Comparative Analysis of Feature Extraction Methods of Malware Detection

Smita Ranveer
PG Student, Dept. of Computer Engineering,
Sinhgad College of Engineering, Pune
Savitribai Phule Pune University, India

Swapnaja Hiray
Associate Professor, Dept. of Computer Engineering,
Sinhgad College of Engineering, Pune
Savitribai Phule Pune University, India

ABSTRACT

Recent years have encountered massive growth in malwares which poses a severe threat to modern computers and internet security. Existing malware detection systems are confronting with unknown malware variants. Recently developed malware detection systems investigated that the diverse forms of malware exhibit similar patterns in their structure with minor variations. Hence, it is required to discriminate the types of features extracted for detecting malwares. So that potential of malware detection system can be leveraged to combat with unfamiliar malwares. We mainly focus on the categorization of features based on malware analysis. This paper highlights general framework of malware detection system and pinpoints strengths and weaknesses of each method. Finally we presented overview of performance of present malware detection systems based on features.

Keywords:

Feature Extraction, Malware Detection, Opcodes, Static Analysis, Dynamic Analysis, Machine Learning.

1. INTRODUCTION

The proliferation of modern computers, internet users, and communication infrastructure in any field is also followed by multiplicative increase in malwares and cyber-attacks caused by them. Malware variants are evolved to gain unauthorized access of systems, to get the economic benefits by illegal ways. Propagation of malware is havoc to internet security, commercial companies, privacy of users and governments.

Malware is derived from malicious software. It is an instance of malicious code with intention to subvert the function of system and has potential to harm a computer or network. It covers a range of threats like virus, trojans, adwares, spywares, etc. They replicate themselves and enter into the system in different ways; either multiple media or through the most popular way of getting downloaded into the system as the genuine application. Since different malware detection system has been introduced till date to circumvent the attacks caused by malwares. A malware detection system identifies malwares and defends the system to perform its function.

Currently existing methods for tackling malware are primarily based on two complementary approaches. These are classified according to the type of features they use for discovering malware activity. The signature-based detection approach relies on the identification of unique string patterns in the binary code. This technique uses static technique for creating signatures. It cannot cope with malwares unseen previously which is called as zero day attacks. When a novel type of malware family is observed, we need to analyse an instance of malware, generate a signature for it and insert it into malware database for reference inducing a classifier. During this period, an instance of this malware might attack several systems or networks. Since signature based detection approach has become inefficient and intractable. Knowing the weakness of detection systems malware designers developed code obfuscation techniques like code reordering, garbage insertion, variable renaming to disguise their content.

Following this intuition, heuristic based approach has been introduced an automated classification system. It is based on rules determined by experts, which relies on dynamic analysis of malicious behavior that deviates significantly from a normal behavior. It precisely deals with unknown malware discovery. However, this detection approach generates greater amounts of false alarms than signature based detection because not each suspicious executable file is malware. It has been observed that each of the two approaches had some limitations. Further antivirus vendors attempted to use individual as well as hybrid analysis approach for mining features and tackling newly emerging malwares [1]. They achieved a precise detection rate and low false positives compared to existing malware detection methods. In [2, 3, 4] investigated malware detection systems based on integrated static and dynamic analysis features using data mining approaches. An appropriate determination of malware variant depends on the feature type employed for discovering malicious activity. The performance of the system depends on the feature type which is the best indicator of malware and requires least time for quantifying the correlation between malicious activities. Hence, we sought to give a summarized view the earlier malware detection system propounded by researchers. Table-1 summarizes aforementioned malware detection systems with their pros and cons in light of emerging threats. Here we synthesized the subset of up-to-date malware detection system incorporating static, dynamic and hybrid analysis approach. Rest of the paper is structured as follows: at first section 2 gives the motivation and section 3 explores the general

framework of malware detection using machine learning approach followed by malware analysis in section 4. In this vein, we mainly focus on the feature categorization based on malware analysis. These features description is briefed in section 5. Further, Section 6 analyzes the performance of existing malware detection systems based on feature extraction techniques on standard dataset which is briefed in Table-II, finally concluding remarks in section 7.

