International Journal of Integrated Engineering:
Special Issue 2018: Data Information Engineering, Vol. 10 No. 6 (2018) p. 82-88.
 DOI: https://doi.org/10.30880/ijie.2018.10.06.011

Zero-Day Aware Decision Fusion-Based Model for Crypto-Ransomware Early Detection

Bander Ali Saleh Al-rimy[1*], Mohd Aizaini Maarof[1], Yuli Adam Prasetyo[2], Syed Zainudeen Mohd Shaid[1], Aswami Fadillah Mohd Ariffin[3]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, MALAYSIA.
[2]School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia.
[3]CyberSecurity Malaysia, Seri Kembangan, Selangor 43300, MALAYSIA.

Received 28 June 2018; accepted 5 August 2018, available online 24 August 2018

Abstract: Crypto-ransomware employs the cryptography to lock user personal files and demands ransom to release them. By utilizing several technological utilities like cyber-currency and cloud-based developing platforms, crypto-ransomware has gained high popularity among adversaries. Motivated by the monetary revenue, crypto-ransomware developers continuously produce many variants of such malicious programs to evade the detection. Consequently, the rate of crypto-ransomware novel attacks is continuously increasing. As such, it is imperative for detection solutions to be able to discover these novel attacks, also called zero-day attacks. While anomaly detection-based solutions are able to deal with this problem, they suffer the high rate of false alarms. Thus, this paper puts forward a detection model that incorporates anomaly with behavioral detection approaches. In this model, two types of detection estimators were built. The first type is an ensemble of behavioral-based classifiers whereas the second type is an anomaly-based estimator. The decisions of both types of estimators were combined using fusion technique. The proposed model is able to detect the novel attack while maintaining low false alarms rate. By applying the proposed model, the detection rate was increased from 96% to 99% and the false positive rate was as low as 2.4 %.

Keywords: Crypto-ransomware; malware; anomaly detection; Cryptography; Ensemble learning

1. Introduction

Although the proliferation of the Internet and communication technology have facilitated many aspects of our daily life and the way we conduct business in the modern economy, they brought several difficulties and threats as well. Malicious software, also called malware is one of these threats that puts many systems and cyberspace resources at risk [1]. The attackers and malware developers have created many variants of malware to evade existing security measures and avoid the detection. Consequently, many types of malware such as Viruses, Trojans, Worms, and Ransomware have been seen in the wild. Motivated by the financial revenue, malware developers have introduced the extortion concept into cyberspace world by creating Ransomware. Since its first occurrence on late 1980s, Ransomware has become a major threat that compromises the accessibility, integrity and confidentiality of user and business data and resources [2]. Enabled by Ransomware-as-a-Service (RaaS), Cryptography and Cyber-currency technologies, the rate of ransomware attacks have increased dramatically in recent years [1, 3-6].

Ransomware is categorized into two main types, viz. Scareware and detrimental Ransomware [1]. The former is fake warnings that mimics anti-virus software and send false allegations to threaten the victim while latter is a real threat which leverages several system utilities to mount the digital extortion against the victims [1, 7-9]. Furthermore, detrimental ransomware is categorized into two types, i.e. Crypto-Ransomware (CRW) and Locker-Ransomware (LRW). CRW leverages the cryptography functions in the host operating system to encrypt user-related files. On the other hand, LRW locks and/or disables some operating system functions such as desktop, applications and input/output utilities [1, 7].

Several studies have been conducted to propose CRW detection solutions. Ahmadian, et al. [10] adopted the signature-based approach to detect the Domain Generation Algorithm (DGA) strings in the ransomware payload. This approach depends on the static analysis by

*Corresponding author: bnder321@gmail.com

which, the ransomware source code was introspected to extract the structural patterns, i.e. static signature that distinguish ransomware from benign programs. Likewise, Andronio, et al. [11] employed the static analysis to look for the threatening text in the payload of the program under analysis which is; if found; a strong indicator of the maliciousness of that program. Similarly, several static features like the imported libraries, functions, and file extensions were utilized by Sgandurra, et al. [12] for crypto-ransomware early detection. However, static analysis is not suitable for early detection as it inspects the payload of the ransomware without executing it whereas early detection depends on the runtime information at the initial phases of the attack[1]. Additionally, like any malware; CRW employs several obfuscation and packing techniques that changes the patterns with each infection to prevent and/or evade the detection [13] which renders the static-based detection not effective.

