IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

CAM: A Combined Analytical Model for Efficient Malware Classification

Om Prakash Samantray[1], Satya Narayan Tripathy[2], Susant Kumar Das[3], Binayak Panda[4]

Research Scholar, Computer Science, Berhampur University, Odisha, India [1]

Lecturer , Computer Science, Berhampur University, Odisha, India [2]

Reader, Computer Science, Berhampur University, Odisha, India [3]

Computer Science, Berhampur University, Odisha, India[4]

Abstract: With the substantial use of internet Technology, the frequency of malware is increasing swiftly despite careful use of anti-malware software. Detecting malware is still is a challenge because invaders use new techniques to escape from the detection methods. The signature based detection which is used in most of the anti-malware software is proved to be unproductive due to the exponential increase in the number and types of malware. The static analysis methods can be used to analyse the binary file and generate the signature as an output. Dynamic analysis executes the file and then considers the behaviour and actions to identify whether the executable is a malware or benign. Considering the positive aspects of both these methods, an integrated analysis can be formulated to analyse and classify an unknown executable file. This proposed method uses machine learning in which known malwares and benign programs are used as training dataset. The binary code as well as dynamic behaviour can be analysed to generate a feature vector. A Combined Analytical Model is proposed by integrating advantages of both static and dynamic analysis. The proposed model improves efficiency and accuracy of malware classification. Experiments done on static, dynamic and the proposed integrated analysis technique to prove that the proposed method has a better accuracy than the individual analysis techniques.

Keywords: Malware Detection, Dynamic analysis, Static analysis, Printable string information (PSI), Support Vector Machine (SVM).

I. INTRODUCTION

With the advancement of online banking and online marketing system, the Internet has become an essential part of people's day-to-day life. Our world is changing and much of our personal communications, banking and overall well-being is now accessible online. As the use of internet has increased vigorously, the security threats also increased relatively with a more pace. The users of Internet including corporates are the victims of the security threats caused by malwares. The malware, intruders, hackers has made today's internet a warzone, where everybody online is part of the fight.

Malware or malicious software is a program that affects a computer system without the user's permission and with an aim to cause harms to the system or steal private information from the system. Software that deliberately fulfils the harmful intent of an attacker is commonly referred to as malicious software or malware. Depending on the Behaviour and the way of their propagation & infection malwares are classified as viruses, worms, Trojan Horses, root-kits, spy-ware, Backdoor, Botnet and Adware etc.

Thousands of new malwares are emerging every day and the existing malwares are evolving in their structure become difficult to detect. According to the latest Internet Threat Report from Symantec, a whopping 317 million new types of malware were discovered in 2014. According to Kaspersky Lab report-2014, in 2014, Kaspersky Lab products detected 22.9 million attacks utilizing financial malware, targeting 2.7 million users. This represents a year-on-year decrease of 19.23% for attacks and 29.77% for the number of users. Although the total number of financial attacks decreased, the share of malware attacks targeting online banking credentials rose 8.89 percentage points to comprise 75.63% of all financial malware attacks in 2014 [1].

Due to increase in new samples every day, automated malware analysis tools and methods are needed to distinguish malicious from benign code. Most of the commercial anti-virus software uses signature based malware classification method [2]. This method compares the unknown malwares with a database of known malicious programs to identify whether the file is malware or benign. The signature is a unique identification of a binary file. Signature of malware is found by using static analysis, dynamic analysis or hybrid analysis and is stored in signature databases. The main disadvantage of this method is, the signature database need to be updated frequently because of the fast emerge of new malwares every day.

The analysis method which analyses malicious software without executing it is called static analysis. In static analysis, String signature, byte-sequence, syntactic library call, n-grams, opcode (operational code), control flow graph and frequency distribution etc. are used as detection

patterns. The disassembler/debugger and memory dumper tools can be used to reverse compile windows executable [3].

In static analysis, models can be designed by extracting and examining features from the binary code of programs and in turn these models can be used to differentiate malware and legitimate executable. The static analysis techniques sometimes proved to be very expensive and unreliable because, binary obfuscation techniques, used by attackers to transform the malware binaries into self-compressed and uniquely structured binary files, which resist reverse engineering process. Likewise, when utilizing binary executable for static analysis, the information such as size of variables or data structures gets lost thereby complicating the malware code analysis [4].

In Dynamic Analysis the behaviour of malicious code is analysed while it is being executed in a controlled environment like virtual machine, simulator, emulator, sandbox etc. Before executing the malware sample, the appropriate monitoring tools like Process Monitor and Capture BAT (for file system and registry monitoring), Process Explorer and Process Hackerreplace (for process monitoring), Wireshark (for network monitoring) and Regshot (for system change detection) are installed and activated [3].