2. MOTIVATION

Network security has always been a major concern for everyone involved in internet and for everyone using computer system. According to the Sophos Security Threat Report 2014 [5], malware and related IT security threats have grown and matured, and the developers of malicious softwares have become far more creative in camouflaging their work. In 2013 there was a rise of a vicious new version of trojans, spywares. McAfee Security Labs catalogues nearly 100,000 malware versions every day, i.e. approximately one new threat per new second of time. Since this is urged to know how to circumvent the malware propagation. Most of the previous surveys briefed malware types and detection techniques. In [6], Saeed et al. gave an overview of malwares and their detection systems; while in [1] Shabtai et al. presented a state of art survey on machine learning techniques employing static features. Hence here we give an abstract view of the recently formulated malware detection systems. The prime motivation of our survey is to summarize the types of widely used features for malware detection.

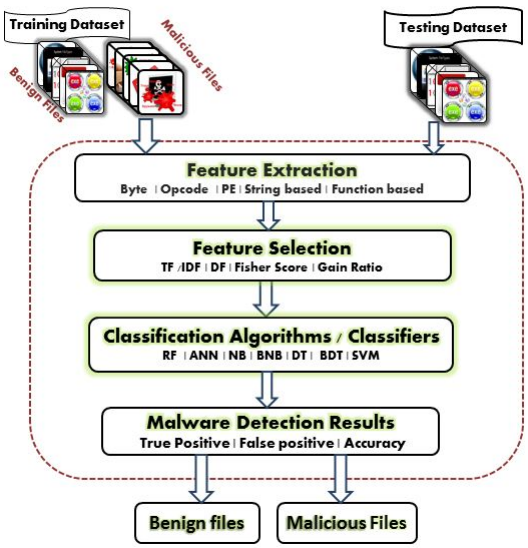

Fig. 1: General framework of malware detection system

3. GENERAL FRAMEWORK OF MALWARE DETECTION SYSTEM

Extensive survey has been done into the detection methods propounded by research community. Malware research can be categorized in terms of static as well as dynamic analysis and in terms of how the features of malware are processed after extraction. We observed that the general framework for malware detection using machine learning exhibits three distinct stages: Feature extraction, feature selection sometimes followed by dimensionality reduction techniques, and then classification using machine learning algorithm. This flow of malware detection process is as shown in Fig. 1. Each stage indicates different measure and methods used in previously existing methods. Firstly the dataset is prepared which consists of malware and benign executables. These files are preprocessed depending on the FE method and next feature selection is done to quantify the correlation of feature for improving performance and reducing number of computations to attain the learning speed. Further after generalizing the feature capability, classifier is trained on the basis of the filtered results of feature selection. Researchers have adopted supervised machine learning approach which uses classifiers Decision trees, Support vector machine, Nave bayes, Bayesian network, KNN algorithm, etc. are mentioned in [6,7,1]. The best classifier is chosen which gives the clear margin, and reduces interference and misclassification between maliciousness and benignancy of executables. The dataset is tested corresponding to the trained classifier and results are generated as malicious or benign softwares. The obtained outcomes are evaluated with consequent performance metrics.

4. MALWARE ANALYSIS

Malware analysis is a technique to study malware behavior and its structure by extracting features which describes its malevolent intention. Several techniques have been introduced to detect unseen variants of malware. The domain of features is characterized by the way of analyzing executables. Traditionally features are categorized on the basis of static and dynamic analysis of program files. For attaining efficiency and robustness, the system adheres to the best feature type which explores a meaningful corpus of malwares. In static analysis, the expected behavior of program is determined over the observations in its binary code or internal structure of files instead of actually executing it [6]. The static feature uniquely identifies the signature of malware or malware families. Static analysis is vulnerable to code obfuscation techniques. Dynamic analysis is test the program real time by actual execution in controlled environment. In dynamic analysis behavior of malicious softwares is monitored in emulated environment and traces are obtained from the reports generated by sandbox. It can deal with code evasion techniques [8]. However, it is resource consuming and time intensive. Further malware detection system utilized hybrid approach which is an integration of static and dynamic analysis. Variety of features is invented by compounding the static and dynamic approach. This taxonomy of features based on malware analysis is depicted in Fig.2.

