2019 4th International Conference on Information Systems and Computer Networks (ISCON)
GLA University, Mathura, UP, India. Nov 21-22, 2019

Malware Detection based on API Calls Frequency

Vidhi Garg
Computer Engineering Department
Marwadi University
Rajkot, India
vidhi7168@gmail.com

Rajesh Kumar Yadav
Computer Engineering Department
Marwadi University
Rajkot, India
rajg1012@gmail.com

Abstract— **Nowadays, one of the main dangers to computer system's security is the malware, a piece of software or a computer program that is designed to detriment and penetrate computers without the owner's permission. Traditional signature-based and anomaly-based malware detection approaches are still in use. However, the signature-based detection approach fails for new anonymous malware. In the anomaly-based detection, if the malicious activity behaves like a normal activity, the detection treats it as a normal one. Today's attackers are using various obfuscation techniques which have become a great challenge for the detectors to detect the malicious content with the traditional malware detection techniques. In this research, supervised learning algorithms are used to detect malware using the concept of API Calls usage frequency in a portable executable format. The experimental results provide the accuracy of 93% in distinguishing malware from benign files.**

Keywords— Machine Learning; Malware Analysis; Signature-based detection; Anomaly-based detection; Supervised Learning

I. INTRODUCTION

The current decade of Data Science and Information Technology is suffering from threats of malware, i.e., malicious software. Malware is a code or software installed on a system without the owner being aware of it with a purpose to steal personal data from the system. This, in turn, threatens the computer's security, it is dangerous because computers are widely used in the domain of education, communication, medicines, banking, entertainment, etc. The malware can reach into the systems in many manners; the most frequent way is to download from the Web. Once the malware is able to enter the systems, based upon the functioning of malware, they infect the system. Sometimes, the malware does not completely infect the system, rather, they affect the system's performance or create an overload of processes.

Malware detection is the procedure of finding whether a code is actually benign or it has some malicious content [1]. As malware has various kinds, behaviors and risk levels, the same malware detection techniques cannot be used for all malware. It is infeasible to have only one malware detection approach to efficiently handle all malware. Thus, security researchers are using two popular malware detection techniques for malware analysis. One technique is *anomal-based detection* and the other is *signature-based detection* [2]. Signature-based malware detection approach sometimes also called misuse detection, since it maintains the database of known software. It is an approach that detects malicious programs by comparing the signature against the database [3]. This approach can identify a familiar attack correctly. To detect a malicious content in the code, the signature-based detection technique searches for a previously defined signature in the code. The deprivation of signature-based malware detection approach is the ineffective performance while dealing with unknown attacks since signatures are not provided for these attacks. Anomaly-based detection approach analyzes user's activity and the statistics of a process in usual conditions, and it verifies that the system is functioning in accordance with pre-established normal behavior. When an activity leads to dissimilar nature from the usually recognized behavior in a system, an attack can be diagnosed. The disadvantage of anomaly-based detection approach is that it requires updating huge amount of information that characterized the system behavior in normal profile. If the malicious activity behaves like the normal activity then anomaly-based detection approach can treats it as a normal activity. However, if the normal activity behaves abnormally it can alarming it as malicious thus, in this approach, false positive rate (FPR) is high [4, 5].

Today, the attackers pack the malicious code in such a way that it is very difficult for the traditional malware detection approaches to detect malwares. Thus, security researchers used various advanced approaches to detect unknown malwares. One such approach *is machine learning based malware detection approach.* It involves processing and learning a large number of examples to obtain useful, unpredictable, and previously undefined information, patterns, and trends from a huge amount of data, which are saved in databases [6]. This approach extracts features of various activities and then classifies. Machine learning techniques are used to group various types of malicious activities into profiles and then use these profiles to identify an attack whenever it arises [7, 8]. Supervised, semi-supervised and unsupervised are three kinds of machine learning methods.