Various techniques that can be applied to perform dynamic analysis are, function call monitoring, information flow tracking, function parameter analysis, instruction traces and auto start extensibility points etc. [4]. Dynamic analysis is more effective as compared to static analysis and does not require the executable to be disassembled and the natural behaviour of malware is revealed in dynamic analysis which makes this method more robust than static analysis.

However, it is time intensive and resource consuming, thus elevating the scalability issues. Sometimes, the malwares may perform in a different way in real environment and sometimes malware behaviour is triggered only under specific conditions which can't be detected in virtual environment. Several online automated tools exist for dynamic analysis of malwares, e.g. Norman Sandbox, CWSandbox, Anubis and TTAnalyzer, Ether and ThreatExpert.

Both static and dynamic analysis techniques have their own merits and demerits. Hence, the best features of both these methods can be integrated and machine learning techniques can be used to provide an efficient classification model for malwares.

II. RELATED WORK

There are numerous machine learning approaches like Association Rule, Decision Tree, Support Vector Machine, Naive Bayes, Random Forest, and Clustering have been proposed in the literature for classification and detection of malwares. A few of these are discussed in this section.

Tian et al. [5] used virtual environment to run the executables and used an automated tool for extracting API call sequences. They used the classifiers available in WEKA library to discriminate malware files from safe files as well as for classifying malwares into their families. They used a data set of 1368 malwares and 456 clean wares to validate their work and achieved an accuracy of over 97%.

Rieck et al. [6] proposed automatic analysis framework for malware behaviour using machine learning. They monitored behaviour of a large number of malware samples in sandbox environment. The observed behaviour is embedded in a vector space and then learning algorithms are applied. They used clustering to identify novel classes of malware with similar behaviour. Classification method is used to assign a new malware to the discovered class. Based on both, clustering and classification, an incremental approach is used for behaviour-based analysis, capable of processing the behaviour of thousands of malware binaries on daily basis.

Anderson et al. [7] proposed a method, which used multiple data sources. For the binary file and disassembled file, kernels based on the Markov chain graphs are used. A graphlet kernel is used for the control flow graph and a Standard Gaussian kernel is used for the file Information feature vector, then multiple kernel learning is employed to find a fair combination of the data sources and support vector machine classifier is used to categorize the dataset into malicious and benign. It is tested on a dataset of 780 malware and 776 benign instances giving an accuracy of 98.07%.

Nari et al. [8] presented a framework which classifies malwares into their respective families based on network behaviour. Network traces in the form of Pcap files are taken as input to the framework and as a result, the network flows are obtained. Then the network activities and deeds of malwares and dependencies between network flows were represented as a graph termed as behaviour graph. Then the properties of behaviour graph such as graph size, root out-degree, maximum out-degree, average out-degree, number of specific nodes were used to classify malwares using classification algorithms available in WEKA library.

Santos et al. [9] proposed a hybrid unknown malware detector called OPEM, which uses a set of features obtained from both static and dynamic analysis of malicious code. The static features are obtained by modelling an executable as a sequence of operational codes and dynamic features are acquired by monitoring system calls, operations and raised exceptions. The approach is then validated over two different data sets by considering different learning algorithms for classifiers Decision Tree, Bayesian network, K-nearest neighbour and Support Vector Machine and it has been found that this hybrid approach enhances the performance of both approaches when run separately.

A similar work is done by Islam et al. [10] where the static include function length frequency and printable sting information and dynamic features include API function names and API parameters. The experiment was conducted using 2939 executable files including 541 clean files separately for every feature and then for integrated

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

method for Meta classifiers SVM, IB1, DT and RF. They proved that that all meta-classifiers achieve highest accuracy for integrated features and meta-RF is the best performer in all cases. They also done a compared study to prove the accuracy of their integrated method over the existing methods.

Yuxin Ding et al. [11] used an Objective-Oriented Association Mining to detect malware. To reduce the number of rules and to improve the quality of the rules, the criteria for API selection and the criteria for association rule selection were proposed. They adopted CBAA classification method to improve the classification accuracy. They proved that the above strategies can remove approximately two third of the redundant rules whereas the accuracy remains same as the original OOA algorithm. The detection speed of their proposed method is approximately two times faster than that of the original method due to decrease in the number of association rules. Hence, the proposed method is effective and faster than traditional OOA mining for malware detection.

