ISSN:0975-9646

SVM Based Effective Malware Detection System

Smita Ranveer[#1], Swapnaja Hiray[*2]

[#]*Dept. Computer Engineering, Savitribai Phule Pune University,*
Sinhgad College of Engineering Pune, India

***Abstract*—Malware is coined as an instance of malicious code that has the potential to harm a computer or network. Recent years have encountered massive growth in malwares as existing signature based malware detection approaches are becoming ineffective and intractable. Cyber criminals and malware developers have adapted code obfuscation techniques which undermines the effectiveness of malware defense mechanism. Hence we propounded a system which focuses on static analysis in addition with automated behavior analysis in emulated environment generating behavior reports to investigate malwares. The proposed method uses programs as opcode density histograms and reduces the explosion of features. We employed eigen vector subspace analysis to filter and diminish the misclassification and interference of features. Our system uses a hybrid approach for discovering malware based on support vector machine classifier so that potential of malware detection system can be leveraged to combat with diverse forms of malwares while attaining high accuracy and low false alarms.**

Keywords—Behavior Analysis, Static Analysis, Opcode Extraction, Malware Detection, Support Vector Machine.

I. INTRODUCTION

The enormous increase of internet users and system users in any field is also followed by multiplicative rise in malwares and cyber-attacks caused by them. Malware is a term derived from malicious software. It is an instance of automated malicious code which has potential to subvert the function of the system. It is in a constant state of malignant evolution, finding new forms, disguises, and vectors to reach, intrude, and compromise its target systems. According to CISCO 2014 annual Threat Security Report [1], backdoors expanded the attack surface area and given a way to cybercrime. As the threats have become more matured and complex. CISCO evaluates 400,000 malware threats every day. Propagation of malware might result in havoc to privacy and security of users. Hence it is essential to have an efficient antivirus shield to the system.

In the light of existing antivirus solutions for tackling malware, basically there are two approaches which rely on the static or dynamic type of malware analysis employed for identifying features. The signature-based detection approach relies on static malware analysis. It documents unique patterned signature, exploring malware features by supervising the malicious code. In spite of the broad use of this method commercially, it is vulnerable to the malwares unseen previously, consistently fails to deal with zero day attacks. On the contrary, the heuristic (also known as behavior based detection approach) detection approach which uses the dynamic type of malware analysis to key out malware features based on the malicious behavior of executables observed on execution in an emulated environment. This method is able to confront the loopholes of signature based malware detection approach up to some extent. This method can precisely deal with the problem of unknown malware discovery arise due to code obfuscation techniques like code reordering, garbage insertion, variable renaming employed by malware designers to disguise their content. However, this detection approach generated an additional challenge of greater amounts of false alarms prohibiting the benign files from execution. This is novel and serious problem as each suspicious executable file is not malware. Behavior based detection approach is also time intensive. Each of the two approaches had some limitations. Since, some of the researchers invented hybrid approach for malware detection which attempts to cope with the weaknesses of both detection approaches. It is a new line of defense to augment efficient but porous antivirus defenses and less-reliable, more resource-intensive heuristics. Such a defense would allow the antivirus layer to block known threats, while keeping even the majority of new threats from reaching the systems.

Following the similar intuition, we proposed an effective detection technique removing the flaws of both existing detection systems against malwares which uses the hybrid approach. It mainly extends the idea of signature based methods in addition with automatic behavior analysis. Support Vector Machine (SVM), a supervised machine learning technique is employed for classification of malicious and benign softwares. Our system detects malware on the basis of two features first by opcode density of executables and system call features obtained by dynamically tracing the behavior of executables during run-time. We focus on reducing feature explosion in the original dataset space and lessen the false alarm rate by analyzing behavior of executables. We applied opcode feature filtering to prune irrelevant and most common opcode features. Our system automatically analyzes the behavior of each malware executables and generates reports exploring system call features. Further it trains and builds reference model of SVM classifier to validate test dataset discriminating malicious and benign executables given as input.