The main objective of this research is to analyze the API Calls of portable executable file which can be leveraged to find unknown malware. It is wholly based on Machine learning based malware detection technique which helps antivirus vendor to detect unknown and polymorphic malware. Rest of this paper is formulated as follows: Section 2 briefly shows the literature survey in the domain of machine learning based malware detection approach. The proposed architecture for malware detection is explained in section 3. In section 4 performance evaluation of proposed architecture is elaborated. Section 5 chapter represents the conclusion and future work.

II. LITERATURE REVIEW

For unknown malware detection and malware classification, it is important to understand the malware features and various techniques used to classify them. In this section, the survey of existing work related to malware

978-1-7281-3651-6/19/$31.00 ©2019 IEEE

detection and classification techniques based on the API Calls is provided.

A. Machine Learning based Approaches

This approach extracts features of various activities and then classifies them as normal or malicious. Machine learning techniques are used to group various types of malicious activities into profiles and then use these profiles to identify an attack whenever it arises. The survey of malware classification using machine learning categories are described follows:

Schultz *et al.* [9] used three types of features- dll used by the exes, dll function calls made by the exes and number of difstinct function calls in each dll. Further, they implemented network level email filter that employed the naïve Bayes (NB), Multi-NB, RIPPER algorithms to discover malicious exes prior to them reaching the users through their mail. Their technique can either block the malware or wrap the malicious exe. They achieve 97.76% accuracy with the multi naïve Bayes method.

Sathyanarayan *et al*. [10] used the static analysis for the API extraction and presented a method for generating signature to identify unknown malware based on their API Calls. They created a single signature for a distinct class of malware. These API Calls' signature were compared with unknown malware's API Calls to identify whether the unknown malware belongs to a distinct class or not. The obfuscation of malware could affect the accuracy of the system.

Wang *et al.* [11] presented virus identification technique on the basis of the API Calls sequence. Their technique designed Bayes algorithm for the identification and detection of apprehensive working by analyzing the common API Calls used by virus detection. The precision of the Bayes algorithm has been validated by using the training and testing samples and found to be 93.01%.

Tian *et al.* [12] presented an effective classification technique for malware classification based upon the printable strings of the executables. They chose five algorithms support vector machine (SVM), NB, decision tree (DT), IB1, random forest (RF) to classify the malware and also used the boosting technique to determine if the results improved or not. They statically extracted the features with the IDA PRO after unpacking the malware and using k-fold cross validation. They achieved 97% overall accuracy.

Tian *et al.* [13] extracted the API Calls from both the normal and malicious records for modeling the various behavior patterns of these files and distinguish the malicious files from the benign files and also classify the malware family by using a binary SVM, RF, DT and IBK classification models and compared their accuracy with and without Adaboost classifier.

Natani and Vidyarthi [14] proposed a scheme to detect malware by mapping API Calls functions to malicious behaviors. The frequency of API Calls was exploited to label file as malicious or benign. The experimental results showed that the true positive rate (TPR) at 1000 iterations using EnsembleAdaBoost was 92.31%, however, FPR was extremely high as compared to EnsembleBag. However, at 5000 iterations AdaBoostM1 generated similar result as EnsembleBag. The proposed method also works for rootkits which often use Native API functions.

Kawaguchi and Omote [15] classified malware efficiently using initial behavior APIs using SVM, C4.5, RF, NB, k-nearest neighbor (kNN) by using activity logs of APIs of FFRI dataset 2014 by dynamically analyzing them and creates the dataset using Cuckoo sandbox and achieved 83.4% accuracy with RF. The proposed method has three states: API extraction, learning and classification phase.

Shijo and Salim [16] presented an integrated dynamic analysis and static analysis approach to enhance the accuracy of detection of malware. Application signatures are used as unique identification feature are classified using SVM classifier and RF classifier. The results spresented that for static analysis the accuracy is 95.8% and for dynamic analysis the accuracy is 97.1% and 98.7% in the combined analysis

Fan *et al.* [17] utilized hooking. The behavior records are used to train the classification model and then classify the malware from the benign using these trained models. The classification algorithm used are NB, J48 and SVM. Attribute selection method is applied to reduce the features so that efficiency can be improved. Detection rate of J48 and NB are upto 95% and SVM shows the worst performance of upto 89% in terms of detection rate.