5. FEATURE EXTRACTION METHODS

The first crucial stage of malware detection mechanism is to determine the representation of malicious software files. Various representation patterns of malware files were mentioned in the literature. Transforming the large, vague collection of inputs into the set of features is called feature extraction. When there is abundant input data to an algorithm and tend to be more redundant and irrelevant, feature extraction is performed. It is required to gain the precise measurement of features which influence the classification of input as benign or malicious. Since feature extraction process transforms the features into an organized, more manageable subset of information. Further it also reduces the dataset for processing resulting low computational overhead [1]. The outcome of the feature extraction phase is a vector containing

the frequencies of features extracted. Features extracted are chosen such that it attains maximum classification accuracy. The time required to get features from input dataset is also depends on the feature extraction methods.

Feature extraction method affects the performance of the system in terms efficiency, robustness, and accuracy. At first Schultz et al. in [9], introduced the notion of applying machine learning techniques for the detection of malwares based on their respective representation of files from the dataset. They employed three FE methods, while further researchers extended this idea of feature extraction to ameliorate the performance and accuracy of the system. Following the aforementioned research background features are described as follows:

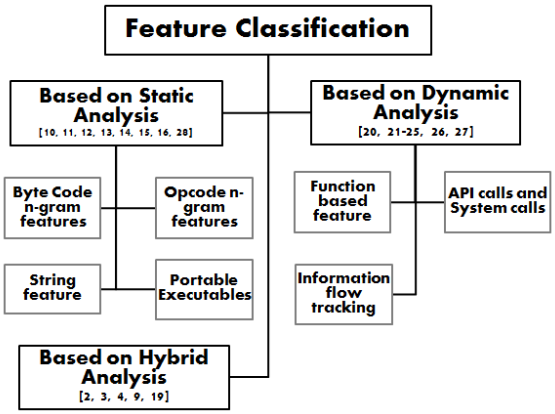

Fig. 2: Taxonomy of feature extraction methods

(1) *Byte n-gram Features*

Byte n-gram features are sequences of n bytes extracted from malwares used as signature for recognizing malware. Although this type of feature does not provide meaningful information, it yields high accuracy in detecting new malware. Abou-Assaleh et al. [10] extracted byte n-gram features from the binary code of the file where the L most occurring n-grams of each class in the training set are selected to denote the profile of the class. Every new instance is associated with a class closest profile using K-Nearest Neighbors (KNN) algorithm. Their experiments achieved 98% accuracy on dataset of benign and malware files. In [11], byte n-grams in combination with opcode n-grams are used as features. They provide an extensive evaluation using a test collection comprised of more than 30,000 files. Different settings of opcode and byte sequence n-gram representations and five types of classifiers yielded an accuracy of up to 99%. In [12], Li et al. propounded a method for detecting file types by analyzing n-gram sequence of their binary content. This method represented compact fileprint for each file type and used mahalanobis distance to determine the closest file type model based on centroids obtained.

(2) *Opcode n-gram Features*

Previous studies represented that opcodes feature extraction was more efficient and successful for classification. They reveal statistical diversities between malicious and legitimate softwares. Some rare opcodes are better predictors of malicious behavior. First all dataset executable files are disassembled and opcodes are extracted. An opcode is the assembly language instruction which describes the operation to be performed. It is short form of operational code. An instruction contains an opcode and operands, optionally upon which the operation should act. Some operations have operands upon opcodes may operate, depending on CPU architecture, registers, values stored on memory and stacks, etc. The action of an opcode takes in arithmetic, logical operations, and data manipulation operation. Opcodes are capable to statistically derive the variability between malicious and legitimate software.

Moskovitch et al. [16] presented mean accuracy of the combinations n-gram opcode sequences. They stated that 2-gram opcode sequence was the best N-gram sequence comparatively, which showed classification accuracy. However, for more than bigram opcode sequence the accuracy is decreased. Santos et al. [13, 15] used opcode sequences for categorizing malicious and benign files with different feature selection and classification algorithms. In [13, 28], opcode sequence of 1-gram and 2-gram sequences for detecting new variants of malware families. They used histograms for each n-gram sequences calculating frequency of similarity ratio for each malware instance. Sekar et al. [29] used n-gram approach and examined performance of system by applying Finite State Automaton (FSA) approach. They estimated two approaches on httpd, ftpd, and nsfd protocols which resulted into a lower false positive rate when compared to the n-gram approach.

