IRBAS www.irbas.academyirmbr.com April 2022
International Review of Basic and Applied Sciences Vol. 10 Issue.2

Ransomware Early Detection Model using API-Calls at Runtime by Random Decision Forests

IZAZ AHMAD KHAN
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.
Email: azaz@bkuc.edu.pk

FAIZAN ULLAH
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

MOHAMMAD ABRAR
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

SABIR SHAH
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

FAWAD TAJ
Department of Computer Science, Vrije University, Amsterdam Netherland.

MUHAMMAD IMRAN KHAN KHALIL
Department of CS & IT, University of Engineering and Technology, Peshawar Pakistan.

ABDU SALAM
Department of Computer Science, Abdul Wali Khan University, Mardan Pakistan.

Abstract

Ransomware is a destructive type of malware that encrypts the user's valuable data or locks the screen of the user's device, causing massive economic losses to users. Signature-based ransomware detection models struggle to detect zero-day ransomware and questioning their suitability for protecting user's files against such attacks. In this study, we propose a model that extracts eighteen useful features vectors from the ransomware dataset. It performs classification on ransomware datasets. We utilize API call series to represent behavior-based features of ransomware. To validate the effectiveness of Random Forest, we tested 78556 ransomware and good ware files. Compared to Naive Bayes and Support Vector Machine, the testing accuracy of the proposed method is 99.57%. In the future, we will use deep learning to detect ransomware and its types at an early stage.

Keywords: *Malware Analysis, Machine Learning, Virus Total, Random Forest, Support Vector Machine.*

Introduction

Although computer applications and the internet make our lifestyle easier, they also pose several serious risks(Adams, 2018). Malicious applications or malware are rapidly attacking the internet. They penetrate industrial systems, gather sensitive data, and disrupt the operations, making cyber security a challenging

issue for researchers nowadays (Zonouz, Rrushi, & McLaughlin, 2014). Malware attacks were initially reported in 1970 (Biddle, Chiasson, & Van Oorschot, 2012). Trojan's horse, worms, viruses, and ransomware are various types of malwares that are now getting the attention of researchers in recent years. In ransomware, the attacker strikes the user's data. It prevents or limits the user's access to their data by either encrypting the user's data or locking the screen. After the damage, the attacker demands the victim to pay some ransom for the hijacked data. It compelled the victim to pay the ransom, otherwise the precious data is going to vanish (Kharraz, Robertson, Balzarotti, Bilge, & Kirda, 2015). The first attack of ransomware was launched in 1989, when a type of Trojan was identified. Unlike existing traditional malware, ransomware is very harmful because it is hard to trace, and encrypts user data (Lee, Moon, & Park, 2017).

Wanna-cry, a form of ransomware that initially appeared in 1980, shocked the world in 2017 and is hunting a huge number of users (Scaife, Traynor, & Butler, 2017). According to a survey conducted by Kaspersky (Zaitsev, 2013), crypto-ransomware, another type of ransomware, has drastically grown to 31.6% in 2015-2016 against a mere 6.6% during 2014-2015. McAfee (Teddy, Bean, Dalcher, Hetzler, & Woodruff, 2016) shows that ransomware will emerge as a major fear soon, mainly because of the Ransomware-as-a-service (Raas) development platforms. Different ransomware infection factors are having varying magnitude, complexity, and effectiveness (Zimba, Wang, & Chen, 2017).These infection vectors enter the user systems more secretly. Some examples are forceful authentication credentials, drive-by freeware applications, and frauds. The unaware users may get trapped by clicking the money-making links or free applications. Likewise, the attack of ransomware may occur through social engineering techniques, where in an infected attachment is sent through an email or botnet attack (Bertino & Islam, 2017). Ransomware uses a variety of infection techniques. Attackers get access to the victim's device by sending attachments or compromised websites (Aljawarneh & Yassein, 2016).