Bayer *et al.* [18] proposed a novel approach that make clusters which are used to discover a partitioning from a set of malware so that their subset show equivalent behavior. The system starts with dynamic analysis of each malware sample which is enhanced with taint tracking and network analysis. These profiles are the input for the clustering algorithm. The proposed approach accurately recognizes the malware that behaves in similar fashion.

Nakazato *et al.* [19] method focuses on the working of individual thread which are invoked by the original process. In the proposed method, malware samples are classified using clusters according to the malware's API Calls. The frequencies of each API call sequences are calculated and TD-IDF score is given to each API sequence which is used for visualizing the malware characteristics.

Shuwei *et al.* [20] proposed clustering based malware detection method based upon Shared Nearest Neighbour (SNN) and fixed length vectors are produced by taking the frequencies of various system calls as input. To estimate the exact distance between the samples, Euclidean distance is calculated. The proposed method has three phases: Feature Extraction (calculate frequencies of various system calls), Calculating similarity (using Euclidean distance, kNN and SNN) and Clustering (based on SNN density).

III. PROPOSED FRAMEWORK

This section demonstrates the design of proposed solution through Malware Detection based on API Calls frequency. It briefly explains how the malware detection is implemented by using supervised machine learning algorithms.

A. Framework for Malware Detection

This approach comprises Data Preprocessing , Extraction of API Calls of various dlls, Feature Selection and Supervised learning algorithms. Fig. 1 shows proposed architecture of malware detection system. All malware are downloaded from dasmalwerk website and in portable executable file format. The type of malware is checked in virustotal. As all the malware are Trojan horse, there is no need to classify them into categories. All the benign files are also in portable executable format. The malware feature extraction process has been done in virtual machine only. In this research, the major focus is on the malicious and benign files.

Data Preprocessing: In data preprocessing, unpacking is done where it is identified whether Malware or benign file is packed or unpacked. If the file is found to be packed then different unpacking tools and techniques are used to unpack it. After this, the unpacked files are provided to feature extraction process which extracts the features of files. In case of unpacked files, the files are directly passed for feature extraction process. PEid [21] and OllyDbg memory map tool are used to detect whether the file is packed or unpacked.

Feature Extraction: For extracting the features, API Calls have been recorded through IDA-Pro during static analysis. All API Calls have been retrieved and regarded as a feature vector. The extracted APIs from the files have been presented into a database which will be used for feature vector generation. 180 API Calls from 9 most frequent dlls have been logged. The API function Calls have been copied in the global list which has the API Calls of every input file. After copying the API function Calls of all 35000 files, the unique API Calls have been found. After calculating the unique API Calls, the feature vector has been generated. The absence or presence of unique API in each record has been checked. The feature vector is created as follow:

$$\text{FEATUREVECTOR}_{\text{MALWARE}_i} \begin{cases} 1 & \text{If } \text{API}_i \text{ is in i} \\ 0 & \text{OTHERWISE} \end{cases} \quad (1)$$

To create the feature vector, feature vector generation algorithm has been used which maps the API database used by all samples and API used by each individual sample and calculates the frequencies of API Calls used by the malware.

Feature Selection: 45 features have been selected by using random forest to determine the variable importance function to get the importance of each feature so that highly correlated features can be removed. The dataset has been divided into training and testing components. The training component has 70% of the instances of the whole dataset and testing component has 30% of the instances.