Kharraz, et al. [4] proposed dynamic-based detection system called UNVEIL which observes the program's behavior based on data from several sources such as CPU, memory, I/O buffer. Similarly, Song, et al. [14] used the data gathered from CPU, memory, and file events during the runtime of the process in question in Android systems and extracted the dynamic patterns that distinguish ransomware from other processes. Furthermore, honeypot approach was employed by Cabaj, et al. [15] to inspect the ransomware communication behavior and infection chain. Entropy was adopted by Kharraz, et al. [4], Mbol, et al. [16] to measure the difference in the data in I/O buffer before and after access. The higher entropy difference, the higher possibility that the data have undergone encryption which might be carried out by ransomware.

The irreversible effect of CRW attack entails that such attack needs to be detected early, i.e. before CRW starts encrypting the files. Several studies proposed early detection solution for CRW attacks. Sgandurra, et al. [12] built machine learning-based detection model for CRW early detection. A logistic regression classifier was trained by the data extracted from the first 30 seconds of CRW runtime. This data includes API calls, files, directories and registry keys operations. The same approach was adopted by Homayoun, et al. [17]. They reduced the runtime period into 10 seconds. Then the collected data was used to train ensemble of classifiers using Bagging and Random Forest. However, the detection models that have been proposed in those studies are misuse-based. That is, they have built the detection models based on CRW known patterns. Such approach fails when encountering new attacks whose patterns were not known to the detection model. These attacks are called novel or zero-day attacks. In the same time, accounting on anomaly detection only renders the detection solution vulnerable to high rate of false alarms. Therefore, there is a need for detection solutions that detect the novel attacks while maintaining low rate of false alarms. To the best of our knowledge, there is no study that tackles such integration for CRW detection. To this end, this study fills this gap and puts forward the integration between the anomaly and behavioral approaches for CRW detection. By incorporating the behavioral-based approach into the anomaly-based model, the proposed solution addresses the limitation of the anomaly detection-based model and decreases the false alarms rate. In our previous study, Al-rimy, et al. [3], we proposed a framework for the integration between anomaly and behavioral approaches to detect CRW attacks. In this study, we extend that work and build an early detection solution by utilizing the APIs generated during the first five second of CRW runtime period and build a hybrid detection solution that combines the anomaly and behavioral models for CRW early detection. To enhance the detection performance, the proposed solution employs ensemble learning techniques to train group of classifiers on different data subsets. In addition, the anomaly-based estimator was trained on the entire benign programs dataset and fused into the ensemble using OR logic. To the best of our knowledge, this is the first study that introduces the integration between the anomaly and behavioral approach for CRW early detection. Such integration guarantees that the proposed detection model is able to detect novel CRW attacks while maintaining low false alarms rate. The rest of this paper is organized as follows. In section 2 methodology of this study is detailed. Results and discussion are presented in Section 3. Then, the paper is concluded by Section 4.

2. The Methodology

The proposed model combines two detection approaches, behavioral and anomaly detection. The behavioral detection was built using the ensemble learning. That is, several classifiers were trained on the behavioral dataset and constitute the base estimators. The decision of those estimators were combined using the majority voting strategy. On the other hand, the anomaly detection model was built and trained on the benign dataset. Unlike the behavioral detection model, the anomaly detection model is a single classifier. The combination between the decision of both behavioral and anomaly based detection models was carried out using the decision fusion technique. This section elaborates the methodology employed by this study and discusses the design of the proposed model.

2.1 Preprocessing

A corpus of crypto-ransomware samples downloaded from virsushare.com public repository was used to conduct the experiments of this study. Additionally, benign programs were downloaded from informer.com, a well-known windows applications repository [13, 18, 19]. Then, both types of programs, i.e. ransomware and benign were undergone dynamic analysis in a controlled environment. The analysis environment was built using Cuckoo Sandbox, a well-known and widely-used analysis platform [20] by which each instance was run for 5 seconds. Then, the runtime data for all instances were

dumped into trace files such that each instance has its own trace file. Fig. 1 shows the flow pre-processing phase.

