

A Novel Approach to Malware Detection using Static Classification

Sanjam Singla

Department of Computer Science
PEC University of Technology Chandigarh, India

Ekta Gandotra

Department of Computer Science
PEC University of Technology Chandigarh, India

Divya Bansal

Department of Computer Science
PEC University of Technology Chandigarh, India

Sanjeev Sofat

Department of Computer Science
PEC University of Technology Chandigarh, India

Abstract—Malware, commonly called computer virus, is one of the top security threats to the computer systems around the globe. These are evolving at a very rapid pace and are continually finding new ways to exploit and infect the systems of various enterprises and businesses. Malwares use different techniques to camouflage themselves to make their lifetime longer. In this paper, we present a simple technique based on static features extracted from Windows PE files. The features used are not only extracted from the header part of the malware but also from the payload i.e. body of malware. The static features used are a combination of Function Call Frequency and Opcode Frequency for differentiating malwares from clean files. This combination of features set makes it a new approach for malware detection which provides an accuracy of 97% for a dataset of 1,230 executables files including 800 malware and 430 cleanwares. For classification purpose, we use machine learning algorithms available in WEKA library. Based on the results obtained, we conclude that both features considered in this work play a significant role in distinguishing malicious files from clean ones.

Keywords- *Static Malware Analysis; Machine Learning; Classification;*

I. INTRODUCTION

Malware is a hostile, intrusive or annoying software program designed to furtively gain access to a computer system without the owner's permission. According to McAfee 2014 Q3 threat report [1] there are over 307 new threats appearing every minute, and more than 5 are detected every second. According to the report, the total malwares broke the 300 million sample barrier in Q3 2014 i.e. the growth by 76% over the past year.

Nowadays, malwares are becoming more targeted and sophisticated. Advanced malwares are more dangerous as compared to traditional malwares as they are unknown, attack stealthily and are persistent in their attack [2]. Malware authors are often looking for one-time development of specific code to generate new variants of existing malware [3], instead of developing new malware from scratch as variants of existing malware can be developed easily and quickly.

Vulnerabilities in browsers and operating systems are often exploited by attackers. Also social engineering techniques are used to run malicious code on unsuspecting users. Malware authors often use obfuscation techniques [4] to evade detection by traditional defences like antivirus, firewalls, IDS, IPS, and gateways which uses signature based detection. Furthermore, different API calls used by malwares to perform malicious activities are also used by benign files to perform some operations and thus making it more difficult for the antivirus vendors to detect malwares accurately. In this paper, we extracted the features from both the header as well as payload of the malware which has not been used yet. We use Function Call Frequency and Opcode Frequency as static features to develop an automatic malware detection and classification system. A python script is used for extraction of mentioned features from the available data set, which is further classified into malwares and cleanwares with the help of machine learning algorithms available in WEKA. Our method gives an accuracy of above 97%.

Rest of the paper is structured as follows: In Section 2, we reiterate the literature in related area. Section 3 provides an overview about proposed malware classification system. Section 4 highlights the experimental set up along with the result analysis. Section 5 provides a brief discussion on results, followed by conclusion in Section 6.

II. RELATED WORK

This section describes the state of art on extracting various features for static analysis of malwares. These techniques were earlier used by developers in code optimization and reverse engineering but later used in malware analysis and detection.

Machine learning approaches like, Association classifiers, Support Vector Machines, Decision Trees, Random Forest, Naive Bayes and Clustering etc, are proposed for identifying and classifying new and unknown samples of malwares into known malware families.