(3) *Portable Executables*

These features are extracted from certain parts of EXE files. Portable Executables (PE) features are extracted by static analysis using structural information of PE. These meaningful features indicate that the file was manipulated or infected to perform malicious activity. In [19], Shafiq et al. propounded a real time approach for malware detection based on structural features mined from PE. They tested performance on two datasets Malfease and VXheavens dataset [30] which remarked that PE features has low processing overheads. These features may include part of the pieces of information given as follows [1]: 1. File pointer: pointer denotes the position within the file as it is stored on disk, CPU type; 2. Import Section: functions from which DLLs were used and Object files, list of DLLs of the executable can be imported. 3. Exports Section: describes which functions was exported. 4. Data extracted from the PE Header that describes physical and logical structure of a PE binary which may include features like code size, debug size as well as creation time, file size, etc. 5. Resource Directory: indexed by a multiple-level binary-sorted tree structure, resources like dialogs and cursors used by a given file.

(4) *String Features*

These features are based on plain text which is encoded in executables like windows, getversion, getstartupinfo, getmodulefilename, messagebox, library, etc. These strings are consecutive printable characters encoded in PE as well as non-PE executables. String features are used in Schultz et al. [9] provided 97.11% accuracy, when compared to using PE features and byte n-grams. Strings features are not very robust as they can be modified easily any time. In [19] proposed a malware detection system, SBMDS, which

Feature Type	Classification Method	Strengths	Weaknesses
Static [13,14, 15,11, 16,17]	Gain ratio, Fisher Score, ANN, DT, NB, BNB, BDT, SVM [11, 16]	Accuracy, Imbalance problem, Unknown Malware	Packed Executables
	Information gain, DT, KNN, SVM, RF, NB, Bayesian Network[13,14]		
	Mutual Information, TF, cosine similarity [15]	Detects Malware variants families	Pack executable Accuracy
	Game Theory, Genetic Algorithm, SVM [18]	Dimensionality problem, False positive	Time
Hybrid [2, 3, 4, 19]	DT, KNN, SVM,RF, NB, Bayesian Network,	Hybrid approach, Scalable, Automation, Robust	Manifest Single program behavior.
	SVM ensemble with bagging [19]	Defend Polymorphism & Metamorphism	Large size data
Dynamic [20,21, 22,23, 24,25, 26,27]	Behavioral analysis using Phylogenic trees[25]	Flexible, Automation, Zero day malware	False Positive
	Behavior analysis using SVM,DT,IB1,RF,[20] + KNN, NB[22, 24]	Reduces runtime and memory overheads, Automation	Incomplete picture malware activity
	Behavior Graph Matching [23, 24]	False positives, Fast generation of behavior graphs	Accuracy, Latest malwares Testing time

Table 1. : SUMMARY OF MALWARE DETECTION SYSTEMS

classifies malware using SVM based on interpretable string features. It outperformed existing antivirus softwares achieved better accuracy and efficiency using string features.

(5) *Function Based Features*
Function based features are extracted over the runtime behavior of the program file. Function based features functions that reside in a file for execution and utilize them to produce various attributes representing the file. Dynamically analyzed function calls including system calls, windows application programming interface (API) calls, their parameter passing, information flow tracking, instruction sets, etc. These functions increase the code reusability and maintenance. It is semantically richer representation. Any malicious software for execution or replications invokes some kernel level system call to communicate with operating system; it is a sign of malicious activity. In [22, 21, 25], addressed automatic behavior analysis using Windows API calls, instruction set, control flow graph, function parameter analysis and system calls are used as features.

In [31] presented an automated malware detection system which classifies malwares into their families monitoring their network behavior. It creates behavior graph from network traces obtained which represents network activities and their network flow dependencies. The graph structure, in-degree, out-degree of nodes and root denotes the features of malware activity. As per [31, 24], J48 decision trees given better TPR, FPR and accuracy results in comparison with other classifiers. Firdausi et al. [24] propounded a malware detection system which monitors the behavior of malicious files in controlled environment using a free online dynamic analysis tool named Anubis. Then the generated results are parsed into vector model for classification on the basis of the trained classifier. The performance is tested on the small dataset of benign and malicious files with and without feature selection. The accuracy of 92.3% and 96.8% with and without feature selection resp. achieved by J48 classifier was better than other classifiers SVM, KNN, and nave bayes. In [20], Tian et al. presented an automated classification system which uses API call sequences as features and discriminates malwares and cleanwares performance an accuracy of 97% achieved over a dataset of malwares and cleanwares.