Rest of the paper is structured as follows: at first section II explores the summarized view of earlier studies. Section III gives the overview of proposed system; Further Section IV gives the brief architecture of proposed system following the step by step explanation of state of art. In this vein, performance metrics and discussion of results are briefed in section V and VI respectively. Finally concluding remarks are stated in section VII.

II. RELATED WORK

Extensive survey has been done in the domain of malware detection systems using both static and dynamic analysis. Moskovitch et al. [2] 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 better classification accuracy. However, for more than bigram opcode sequence the accuracy is decreased. Their further research in [3] states that 99% can be achieved on considering the malicious file percentage while their weakness observed on packed executables. Santos et al. [4, 5] used opcode n-gram sequences for categorizing malicious and benign files with different feature selection and classification algorithms. They obtained good detection rate while keeping low False Positive Rate (FPR). In [3, 6], 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. [7] 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. These systems have to deal with computational overhead when n-gram analysis is performed. In [8] presented an automated malware detection system which classifies malwares into their families monitoring their network behavior. Firdausi *et al.* [9] propounded a malware detection system which monitors the behavior of malicious files in controlled environment using a free online dynamic analysis tool named Anubis. 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 naïve bayes. As per [8, 9], J48 decision trees given better TPR, FPR and accuracy results in comparison with other classifiers. In [10], 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.

An automatic behavior analyzing system proposed by Rieck et al. in [11] 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. [12] presented a malware detection system which uses system call and their parameters values as the features and 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 [13] proposed a similar technique of clustering malware families using supervised machine learning technique. These detection approaches have high false positive rate. Malware developers have applied code obfuscation techniques. Our system monitors behavior of executables in controlled environment.

In this vein, we have chosen to reduce the computational overhead required when n-gram analysis is performed.

III. SYSTEM OVERVIEW

We proposed a malware detection system which uses a supervised machine learning approach for discovering malwares. The SVM based malware detection system extends the idea of signature based detection system with a combination of behavior monitoring approach. It utilizes static and dynamic analysis of malwares by taking the run time traces of the executables. It applies signature and behavior based methods parallelly for extracting opcodes and system call features and further classifiers are trained on the basis of these feature vectors generated. At first executables under investigation are in test environment with monitoring its behavior and runtime opcodes. The programs in the dataset are disassembled; data is parsed and represented using opcode density histograms gained through dynamic analysis. A support vector machine is used to create a reference model, which is used to evaluate two set of features, opcode and system call features obtained through behavior monitored. The reference model is constructed by configuring the SVM to perform an exhaustive search by traversing through all the features, searching for those opcodes that have a positive impact on the classification of benign and malicious software.

IV. PROPOSED MALWARE DETECTION SYSTEM

The proposed SVM based malware detection system being implemented with machine learning techniques is based on SVM classifier. It extends the notion of feature filtering [14] and attempts to improve the performance with addition of behavior detection mechanism [9]. The malware detection system being implemented can be best described as follows:

A. Dataset Preparation

The dataset is prepared by using two sets of executables one is malicious and benign executables. Benign files are system files of windows operating system taken from system32 directory or program files directory. The malicious executables were downloaded from the VXheavens website [15], which cover malwares such as virus, adwares, worms, Trojan horses, etc. These malwares perform a range of malicious activities such as back-door downloaders, system attack, fake alerts, fake warnings, ad-ware, and information stealer. Our system uses SVM classifier machine learning technique which implements two phases training and testing. We used part of dataset for training the classifier and part can be used for validation as test data, SVM assigns a benchmark with measured target value for each benign and malicious.

B. Feature Extraction

The program under investigation needs to be monitored during execution. Dataset executables are disassembled by using the debugging tool OllyDBG, which is an open source disassembler used to extract the assembly language code of each and every dataset executable. Opcodes are obtained by disassembling executables. An opcode is operational code, a machine language instruction that specifies the operation to be performed. The operands associated with each opcode are omitted. Further opcode

occurrences are measured and are parsed into density histograms for each executable. The next step is streamlining the extracted sequences of opcode in the same logical order as they appear in the executable file. Final step is to map density histogram for each dataset executables.