An increase in the infection rate of ransomware is because of monetary benefits. Ransomware achieved this by attracting many adversaries and build new variants. Technology users need new protection techniques and approaches for detection and prevention of these attacks. In this article, we propose a robust solution for ransomware detection at the runtime. This task is accomplished by monitoring network, file systems, and associated registries. In this study, API-call data to denote behavior-based features. By using relevant classifiers, we apply 18 feature vectors during runtime in the proposed method.

The rest of the article is organized as follows. Section 2 describes the recent literature on Ransomware classification and its detection. Section 3 elaborates the proposed methodology. Experiments and evaluation are discussed in section 4. The last section concludes the paper and provides the future direction.

Related Work

This section presents an overview of the work done for the detection and classification of ransomware. Table 1 summarizes the existing models. The research community uses classifiers that fall into one of four major categories: Statistical, Event-based detection, Machine learning-based solution, and Data centric-based detection solution.

Statistical

This category is widely used to create a ransomware detection model. Such models develop a reference model in the form of a probability distribution. This distribution represents the data in the reference model and then compares every suspected data with the reference model (Alhawi, Baldwin, & Dehghantanha, 2018). Any divergence from the threshold is treated as outliers. In (Homayoun et al., 2019) utilizes statistical Bayesian network to create a model for Detecting a High Serval able Ransomware (HSR). The authors train the model with 20 dynamic and static features of the HSR family. Like a specification-based technique which is derived from an anomaly-based approach, 20 features are utilized as fundamental in a

rule-based manner to detect a malicious pattern. Despite anomaly detection systems, these principle systems describe suspicious characteristics of ransomware instead of the normal aspect of the proposed approach. However, the exact definition of HSR will not always be the same because the adversaries and advancement come in ransomware with time.

The authors present a technique for mobile applications in (Song et al., 2020). They collect the data by scanning I/O, CPU, and memory usage. The acquired data is then compared with a pre-defined list of critical files that are needed for detection.

Event-based Detection

To detect imminent ransomware, event-based detection is used, which looks for an individual event to occur. (Lei, Qin, Wang, Li, & Ye, 2019) proposes a scanning-based solution, which monitors the communication to highlight the encryption key. Domain Generation Algorithm access gain to information transmitting between C&C remote server and malicious code, before the ransomware starts encryption. The same technique is proposed in (Victoriano, 2019) to use a dynamic approach for detection of threats containing text from payload having Crypto-ransomware.

The authors in (Ma, Han, & Wang, 2021) developed a Software Defined Network (SDN) that scans a series of HTTP messages and their data to detect Crypto-wall families. Similarly, (Mehnaz, Mudgerikar, & Bertino, 2018) presents a system that scans Microsoft Cryptographic API to detect ransomware before encrypting the user files. However, monitoring outliers, information is ignored by encoding such connections (Abri, Siami-Namini, Khanghah, Soltani, & Namin, 2019).

Event-based approaches for ransomware detection have their limitations. As event-based techniques need prior information of encryption used by ransomware (Maiorca, Mercaldo, Giacinto, Visaggio, & Martinelli, 2017). Similarly, event-based techniques are not enough to detect ransomware because sometimes they may not occur, but they already did their damage. These techniques have a high rate of false detection in some cases.

Machine Learning-based Solution

Machine learning techniques involve pattern learning to create a model from the data. When new data is fed into this model, it predicts the outcome. Machine learning-based approaches show a high rate of effectiveness in detecting ransomware. Researcher proposed different approaches based on machine learning techniques. The dynamic system suggested in (Al-rimy, Maarof, & Shaid, 2017) utilizes a collection of features and uses them during the execution of the first half of the first minute of the malicious code. These features are then given as input to a classifier called Mutual Information Criterion (MIC). However, the rate of detection is low and the false positive rate of the system is at the higher side. To detect ransomware in Android-based application, the work in (Alqahtani, Gazzan, & Sheldon, 2020) develops Rpack Droid, which uses extracted information as API calls. The system scans many commands comprising different classes.