Biley et al. [26] investigated an antivirus (AV) technique which eliminates the drawbacks of earlier AV products and qualifies consistency, conciseness and completeness across malware. System state changes describe the malware behavior fingerprint in terms of files registry, process creation, network flows, etc. It uses clustering and classification of malware samples. However the virtualized environment was static. An automatic behavior analyzing system proposed by Rieck et al. in [20] which gives an incremental and timely defense method for clustering and classification of malware binaries in similar behavior and identifying novel classes of malwares using machine learning method. It avoids runtime overhead and gives accurate discrimination of novel malware.

Park et al. [23] presented a malware detection system which uses system call and their parameters values as the features and generates directed subgraph for each programs behavior during execution. It creates a maximal behavior subgraph for measuring their similarity between their programs and known malware families. They evaluated performance over 6 known malware families and provided fair dissimilarity rates keeping low false positives still the accuracy needed to be improved as some malwares succeed to get kernel privileges. Lee et al. in [27] proposed a similar technique of clustering malware families using supervised machine learning technique. It also analyzes sample datasets behavior according to system call and parameters in virtual environment and generating a behavior profile for network activities. Further they computed similarities between those profiles and grouping of different samples is done by applying k- medoids clustering.

(6) *Hybrid Analysis Features*
These features are obtained by combining both techniques

Feature		Performance Metrics (High Accuracy, TPR, & Low FPR is better)		
Feature Type	Feature Signature	TPR	Accuracy (%)	FPR
Static	Opcode n-gram + Byte Code n-gram [16]	-	95	0.06
	Opcode n-gram [11]	-	99	0.03
	Opcode n-gram [13]	0.95	92	0.03
	Byte code n-gram + Opcode n-gram [14]	0.95	96	0.1
	Portable Executable Header [17]	-	99	0.05
Hybrid	Opcode n-gram + Application Programming Interface Function calls [2]	0.97	96.22	0.07
	Function Length Frequency + Printable String Information + Application Programming Interface calls [3]	0.98	97.05	0.055
	Application Programming Interface Function calls + Portable Executable Header + String [19]	-	93.7	0.15
	Function Length Frequency + Printable String Information [4]	-	98.86	-
Dynamic	System Call [24]	0.95	96.8	0.04
	System state change [26]	-	91.6	-

Table 2. : PERFORMANCE EVALUATION OF MALWARE DETECTION SYSTEMS

static analysis as well as dynamic analysis. It reduces the effect of countermeasures of each static and dynamic technique for analyzing malwares and improves the performance and detection rates. Islam et al. [3] extracted static features of functions such as function length frequency and printable string Information (FLF and PSI) based on the functions of different lengths and the number of distinct printable strings present in unpacked malware executables. Further they extracted Application Programming Interface (API) function calls and parameters by dynamic analysis. They provided superior results in terms of accuracy on combining the function based features and string features. Similarly a combination of string and function features is used for classification of malwares in [4]. They used different function length frequency ranges and printable string information performed better over seen malware set. Santos et al. [2] introduced a hybrid approach eliminating the need for each individual static and dynamic malware analysis using both emulation (Qemu) and simulation (Wine) techniques for attaining the transparency without interference to the system. They extracted opcode sequences statically and Windows API calls dynamically; characterizing their behavior in groups of system information, persistence, file creation, process or thread creation, adding registry keys, errors, etc. This method employed classification algorithms such as KNN, SVM, Decision trees, bayesian networks, etc. to discriminate malwares and benign softwares. This provided more accurate results leading to notable increase in performance metrics.

6. PERFORMANCE EVALUATION

Every malware detection system is obliged to provide a timely defense against cyber-attacks caused by malwares with high precision. The performance evaluation is done by using classical metrics such as classification accuracy, False Positive Rate (FPR) and True Positive Rate (TPR) with least processing time. TPR is ratio of the number of correctly detected malware to the total number of malware in the testing set. FPR ratio of the number of benign files detected as malware to the total number of benign files in the testing set. The efficiency and robustness of the system is defined by high accuracy, high TPR and low FPR, such system is effective in the real life scenarios.