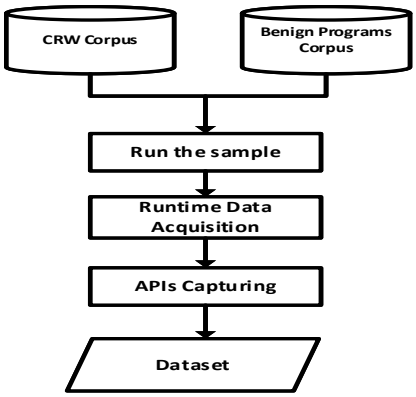

Fig. 1: Pre-processing phase.

The data includes Application Programming Interface API calls that were used by the running program to interact with the underlying operating system along with the timestamp that represent the time by when these APIs were called. Then, the API calls of each trace file were sorted ascendingly, i.e. from the newest to the oldest according to the time stamp.

2.2 Features Extraction

To extract the features that represent the dataset, 1-gram, a variant of the commonly used n-gram technique was employed. That is, each API was treated as one feature. These features are called API-gram features. Then, the Term Frequency-Inverse Document Frequency (TF-IDF) was utilized to calculate the weight of each API [21] based on (1).

$$w(x_k) = tf(x_k) \cdot \log \frac{N}{idf(x_k)} \quad (1)$$

$w(x_k)$ is the API weight, N is the total number of instances in the dataset, x_k is the k^{th} feature in the instance, *tf* calculates frequency the feature x_k that was called by a particular instance and *idf* determines how many instances called a that specific feature x_k at least once. For each instance, TF-IDF built a feature vector V_i which were used as input for the learning algorithm at both training and testing phases.

2.3 Features Selection

The usage of n-gram technique to extract the features generates a high dimensional features space which affects the estimators' accuracy and renders them vulnerable to overfitting. As such, the Mutual Information (MI) was adopted to select the distinguishing features of the dataset. MI utilizes the entropy measurement to assess the amount of information each feature carries about the class label. The calculation of entropy was carried out using (2).

$$H(x_i) = -\sum_{i=0}^{N} p(x_i) \log_2 p(x_i) \quad (2)$$

where $p(x_i)$ is the probability density function (PDF) of the feature x_i and N is the number of features. Then, MI was calculated according to (3).

$$MI(X;Y) = H(Y) - H(Y \mid X) \quad (3)$$

2.4 Training/Testing

After extracting the API-gram features and calculating TF-IDF weights, the dataset was split into two parts, i.e. training set and testing set. The size of raining set is 70 % and testing set is 30%. In addition, the benign set which contains data from benign programs only was created as well. The training set was used to build the estimators of module 1 while the benign set was used to build the anomaly-based estimator. After training, the performance of base estimators as well as entire ensemble was validated using the 10-fold cross-validation technique.

2.5 Designing the Proposed Model

To boost the detection accuracy, the proposed ensemble consists of *n* estimators each of which was trained on different data subset. These subsets were selected according to bootstrap aggregation (bagging) technique. That is, the subsets were built by randomly choosing group of instances with replacement for each subset. This leads to different data subsets and thus, different models [22]. The number of instances are equal for all subsets. Additionally, the sampling is stratified, i.e. the benign-to-ransomware ratio of each subset is maintained and consistent with that of the original dataset. Then, each data subset was used to train one base estimator.

A) Ensemble-Based Behavioral Model (BE)

In this type, one classification algorithm was used to build homogenous base estimators. Particularly, Support Vector Machine (SVM) was employed to build these estimators as it proved its efficacy and powerful generalization capability in many machine learning tasks including malware detection [23, 24]. For binary classification, SVM constructs a hyperplane which maximizes the margin that separates the two classes. Intuitively, a good separation is achieved by finding a hyperplane with the largest functional margin, i.e. the distance to the nearest data points of any class [25]. As such, the generalization error is inversely proportional to the functional margin. For such separation, SVM employs the kernel function technique that transforms the original data into high dimensional feature space [26]. The formula (4) is used for SVM classification [27]:

$$h(x) = sign\left(\sum_{i \in SV} \alpha_i y_i K(x_i, x) + b \right) \quad (4)$$

where SV are the support vectors, $K(x_i, x)$ is the kernel function, x is the features vector of the input sample, x_i

the i^{th} feature in the vector x, α is the lagrange multiplier that determines the parameters w, b of the maximal margin classifier. The detection result *P(x)* is ransomware (*rw*) if the *h(x)* is positive, i.e. *h(x) > 0*, and benign otherwise as illustrated by (5).

