A behaviour based ransomware detection using neural network models

Eleni Ketzaki
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
eketzaki@iti.gr

Petros Toupas
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
ptoupas@iti.gr

Konstantinos Giannoutakis
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
kgiannou@iti.gr

Anastasios Drosou
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
drosou@iti.gr

Dimitrios Tzovaras
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
Dimitrios.Tzovaras@iti.gr

***Abstract*— This study proposes a behaviour based methodology for ransomware detection. Ransomware is the type of malware that restricts access to files or blocks an infected device asking victims to pay fees in order to remove the restriction. The proposed detection procedure is based on the usage of neural network methodologies for the ransomware detection assuming features that related only with the utilization of the device resources. In the first part of the study, the System Monitor Service is proposed, that records the utilisation of the workstations' resources and extracts the corresponding features that describe their behaviour. The above tool monitors in real time the CPU, the memory, the disk space, the rate of reads and writes, the number of changed, created and deleted files. The second part of the methodology concerns the development of a neural network model that detects ransomware. Based on real data that arose from the System Monitor service, a model that fulfils the modern needs regarding the performance of the agents has been developed. The proposed methodology is ideal for Small and Medium Enterprises (SMEs) that constitute a particular target of the ransomwares for financial reasons.**

Keywords— ransomware, malware detection, neural network

I. Introduction

Ransomware constitutes the type of malware that block access to files or devices and forces the victims to pay ransoms in order to remove the restrictions from infected devices [1]. The ransomware is designed mainly to damage or lock files in order to extract money. The attacker exploits the fear of the victim for losing valuable data or publishing sensitive data and claims money [2]. The main targets of ransomware are professional users or companies for financial reasons. The amount of money that the victim asked to pay for ransoms varies, it usually ranges between 300$ -700$ for users and 10,000$ - 17,000$ for companies [3], [4].

During a ransomware attack the following phases take place: the distribution phase, the infection phase, the communication phase, the searching files phase, the encryption phase and the demanding ransomware phase. During the distribution phase, the malicious code or the mail attachment is disseminated into the victim's machine [5]. In the infection phase, a series of actions are taking place in the host machine. At first, the attacker generates a unique computer ID, disables shadow copies, installs the program to startup and retrieves the external IP [6]. In the communication phase, the ransomware contacts its command and control server to get an encryption key [7]. Then in the searching files phase the malicious process searches for user-related files with specific extensions, such as pdf, docx, xlsx, pptx and jpg [8]. The ransomware moves the targeted files into a different location, and then the encryption phase takes place. The encrypted files are renamed and the original files are deleted. The last part of the ransomware is the demanding ransom phase where the malicious process displays the claim that contains ransom demands to the victim, on either a text file or the desktop screen

The proposed methodology examines the behaviour of the SMEs devices in order to develop a neural network model for ransomware detection. The contribution of the proposed methodology is of considerable significance because it proposes a detection procedure based only on metrics related to the performance of the device with no need to analyse the source code of the malicious software. The idea of the above procedure inspired by the standard phases which happen during a ransomware attack. The usage of the deep neural network as a classification tool makes this proposal more flexible to the detection of new ransomwares since it does not relate to the type of the ransomware attack. The System Monitor Service is a tool that has been developed for recording and monitoring the performance of and records the features that related to their performance. A real dataset has been created from the records that have been arisen from the system monitor service. The features of the dataset had been used for the development of a deep neural network model that detects the ransomware type of attack effectively.

Our proposal is ideal for SMEs that constitute the main target of a ransomware attack and can benefit both morally and financially for this proposal. The rest of the manuscript is organized as follows: Section II provides related work for the ransomware detection methodologies. Section II describes the architecture of the proposed methodology for ransomware detection. Section IV provides the experimental results of the proposed methodology and consists of two parts: the first part concerns the creation of the dataset that is based on real data and the second part concerns the development of an efficient deep neural network model that can be used as a tool for ransomware detection. The last section, Section V concludes this study and contributes to the proposed methodology.

II. Related Work

The ransomware uses various techniques that differ in complexity and effectiveness to spread out and attack to as

many users as possible. There are two main approaches for ransomware detection: the static and dynamic analysis [9].