The aforementioned researches evaluated their system on the standard dataset which consists of two sets of executables benign and malicious. The malicious executables dataset is downloaded from the VXheavens website [30], which covers malwares such as virus, adwares, worms, Trojan horses, etc. Here we provide comparative assessment of performance measures over results generated by systems on the malware dataset. Table II gives the overview of referenced malware detection system. We found some insights from our review which are as follows: First we observed that systems using opcode and PE features adhere to low FPR and high accuracy i.e. above 95% with some fluctuations [11, 14, 17, 18]. They were unable to cope with packed executables, while disassembly of executables is not always feasible.

PE-miner approach in [17] was robust and reliable against packed executables in real time with low processing overheads. Behavioral features API call and system call tracing is effective on zero day malwares while they increase the FPR which can undermine the efficacy of the system. Combining the features in a single method step up the performance and provides accuracy up to 99% along with high TPR keeping the low FPR. Features based on dynamic analysis are less vulnerable to code evasion techniques. Though features based on dynamic analysis are best indicators of malware, they are time consuming and resource intensive. Since, precise and effective results are achieved by hybrid approach which eliminates the loopholes of each method. In [2, 3, 4] malware detection system employing hybrid features showed high accuracy and TPR in comparison with those using static and dynamic features.

7. CONCLUSION

This paper gives an overview of malware detection techniques based on static, dynamic and hybrid analysis of executables. We presented a comparative assessment of features and illuminated their effect on performance of the system. We found that, high accuracy and TPR can be achieved by selecting an appropriate feature extraction method. Although opcode and PE features enhanced the speed and accuracy of malware detection system, they give rise to false positives. Hybrid analysis features maintain low false positive rate and yield precise results in least processing time. These methods used for malware classification should be able to

deal with huge and daily emerging malware variants which can preserve the performance and accuracy of the system in real time.

8. REFERENCES

[1] A. Shabtai, R. Moskovitch, Y. Elovici, C.Glezer, Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey, Information security technical report 14, 2009.

[2] Santos, I., Devesa, J., Brezo, F., Nieves, J. and Bringas, P.G. (2013) OPEM: A Static-Dynamic Approach for Machine Learning Based Malware Detection, Proceedings of International Conference CISIS12-ICEUTE12, Special Sessions Advances in Intelligent Systems and Computing, 189, 271-280.

[3] R. Islam, R Tian, Lynn, M. Batten , S. Versteeg, Classification of malware based on integrated static and dynamic features, Journal of Network and Computer Applications 36,646656,2013.

[4] Islam R, Tian R, Batten L, Versteeg S. Classification of malware based on string and function feature selection, Cybercrime and Trustworthy Computing Workshop (CTC) 2010:917.

[5] Sophos labs, Security Threat Report 2014.

[6] I.A. Saeed, A. Selamat, Ali M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, International Journal of Computer Applications, Volume 67 No.16, April 2013.

[7] Mathur, K. and Hiranwai, S. A Survey on Techniques in Detection and Analyzing Malware Executables. International Journal of Advanced Research in Computer Science and Software Engineering, 2013, 3: 422428.

[8] Ekta Gandotra, Divya Bansal, Sanjeev Sofat, Malware Analysis and Classification: A Survey, Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, India Journal of Information Security, 2014, 5,56-64 Published Online April 2014 in SciRes.

[9] Schultz, M., Eskin, E., Zadok, F., Stolfo, Data mining methods for detection of new malicious executables. In: Proceedings of the 22nd IEEE Symposium on Security and Privacy. (2001) 3849.

[10] Tony Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan. Detection of new malicious code using n-grams signatures In Proceedings of Second Annual Conference on Privacy, Security and Trust, pp. 193196, 2004.

[11] R. Moskovitch, C. Feher, N. Tzachar, E. Berger, M. Gitelman, S. Dolev and Y. Elovici. Unknown Malcode Detection Using OPCODE Representation. Proc. Of the 1-st European Conference on Intelligence and Security Informatics (EuroISI08), 2008.

[12] W. Li, K. Wang, S. Stolfo, B. Herzog. Fileprints: Identifying file types by n-gram analysis. Proc. of the IEEE Workshop on Information Assurance and Security,2005.