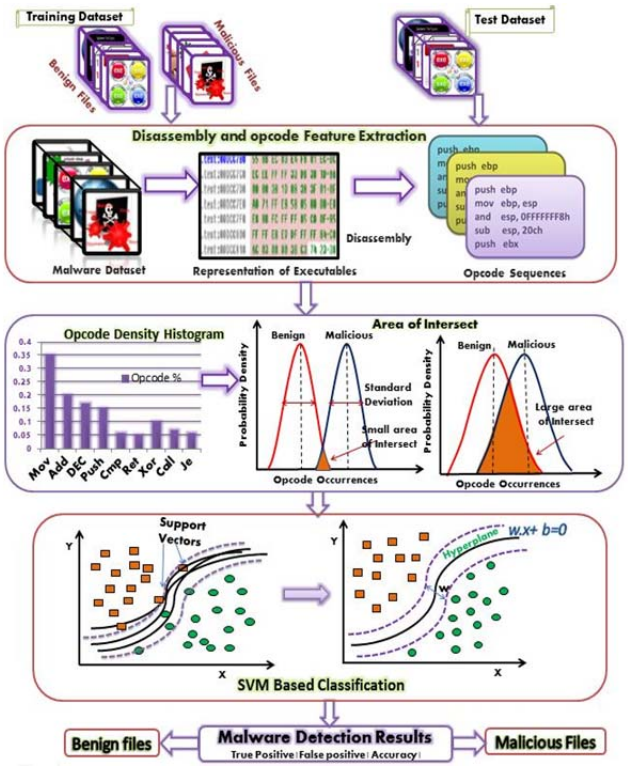

Fig. 1.Architecture of SVM based effective malware detection system

C. Behavior Monitoring

Each and every dataset file is executed in an automated environment using dynamic analysis parallelly so that the behavior of programs can be monitored. This performs automatic behavior analysis on execution of files in sandbox generating XML reports on the basis of behavior profile. System call features are extracted for discrimination of malicious and benign files.

D. Opcode Feature Filtering

An initial assessment of the data showed that the distribution of the various opcodes does not conform to any consistent distribution shape; rather it has been seen that feature explosion occurs in which some of the opcodes were irrelevant and very common. We employed feature filtering approach to reduce the explosion of features and diminish the interference and misclassification of benign and malicious softwares. The proposed system investigates Principal Component Analysis to find the subspace to determine the importance of the individual opcodes and weed out irrelevant opcodes.

1. Subspace Analysis using PCA: It determines the importance of the individual opcodes while ranking their relative importance as classification feature. It investigates the eigenvalues and eigenvectors in subspace. Principal Component Analysis (PCA) is a transformation of the covariance matrix and it is termed as:

$$C_{ij} = \frac{1}{n-1}\sum_{m=1}^{n}(X_{jm} - \bar{X}_i)(X_{jm} - \bar{X}_j)$$

Where,

C = Covariance matrix of PCA transformation;
X = dataset value;
$\bar{X}$ = Dataset mean;
n and m are data length;

PCA transforms the obtained covariance matrix into eigen vectors. It finds new principal components of opcodes over original set of opcodes and determines the number of PCs that correlate to greater than some threshold. Further the system searches for the most significant eigen values and eigen vectors.

E. SVM Based Classification

SVM is a technique for data classification; it can generate a nonlinear decision plane and classifies data which has non-regular distribution. It avoids attributes with greater numeric ranges dominating those with smaller numeric ranges and it avoids numerical difficulties during the calculation as kernel values usually depend on the inner products of feature vectors. SVM works in two phases training phase and testing phase.

During the training phase an SVM takes a set of input points in the form of Attribute Relation File Format (ARFF), each of which is marked as belonging to one of two categories, and builds a model representing the input points in such way that the points of different categories are divided by a clear gap that is as wide as possible. Thereafter, a new data point is mapped into the same space and predicted to belong to a category based on which side of the gap it falls on. A linear SVM model separates data belonging to different categories by using a hyperplane so that the distance from its nearest data point on each side is maximized. The kernel trick allows the SVM algorithm to become nonlinear to separate points by a hyperplane in a transformed feature space. The SVM is configured and trained to traverse through two types of features. At first SVM highlights those files whose system calls are having deviating behavior than normal behavior of benign files and other one is SVM pinpoints those files having opcodes that are having positive impact on the classification of benign and malicious software. Finally during testing phase SVM validates the dataset discriminating the files into sets of benign and malicious files.

