IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-2 E-ISSN: 2347-2693

Polymorphic Malware in Executable Files and the Approaches towards their Detection and Extraction

Faiz Baothman[1], Muzammil H Mohammed[2*]

[1]Dept. of Computer Science, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia
[2*]Dept. of Information Technology, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

**Corresponding Author: m.muzammil@tu.edu.sa*

Received: 23/Jan//2018, Revised: 05/Feb2018, Accepted: 14/Feb/2018, Published: 28/Feb/2018

Abstract- The malwares which are present with subtle with polymorphic techniques like self-mutation and emulation based mostly analysis evasion. Most anti-malware techniques are engulfed by the polymorphic malware threats that self-mutate with completely different variants at each attack. This analysis aims to contribute to the detection of malicious codes, particularly polymorphic malware by utilizing advanced static and advanced dynamic analysis for extraction of a lot of informative key options of a malware through code analysis, memory analysis and activity analysis. Correlation based mostly feature choice rules are rework features; i.e. filtering and choosing best and relevant options. A machine learning technique known as K-Nearest Neighbor (K-NN) are used for classification and detection of polymorphic malware analysis, results are supported the subsequent measuring metrics— True Positive Rate (TPR), False Positive Rate (FPR) and therefore the overall detection accuracy of experiments.

Keywords: Malware Detection, Static Analysis, Dynamic Analysis, Polymorphic Malware, Machine Learning

I. INTRODUCTION

Presently the planet depends on info technology (IT) because it facilitates human daily activities. Multiple devices like personal computers, laptops, tablets, etc., have gained quality once used for accessing IT. Such devices widely employed in offices, homes, etc., for multiple services. However, there's an excellent concern relating to security within the use of IT. Plenty of malware like rootkits, spyware, trojan horses, bots and alternative sorts discharged by attackers. In line with the Symantec report, there have been 317 million items of malware injected in year 2014, which suggests that just about new threats were created on a daily basis. Several developers have tried to beat this case through creation of anti-malware programs— like Symantec antivirus, Lavasoft [1] and plenty of others. However, these anti-malware have quite restricted potency in distinguishing and eliminating threats [2]. This gap has attracted a lot of analysis interest within the space, particularly on malware analysis to realize new reliable and more brilliant algorithms. Malware have become a lot of subtle with polymorphic behaviors [4], [5], [6] so as to cover themselves from analysis and detection. Polymorphism is that the capability of the malware to vary identity at any instance of infection. It's not a replacement malware, however it's a variant of existing malware that is packed and contains some code obfuscation. These variants of existing malware can confuse anti-virus and may then be detected as benign owing to the shortage of applicable signatures to contain them. An enormous downside that arises is a way to expeditiously traumatize such polymorphic malware. Previous researchers have met variety of challenges in addressing this issue. Most planned solutions are hoping on extracting activity options from malware and use totally different machine learning strategies to implement detection approaches. It's in this context that this analysis aims at planning a completely unique approach in terms of feature engineering and detection mechanisms. This approach can integrate advanced elements of 2 powerful analysis techniques for a comprehensive malware dissection and have extraction method. These techniques referred to as advanced static and advanced dynamic analyses. Structural and activity options are extracted a Machine learning technique referred to as K-NN are employed in the method of planning or implementing a detection approach. The objectives are to realize high detection accuracy that considerably reduces false alarms and will increase the speed of properly detected malware and outperforms previous approaches. The study can primarily concentrate on malicious transportable Possible (PE) Executable files. The letter of the alphabet file format may be an arrangement that contains necessary info for the OS loader to manage possible code [7].

II. RELATED WORK

The analysis related to Malware helps to look at the capabilities of a worm so as to raise to investigate the character of security breach incident and interference of any more infections [7]. There are two measures normally used

for malware analysis techniques, i.e. static analysis [7] and dynamic analysis [7].