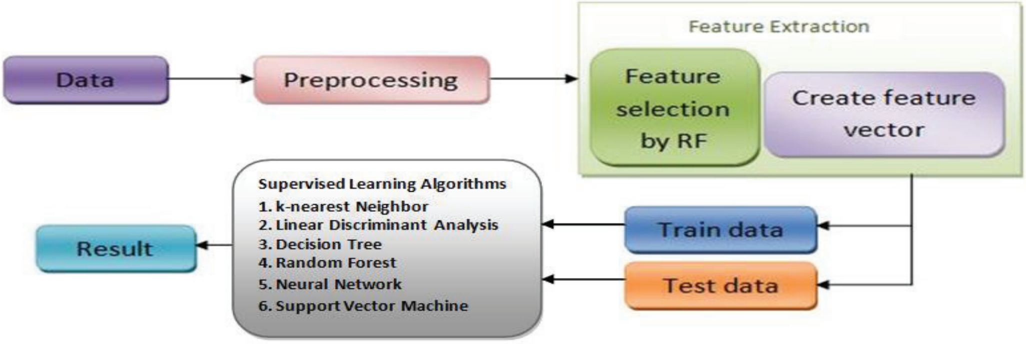

Fig. 1. Architecture of proposed malware detection system

Supervised Learning Algorithms: This technique uses the training data to predict a hidden pattern [22, 23]. The training data is a combination of input variables and output classes or labels. Supervised machine learning models make predictions of the classes for the unseen new data. Various supervised machine learning examples are classification, regression and attribute prioritization.

This research has been used kNN, linear discriminant analysis (LDA), DT, RF, neural network (NN), SVM models. Table 1 contains the various machine learning models with optimal tuning parameters that have been trained on the dataset.

TABLE I. SUPERVISED LEARNING ALGORITHMS

Model	Required Package	Tuning Parameter
kNN	class	k=5
LDA	MASS	none
DT	rpart	none
RF	random forest	mtry=3,ntree=500
NN	nnet	size=10
SVM	kernlan	kernel=”radial”, gamma=0.1, cost=10

IV. EXPERIMENTAL RESULTS

The current work indents to discover unknown malware with the supervised learning technique. 35000 malicious and benign files each containing API Calls approximately 100 to 700 in number have been analyzed. The main objective of this research is to identify malicious files through API Calls usage and their occurring frequencies. The algorithms have been implemented in R (environment for statistical computing and graphics) [24].

A. Performance Evaluation Metrics

Accuracy: The complete amount of observations properly categorised separated by the complete amount of observations in the whole dataset [25] is known as accuracy.

$$\text{Accuracy} = \frac{TP+TN}{P+N} \quad (2)$$

Sensitivity: Also called the TPR or the recall measures the ratio of correctly recognized positives.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad (3)$$

Specificity: Also known as the TNR (true negative rate) calculates the ratio of correctly recognized negatives.

$$\text{Specificity} = \frac{TN}{TN+FP} \qquad (4)$$

Where P: total positive numbers.

N: total number of negatives.

TP: positive value which is predicted positives.

FN: positives which are predicted as negatives.

FP: negatives which are predicted as positives.

TN: negative values which are predicted as negatives.

A comparison of performance evaluation metrics (measured in %) of different model has been given in Table 2.

TABLE II. COMPARISON OF PERFORMANCE EVALUATION METRICS

Model	Accuracy	Sensitivity	Specificity
kNN	73.33	70.04	94.54
LDA	76	78.62	95.22
DT	84	81.95	96.74
NN	92.67	91.27	98.46
RF	90.67	87.67	98.7
SVM	93	91.51	98.75

Fig. 2 shows the graph of accuracy of each model. The highest accuracy is given by SVM and least accuracy is given by kNN.

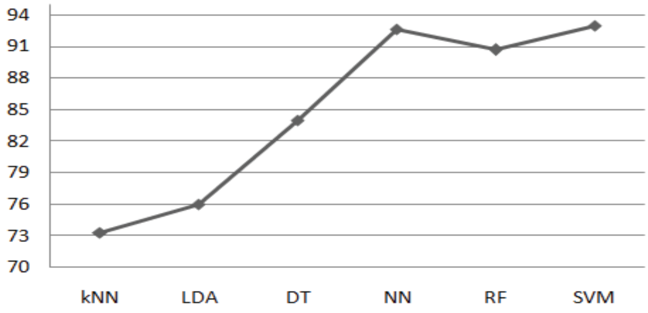

Fig. 2. Performance evaluation with accuracy

Fig. 3 shows the classification model vs. sensitivity graph. The highest sensitivity of 91.51% is given by SVM and least sensitivity is given by kNN.