V. PERFORMANCE METRICS

The proposed system statistically measured the performance of SVM based malware detection system for discriminating benign and malicious software. The statistical measures include True Positive Rate (TPR), False Positive Rate (FPR) and accuracy. TPR is defined as the ratio of the number of correctly detected malware to the total number of malware in the testing set.

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

FPR is termed as ratio of the number of normal files classified as malware to the total number of normal files in the testing set.

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

Our proposed system's detection accuracy is evaluated. It gives the ratio of the total number of normal files detected as normal and malware detected as malware to the total number of files in the testing set. The proposed system attains low FPR and accelerates detection accuracy.

VI. EXPERIMENTS AND RESULTS

The performance of the proposed system is measured and validated on the basis of experiments by using the dataset of various combinations of malicious and benign executables at the time of training and testing. We first used dataset of 75 benign files and 35 malware files for training and tested the system to compare our method, we have validated test dataset and measured the performance of the SVM based malware detection system on static, opcode based malware detection as well as on system call features, which is behavior based malware detection approach. Fig.2 and 3, show the obtained results in terms of TPR and FPR respectively. The conduction of tests showed that TPR and FPR results were improved when using the combination of both static and dynamic features. In terms of TPR, opcode based static approach yield highest 0.95 classification rate, while behavior based approach yielded 0.93 which was low in comparison with hybrid approach for the same test set. In terms of FPR, opcode based static approach yield 0.08 false alarms while behavior based approach yielded 0.3 which was high in comparison with hybrid approach for the same test set. The proposed system is able to attain a TPR up to 0.95-1 and FPR up to 0.03-0 with quite variation on changing the number of malicious files in training dataset. On observing these results, we have conclude that it is possible to reduce impact of countermeasures of static and dynamic methods, we can improve the performance of the system in terms of TPR while maintaining low FPR and notable rise in accuracy.

Besides, we evaluated performance of the system to see the effect of MFP among the training set files. We tested the dataset combinations by creating five levels of MFP in the training set (5, 10, 15, 30, and 50%) and measured the variations in the TPR and FPR of the system as depicted in Fig. 4. We observed that the system's performance was generally low and dropped significantly for 5%, 15% and 50% MFP in the training dataset. Additionally, it has been seen that the FPR grows with the increasing level of the MFP in the training dataset.