Relates to Static analysis [8], [6] may be a method whereby data regarding worm is extracted while not being dead. Non execution of the malicious code makes static analysis safer compared to dynamic analysis during which malicious code should be dead on the machine used for analysis [7]. Basic static analysis will show basic data regarding the worm like its version, file size, file format, any suspicious imports, etc. Basic static analysis is easy and fast, however not terribly effective as vital details may be uncomprehensible [7]. Advanced static analysis deals with code/structure analysis during which the data of programming language, compiler code and software ideas square measure needed [7]. Malware practicality is analyzed through inspecting the inner code of the malware [5].

Next Dynamic analysis [8], [9] is that the method of analyzing a worm through execution and monitor its run time practicality of such an execution. Basic dynamic analysis consists of perceptive the behavior of a malware and doesn't need deep programming skills whereas the advanced dynamic analysis makes a profound examination of the inner state of a running worm whereas extracting elaborate data [7]. The code is analyzed at run time and any code hidden through packing is unconcealed [10]. The identity of a malware is programmatically known as operate calls, parameter analysis and data flow square measure all unreal [5]. The analysis on malware variants or polymorphic malware relies on activity analysis during which malware functionalities square measure investigated at run time [9]. And developed a technique to notice malicious files supported activity ordered patterns during which the behavior of malicious executables were analyzed. API calls were extracted and a log was created. The repetitive patterns within the API decision log were thought of to create the initial dataset for classification. The Fisher score algorithmic rule was used for feature choice in their analysis whereas support vector machines was combined with call tree algorithms and used for malware detection. The coaching dataset contained 806 malware and 306 benign files. A malware detection accuracy of ninety fifth was achieved. Cesare et. al [10] detected new malware samples and variants of existing ones through generating signatures for any fresh known malware. It handles unpacking. The sample consisted of 15409 malware out of that, their results showed eighty eight.26% were classified as variants of existing ones and thirty four% were classified as famed malware. The combined static and dynamic analysis in [6] was done on a malicious file referred to as TT.exe that breaks into a system and performs malicious activities. The benefits of mixing each ways are found to be on the far side preliminary analysis as a malware will deeply be compound to reveal a lot of its functionalities. Classification of analysis supported machine learning [11] targeted on malware classification and clump. 1270 malware samples of various format, namely; pdf, executables, html, zipped, jpeg, etc. were investigated. Logic Model Tree and K-Means algorithms were used for the task of classification and clump severally. The results show that eighteen of analyzed malware were embedded with networking capabilities to attach to the outer world, whereas eighty two aimed to corrupt the system domestically or network resources. Malware were additionally sorted with success consistent with their file format sorts. Comar et. al in [11] combined supervised and unattended learning methodology to capture packets from a live network affiliation and use the data of existing attacks to classify new network flow as either new attack, existing or variants of existing. K-Nearest Neighbor (K-NN) and Support Vector Machine (SVM) algorithms are utilized in the classification method. 216,899 flows are captured, out of that four,394 (2%) were found malicious and classified in thirty eight famed malware categories.

Liang et. al, in [12] planned a completely unique methodology to notice variants supported activity dependency. options were extracted exploitation Temu dynamic analysis code and were be spoken noise removal. In their analysis, Jaccard algorithmic rule was used for similarity calculation. Their experiments were done exploitated completely different malware and 6 variants of a Trojan malware referred to as Ghost. Results showed that the 2 completely different malware had a weighted similarity of twenty seventh, whereas the six variants had a robust weighted similarity starting from eighty six% to 96.2%.Naidu et. al, in [13] planned a way that mechanically generates super- signatures to contain polymorphic malware. They used hex characteristics as options furthermore as Needleman-Wunsch and Smith-Waterman algorithms for string matching. Experiments were done on multiple variants of malware "JS.and Cassandra" detection rate was ninety six%. Table one below, discusses regarding completely different techniques furthermore as their strengths and limitations.

TABLE 1: Detection Techniques Comparison.