[13] Moskovitch R, Stopel D, Feher C, Nissim N, Elovici Y. Unknown malcode detection via text categorization and the imbalance problem In: IEEE Intelligence and Security Informatics, Taiwan; 2008.

[14] I.Santos, F. Brezo, X. Ugarte-Pedrero, P. G. Bringas, Opcode sequences as representation of executables for data-mining-based unknown malware detection, Information Sciences, vol. 231, pp. 64-82, 2013.

[15] A.Shabtai, R. Moskovitch, C. Feher, S. Dolev, and Y. Elovici, Detecting unknown malicious code by applying classification techniques on opcode patterns, Security Informatics, vol. 1, pp. 122, 2012.

[16] I. Santos, F. Brezo, J. Nieves, Y. K. Penya, B. Sanz, C. Laorden, and P. G. Bringas, Opcode-sequence-based malware detection, in Proc. 2nd Int. Symp. Eng. Secure Software and Syst. (ESSoS), Pisa, Italy, . vol. LNCS 5965, pp. 3543, Feb.34, 2010.

[17] M. Z. Shafiq, S. M. Tabish, F. Mirza, and M. Farooq, Pe-miner: Mining structural information to detect malicious executables in realtime, in Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection, ser. RAID 09. Berlin, Heidelberg: Springer- Verlag, 2009, pp.121141.i.org/10.4236/jis.2014.5-2006.

[18] Mikhail Zolotukhin, Timo Hamalainen, Support Vector Machine Integrated with Game-Theoretic Approach and Genetic Algorithm for the Detection and Classification of Malware, Globecom 2013 IEEE Workshop - First International Workshop on Security and Privacy in Big Data

[19] Y. Ye, L. Chen, D. Wang, T. Li, Q. Jiang, and M. Zhao, Sbmds: an interpretable string based malware detection system using svm ensemble with bagging, Journal in Computer Virology, vol. 5, no. 4, pp. 283293, 2009.

[20] Rieck, K., Trinius, P., Willems, C. and Holz, T. (2011) Automatic Analysis of Malware Behavior Using Machine Learning. Journal of Computer Security, 19, 639-668.

[21] Tian R, Batten L, Islam R, Versteeg S. An automated classification system based on the strings of Trojan and virus families, In: Proceedings of the 4th international conference on malicious and unwanted software: MALWARE 2009; 2009. p. 2330.

[22] Tian, R., Islam, M.R., Batten, L. and Versteeg, S. (2010) Differentiating Malware from Cleanwares Using Behavioral Analysis, Proceedings of 5th International Conference on Malicious and Unwanted Software (Malware), Nancy,October 2010, 23-30.

[23] Park, Y., Reeves, D., Mulukutla, V. and Sundaravel, Fast Malware Classification by Automated Behavioral Graph Matching. Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research, Article No. 45,2010.

[24] Firdausi, I., Lim, C. and Erwin, Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection, Proceedings of 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), Jakarta, 2010, 201-203.

[25] Wagener G, State R, Dulaunoy A. Malware behaviour analysis, Journal in Computer Virology 2008;4(4):27987.

[26] Biley, M., Oberheid, J., Andersen, J., Morley Mao, Z., Jahanian, F. and Nazario, Automated Classification and Analysis of Internet Malware, Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, 4637, 178-197.

[27] Lee, T. and Mody, J.J. Behavioral Classification Proceedings of the European Institute for Computer Antivirus Research Conference (EICAR2006).

[28] D. Bilar, Opcodes as predictor for malware. International Journal of Electronic Security and Digital Forensics, pp. 156-168, 2007.

[29] R. Sekar, M. Bendre, D. Bollineni, and Bollineni, R. Needham and M. Abadi, Eds., A fast automaton-based method for detecting anomalous program behaviors, in Proc. 2001 IEEE Symp. Security and Privacy, IEEE Comput. Soc., Los Alamitos, CA, USA, 2001, pp.144155.

[30] VXheavens Website:url:http://vx.netlux.org.

[31] Nari, S. and Ghorbani, Automated Malware Classification Based on Network Behavior. Proceedings of International Conference on Computing, Networking and Communications (ICNC), San Diego, 28-31 January 2013, 642-647.