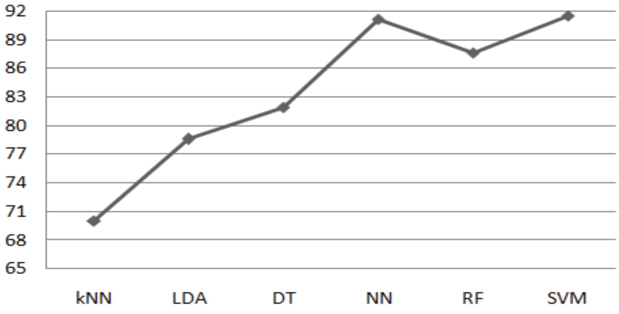

Fig. 3. Performance evaluation with sensitivity

Fig. 4 shows the graph of specificity of each model. The SVM achieved 98.75% specificity and least specificity is given by kNN.

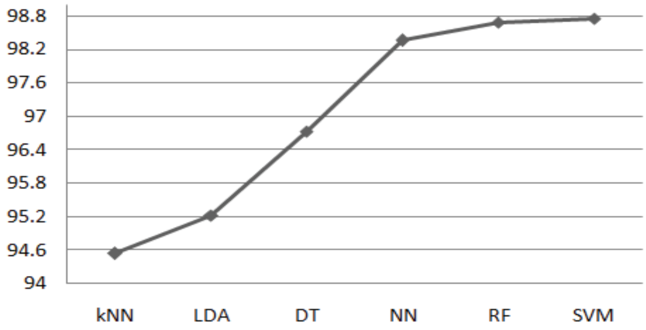

Fig. 4. Performance evaluation with specificity

V. CONCLUSION AND FUTURE SCOPE

Malware detection is an expending research domain owing to the increase of malware with each passing day. Signature-based malware detection approach is facing problems for detecting unknown malware for example, zero day attack. Anomaly-based malware detection approach treats a malicious activity as normal if it behaves like a normal activity. Hence, it is imperative to use traditional malware detection approaches alongside unconventional approaches to diagnose new malware. Machine learning is an appropriate technique to complement classical malware detection approaches. This research acquired API Calls from 35000 malicious and benign files and categorized them into malicious and benign files. In this research 93% accuracy is achieved.

In the future, dataset will be input to additional machine learning approaches like unsupervised and semi-supervised machine learning algorithms. Furthermore, this research will be extended to focus on APIs sequence and try to classify malware into categories

REFERENCES

[1] M. Christodorescu, S. Jha, S.A. Seshia, D. Song and R.E. Bryant, “Semantics-aware malware detection,” In Security and Privacy, 2005 IEEE Symposium on, pp. 32-46, 2005.

[2] H.D. Huang, C.S. Lee, M.H. Wang, and H.Y. Kao, “IT2FS-based ontology with soft-computing mechanism for malware behavior analysis”, Soft Computing. vol. 18, no. 2, pp. 267-284, 2013.

[3] P. Vinod, R. Jaipur, V. Laxmi and M. Gaur, “Survey on malware detection methods”, In Proceedings of the 3rd Hackers’ Workshop on computer and internet security (IITKHACK’09), pp. 74-79, 2009.

[4] M. Alazab, R. Layton, S. Venkataraman and P. Watters, “Malware detection based on structural and behavioural features of api calls”, 2010.

[5] M.A. Jerlin and C. Jayakumar, “A Dynamic Malware Analysis for Windows Platform - A Survey”, Indian Journal of Science and Technology, vol. 8, no. 27, 2015.