The static analysis is a passive approach to examine ransomware without running its code [10]. On the other hand, dynamic analysis is the type of analysis that takes place during the execution of the ransomware in a controlled simulated environment, in order to observe the real behavior and interaction of the program with the operating system [11].

Chen et al. [12] propose a dynamic analysis scheme to detect Android ransomware attacks by encrypting private data. In the first part, the implementation includes the distinction between normal applications and ransomware applications and then the creation of a detection system that monitors the operations on sensitive resources and extracts three UI indicators.

Kharraz et al. [13] refer to an automated payload analysis to limit the manual process, and proposed UNVEIL, which was designed to detect ransomware [13]. UNVEIL creates an artificial user environment, which is tempting to ransomware attacks, and then detects the possible interaction with user data, in order to identify unknown malware. Furthermore, they proposed improved techniques of monitoring using UNVEIL. The UNVEIL intervene in the interaction of user-mode processes with the file system and has access to data buffers involved in I/O requests, in order to monitor the system-wide activities using a kernel-level module.

Morato et al. [19] proposed an algorithm based on the behaviour of reading, writing and removing files. Their methodology proved that more than 99% of the ransomware they are detected before ten files are deleted.

The main types of features that have been used to develop detection tools are the third-party calls [14], API packages, string-based features [15], permissions and network addresses [16], and encryption processes, threatening texts or locking devices [17], measures of hardware metrics, such as a processor or memory usage [18]. Moreover, the observation of ransomware spreading could also drive to ransomware detection methods. The spreading of malware can examined by continuous monitoring of abnormal file system and registry activities, or paying closer attention to permissions requested by the applications

III. Proposed methodology

The proposed methodology inspired by the standard phases that are observed under a ransomware attack. We assume that the distribution phase, the infection phase and the communication phase increase the CPU and the features related to the memory performance. On the other hand, the searching file phase and the encryption phase may affect the features related to the number of files and the rate of reading and writing.

The architecture of the proposed methodology is described in Fig.1 and consists of three main parts. The first part of the procedure concerns the development and the installation of the System Monitor (SysMon) service that depicts and records the performance of the system. The second part is the feature extraction part which is responsible for the development of a dataset based on the log files that have been arisen from the SysMon service. The third part of the procedure describes the development of the neural network model for ransomware detection. The output of the model discriminates the normal usage of the system from the performance of the system that is depicted during a ransomware attack.

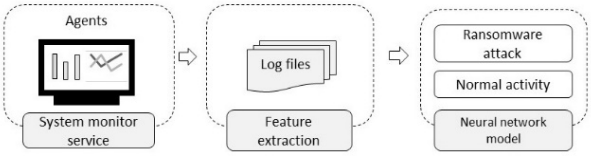

Fig. 1. The architecture of the ransomware detection methodology consists of three parts; the usage of the SysMon service the feature extraction part and the development of the neural network model for ransomware detection

A. *Description of the System Monitor service and the feature extraction part.*

The SysMon service has been developed for the purposes of the FORTIKA project, it examines real-time the performance of the system and records every five seconds the values of the nine features described in Table 1. The purpose of the SysMon service is to produce the log files that depict the performance during normal usage and under ransomware attack.

TABLE I. Description of the features that are available from the SysMon service per timestamp.

Feature	Description of the feature
CPU	Utilization of the used CPU
Physical memory	Physical memory of the overall used memory
Virtual memory	Used virtual memory of the overall used memory
Disk space	Used space for every disk of the overall used space
Reads	Rate of reads per second
Writes	Rate of writes per second
Created files	Number of created files
Deleted files	Number of deleted files
Renamed files	Number of renamed files
Changed files	Number of changed files

B. *Description of the deep neural network development*

The second part of the proposed methodology concerns the development of the neural network model that detects the ransomware attacks. It is based on the usage of a real generated dataset that was obtained from the log files produced from the SysMon service. Both situations related to normal performance and malicious attack should be examined to provide the input features for the proposed neural network model. The output of the proposed model predicts whether the performance of a device concerns normal usage or it indicates possible malicious activity

IV. Experimental results

The experimental procedure consists of two main parts. The first part of this section describes the creation of the dataset that based on real data. A well designed dataset derives to reliable predictions and constitutes a fundamental part of the experimental procedure. The dataset will provide the input features for the training of the neural network model.