Technique	Characteristics	Strengths/Contribution	Limitations
Malware detection by behavioral sequential patterns][13]	-Uses API calls based features. - Random forest and SVM are used for classification	-Effective in detecting malware variants.	-Static features not considered. -High rate of False Positive detections

Malicious data classification using structural information and behavioral specifications in executables[8]	-Uses common static and API call features -J48 algorithm is used for classification	-Can detect similarities among malware samples	-can't handle analysis features
A Behavior-Based Malware Variant Classification Technique[p70]	-API calls based features are used. -Weighted similarity among malware behaviors is calculated using Jaccard similarity algorithm	-Effective at detecting similarity among malware variants	-Static features not considered. -High rate of False Positive detections
Combining supervised and unsupervised learning for zero-day malware detection[14]	-Network flow based features are extracted using IDS/IPS -Uses one class SVM algorithm for classification.	-Effective at detecting polymorphic. -Can detect new malware from a suspicious flow	-Limited to network based features - High rate of False Positive detections
Needleman-Wunsch and Smith-Waterman Algorithms for Identifying Viral Polymorphic Malware Variants[12]	-static Hexadecimal based features are extracted. - Needleman-Wunsch and Smith-Waterman Algorithms are used for creating effective signatures.	-Effective at generating appropriate signatures to contain polymorphic malware.	-only static features are considered. -Can have false positive detections
Proposed solution: Integrated Feature Extraction Approach towards Detection of Polymorphic Malware in Executable Files	Comprehensive dissection of malware using Advanced static analysis and advanced dynamic analysis as discussed in Table 2 and 3. Correlation based feature selection (CFS) algorithm. CFS helps in creating good feature subsets that are highly correlated with the predicted class. This method is chosen because it is fast, produces high ranking and correlated features compared to alternative methods used in other techniques. As we'll have a b set, the accurate automated selection is also well done with CFS. K-NN classifier to detect polymorphic malware with high accuracy. Comparing to other methods used in previous methods, K-NN is selected due to its good performance and robustness in dealing with large datasets with many features[15].	This method will address the limitations of previous techniques by developing an approach with the following components: -Detection of polymorphic malware with high accuracy. -Significantly minimized false detection alarms. -Consideration of hidden malware functionalities, especially analysis/detection avoidance capabilities. -Increased detection performance	

III. METHODOLOGY

3.1 Proposed Detection Approach

The detection approach is illustrated by the flow diagram in figure one. A malware sample is analyzed by advanced dynamic analysis and advanced static analysis. Dynamic analysis results in the extraction of behavioral options. For static analysis a sample is 1st investigated to spot packing traces. If it's packed structural options square measure extracted. All options square measured combined to create

a giant feature dataset. These options can then be filtered to pick out a reduced dataset that includes most optimum options that measure for relevant for classification task. Lastly, the classification method are done supported antecedently preprocessed options so as to discover polymorphic malware.