III. THE COMBINED ANALYTICAL MODEL

Most of the Malware classification methods use either Static or Dynamic approach for malware Analysis. The Proposed Combined approach integrates the best features of both the static and dynamic methods. The static features of training dataset are obtained from the binary code of the executable whereas the actions and behaviours are obtained from Dynamic analysis. The Static analysis gives emphasis on the printable string information (PSI) of the binary file and Dynamic analysis gives emphasis on System Call Sequences. Then both these extracted information (features) can be combined to generate a more accurate feature vector which can be helpful in classifying the files either as Malware or Benign. The combined analysis approach is represented in figure-1.

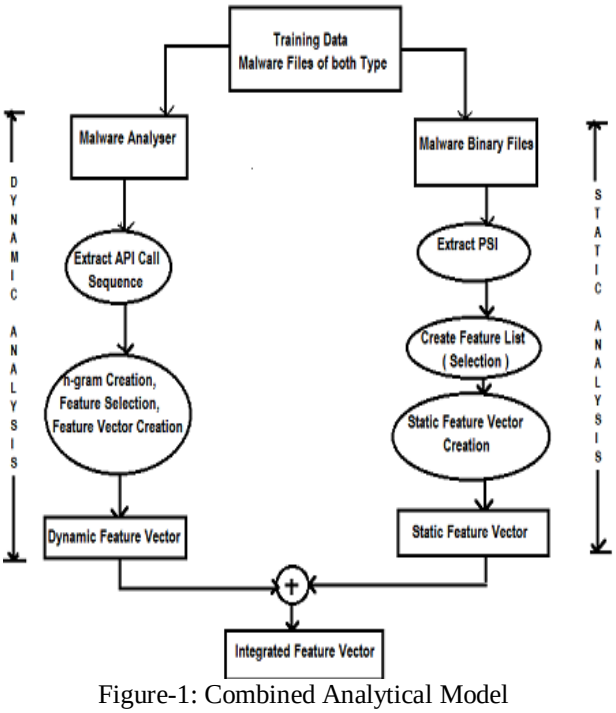

Figure-1: Combined Analytical Model

A. Static Feature Vector Creation Algorithm

In this work we have extracted un-encoded strings or printable string information (PSI) from the binary files and used them as the static feature. All PSI are not significant as few of them might have been inserted by code obfuscation techniques. Hence, the extracted PSIs are processed to pick out only meaningful strings which may be helpful for classification. The count of each PSI is found out and PSI having a count value less than a predefined threshold value is eliminated. These selected strings constitute a global list of PSI named as feature list where each entry in the list is termed as a feature. Then a binary vector is created which contains true/false value to represent the match or mismatch of selected PSI with the sample malware and benign files.

Algorithm-1 illustrates Static feature vector creation Algorithm which takes the Sample malware and benign files or sample Dataset (D) as input and produces the feature vector as output which can further be used as input to different learning and classification algorithms.

Algorithm-1:

1. *Start.*
2. *repeat for* each f_i in *D*
   - 2.1. *Extract and Process PSI from fi to generate significant PSI*
   - 2.2. *repeat for each (PSI j) for all j (j=no. of PSI in fi)*
   - 2.3. *Calculate Count (PSI j).*
   - 2.4. *if (Count (PSI j)>=threshold) then*
   - 2.5. *add PSI j to the feature List.*
   - 2.6. *end of for loop*
3. *end of for loop*
4. *Create a binary feature vector with each PSI in the feature list as attributes;*
5. *repeat for* each f_i in *D*
   - 5.1. *repeat for each PSIj in feature list*
   - 5.2. *if (PSIj ∈ fi) then*
   - 5.3. *Set value of the attribute in the vector true.*
   - 5.4. *else*
   - 5.5. *Set value of the attribute in the vector false.*
   - 5.6. *end of for loop*
6. *end of for loop*
7. *stop*

B. Dynamic Feature Vector Creation Algorithm

Dynamic analysis is used to extract the API calls made by a binary file while in execution. In this work Cuckoo malware analyser is used to run and analyse malware files and generate analysis result of the malware behaviour while in execution. Information about API calls, registry modifications, heap memory address, process address etc. are maintained in a log file. Mere finding the presence or absence of API calls in log file may not be enough for malware detection because the same set of API calls might be used by attackers in code obfuscation techniques. Hence a better option is to consider the API call sequence. We used API-call grams, n-gram based method to analyse

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 1, January 2016

the call sequence. We have considered 3-API –call-grams and 4-API-call-grams for simplicity and to get more similar n-grams between files. Hence, the set of 3 and 4 API-call-grams are generated for each file from the log file. The grams having count less than threshold are eliminated to make the process more efficient. We can represent both these type of call grams in the form of a two dimensional table where each entry is an API-call-gram having more frequency than the threshold from the n-gram set corresponding to a binary file in the dataset. The selected API-call-grams are called as feature and the table is called as feature vector.