The second part of the experimental procedure concerns the development of the proposed neural network model. This part includes also the preprocessing methodologies that prepare the values of the features for the training process of the deep neural network. Finally, the high values of the accuracy, the precision, the recall and the F1-score in the last part of the section validate the proposed methodology

A. Dataset creation

The SysMon service installed in eighteen devices for the creation of the dataset. The sum of the log files that produced from the monitoring agents during normal activity or under malicious performance have been collected and stored in order to be re-used as training data to the neural network. The normal behavior concerns actions that are associated with the normal usage of their system. During a normal behavior scenario the performance may be either low or high. We assume that the normal usage is obtained as a results of multiple actions such as the browsing of web pages, the downloading of files, the reading or writing documents and the training process of neural network models. During the experimental procedure 209,627 incidents that concern normal behaviour from all the agents were recorded. The RanSim tool was used to simulate the ransomware attack scenarios [1]. It is a free ransomware simulator tool which simulates fifteen ransomware infection scenarios one crypto mining infection scenario [19]. The RanSim tool assumed to simulate the procedure under a ransomware attack and used to produce the performance of the agents that are under an attack. The simulation procedure consists of 8,862 incidents that concern the performance of a ransomware attack.

TABLE II. THE MEAN VALUE AND THE ST. DEVIATION FOR THE INPUT FEATURES DURING NORMAL ACTIVITY AND UNDER RANSOMAWARE ATTACK

Features	Malicious		Normal	
	Mean value	St. Deviation	Mean value	St. Deviation
CPU	37.69	20.10	12.69	16.92
Physical memory	52.94	0.24	54.46	8.90
Virtual memory	62.37	2760174.63	61.93	12.84
Disk space	57.11	717084.21	54.58	24.65
Reads	0.78	808252.14	19.16	118.07
Writes	39.01	71981.40	22.30	117.53
Created files	497325.90	3.75	19580.09	157568.55
Deleted files	485975.72	15.26	23384.65	574465.42
Renamed files	42676.45	5.37	7982.76	46336.01
Changed files	2162864.22	98.19	128500.75	525370.04

Adding the log files that are annotated with normal or malicious performance we construct a dataset that consists of total 218,489 incidents, each incident consists of the values for the nine features described in Table I and an extra feature that denotes whether that belongs to normal usage or not. Table II provides a brief overview regarding the mean value and the standard deviation for each feature of the dataset.

Comparing the mean values of the features, it obtains that there is a significant differentiation between the incidents that concern malicious performance and the incidents that concern normal behaviour. More specifically the mean percent of the CPU used during a ransomware attack is 37.68% with standard deviation 20.09 while the mean percent of the CPU used during normal performance is lower 12.68% with standard deviation 16.91.

The low value of standard deviation indicates that the values tend to be close to the mean value, and a high standard deviation indicates that the values are spread out. The experimental results show that there is a higher standard deviation for the CPU, the virtual memory, the disk space, the reads and the writes during a ransomware attack.

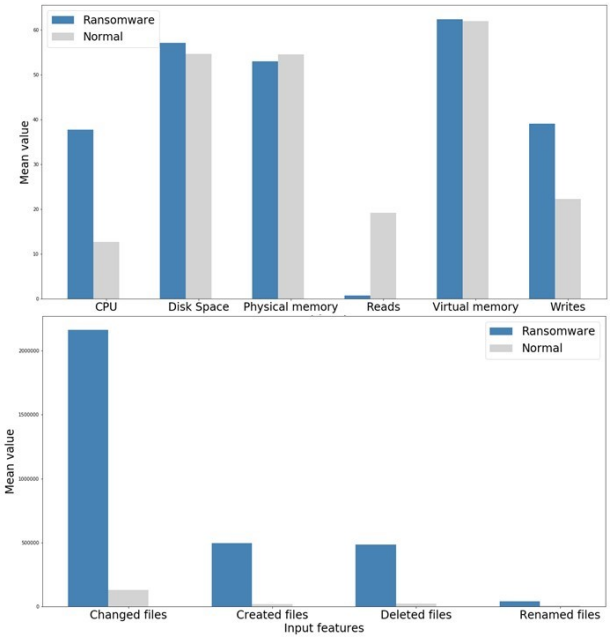

Fig. 2. Comparison of the mean value for the Features CPU, Disk Space, Physical memory, Reads, Virtual memory and Writes for a normal performance and for performance under ransomware attack.