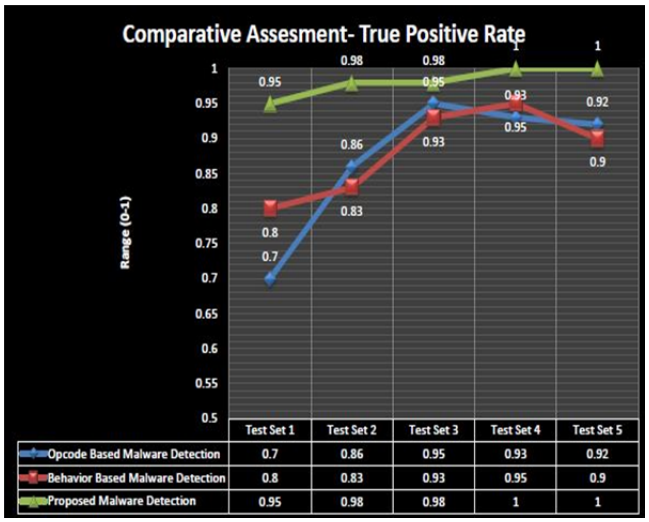

Fig. 2. Performance evaluation in terms of TPR

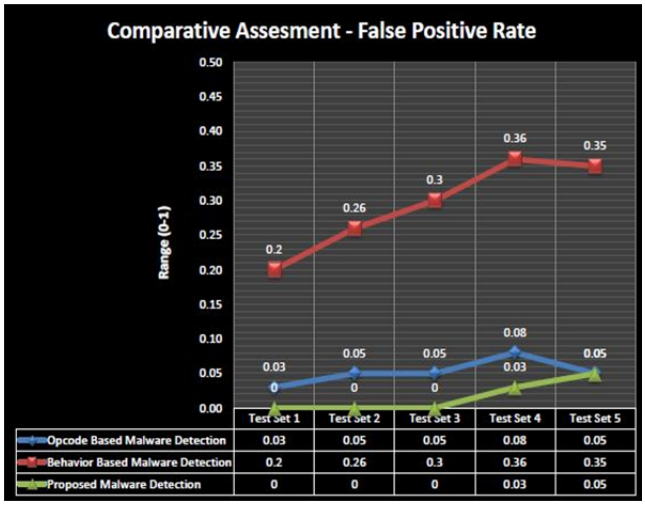

Fig. 3. Performance evaluation in terms of FPR

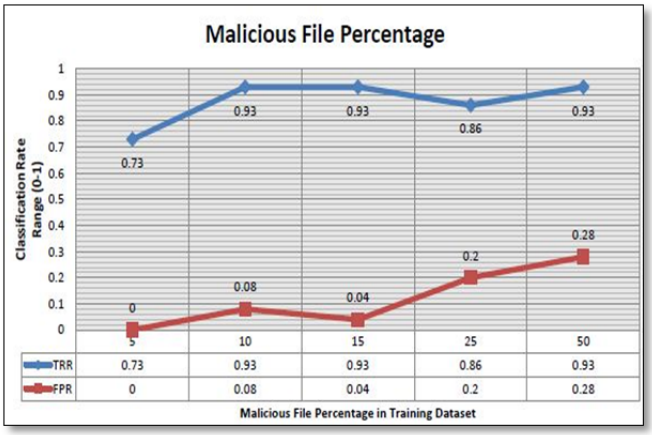

Fig. 4. Performance evaluation on the basis of MFP in training dataset.

In particular, the best overall results were obtained by SVM trained with large number of training files. The obtained results validate our initial hypothesis that building an unknown malware detector based on hybrid approach is feasible. The hybrid approach for malware detection using machine learning classifier (SVM) achieved high performance in classifying unknown malware

VII. CONCLUSION

In this paper, we present a method for malware detection. Specifically, we propound a system for representing malware that relied on opcodes density histograms in order to construct a vector representation of the executables and also system call features obtained by executing the malwares in automated environment. Our system implements SVM as a means of discovering malware consequently reducing the training efforts. Our malware detection system eliminates the flaws of both signature based and behavior based detection techniques incorporating the hybrid analysis for efficient detection of malware. Our experiments show that this method provides a good detection ratio of unknown malware while keeping a low false positive rate. The future development of this malware detection system will be concentrated on facing packed executables.

REFERENCES

[1] Cisco labs, "CISCO Internet Threat Security Report 2014".

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] R. Sekar, M. Bendre, D. Bollin

[8] eni, 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. 144–155.

[9] 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.

[10] Firdausi, I., Lim, C. and Erwin, "Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection," *Proceedings of* 2*nd International Conference on Advances in Computing, Control and Telecommunication Technologies* (*ACT*), Jakarta, 2-3 December 2010, 201-203.

[11] Tian, R., Islam, M.R., Batten, L. and Versteeg, S. (2010) "Differentiating Malware from Cleanwares Using Behavioral Analysis," *Proceedings of* 5*th International Conference on Malicious and Unwanted Software* (*Malware*), Nancy, 19-20 October 2010, 23-30.

[12] 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.

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

[14] Lee, T. and Mody, J.J. "Behavioral Classification" *Proceedings of the European Institute for Computer Antivirus Research Conference* (EICAR'2006).

[15] Philip OKane, Sakir Sezer, Kieran McLaughlin, and Eul Gyu Im, SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection, IEEE Transactions on Information Forensics and security Vol. 8, No. 3, MARCH 2013.

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