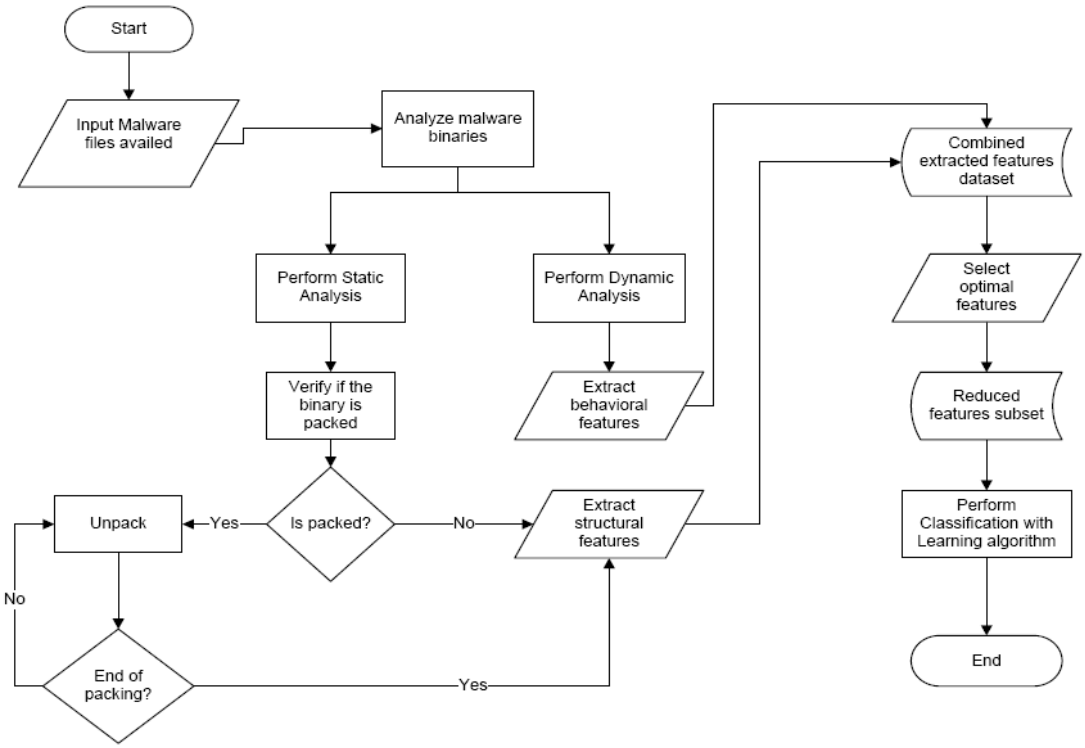

FIGURE 1: The Detection Approach Flowchart.

3.2 Extraction of options from Malware knowledge Samples Malware related samples for researchers are collected from on-line repositories [17] such as— Open Malware, Malware repository, Malware Samples[17].Having nonheritable relevant malware samples, the analysis seeks to own as a lot of descriptive info as doable a few given malware through feature extraction. Options for identification characteristics of malware won't to build the detection data. To extract these options this analysis can use a mixture of advanced static and advanced dynamic analysis techniques. Tools to be used are shown in table two and therefore the main options to be extracted as shown in table two.

TABLE 2: Tools and Their Characteristics.

Tools	Description
PEiD and UPX	For identifying packer and compiler information and regeneration of original unpacked file.
IDA pro	For disassembling the malware binary for further analysis
Process Monitor	For viewing real-time file system, process activity and registry, Network activity, API calls, Mutex, Self-modifying code traces
Dependency Walker	For exploring the Dynamic Link Libraries (DLL) and imported functions.
Regshot	To capture and compare registry snapshots to discover any modifications
ApateDNS	For controlling DNS requests and response in case of malware network activity
Wireshark:	For capture and analyze network traffic
INetSim	for simulating network services such as DNS, HTTP, HTTPS, FTP, IRC, DNS, SMTP

Not all options extracted are relevant for this analysis. Therefore, once extracting options, some options with low impact are removed as a result of they may have a negative impact on the general accuracy. Techniques like Fisher score algorithmic rule [8], correlation primarily {based} algorithmic rule [15] and tree based algorithmic rule [18] square measure smart at feature choice method. Fisher score algorithmic rule selects high ranking options and tree-based feature transformation approach selects and removes noise from information. Correlation based mostly feature choice (CFS) algorithmic rule helps in making smart ranking feature subsets that square measure extremely correlative with the expected category, particularly just in case of huge feature dataset. CFS is so chosen to be used for this analysis. Most relevant options are maintained and can be candidate instances of the coaching dataset. Relevant options can then type Associate in Nursing best feature set to be employed in classification. The benefits of feature choice include: reduced overfitting (avoiding the worst case situation in prediction), important reduction of coaching time and improved accuracy. CFS algorithmic rules are used for choosing best options. The advantage of a feature set S with k options is computed in step with equation one.

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} \quad (1)$$

Where the average is value of feature classification correlations, and is the average value of feature-feature correlations.