Algorithm-2 illustrates Dynamic feature vector creation algorithm. The algorithm takes the Sample malware and benign files or sample Dataset (D) as input and produces the Dynamic feature vector as output.

Algorithm-2:

1. *Start*
2. *repeat for each fi in D*
   - 2.1. *Generate log file and extract 3-gram, 4-gram API call sequence*
   - 2.2. *repeat for each 3-API-call-grams and 4-API-call-grams*
   - 2.3. *calculate count(3-API-call-gram) and count(4-API-call-gram)*
   - 2.4. *if (count(3 or 4-API-call-gram) > threshold) then*
   - 2.5. *add API-call-gram to the corresponding feature list.*
   - 2.6. *end of for loop*
3. *end of for loop*
4. *Create a binary feature vector with two attributes. Such as, 3-API-call-grams and 4-API-call-grams.*
5. *repeat for each f_i in D*
   - 5.1. *repeat for each 3-API-call-grams and 4-API-call-grams in feature list,*
   - 5.2. *if (API-call-gram is present in Table associated with f_i) then*
   - 5.3. *Set value of the attribute in the vector true;*
   - 5.4. *Else*
   - 5.5. *Set value of the attribute in the vector false;*
   - 5.6. *end of for loop*
6. *end of for loop*
7. *stop*

In this work, we have combined static and dynamic features to get the integrated feature vector which in turn is used in training phase and classification. Although there are so many machine learning algorithms for malware classification exist, we have chosen Support Vector Machine (SVM) and Random Forest (RF) methods because these are efficient techniques as specified by few previous research works related to this field.

IV. EXPERIMENT ENVIRONMENT AND RESULTS

In this work, we have used Debian-based Linux operating system, Ubuntu as the base for environmental set up. We know that in Linux, the powerful utility "strings", finds and display the printable strings in a given executable, binary, or object file. Hence, we have run the utility for each binary file and analysis output for each file is recorded into a file with the same name as the name of the binary file. We have statically analysed 537 virus files collected from Virus sign, Virus share & Malshare websites [12, 13, 14] and 390 clean files using string utility and identified PSI from each file and recorded in individual files. From the output file containing PSI, we have extracted all the strings of length greater than 8 bytes and using the algorithm we have created the static feature set. We got 3253 static features from our static analysis. The same set of binary files were dynamically analysed in a controlled environment using Cuckoo malware analysis system. The environment was set up on the same OS and the analyser system was configured to work with Virtual machine (VMware) [17] in which three Windows host machines were installed. The binary files were executed in these host machines. The analyser produced the log which contains information about the API call sequence. Then the dynamic feature vector was created following the Dynamic feature vector creation algorithm. In our experiment, 1722 number of 4-gram and 2026 number of 3-gram features were selected to create the feature vector. The static and dynamic feature vectors are concatenated to get the integrated feature vector which is used for classification. The WEKA machine learning tool [16] is used for classification.

Table: 1 shows the classification results of static, dynamic and Proposed CAM methods using SVM and Random Forest algorithms.

TABLE I: CLASSIFICATION ACCURACY RESULTS OF STATIC, DYNAMIC AND CAM APPROACH

Analysis Type / Classification Type	RF			SVM		
	TPR	FPR	Accuracy (%)	TPR	FPR	Accuracy (%)
Static Analysis	0.942	0.152	94.24	0.962	0.087	96.19
Dynamic Analysis	0.957	0.110	95.76	0.974	0.099	97.39
Combined Analysis (CAM)	0.969	0.059	96.99	0.982	0.039	98.21

V. CONCLUSION

In this paper we have presented a Combined Analytical Model that integrates both static and dynamic features for malware classification and detection. According to the experimental results, the proposed model has a better accuracy than individual static and dynamic analysis methods. We further showed that the SVM technique is more suitable to classify our sample malware dataset as compared to RF technique. The dynamic analysis is better than static analysis in either of these classification techniques.

To continue our work, we will collect more malware samples to extract more static and dynamic features and reduce the number of appropriate features to improve the efficiency of classification or accuracy of detection. Feature selection algorithms and dimension reduction techniques can be used to reduce the number of features.