$$P(x)=\begin{cases} rw, & h(x)<0 \\ benign, & h(x)\geq 0 \end{cases} \quad (5)$$

The decision of SVM base estimators were combined using majority voting scheme as it is simple and straight forward. Particularly, each base estimator got one vote, either rw or benign. These votes were calculated and the class label that have at least one more than half votes won. Fig. 2 illustrates the design of the ensemble. Testing error curve was utilized to determine the number of estimators such that the detection error is minimum. Due to space limitation of this paper, we do not elaborate on the process of choosing the optimal number of base estimators. However, it turned out that n=3 gives comparable detection rate.

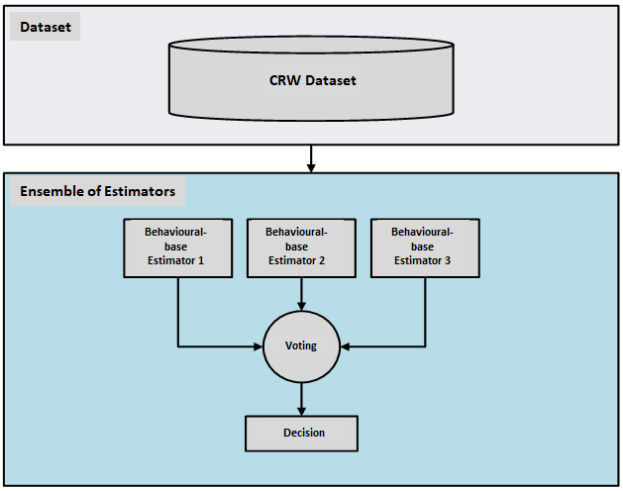

Fig. 2: Behavioral-based ensemble model

B) Anomaly-Based Estimator

To detect zero-day, i.e. novel ransomware attacks; an anomaly based estimator were built using One-Class SVM (OC-SVM) algorithm. In this algorithm, the training data has only one type of samples. These samples were benign programs such that the estimator represents the normal behavior. Any deviation from the normal reference is considered anomaly, i.e. ransomware. OC-SVM determines a hyper sphere of minimum volume that covers all the training points [28]. Unlike CRW behavior that could be separated into multiple regions based on the malicious family which make it easy to build multiple classifiers on smaller subsets, the boundaries of the behavior of benign programs have no such separation. As such, we opted to represent these benign programs by only one anomaly estimator.

C) Decision Fusion

The decision of the anomaly estimator is then fused with the vote result of the homogenous ensemble using OR logic. That is, if any or both decisions was ransomware, then the final decision is ransomware. Otherwise, the decision is benign as (6).

$$D(m_1,m_2)=\begin{cases} 1, & any(m_1,m_2)=1 \\ 0, & otherwise \end{cases} \quad (6)$$

m_1, m_2 are the homogenous ensemble and anomaly estimator respectively. $D(m_1, m_2)$ is the decision fusion. Fig. 3 shows the pseudo code for decision fusion. Fig. 4 illustrates the combination of the ensemble model with the anomaly estimator.

3. Results and Discussion

In this section, we evaluate the proposed method experimentally. The proposed method was evaluated against four performance metrics, i.e. accuracy, F1-measure (F1), Detection Rate (DR) and False Positive (FP) alarms. Furthermore, a comparison was conducted between the proposed method and two base-line methods, i.e. Logistic Regression and SVM. In addition, the comparison was carried out between the proposed method before fusion (BE) and after fusion (FDE).