Data Centric-based Detection Solution

Data centric-based detection is based on the idea of monitoring the resources that are subject to attack rather than running after the malicious process that executes the attacks (Al-rimy et al., 2017). Many studies have floated the proposal for a data centric-based ransomware detection solution (Alqahtani et al., 2020). These solutions are based on continuously clicking the resources subject to attack in order to detect any kind of abnormal changes for achieving the goal. Several tools, like entropy and similarity measures, are employed for situations when changes occur to the targeted file or resources at risk by the cryptography.

By employing the principles, the work in (Ramesh & Menen, 2020) leverages Shannon entropy to measure all the changes that are happening to the files that are being monitored whenever the files are being accessed. The proposed technique statically monitors the resources at risk or the user files for any changes to its structure before and after the file or resources are accessed. Besides, the similarity measurement is based on the hypothesis, which has no resemblance to the original copy with strong encryption. However, for such an irreversible attack, the proposed technique is not a very effective method of detection, as it depends mostly on static analysis.

For addressing the issue, the authors in proposed a technique that dynamically monitors the difference occurring at the entropy of the read and write data, and also monitors the I/O buffer contents. Likewise, proposes a data centric-based behavior model that focuses on the APIs which deal with the user-specific files, also known as relative entropy to detect the Torrent locker ransomware attacks, which encrypt the first part of the user-related file.

Decoy technique is another form of data centric detection, proposed in (Ahmed, Kocer, & Al-rimy, 2020) to detect the malicious and suspicious events that temper the user-specific files. It becomes effortless to detect any changes happening to the user's intended files by monitoring these decoys. The idea reveals that decoys are implanted in a specific area of user interest. The issue with this proposal was that the functionality of the technique was limited to a particular size of the file, as the decoy technique does not guarantee that suspicious file programs would access the file before the actual user-specific files. The authors propose to install decoy files in the deepest point of each folder to deviate the ransomware to encrypt these files first. Once the decoy files are touched, an alarm is raised by the detection system, and in the process, almost no data loss occurs. However, interestingly, the decoy technique cannot distinguish between benign and suspicious encryption.

Table 1: Comparative analysis of different techniques used for Ransomware detection.

	Data Source		Statistics		PC				Mob			Statistics		Machine Learning-based										Outcome			Accuracy parameters						
Ref	Static Analysis	Dynamic Analysis	Honeypot	Statistic	Linux	Mac	Windows	Raspberry	Android	IOS	Cloud	Statistic	Honeypot	Bayesian	Clustering	Decision Tree	Deep Learning	Dimension Reduction	Ensemble	Instance-Based	Neural Network	Regularization	Regression	Random Forest	Binary Classification	Multi-class Classification	Clustering	True.Positive Rate	False.Positive Rate	True.Negative Rate	False.Negative Rate	Precision	Accuracy	measure	Time
1																							✔						✔						
2		✔																														✔			
3																						✔													
8					✔																				✔										
9	✔		✔					✔					✔												✔					✔		✔	✔		
10		✔		✔			✔					✔																							
11							✔									✔				✔				✔	✔			✔				✔	✔	✔	
12	✔	✔					✔		✔														✔												
13		✔									✔						✔								✔	✔		✔	✔					✔	
14		✔					✔											✔				✔	✔					✔	✔						
15			✔																				✔												