Thonnard *et al.* [5] developed a framework for detecting attack patterns in Honeynet data in order to find clusters of network traces sharing similar patterns within an attack data set, but limitation of his work is that the data collected through honeypots/ honeynets don't monitor network's normal traffic and were not able to cover the whole range of threats. Data mining concept for detecting malwares was first introduced by Schultz *et al.* [6]. Three different static features were used by them i.e. strings, byte sequence, Portable Executable (PE). In PE approach, features extracted were like, list of DLL function calls, and number of diverse system calls made within each DLL. Strings are taken out from executables based on static text based string matching which are encoded in program files. In byte sequence approach, sequence of n bytes extraction is done from an executable file. Later on Kolter *et al.* [7] improved the results by using overlapping byte sequence instead of non-overlapping byte sequence. With the help of this every possible sequence of bytes along with the frequency was extracted. In [8], [9], [10], [11], authors used byte n gram sequence to solve the malware classification problem. A new method for visualizing and classifying malware binaries as gray scale image using image processing techniques was proposed by Nataraj *et al.* [12]. A K-nearest neighbor method along with Euclidean distance method was used for malware classification. This method is fast but has limitation. As this method uses global image based feature, an attacker can implement countermeasures to thrash the system. For automated malware classification, Kong *et al.* [13] presented another framework which was based on structural information of malwares. After extracting the features based on function call graph, similarity is calculated for the two programs and discriminate distance metric learning was used which groups the malware samples belonging to alike family. In [14], [15], the authors analyzed the sequence of API calls used in PE code. Tian *et al.* [16] classified Trojan based on function length frequency, where length of function is the measure of the number of bytes in the code. Siddiqui *et al.* [17] classified worms via variable length instruction chain along with machine learning technique. In [18],[19], proposed methods for analyzing entropy of PE sections in an executable. Saini *et al.* [20] used suspicious section count and function call frequency as the features to distinguish malwares from clean ones. They have used machine learning algorithms as available in WEKA library for classification purpose. They achieved an accuracy of more than 98%.

In our work, we used four classification algorithms for distinguishing malwares from cleanwares: Decision Table (DT), J48 tree, KStar, MultilayerPerceptron (MLP), from WEKA (Waikato Environment for Knowledge Analysis) [21]. These algorithms have a wide range of techniques available for classification.

III. MALWARE ANALYSIS AND CLASSIFICATION SYSTEM

In this section, we present our approach to the classification problem using machine algorithm available in WEKA, over a data set of 1,230 executables after extracting the static features as mentioned in following sections.

A. The Data Set

We used a sum of 1,230 executable files including 800 malware and 430 clean files. The malware samples are provided by University of California, Santa Barbara, (<http://seclab.cs.ucsb.edu/>) on request and rest are collected from available online data sources like <http://www.nothink.org/honeypots/malware-archives>. The clean files are collected from the System32 directories of Windows 2000, Windows 2003, Windows XP, Windows 7 and Windows 8.

B. Static Features Used

This section explains the features which we used for distinguishing malwares from cleanwares. We not only used the extracted features from the header of the malwares but also from the payload i.e. body of the malwares. Function Call Frequency and Opcode Frequency features are extracted from each executable including malware and clean ware using a python script, which are then passed to automatic classification system for further analysis.

C. Function Call Frequency

Malware authors use obfuscation techniques to evade detection by Antivirus vendors. Obfuscation techniques used like dead code insertion, instruction replacement, register substitution, code transportation, and instruction permutation. Normally legitimate programs include a number of strings but obfuscated programs contain very few strings. If a particular searched file is packed or obfuscated then the number of strings will be less, thus suggesting that it may be malicious. Earlier, in [20], [23] authors have used function call frequency as one of the features to distinguish malwares from clean ones, we also in this work used the same feature but with a different combination for distinguishing the data.

In our approach, we analyzed the number of function calls made by malware and clean samples. We first created a global list of all known function calls and then match all the readable strings extracted from the analyzed files, which gives the exact numbers of function calls made by that executable. For instance, global list is considered to be made up of {'CheckRemoteDebuggerPresent', 'IsDebuggerPresent', 'LoadLibrary', 'WriteFile', 'CreateProcess'} functions based on which classification is done.

D. Opcode Frequency

Opcode basically means operational codes that are used in the files in order to perform some operations. In order to detect the advanced malwares like polymorphic and metamorphic, the opcode frequency can be used in analysis for their detection.

In our approach, we analyzed the number of opcode calls made by malware and clean samples. We first created a global list of all opcodes that are not common in benign files and then match all the readable strings extracted from the analyzed files, which gives the exact numbers of opcode calls made by that executable. Due to uncommon opcode list it is easy to distinguish between clean and malicious file as frequency of these opcodes is high in malwares as compared to that in clean files. For instance, the global list i.e. uncommon is made up of {'int', 'nop', 'rdtsc', 'sbb', 'shld', 'fdvip', 'imul', 'pushf',

'setb', 'fild'}. However, some common opcodes like {'mov', 'pop', 'call', 'test', 'jmp', 'lea'} are also considered for achieving better accuracy in the classification process.