Decision Fusion

Input: CRW Dataset
API calls generated by the running process $C = \{c_1, c_2, \ldots\ldots, c_n\}$
Output: *Decision*

```
1. Buffer ← C
2.Decision ← Decision(ensemble)
3. if    Decision is CRW
5.           end
6.       Otherwise
7.           if Decision(anomaly) is Anomaly:
8.                Decision is CRW
9.                end
10.          else:
                  Decision is benign
12.               end
```

Fig. 3: Pseudo Code of the Decision fusion

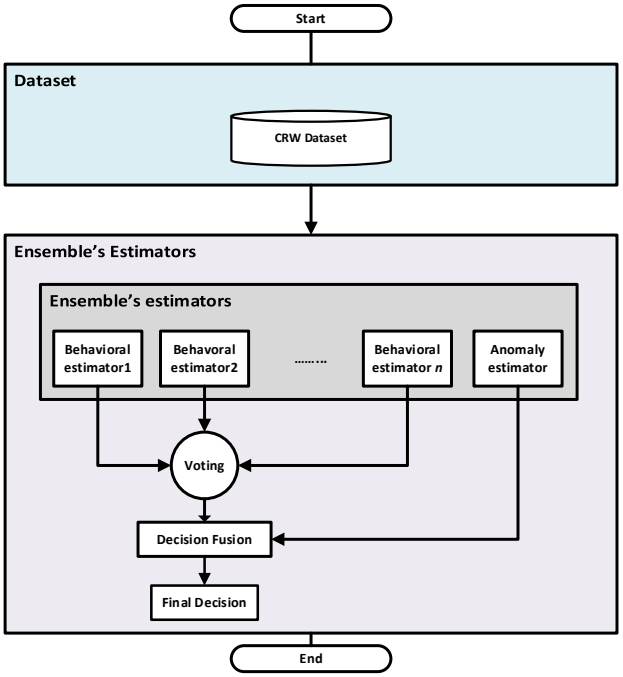

Fig. 4: The combination of behavioral and anomaly estimators in the ensemble.

3.1 The Experimental Environment Setup

The experiments were conducted on an Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHZ and 16 GB RAM. The Cuckoo Sandbox analysis platform were installed on Linux ubuntu 4.4.0-59-generic with MS Windows XP Professional SP3 guest machine. The ensemble estimators and results analysis were implemented using Python libraries including Sklearn, Pandas and Numpy. CRW and benign programs were run in one by one. After each run, the gust machine was restored into the original, clean state. Extracted data was gathered and the features were extracted and selected during the preprocessing phase. Once ready, the dataset was used to train the detection model.

3.2 Dataset

The corpus of crypto-ransomware binaries used in this study were downloaded from virusshare.com public repository [12, 13, 18, 19]. The corpus consists of 38,152 samples. These samples represent different families such as Cerber, TeslaCrypt, CryptoWall, Petya and WannaCry. Those samples were collected during the period from Sep 2016 to Aug 2017. In addition, 1000 benign programs were downloaded from informer.com [12, 19, 29, 30], a popular Windows-based applications repository. For the purpose of this study, the collected ransomware corpus was divided into two sets, training set and holdout set. The sampling was carried out so that the training set contains 90% of the samples whereas holdout set contains 10%. The training set was used to train, validate and test the performance of the detection model using 10-fold cross validation. Holdout set contained crypto-ransomware samples that have not been previously seen or used in any training-validation-testing process at the time the model was built. So, crypto-ransomware instances in the holdout represent the zero-day attacks. The purpose of such division is to determine how accurate the detection model in detecting the new ransomware samples. Then, both ransomware and benign programs were run in the sandbox. After submitting the sample to the analyzing machine, the sandbox agent in the guest machine hooks the process created by that sample and captures the APIs along with the parameters and dumps them into a trace file in the host machine specified for that sample. These files constitute the corpus by which the dataset was built and the features were extracted and selected.

3.3 Experimental Results

Fig. 5 shows that the FDE method surpassed the other three methods. Similarly, Table 1 shows that FDE outperformed the homogenous ensemble in three measurements, i.e. accuracy, DR and F1 while BE gave lower FP rate than FDE. Equations (7,8,9,10) calculate the F-Measure (F1), accuracy (acc), DR and FP measurements respectively.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$DR = \frac{TP}{TP + FN} \tag{9}$$

$$FPR = \frac{FP}{FP + TN} \tag{10}$$

From the results, it can be observed that using ensemble of estimators enhanced the detection performance in terms of the detection accuracy, F1-measure, detection rate and false positive rate. Additionally, fusing the anomaly-based approach with the behavioral-based approach increased the detection rate from 0.96 to 0.99. However, the false positive rate of the FDE method increased. The reason is that the anomaly-based estimator generated high false alarms which generally one of the main limitations of this approach. Such high rate of false alarms is because the anomaly approach builds the normal profile based on the behavioral aspects of the benign programs. However, the normal behavior is so diverse due to the large number of benign applications which is difficult to include in single profile. Therefore, the anomaly based detection suffers the high rate of the false alarms as mentioned in Section 1.