CFS will finally be computed according to equation 2

$$CFS = \max_{S_k}\left[\frac{r_{cf_1}+r_{cf_2}+\cdots+r_{cf_k}}{\sqrt{k+2\left(r_{f_1f_2}+\cdots+r_{f_if_j}+\cdots+r_{f_kf_1}\right)}}\right] \quad (2)$$

3.3 Designing the Detection

This task can principally include building classification models which will optimally generalize the predictions in detective work polymorphic malware. The selection of a classifier depends on the kind of options, dataset size and additionally the matter to be resolved [16]. Classifiers like call Trees (DT) [16], Support Vector Machines (SVM) and K-nearest neighbor (K-NN) perform well in several things [16]. SVM and K-NN ar appropriate for this analysis as they'll support similarity operate testing for prediction. SVM is appropriate to figure with few knowledge points as a result of its slow [16]. K-NN is good for several knowledge points and it's quick [16]. Therefore, to perform classification, the analysis proposes to use K-nearest neighbor (K-NN) formula [19] as a result of it's the power to work out similarities among instances. K-NN are enforced and customised to satisfy the challenges of classification. Distances are calculated between the targeted instance and every one alternative instances. The shortest distance shows the strongest similarity. Once there's a powerful similarity, it's implies that there are variants in instances [19]. These variants are signs of polymorphism. Nearest neighbors will be computed as follows:

1. Let $x_i^{j)}$ represents all training examples, where i is the number of features and j is the number of instances.

2. Let k be the number of nearest neighbors determined beforehand in building (K-NN) model,

$$dist\left(x_i^{(b)}, x_i^{(j)}\right) = \sqrt{\sum_{a=1}^{i}(x_a^{(b)} - x_a^{(j)})^2}, \text{ where } a \le i \text{ and } b \le j$$

3.4 Evaluation and Validation

To evaluate the results, main performance metrics particularly True positive (TP), False positive (FP), True negative (TN), and False negative (FN) are going to be calculated. True Positive rates (TPR) can offer the proportion of properly known as polymorphic samples. False Positive Rates (FPR) can offer the proportion of incorrectly known as polymorphic samples. the accuracy of the model are going to be calculated supported total variety within the sample and people that were properly detected as shown in equation vi. Performance metrics measurement calculated as follows:

$$TPR = \frac{TP}{TP+FN}$$

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

Overall accuracy is the proportion of the total number of predictions that are correct and will be computed as follows:

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

IV. Result Analysis

The expected outcome may be a polymorphic malware detection approach that will increase overall detection performance in terms of accuracy and speed. Accuracy is measured by the high rate of malware properly known as polymorphic in addition as considerably decreased rate of false detection alarms. Detection speed are high as a result of the optimized feature engineering method.

V. CONCLUSION

The analysis desires to address the problem of polymorphic malware detection. This may be done by collection of malware samples, analyzing them and extract options victimization advanced static and advanced dynamic analyses techniques. Feature choices are going to be done victimization Correlation Feature choice formula. Classification are going to be done victimization machine learning technique known as K-NN. Analysis of detection performance are going to be done supported mensuration overall accuracy, true positive rate additionally as false negative rates. Future work on the implementation of the planned approach and supply simulation results; and on the customization of various machine learning algorithms for a lot of optimized higher detection rates.

REFERENCES

[1] Lavasoft, "Detecting Polymorphic Malware." [Online]. Available: http://www.lavasoft.com/mylavasoft/securitycenter/whitepapers/detecting-polymorphic-malware. [Accessed: 01-Sep-2016].

[3] A. Sharma and S. K. Sahay, "Evolution and Detection of Polymorphic and Metamorphic Malwares: A Survey," International Journal of Computer Applications, vol. 90, no. 2, pp. 7–11, 2014.

[4] S. K. Pandey and B. M. Mehtre, "A lifecycle based approach for malware analysis," Proceedings - 2014 4th International Conference on Communication Systems and Network Technologies, CSNT 2014, pp. 767–771, 2014.

[5] Y. Prayudi and S. Yusirwan, "the Recognize of Malware Characteristics Through Static and Dynamic Analysis Approach As an Effort To Prevent Cybercrime Activities," Journal of Theoretical and Applied Information Technology (JATIT), vol. 77, no. xx, pp. 438–445, 2015.

[6] M. Sikorski and A. Honig, Practical Malware analysis: The hands-on guide to dissecting malicious software. San Francisco: No Starch Press, Inc., 2012.