IV. EXPERIMENTAL SETUP AND RESULT ANALYSIS

This section describes complete set-up for malware analysis and classification. It also includes the experimental result analysis. The experiment is performed on local system with configuration having Ubuntu 14.04 as operating system with 16GB RAM and 1TB hard disk. After extracting the static features from the data set, we used four classifiers: Decision Table and J48 tree, KStar, MLP available in WEKA library for differentiating malwares from cleanwares as mentioned in section 2. WEKA is a set of machine learning programs implemented in Java. It is developed at the University of Waikato and is freely available under the GNU General Public License. The standard input format accepted by WEKA is ARFF (Attribute-Relation File Format). We used WEKA 3.7 for Windows OS version.

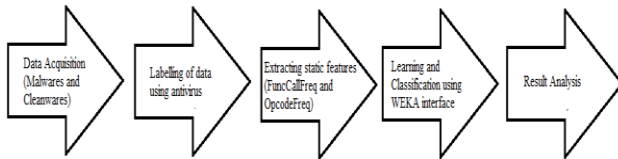


Figure 1. Malware analysis and classification system

Figure 1 illustrates the automated detection and classification system. We collected a large set of executable programs from a wide range of sources and separated them into malicious and benign executables using antivirus. For each executable file, including both malware and cleanware, we extracted Function call frequency and opcode Frequency as features from the complete data set using a Python script. These features are then used for differentiating malwares from cleanwares using WEKA [21] interface as mentioned in section 2. In our experiments, we used 10-fold cross validation for estimating our results as it is best method for estimation for likely predictions over unseen data in Data Mining. In 10-fold cross validation, the novel sample is arbitrarily partitioned into 10 sub samples each having equal size. Out of 10 sub samples, 9 are used for training and 1 is retained as validation data for testing purpose. The process is then repeated 10 times with each of the 10 sub samples used precisely once as the validation data. The average of these results is calculated in order to achieve a good measure of how the algorithm performs over the entire data set.

A. Experimental Results

The classification is done using machine learning algorithm available in WEKA. Detection rate is measured using three parameters: False Positive (FP), False Negative (FN) and the Detection Rate Accuracy.

Table 1 shows the weighted average of the experimental results for the Function Call Frequency (FuncCallFreq) feature.

Table 1. Classification results using FuncCallFreq

Classifier	Classification Results		
	FP	FN	Accuracy (%)
MLP	0.294	0.025	88.14
Kstar	0.068	0.03	91.56
J48	0.021	0.031	96.77
DT	0.014	0.031	97.37

MLP classifier shows the worst accuracy whereas DT shows the highest accuracy rate of 97.37% followed by J48 which gives 96.77% accuracy.

Table 2 shows the weighted average of the experimental results for the Opcode Frequency (OpcodeFreq) feature.

Table 2. Classification results using OpcodeFreq

Classifier	Classification Results		
	FP	FN	Accuracy (%)
MLP	0.250	0.098	84.71
Kstar	0.247	0.092	85.33
J48	0.198	0.097	86.72
DT	0.205	0.101	86.26

MLP classifier shows the worst accuracy whereas J48 shows the highest accuracy rate of 86.72% followed by DT which gives 86.26% accuracy. To further improve our results, we combined both the features and tested them on different classifiers.

Table 3 shows the weighted average of the experimental results for both the features taken together.

Table 3. Classification results using both FuncCallFreq and OpcodeFreq

Classifier	Classification Results		
	FP	FN	Accuracy (%)
MLP	0.275	0.026	88.69
KStar	0.156	0.026	92.84
J48	0.033	0.027	97.07
DT	0.014	0.031	97.47

With both the features combined together, DT gives the best accuracy i.e. 97.47 % followed by J48 which provides an accuracy of 97.07%. After comparing table 3 with table 1 and 2, we see that on an average the number of both FP and FN are reduced when combined features are considered.