To measure the superiority of the ensemble-based method (BE) to detect zero-day attacks, the detection performance of the method was compared with the anomaly-based model in terms of F1-measure, accuracy, precision and recall. The experiments were carried out using the holdout set. Table 2 suggests that the BE outperformed the anomaly-based model in detecting the

zero-day attacks thanks to the high accuracy of the ensemble learning approach due to the diversity of its base estimators. Such diversity was achieved by using heterogeneous base estimators when building the detection model. To determine the degree of significance of the enhancement, t-test was conducted with a threshold adjusted at the standard value 0.05. The p-value was 0.0473 which is less than the threshold. This suggests that the improvement was significant.

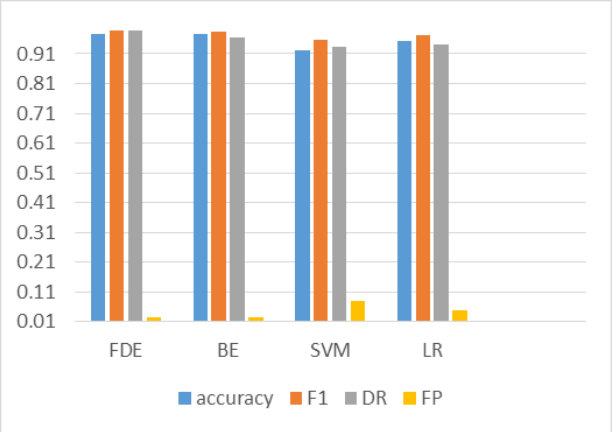

Fig. 5: Comparison between the proposed model with related work.

Table 1: Comparison between the detection performances of the proposed model and the behavioral approach.

	FDE	BE
Accuracy	**0.97717**	0.9757489
F1	**0.986971**	0.986122
DR	**0.99**	0.9638
FP	0.0242	**0.02265**

Table 2: Comparison between the detection performances of the proposed model and the anomaly approach.

	BE	Anomaly
F1	0.995066	0.942771
Accuracy	0.991441	0.891738
Precision	0.993432	1
Recall	0.996705	0.891738

4. Conclusion

In this paper we proposed an ensemble fusion method for crypto-ransomware early detection. This method combines estimators of anomaly and behavioral approaches to detect novel attacks and maintaining low rate of false alarms. The proposed method combines the homogenous estimators using majority voting while fuses anomaly and behavioral estimators using OR logic. This method achieved 99% detection rate and 2.4% false positive alarms on a real world dataset with more than 12000 applications. These results show that the proposed method is effective crypto-ransomware early detection solution. Such model can be applied on different platforms like PC, mobile and IoT devices. One limitation of the proposed model is the reliance on fixed time-based threshold to define the early phase of ransomware attacks. To overcome such limitation, we are currently working on building detection technique that defines the early phase of the attack dynamically based on the behavioral information of the crypto-ransomware sample.

References

[1] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, vol. 74, pp. 144-166, 2018/05/01/ 2018.

[2] J. A. Gómez-Hernández, L. Álvarez-González, and P. García-Teodoro, "R-Locker: Thwarting ransomware action through a honeyfile-based approach," *Computers & Security*, vol. 73, pp. 389-398, 2018/03/01/ 2018.

[3] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "A 0-Day Aware Crypto-Ransomware Early Behavioral Detection Framework," in *Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017)*, F. Saeed, N. Gazem, S. Patnaik, A. S. Saed Balaid, and F. Mohammed, Eds., ed Cham: Springer International Publishing, 2018, pp. 758-766.

[4] A. Kharraz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," 2016.

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

[6] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," in *12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2015* vol. 9148, F. Maggi, M. Almgren, and V. Gulisano, Eds., ed: Springer Verlag, 2015, pp. 3-24.