[7] M. Ahmadi, A. Sami, H. Rahimi, and B. Yadegari, "Malware detection by behavioural sequential patterns," Computer Fraud & Security, vol. 2013, no. 8, pp. 11–19, 2013.

[8] S. Kumar, C. Rama Krishna, N. Aggarwal, R. Sehgal, and S. Chamotra, "Malicious data classification using structural information and behavioral specifications in executables," 2014 Recent Advances in Engineering and Computational Sciences, RAECS 2014, pp. 1–6, 2014.

[9] S. Cesare, Y. Xiang, and W. Zhou, "Malwise-an effective and efficient classification system for packed and polymorphic malware," IEEE Transactions on Computers, vol. 62, no. 6, pp. 1193–1206, 2013.

[10] D. Arish and M. Singh, "Behavior Analysis of Malware Using Machine Learning," in Contemporary Computing (IC3), 2015 Eighth International Conference on, 2015, pp. 481–486.

[11] G. Liang, J. Pang, and C. Dai, "A Behavior-Based Malware Variant Classification Technique," International Journal of Information and Education Technology, vol. 6, no. 4, pp. 291–295, 2016.

[12] V. Naidu and A. Narayanan, "Needleman-Wunsch and Smith-Waterman Algorithms for Identifying Viral Polymorphic Malware Variants," 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), no. August, pp. 326–333, 2016.

[13] M. Ahmadi, A. Sami, H. Rahimi, and B. Yadegari, "Malware detection by behavioural sequential patterns," Computer Fraud and Security, vol. 2013, no. 8, pp. 11–19, 2013.

[14] P. M. Comar, L. Liu, S. Saha, P. N. Tan, and A. Nucci, "Combining supervised and unsupervised learning for zero-day malware detection," Proceedings - IEEE INFOCOM, pp. 2022–2030, 2013.

[15] J. Park, S. Choi, and D. Y. Kim, "Malware Analysis and Classification: A Survey," Lecture Notes in Electrical Engineering, vol. 215, no. April, pp. 449–457, 2013.

[16] L. Zeltser, "Malware sample sources for researchers." [Online]. Available: https://zeltser.com/malware-sample-sources. [Accessed: 28-Feb-2016].

[17] Emmanuel Masabo Makerere ,Kyanda Swaib Kaawaase, Julianne Sansa-Otim Makerere University, Kampala, Uganda Damien Hanyurwimfura University of Rwanda, Kigali, Rwanda

[18] V. Kumar and S. Minz, "Feature Selection: A literature Review," Smart Computing Review, vol. 4, no. 3, pp. 211–229, 2014.

[19] A. Azab, R. Layton, M. Alazab, and J. Oliver, "Mining malware to detect variants," Proceedings - 5th Cybercrime and Trustworthy Computing Conference, CTC 2014, pp. 44–53, 2015.

Author's Profile

Dr. Faiz Baothman received his Bachelor's degree in Electronic Engineering from Maulana Azad College of Technology, India in 1984 and Masters in Computer Science from IIT, Roorkee, India in 1995. He did his Ph.D. in Real Time Database Systems from IIT, Roorkee, India in 1999.He started his professional career as a academician in various reputed universities and in different countries. He has published 10 research papers in many National, International journals. He has 20 years of experience in both teaching and research fields. Presently he is associated with Taif University, Taif, Saudi Arabia as Associated Professor in Computer Science Department. His areas of interests are Concurrency, Database and Software Engineering.

Dr. Muzammil Hussain Mohammed received his Bachelors of Science from Sri Venkateswara University, Tirupati, India in 1997 and Masters in Computer Applications from Madurai Kamraj University, Madurai, India in 2001. He did his Ph.D in Wireless Area Programming from Dr .B.R Ambedkar University, India in 2010.He started his professional career as a academician in various reputed colleges in India. He has published 18 research papers in many National, International journals and many conferences. He has 17 years of experience in both teaching and research fields. Presently he is associated with Taif University, Taif, Saudi Arabia as Associated Professor in Information Technology Department. His areas of interests are Software Engineering, Cloud Computing, Software Modeling and Cost Estimation, Web Services and Database. He is a member of IEEE and ACM.