Further, instead of using one classification technique we can apply some other classification method like Bayesian classification along with SVM technique to implement as a double classification to improve the accuracy in malware detection preserving the performance of the system in real time.

REFERENCES

[1] KASPERSKY LAB REPORT Financial cyber threats in 2014.
[2] R. Islam, R. Tian, L. M. Batten, and S. Versteeg. Classification of malware based on integrated static and dynamic features. Journal of Network and Computer Applications. vol. 36, pp. 646-656, 2013.
[3] Malware Analysis and Classification: A Survey Ekta Gandotra, Divya Bansal, Sanjeev Sofat Journal of Information Security, 2014, 5, 56-64.
[4] Egele, M., Scholte, T., Kirda, E. and Kruegel, C. (2012) A Survey on Automated Dynamic Malware-Analysis Techniques and Tools. Journal in ACM Computing Surveys, 44, Article No. 6.
[5] Tian, R., Islam, M.R., Batten, L. and Versteeg, S. (2010) Differentiating Malware from Clean wares Using Behavioral Analysis. Proceedings of 5th International Conference on Malicious and Unwanted Software (Malware), Nancy, 19-20 October 2010, 23-30.
[6] Rieck, K., Trinius, P., Willems, C. and Holz, T. (2011) Automatic Analysis of Malware Behaviour Using Machine Learning. Journal of Computer Security, 19, 639-668.
[7] Anderson, B., Storlie, C. and Lane, T. (2012) Improving Malware Classification: Bridging the Static/Dynamic Gap. Proceedings of 5th ACM Workshop on Security and Artificial Intelligence (AISec), 3-14.
[8] Nari, S. and Ghorbani, A. (2013) Automated Malware Classification Based on Network Behaviour. Proceedings of International Conference on Computing, Networking and Communications (ICNC), San Diego, 28-31 January 2013, 642-647.
[9] 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 CISIS'12-ICEUTE'12, Special Sessions Advances in Intelligent Systems and Computing, 189, 271-280.
[10] Islam, R., Tian, R., Battenb, L. and Versteeg, S. (2013) Classification of Malware Based on Integrated Static and Dynamic Features. Journal of Network and Computer Application, 36, 646-556.
[11] Yuxin Ding, Xuebing Yuan, Ke Tang, Xiao Xiao , Yibin Zhang. (2013) A fast malware detection algorithm based on objective-oriented association mining, Elsevier, computers & security, 3 9, 3 1 5-3 2 4
[12] http://virussign.com/downloads.html.
[13] Virus Share Malware dataset.2014. *http://virusshare.com*/
[14] Mal share sample malware dataset. http://malshare.com/
[15] Weka ML Tool: Data Mining OSS, http://www.cs.waikato.ac.nz /ml/weka/downloading.html
[16] VMware. Accessed 2014. www.vmware.com.

BIOGRAPHIES

Om Prakash Samantray got the M.Tech degree in Computer Science & Engineering from Biju Patnaik University of Technology, Odisha, India in 2010. Currently, he is pursuing Ph.D. in Computer Science from Berhampur University, Odisha, India. His research interests include information security, Computer network security, Data warehousing & mining and big data.

Dr. Satya Narayan Tripathy received his M.C.A. and Ph.D. degrees in Computer Science from Berhampur University, Berhampur, Odisha, India in the years 1998 and 2010, respectively. He has been teaching in the Department of Computer Science, Berhampur University since 2011. Currently, he is a Lecturer in the Department of Computer Science, Berhampur University. Dr. Tripathy serves on the advisory boards of several organizations and conferences. He is a Life Member of Computer Society of India (LMCSI), Life Member of Orissa Information Technology Society (LMOITS) and Member of several professional bodies. His research interests include computer network security, wireless ad hoc network, network security in wireless communication and data mining.

Dr. Susant Kumar Das received his Ph.D. degree from Berhampur University, Odisha, India in 2006. Dr. Das is currently a Reader at the Department of Computer Science. He is a life member of IEEE, ISTE, SGAT, OITS and member of several professional bodies. His research interests include Data Communication & Computer Networks, Computer Security, Internet & Web Technologies, Database Management Systems and Mobile Ad- Hoc Networking & Applications.

Binayak Panda got the M.Tech degree in Computer Science & Engineering from Biju Patnaik University of Technology, Odisha, India in 2010. Currently, he is pursuing Ph.D. in Computer Science from Berhampur University, Odisha, India. His research interests include information security, Software Engineering and Data warehousing & data mining.

DOI 10.17148/IJARCCE.2016.5171