Figure 2 shows the comparison of classifiers with respect to the features considered in our experiment.

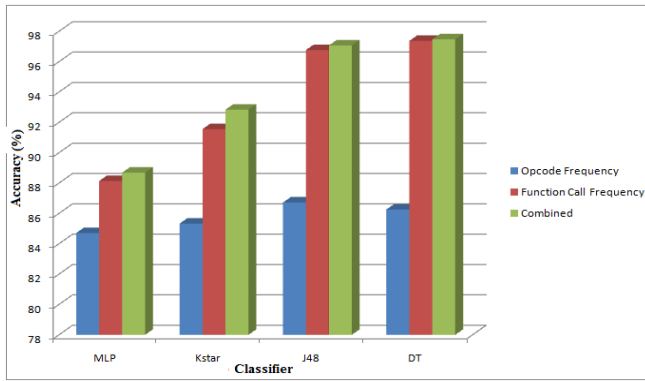


Figure 2. Comparison of accuracy of different classifiers using FuncCallFreq and OpcodeFreq

V. CONCLUSION

In this paper, the function call frequency and opcode frequency are used as static features for distinguishing malwares from cleanwares. The results obtained for our experiments indicate that the function call frequency is more robust feature than opcode frequency. Further, when we combine both features, DT proves to be the best machine learning technique equipped to classify our dataset. Though we get good results with static analysis, yet we understand that our technique may not be effective to classify zero day malwares and furthermore disassembling for extracting opcodes is a time consuming task. In future, we would come up with an optimized dynamic analysis technique, which will be able to classify advanced and stealth malwares.