	Data Source		Statistics		PC				Mob			Statistics		Machine Learning-based										Outcome			Accuracy parameters							
16	✔						✔												✔		✔				✔				✔		✔		✔		✔
17		✔					✔																	✔			✔							
18		✔					✔																	✔			✔				✔			✔
19		✔					✔															✔		✔			✔				✔	✔		
20	✔								✔								✔	✔	✔					✔							✔			
21																	✔					✔		✔								✔	✔	
22		✔					✔											✔		✔				✔			✔				✔		✔	
23		✔					✔											✔		✔		✔		✔			✔	✔				✔		
24	✔						✔							✔		✔		✔	✔	✔			✔	✔			✔				✔	✔	✔	
27		✔		✔					✔			✔										✔		✔										
28		✔	✔				✔						✔											✔							✔			
30							✔									✔		✔	✔	✔				✔			✔	✔			✔		✔	✔
31	✔	✔							✔															✔								✔		
32	✔								✔															✔			✔				✔		✔	✔
33	✔	✔					✔											✔		✔		✔		✔			✔	✔						
Total	8	13	2	2	1	0	14	1	5	0	1	2	3	1	1	3	1	4	7	4	5	3	9	3	18	1	1	11	6	2	1	10	9	7	4

Methodology

In this section, we propose a novel schema. The primary motivation of the proposed system is to detect ransomware during runtime. We obtain a ransomware dataset from the website www. Virustotal.com. There are two types of uniform resource locators (URLs) that are benign and ransomware. Figure 1 illustrates the different phases of research methodology.

As shown in Figure 1, initially we collect preprocessed data from the virus total. The next phase focuses on feature extraction from the data. Training and testing are performed next. In the last phase, three commonly used classifiers, namely Random Forest, Logistic Regression, and Naïve Bayes, were applied accordingly.

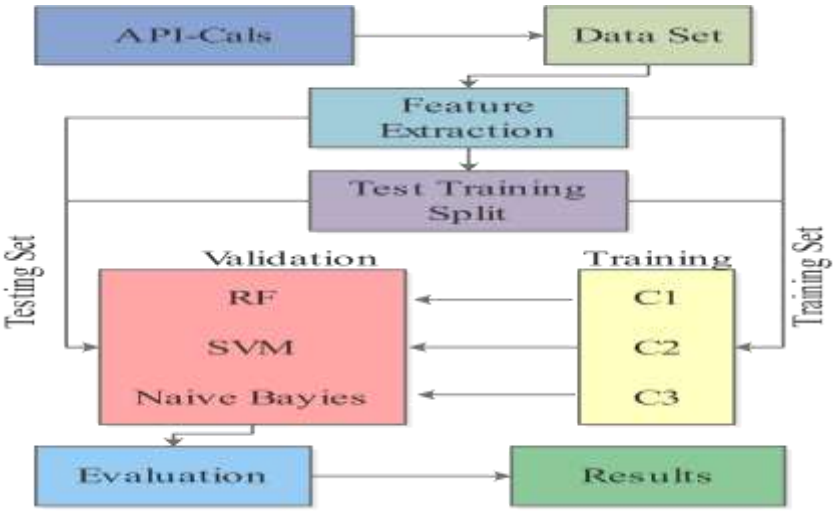

Fig.1 Proposed methodology of Ransomware Detection

Data Sets

The dataset has 43191 benign samples, while 35369 samples are malware out of 78550. There are 18 features in the dataset. We utilize 14 features in our previous study [25]. Now the reaming four features are used for classification.

Features Extraction

In the feature extraction phase, out of 18 features this time the remaining features i.e., File name, MD5 Hash, Packer method PE/UPX, and Registry entries are used. We did not consider these attributes because of computational complexity in our previous study. Now, these 4 attributes are fed to the classifier with better accuracy.

Table 2 : List of Extracted features from API-Calls

No	Features/ attributes	Description
1	File name	Information about API file. The file is lock screens or non-encrypting ransomware.
2	MD5 Hash	A message-digest is used for integrity. Message Digest Algorithm-5 (MD5) is a hash function. MD5 is the type of Hash that is most commonly utilized for malware analysis.
3.	Packer method PE/UPX	Ultimate Packer for Executable (UPX) is an open-source tool and executable packer, which is mostly present in the malware scene. UPX operates on all foremost systems applications and both x64 and x86 environment.
4.	Registry entries	Essentially it contains information about extensions and data. The data will have diverse attached codes with respective extensions.
5.	debugSize, DebugRVA, Major Image Version, ExportRVA, Export Size, IatRVA, Minor Linker Version, Major Linker Version, Size Of Stack Reserve, Dll Characteristics, ResourceSize, and Machine (Ullah et al.)	