[7] P. Pathak and Y. M. Nanded, "A Dangerous Trend of Cybercrime: Ransomware Growing Challenge," 2016.

[8] K. Cabaj and W. Mazurczyk, "Using Software-Defined Networking for Ransomware Mitigation: The Case of CryptoWall," *Ieee Network*, vol. 30, pp. 14-20, Nov-Dec 2016.

[9] J.-L. Richet, "Extortion on the Internet: the Rise of Crypto-Ransomware," 2015.

[10] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, "Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares," in *12th International ISC Conference on Information*

Security and Cryptology, ISCISC 2015, 2015, pp. 79-84.

[11] N. Andronio, S. Zanero, and F. Maggi, "HELDROID: Dissecting and detecting mobile ransomware," in *18th International Symposium on Research in Attacks, Intrusions, and Defenses, RAID 2015* vol. 9404, H. Bos, G. Blanc, and F. Monrose, Eds., ed: Springer Verlag, 2015, pp. 382-404.

[12] 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.

[13] C. Le Guernic and A. Legay, "Ransomware and the Legacy Crypto API," in *Risks and Security of Internet and Systems: 11th International Conference, CRiSIS 2016, Roscoff, France, September 5-7, 2016, Revised Selected Papers*, 2017, p. 11.

[14] S. Song, B. Kim, and S. Lee, "The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform," *Mobile Information Systems*, vol. 2016, 2016.

[15] K. Cabaj, P. Gawkowski, K. Grochowski, and D. Osojca, "Network activity analysis of CryptoWall ransomware," *Przeglad Elektrotechniczny*, vol. 91, pp. 201-204, 2015.

[16] F. Mbol, J.-M. Robert, and A. Sadighian, "An Efficient Approach to Detect TorrentLocker Ransomware in Computer Systems," in *Cryptology and Network Security: 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, S. Foresti and G. Persiano, Eds., ed Cham: Springer International Publishing, 2016, pp. 532-541.

[17] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[18] J. B. Christensen and N. Beuschau, "Ransomware detection and mitigation tool," 2017.

[19] Z.-G. Chen, H.-S. Kang, S.-N. Yin, and S.-R. Kim, "Automatic Ransomware Detection and Analysis Based on Dynamic API Calls Flow Graph," presented at the Proceedings of the International Conference on Research in Adaptive and Convergent Systems, Krakow, Poland, 2017.

[20] 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*, vol. 81, pp. 1012-1026, 9// 2015.

[21] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, pp. 1-47, 2002.

[22] X. Yang, D. Lo, X. Xia, and J. Sun, "TLEL: A two-layer ensemble learning approach for just-in-time defect prediction," *Information and Software Technology*, vol. 87, pp. 206-220, 2017/07/01/ 2017.

[23] S. M. Hosseini Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90-102, 7/26/ 2016.

[24] S. Das, Y. Liu, W. Zhang, and M. Chandramohan, "Semantics-Based Online Malware Detection: Towards Efficient Real-Time Protection Against Malware," *Ieee Transactions on Information Forensics and Security*, vol. 11, pp. 289-302, Feb 2016.

[25] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," in *Advances in neural information processing systems*, 1993, pp. 147-155.

[26] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "A Survey on Malware Detection Using Data Mining Techniques," *ACM Comput. Surv.*, vol. 50, pp. 1-40, 2017.

[27] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, and J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection," *Future Generation Computer Systems-the International Journal of Escience*, vol. 55, pp. 376-390, Feb 2016.

[28] W. L. Shang, P. Zeng, M. Wan, L. Li, and P. F. An, "Intrusion detection algorithm based on OCSVM in industrial control system," *Security and Communication Networks*, vol. 9, pp. 1040-1049, Jul 2016.

[29] A. Ioanid, C. Scarlat, and G. Militaru, "The Effect of Cybercrime on Romanian SMEs in the Context of Wannacry Ransomware Attacks," in *12th European Conference on Innovation and Entrepreneurship ECIE 2017*, 2017, p. 307.

[30] S. K. Pandey and B. M. Mehtre, "Performance of malware detection tools: A comparison," in *2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014*, 2015, pp. 1811-1817.