REFERENCES

- [1] McAfee labs threats report:
<http://www.mcafee.com/in/resources/reports/rp-quarterlythreat-q3-2014.pdf>
- [2] E. Gandotra, D. Bansal and S. Sofat, "Malware analysis and classification: A survey," *Journal of Information Security*, Vol. 5, No. 2, pp. 56-64, April 2014.
- [3] S. Treadwell and M. Zhou, "A Heuristic Approach for Detection of Obfuscated Malware," in *Proceedings of the 3rd International Conference on Intelligence and Security Informatics*, IEEE, pp. 291-299, 2009.
- [4] K. Yim, "Malware Obfuscation techniques: A brief survey," in *Proceedings of International Conference on Broadband Wireless Computing Communication and Applications (BWCCA)*, Fukuoka, pp. 297-300, November 2010.
- [5] Thonnard and M. Dacier, "A framework for attack patterns' discovery in honeynet data," *Digital Investigation: The international Journal of Digital Forensics and Incident Response*, vol. 5, pp. 128-139, September 2008.
- [6] M. Schultz, M. Eskin, E. Zadok, and F. Stolfo, "Data Mining Methods for Detection of New Malicious Executables," in *Proceedings of 2001 IEEE Symposium on Security and Privacy*, IEEE, Oakland, CA, pp. 38-49, May 2001.
- [7] J. Kolter, M. Maloof, "Learning to Detect Malicious Executables in the Wild," in *Proceedings of 10th ACM International Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, ACM New York, NY, USA, pp. 470-478, 2004.
- [8] R. Moskovitch, D. Stopel, C. Feher, N. Nissim, and Y. Elovici, "Unknown Malcode Detection via Text Categorization and the Imbalance Problem," in *Proceedings of 6th IEEE International Conference on Intelligence and Security Informatics (ISI)*, Taiwan, pp. 156-161, 2008.
- [9] Santos, Y. Penya, J. Devesa and P. Bringas, "N Grams-Based File Signatures for Malware Detection," in *Proceedings of 11th International Conference on Enterprise Information Systems (ICEIS)*, AIDSS, Milan, Italy, pp. 317-320, 2009.
- [10] Y. Zhou and W. Inge, "Malware Detection using Adaptive Data Compression," in *Proceedings of 1st ACM workshop on AISec*, ACM New York, NY, USA, pp. 53-60, 2008.
- [11] Y. Ye, Y. Li, Y. Chen and Q. Jiang, "Automatic Malware Categorization using Cluster Ensemble," in *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, July, pp. 25-28, 95-104, 2010.
- [12] L. Nataraj, S. Karthikeyan, G. Jacob and B. Manjunath, "Malware Images: Visualization and Automatic Classification," in *Proceedings of 8th International Symposium on Visualization for Cyber Security*, ACM New York, USA, 2011.
- [13] D. Kong and G. Yan, "Discriminant malware distance learning on structural information for automated malware classification," *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, ACM New York, USA, pp. 347-348, 2013.
- [14] C. Wang, J. Pang, R. Zhao and X. Liu, "Using API Sequence and Bayes Algorithm to Detect Suspicious Behavior," in *Proceedings of International Conference on Communication Software and Networks*, IEEE Computer Society Washington, DC, USA, pp. 544-548, 2009.
- [15] J. Xu, A. Sung, P. Chavez and S. Mukkamala, "Polymorphic Malicious Executable Scanner by API Sequence Analysis," in *Proceedings of 4th International Conference on Hybrid Intelligent Systems*, IEEE Computer Society Washington, DC, USA, pp. 378-383, 2004.
- [16] R. Tian, L. Batten, R. Islam and S. Versteeg, "An automated classification system based on the strings of Trojan and virus families," in *Proceedings of 4th International Conference on Malicious and Unwanted Software*, Montréal, Quebec, Canada, pp. 23-30, 2009.
- [17] M. Siddiqui, M.C. Wang and J. Lee, "Detecting Internet Worms Using Data Mining Techniques," *Journal of Systemics, Cybernetics and Informatics*, Vol. 6, pp. 48-53, 2009.
- [18] R. Lyda and J. Hamrock, "Using Entropy Analysis to Find Encrypted and Packed Malware," *IEEE Security & Privacy* 5, pp. 40-45, 2007.
- [19] X. Pedrero, I. Santos, B. Sanz, C. Laorden and P. Bringas, "Countering Entropy Measure Attacks on Packed Software Detection," in *Proceedings of 9th IEEE Consumer Communications and Networking Conference (CCNC2012)*, Las Vegas, NV, pp. 164-168, 2012.
- [20] Saini, E. Gandotra, D. Bansal and S. Sofat, "Classification of PE files using static analysis," *SIN'14*, Glasgow, Scotland, UK, ACM 2014.
- [21] S. Michael and H. Andrew, "Basic Static Techniques, 'Practical Malware Analysis' No Starch Press, San Francisco, pp. 21-24, 2012.
- [22] Avira antivirus:
<http://www.avira.com/en/avira-free-antivirus>
- [23] E. Gandotra, D. Bansal and S. Sofat "Integrated Framework for Classification of Malwares," *SIN'14*, Glasgow, Scotland, UK, ACM 2014.

AUTHORS PROFILE

Sanjam Singla is a master student in CSE department of PEC University of Technology, Chandigarh. He completed his graduation from RBIEBT, Mohali Campus under Punjab Technical University, Jalandhar in year 2013.

Eakta Gandotra is a doctoral student in CSE department of PEC University of Technology, Chandigarh. She completed her M. Tech. in Computer Science and Engineering from Kurukshetra University, Kurukshetra in year 2008. She is pursuing her Ph.D. in the area of Cyber Security.

Divya Bansal is Associate Professor with PEC University of Technology, Chandigarh. She is also the Associate Coordinator for Cyber security Research Centre, Chandigarh where she leads a research and development team working on wireless security. She did her Masters in Engineering in

Computer Science & Engineering from PEC University of Technology and also her Ph.D. in the area of security in wireless networks from PEC. Her research interests include areas in Information Security such as Cyber Crime & Investigations, Cyber Warfare and Cryptology, Cloud Computing, Wireless Networks. She has done several government sponsored research projects under the capacity of Principal Investigator. She has over 50 research publications in reputed journals and conference proceedings.

Sanjeev Sofat is a Distinguished Professor with PEC University of Technology, Chandigarh. He is also the Coordinator for Cyber Security

Research Centre, Chandigarh. He completed his M.E in Computer Science from ISM, Dhanbad in the year 1993. He received his Ph. D. degree from Kurukshetra University in 2005. His current areas of research are Computer Networks & Information Security. He is a member of IEEE and CSI.