Training and Testing

We partition the data into two segments. We train the model using the first segment which is 70%. For testing purpose we use the second segment which is 30%.

Classification

In this research, we utilize three different machine-learning algorithms i.e., , SVM, and Naïve Bayes.

Experiments and Results

In this phase, the experimental environment, experiments, and results are discussed. We statistically analyze the datasets to understand the data and then apply different classification techniques to classify the malicious and benign files. Finally, the performance evaluation measures are used to assess the performance of the classifiers.

Experimental Environment

We Carried out the experiments on a core i5 machine with 8GB of memory and a 2.4GHz CPU. We implement the Random Forest, SVM and Naïve Bayes in Python.

Evaluation Matrices

A standard evaluation measure of confusion matric is used to evaluate the performance of this model, as given in Table 3. These parameters include, specificity, sensitivity, and accuracy.

Table 3: Evaluation Measure For The Assessment Of Model

Predication Classes			
Predication Classes		True	False
	True	True-Positive	False-Positive
	False	False-Negative	True-Negative

$$Accuracy = \frac{True.Positive + True.Negative}{True.Positive + True.Negative + False.Postive + False.Negative} \quad (1)$$

$$Sensitivity = \frac{True.Positive}{True.Positive + False.Negative} \quad (2)$$

$$Specificity = \frac{True.Negative}{True.Negative + False.Postive} \quad (3)$$

$$f1 = 2\left(\frac{Sensitivity * Specificity}{Sensitivity + Specificity}\right) \quad (4)$$

Highlights of Accuracy

Table4 shows the individual accuracy of each classifier. Random forest with 99.34% is leading the rest of the classifiers.

Table 4: Individual Accuracy 0f Classifier

S.No	Classifier	Accuracy
1.	Naive Bayes	99.24%
2.	Random Forest	99.34%
3.	SVM	98.37%

Highlights of Sensitivity

The sensitivity of each classifier is shown in Table 5, with Random Forest achieving the highest level of sensitivity at 99.57%.

Table 5: Individual Sensitivity of Classifiers

S.No	Classifier	Sensitivity
1.	Naive Bayes	99.50%
2.	Random Forest	99.57%
3.	SVM	98.11%

Highlights of Specificity

Figure 2 verifies the highest level of specificity of the Random forest i.e., 99.62%. It further depicts the accuracy of our proposed ransomware detection system.