Figure 2, depicts the comparison of the mean values for nine features during a ransomware attack and during a normal performance. It derives that the mean value, during a ransomware scenario, increases for most of the features that related with the performance of the agent's system.

In order to examine if there is a significant difference among the normal performance and the performance of the agent during a ransomware attack the non-parametric Mann Whitney test has been used since the Kolmogorov Smirnov test for normality prove that in the proposed dataset all the variables that describe the features of the Table II, do not follow Normal distribution. More specific the value of the statistic and of the significant value are; for the CPU (Z=0.74199, p=0.0) for the Disk Space feature (Z=1.0, p=0.0), for the physical memory feature (Z=1.0, p=0.0), for the virtual memory feature: (Z=1.0, p=0.0), for the reads feature (Z=0.5, p=0.0), for the writes feature (Z=0.85256, p=0.0) for the Created file feature (Z=0.67049, p=0.0) for the Deleted file feature (Z=0.62395, p=0.0) for the Renamed file feature (Z=0.51514, p=0.0) and for the Changed file feature (Z=0.99988, p=0.0).

The statistic and the significant value of the Mann Whitney test calculated to examine whether there is statistical difference between normal performance and a ransomware attack. The statistic and the significant values for the features are as follow for the CPU (U=48213217.5, p=0.0), for the Disk Space feature (U=0.0, p=0.0), for the physical memory feature (U=0.0, p=0.0), for the virtual memory feature: (U=0.0, p=0.0), for the reads feature (U=338064699.0, p=4.1436e-06), for the writes feature (U=3539875.5, p=0.0) for the Created file feature (U=151889675.0, p=0.0) for the Deleted file feature (U=173321127.0, p =0.0) for the Renamed file feature (U=223532435.0, p =0.0) and for the Changed file feature (U=52678.5, p =0.0). The significant value for every feature is less than 0.05 that means we can

reject the null hypothesis of the test and assume that there is significant difference for the values of the features between a normal performance and the performance under a ransomware attack.

B. *Development of deep neural network model and accuracy measures*

This section describes the architecture of the proposed neural network model and the metrics that validate the proposed model. Since there is a difference in the range of the mean values (Fig. 2) the MinMaxScaler process for the Sklearn Preprocessing libraries of Python has been used for the normalization of data.

The architecture of the neural network model consists of one input layer which has ten features; two hidden layers follow the input layer with 32 and 16 nodes respectively. The final layer is the output layer producing the probabilities for the two classes. The proposed model has been trained for 30 epochs and the Relu, activations functions have been used for each one of the hidden layers and the Softmax activation function has been used for the output layer. Table III, describes the confusion matrix for the proposed neural network model.

TABLE III. THE CONFUSION MATRIX FOR THE PREDICTIONS OF THE NEURAL NETWORK MODEL. THE COLUMNS OF THE TABLE DESCRIBE THE PREDICTED NEGATIVE AND POSITIVE VALUES AND THE ROWS OF THE TABLE DESCRIBE THE REAL VALUES.

	Negative	**Positive**
Negative	41901	3
Positive	7	1787

TABLE IV. DESCRIPTION OF THE ACCURACY, PRECISION, F1-SCORE AND RECALL THAT CONSTITUTE THE VALIDATION METRICS FOR THE EXPERIMENTAL RESULTS.

Metrics	***Accuracy***	***Precision***	***F1-score***	***Recall***
Percent	99.98	99.83	99.6	99.97

The accuracy, the precision, the F1-score and the recall validate the experimental results that obtain from the proposed method. The evaluation of the proposed deep neural network model was based on a 10-fold cross validation. The accuracy of the model is 99.98%, the precision 99.832%, the F1-score 99.609% and the recall 99.977% (Table IV).