[6] S. Kilgallon, L. De La Rosa and J. Cavazos, “Improving the effectiveness and efficiency of dynamic malware analysis with machine learning”, In Resilience Week (RWS), IEEE pp. 30-36, 2017.

[7] M. Barreno, B. Nelson, A.D. Joseph and J.D. Tygar, “The security of machine learning”, Machine Learning, vol. 81, no. 2, pp.121-148, 2010.

[8] I. Santos, Y. Penya, J. Devesa and P. Bringas, “N-Grams-based file signatures for malware detection,” In: Proceedings of the 11th International Conference on Enterprise Information Systems (ICEIS), Volume AIDSS, pp. 317–320, 2009.

[9] M. Schultz, E. Eskin, F. Zadok and S. Stolfo, “Data mining methods for detection of new malicious executables”, Proceedings 2001 IEEE Symposium on Security and Privacy, 2001

[10] V. S. Sathyanarayan, P. Kohli and B. Bruhadeshwar, “Signature Generation and Detection of Malware Families”, Information Security and Privacy Lecture Notes in Computer Science, pp. 336-349, 2008.

[11] C. Wang, J. Pang, R. Zhao. and X. Liu, “Using API Sequence and Bayes Algorithm to Detect Suspicious Behavior”, 2009 International Conference on Communication Software and Networks, 2009.

[12] R. Tian, L. Batten, R. Islam and S. Versteeg, “An automated classification system based on the strings of trojan and virus families”, 2009 4th International Conference on Malicious and Unwanted Software (MALWARE), 2009.

[13] R. Tian, R. Islam, L. Batten and S. Versteeg, “Differentiating malware from cleanware using behavioural analysis”, 2010 5th International Conference on Malicious and Unwanted Software, 2010.

[14] P. Natani and D. Vidyarthi, “Malware Detection Using API Function Frequency with Ensemble Based Classifier”, Communications in Computer and Information Science Security in Computing and Communications, pp. 378-388, 2013.

[15] N. Kawaguchi and K. Omote, “Malware Function Classification Using APIs in Initial Behavior”, 2015 10th Asia Joint Conference on Information Security, 2015.

[16] P. Shijo and A. Salim, “Integrated Static and Dynamic Analysis for Malware Detection”, Procedia Computer Science, vol. 46, pp. 804-811, 2015.

[17] C.I. Fan, H.W. Hsiao, C.H. Chou and Y.F. Tseng, “Malware Detection Systems Based on API Log Data Mining”, 2015 IEEE 39th Annual Computer Software and Applications Conference, 2015.

[18] U. Bayer, P.M. Comparetti, C. Hlauschek, C. Kruegel and E. Kirda, “ Scalable, Behavior-Based Malware Clustering”, In NDSS, vol 9, pp. 8-11, 2009.

[19] J. Nakazato, J. Song, M. Eto, D. Inoue and K. Nakao, “A Novel Malware Clustering Method Using Frequency of Function Call Traces in Parallel Threads”, IEICE Transactions on Information and Systems, vol 94, no. 11, pp. 2150-2158, 2011.

[20] W. Shuwei, W. Baosheng, Y. Tang and Y. Bo, “Malware Clustering Based on SNN Density Using System Calls”, Cloud Computing and Security Lecture Notes in Computer Science, pp. 181-191, 2015.

[21] M. Sikorski and A. Honig, “Practical malware analysis: the hands-on guide to dissecting malicious software”, San Francisco: No Starch Press, 2012.

[22] S.B. Kotsiantis, I. Zaharakis and P. Pintelas, “Supervised machine learning: A review of classification techniques” , 2007.

[23] R. Gentleman, W. Huber and V.J. Carey, “Supervised machine learning”, In: Bioconductor Case Studies, Springer New York, pp. 121-136, 2008.

[24] B. Lantz, “Machine learning with R”, Packt Publishing Ltd., 2013.

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