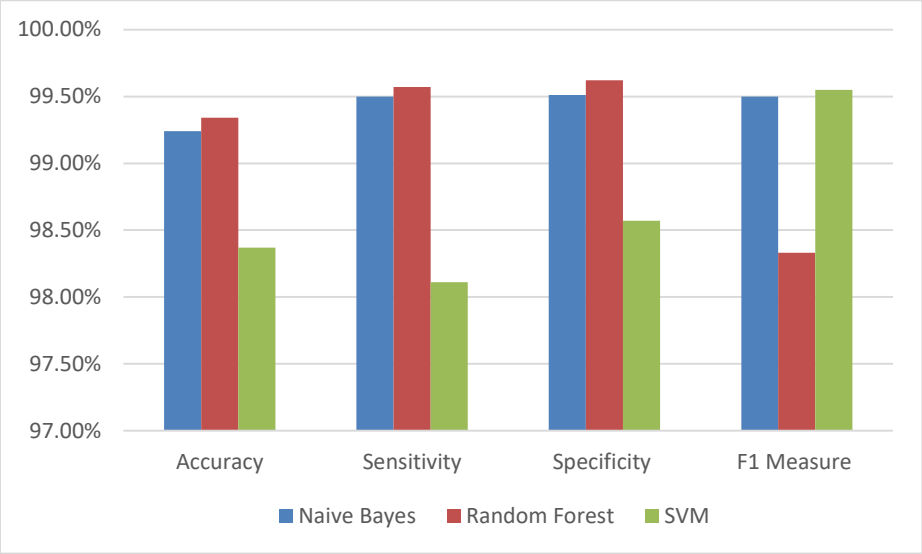

Fig 2 : Individual Specificity of Classifiers

Highlights of F1 Measure

The f-measure for each classifier is displayed in Table 6. Random forest succeeds with a greater value, i.e., 99.55%.

Table 6: F- Measure Based Comparison of Classifiers

S No	Classifier	F1 Measure
1	Naive Bayes	99.50%
2.	SVM	98.33%
3.	Random Forest	99.55%

Performance Comparison with State-of-the-Art Techniques

The results show that the proposed technique availed a higher accuracy as matched to the already developed models. Table7 presents the results of the contrast of the suggested algorithm with other multiple methods.

Table 7: Comparison of the Performance Results with Existing Research

S.No	Ref.	Classifier	Accuracy
1	(Zhang et al., 2019)	Decision Tree + K-Nearest Neighbor + Naive Bayes	97.33%
2	(Cusack, Michel, & Keller, 2018)	K-Nearest Neighbor+ Decision Tree	93%
3	(Carlin, O'kane, Sezer, & Burgess, 2018)	Support vector Machine + Decision Tree	98.5%
4	(Poudyal, Subedi, & Dasgupta, 2018)	Naive Bayes + Decision tree	97.95%
5	Proposed	Methodology	99.57%

Conclusion and Future Direction

In this study, a runtime Ransomware detection model is proposed. A pre-processed dataset gathered from Virus-total containing benign and Ransomware files is used. Benign is a type of good-ware file while Ransomware is a type of malware that encrypts a user's data until a ransom is paid to the attacker. Three different machine learning algorithms namely, i.e., Random Forest, Naive Bayes, and SVM, are utilized to detect the Ransomware and benign files in the experiments. Among the three algorithms, the modified decision tree performed well in accuracy, sensitivity, specificity, and f1-measure. Our experimental outcomes demonstrate that the presented malware classification's testing and training accuracy is at 99.57%.

References

Abri, F., Siami-Namini, S., Khanghah, M. A., Soltani, F. M., & Namin, A. S. (2019). *Can machine/deep learning classifiers detect zero-day malware with high accuracy?* Paper presented at the 2019 IEEE international conference on big data (Big Data).

Adams, C. (2018). Learning the lessons of WannaCry. *Computer Fraud & Security, 2018*(9), 6-9.

Ahmed, Y. A., Kocer, B., & Al-rimy, B. A. S. (2020). Automated Analysis Approach for the Detection of High Survivable Ransomware. *KSII Transactions on Internet and Information Systems (TIIS), 14*(5), 2236-2257.

Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2017). *A 0-day aware crypto-ransomware early behavioral detection framework.* Paper presented at the International Conference of Reliable Information and Communication Technology.

Alhawi, O. M., Baldwin, J., & Dehghantanha, A. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection *Cyber Threat Intelligence* (pp. 93-106): Springer.

Aljawarneh, S. A., & Yassein, M. O. B. (2016). A conceptual security framework for cloud computing issues. *International Journal of Intelligent Information Technologies (IJIIT), 12*(2), 12-24.

Alqahtani, A., Gazzan, M., & Sheldon, F. T. (2020). *A proposed Crypto-Ransomware Early Detection (CRED) Model using an Integrated Deep Learning and Vector Space Model Approach.* Paper presented at the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC).

