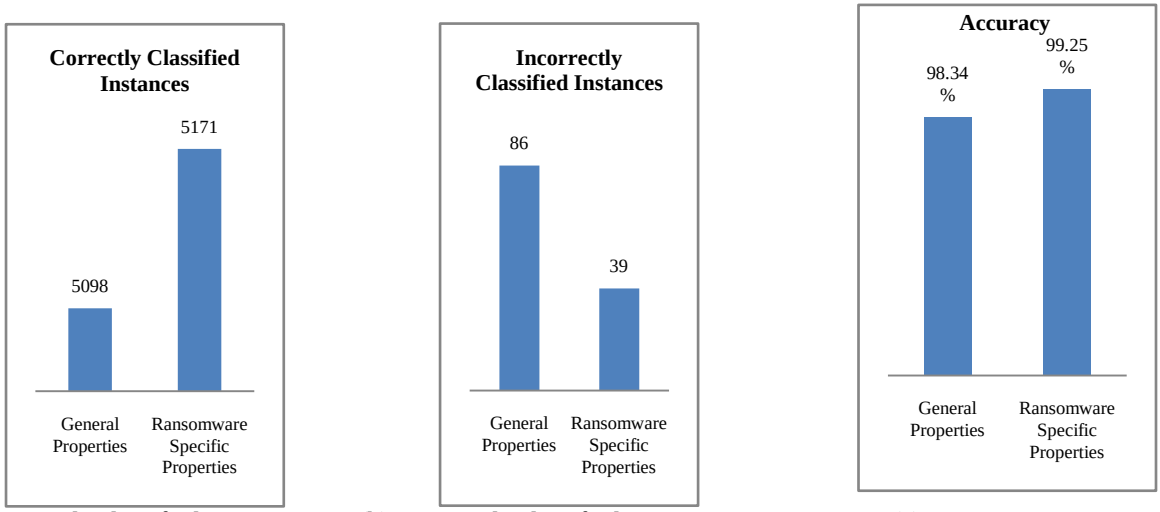


Fig. 10. Classification using Random forest based on ransomware specific malware properties

The performance comparison of the classification based on the general malware properties and based on additional ransomware-specific properties are compared as shown in Fig. 11.



(a) Correctly classified instances (b) Incorrectly classified instances (c) Accuracy

Fig. 11. Performance comparison of classification based on general properties and additional ransomware-specific properties

The comparison shows clear increase in correctly classified instances and decrease in incorrectly classified instances as we include the ransomware-specific characteristics in general malware features for classification. The overall increase in accuracy is shown in Fig. 11-(c). The experimentation results illustrate that including the static properties identified in ransomware-specific analysis with the general malware features can achieve higher accuracy in classification. Apart from these static and debug-time ransomware-specific properties, we have identified 7 run-time properties, typically present during ransomware execution. These properties can be further validated and used for the classification of ransomware.

CONCLUSIONS

The proposed work aimed to identify typical characteristics present in ransomware by static analysis and debug-time analysis. Existing major ransomware detection approaches consist of signature-based detection or analysis of encryption and network communication. The proposed approach and experiment indicates that by including the ransomware-specific properties, conventional malware detection techniques can be extended to detect ransomware with

higher accuracy. Static and debug-time analysis of ransomware identified 9 specific properties, when added to 60 generic properties for malware detection, the classification accuracy is increased. Along with these properties, 7 dynamic behavior patterns specific to ransomware are identified. This research work can be further enhanced by addressing the challenges present in this work such as evasive behavior of certain ransomware and their system locking property.

REFERENCES

[1]. Crowe, J. (2017). Must-Know Ransomware Statistics 2017. Retrieved From Stats & Trends.
[2]. Ablon, L., Libicki, M. C., & Golay, A. A. (2014). Markets for cybercrime tools and stolen data: Hackers' bazaar. Rand Corporation.
[3]. Azmoodeh, A., Dehghantanha, A., Conti, M., & Choo, K. K. R. (2018). Detecting crypto-ransomware in IoT networks based on energy consumption footprint. Journal of Ambient Intelligence and Humanized Computing, 9(4), 1141-1152. –[16]
[4]. Bhardwaj, A., Avasthi, V., Sastry, H., & Subrahmanyam, G. V. B. (2016). Ransomware digital extortion: a rising new age threat. Indian Journal of Science and Technology, 9(14), 1-5.
[5]. Vidyarthi, D., Choudhary, S. P., Rakshit, S., & Kumar, C. S. (2017). Malware Detection by Static Checking and Dynamic Analysis of Executables. International Journal of Information Security and Privacy (IJISP), 11(3), 29-41.
[6]. Kharaz, A., Arshad, S., Mulliner, C., Robertson, W., & Kirda, E. (2016). {UNVEIL}: A Large-Scale, Automated Approach to Detecting Ransomware. In 25th {USENIX} Security Symposium ({USENIX} Security 16) (pp. 757-772).
[7]. Singh S. (2018). Ransomware Command and Control Detection Using Machine. Aclavio. https://www.acalvio.com/ransomware-command-and-control-detection-using-machine-learning; 15 Jan 2018. Accessed: 10-Sep-2018.
[8]. Kramer, S., & Bradfield, J. C. (2010). A general definition of malware. Journal in computer virology, 6(2), 105-114.
[9]. Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. Network Security, 2016(9), 5-9.
[10]. Subedi, K. P., Budhathoki, D. R., & Dasgupta, D. (2018, May). Forensic analysis of ransomware families using static and dynamic analysis. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 180-185). IEEE.
[11]. Zimba, A., Wang, Z., & Chen, H. (2018). Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems. ICT Express, 4(1), 14-18.
[12]. Abrams, L. (2018). The Week in Ransomware - August 10th 2018 - BitPaymer & KeyPass. BleepingComputer. Extracted from https://www.bleepingcomputer.com/news/security/the-week-in-ransomware-august-10th-2018-bitpaymer-and-keypass/. August 11, 2018. Last access date 11-02-2019.
[13]. Zavarsky, P., & Lindskog, D. (2016). Experimental analysis of ransomware on windows and android platforms: Evolution and characterization. Procedia Computer Science, 94, 465-472.
[14]. Chen, Q., & Bridges, R. A. (2017, December). Automated behavioral analysis of malware: A case study of wannacry ransomware. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 454-460). IEEE.
[15]. Data Unit Blog. (2016). The Relationship between TOR and Ransomware. Extracted from blog.dataunit.be/the-relationship-between-tor-and-ransomware. Accessed: 10-Sep-2018.
[16]. Gandotra, E., Bansal, D., & Sofat, S. (2014). Malware analysis and classification: A survey. Journal of Information Security, 5(02), 56.
[17]. Bayer, U., Moser, A., Kruegel, C., & Kirda, E. (2006). Dynamic analysis of malicious code. Journal in Computer Virology, 2(1), 67-77.