V. CONCLUSIONS

In this study, we developed an efficient methodology for ransomware detection. The proposed procedure is a behaviour-based methodology based on the usage of neural network models. The main contribution of the proposed methodology is that it based only on metrics related to the performance of the devices with no need to analyze the source code of the malicious software. The development and the training of the neural network model based on real dataset the data of which have been created based on the System Monitor service. The efficiency of the proposed methodology proved from the obtained high values of the accuracy of the model. Our proposal is ideal for the SMEs that constitute a particular target of the ransomware attack for financial reason. An efficient ransomware detection methodology is of great importance since it will benefit companies financially and will protect their sensitive data effectively from ransomwares.

ACKNOWLEDGMENT

This work has been partially supported by the European Commission through project FORTIKA funded by the European Union Horizon 2020 program under Grant Agreement nº 740690. The opinions expressed in this paper are those of authors and do not necessarily reflect the views of the European Commission..

REFERENCES

[1] N. Andronio, S. Zanero, F. Maggi, "Heldroid : Dissecting and detecting mobile ransomware," International Symposium on Recent Advances in Intrusion Detection. Springer, Cham. pp. 382-404. 2015

[2] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 3-24. Springer, Cham., July 2015

[3] C. Everett, "Ransomware: to pay or not to pay?," Computer Fraud and Security, pp. 8-12, 2016

[4] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," Computers and Electrical Engineering, 66, pp.353-368, 2018

[5] D. Kim, W. Soh, and S. Kim, "Design of quantification model for prevent of cryptolocker," Indian Journal of Science and Technology, 8 (19), 2015

[6] R. Leong, "Understanding Ransomware and Strategies to Defeat it," White Paper (McAfee Labs), pp. 1-16. 2016

[7] Zimba, A. "Malware-free intrusion: a novel approach to Ransomware infection vectors, " International Journal of Computer Science and Information Security, 15 (2), 317. (2017)

[8] F. Mbol, J.M. Robert and A. Sadighian, "An efficient approach to detect torrentlocker ransomware in computer systems," International Conference on Cryptology and Network Security Springer, Cham., pp. 532-541, November 2016

[9] P.V. Shijo, and A. Salim, "Integrated static and dynamic analysis for malware detection," Procedia Computer Science, 46, pp. 804-811, 2015

[10] P. Wang, and Y. S. Wang, "Malware behavioural detection and vaccine development by using a support vector model classifier," Journal of Computer and System Sciences, 81(6), pp. 1012-1026, 2015

[11] R. Kaur, and M.A. Singh, "Survey on Zero-Day Polymorphic Worm Detection Techniques," IEEE communications surveys and tutorials, 16 (3), pp. 1520- 1549, 2014.

[12] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du and G.J. Ahn, "Uncovering the face of android ransomware: Characterization and real-time detection," IEEE Transactions on Information Forensics and Security, 13(5), pp. 1286-1300, 2018

[13] A. Kharraz, W. Robertson, E. Kirda, "Protecting against Ransomware: A New Line of Research or Restating Classic Ideas?," IEEE Security and Privacy, 16(3), pp.103-107, 2018

[14] A. Martín, H. D. Menéndez, and D. Camacho," MOCDroid: multi-objective evolutionary classifier for Android malware detection," Soft Computing, 21(24), pp.7405-7415, 2017

[15] A. Martín, H. D. Menéndez, and D. Camacho, "String-based malware detection for android environments," International Symposium on Intelligent and Distributed Computing pp. 99-108. Springer, Cham, October 2016

[16] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and Siemens, C. E. R. T. :DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket." In Ndss 14, pp. 23-26. February 2014

[17] S. Song, B. Kim, and S. Lee, "The effective ransomware prevention technique using process monitoring on android platform," Mobile Information Systems, 2016

[18] O.D. Morató, I.E. Berrueta, L.E. Magaña and A.M. Izal, "Ransomware early detection by the analysis of file sharing traffic," Journal of Network and Computer Applications, 124, pp.14-32, 2018

[19] Knowbe4 RanSim Homepage. https://www.knowbe4.com/ransomware-simulator