Bertino, E., & Islam, N. (2017). Botnets and internet of things security. *Computer, 50*(2), 76-79.

Biddle, R., Chiasson, S., & Van Oorschot, P. C. (2012). Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR), 44*(4), 1-41.

Carlin, D., O'kane, P., Sezer, S., & Burgess, J. (2018). *Detecting cryptomining using dynamic analysis.* Paper presented at the 2018 16th Annual Conference on Privacy, Security and Trust (PST).

Cusack, G., Michel, O., & Keller, E. (2018). *Machine Learning-Based Detection of Ransomware Using SDN.* Paper presented at the Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization.

Homayoun, S., Dehghantanha, A., Ahmadzadeh, M., Hashemi, S., Khayami, R., Choo, K.-K. R., & Newton, D. E. (2019). DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer. *Future Generation Computer Systems, 90*, 94-104.

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). *Cutting the gordian knot: A look under the hood of ransomware attacks.* Paper presented at the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment.

Lee, J. K., Moon, S. Y., & Park, J. H. (2017). CloudRPS: a cloud analysis based enhanced ransomware prevention system. *The Journal of Supercomputing, 73*(7), 3065-3084.

Lei, T., Qin, Z., Wang, Z., Li, Q., & Ye, D. (2019). EveDroid: Event-aware Android malware detection against model degrading for IoT devices. *IEEE Internet of Things Journal, 6*(4), 6668-6680.

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications, 165*, 113973.

Maiorca, D., Mercaldo, F., Giacinto, G., Visaggio, C. A., & Martinelli, F. (2017). *R-PackDroid: API package-based characterization and detection of mobile ransomware.* Paper presented at the Proceedings of the symposium on applied computing.

Mehnaz, S., Mudgerikar, A., & Bertino, E. (2018). *Rwguard: A real-time detection system against cryptographic ransomware.* Paper presented at the International Symposium on Research in Attacks, Intrusions, and Defenses.

Poudyal, S., Subedi, K. P., & Dasgupta, D. (2018). *A framework for analyzing ransomware using machine learning.* Paper presented at the 2018 IEEE Symposium Series on Computational Intelligence (SSCI).

Ramesh, G., & Menen, A. (2020). Automated dynamic approach for detecting ransomware using finite-state machine. *Decision Support Systems, 138*, 113400.

Scaife, N., Traynor, P., & Butler, K. (2017). Making sense of the ransomware mess (and planning a sensible path forward). *IEEE Potentials, 36*(6), 28-31.

Song, W., Li, X., Afroz, S., Garg, D., Kuznetsov, D., & Yin, H. (2020). Automatic Generation of Adversarial Examples for Interpreting Malware Classifiers. *arXiv preprint arXiv:2003.03100*.

Teddy, J. D., Bean, J. D., Dalcher, G. W., Hetzler, J., & Woodruff, A. A. (2016). Server-assisted anti-malware client: Google Patents.

Ullah, F., Javaid, Q., Salam, A., Ahmad, M., Sarwar, N., Shah, D., & Abrar, M. Modified Decision Tree Technique for Ransomware Detection at Runtime through API Calls. *Scientific Programming, 2020*.

Victoriano, O. B. (2019). *Exposing Android Ransomware using Machine Learning.* Paper presented at the Proceedings of the 2019 International Conference on Information System and System Management.

Zaitsev, O. V. (2013). System and method for computer malware detection: Google Patents.

Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., & Sangaiah, A. K. (2019). Classification of ransomware families with machine learning based on N-gram of opcodes. *Future Generation Computer Systems, 90*, 211-221.

Zimba, A., Wang, Z., & Chen, H. (2017). *Reasoning crypto ransomware infection vectors with Bayesian networks.* Paper presented at the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI).

Zonouz, S., Rrushi, J., & McLaughlin, S. (2014). Detecting industrial control malware using automated plc code analytics. *IEEE Security & Privacy, 12*(6), 40-47.