iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 7 **Issue:** VI **Month of publication:** June 2019

**DOI:** http://doi.org/10.22214/ijraset.2019.6010

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, Jun2019- Available at www.ijraset.com

# Enhancing Malware Detection with Static Analysis using Machine Learning

Fatema Hamid[1], Seema Joshi[2]
[1,2]GTU-Graduate School of Engineering and Technology, Gandhinagar

***Abstract:*** ***As malware increases nowadays, it is necessary to safeguard your system from the malware. Malware is being protected by traditional methods but it only protects system from the malware whose signature is known. So we aim to prepare a software which detects the malware (signature is unknown) by the malware( signature is known) and after detecting the malware, steps must be taken to retrain the model and disallow the malware to compromise the system. For detecting the malware already known machine learning methods such as random forest classifier, multi view ensemble learning etc. will be used.***

***Keywords: Malware Analysis, Static Malware Analysis, Machine learning, Cyber security.***

## I. INTRODUCTION

Malware is defined as the malicious software whose intense is to harm or damage the computer system without the owner's permission [1]. Malware can be classified as standalone malware and file infectors malware [1]. Malware can also be classified as per the action i.e. worms, backdoors, Trojans, rootkits, spyware, adware etc. [1,2] . No class of malware is unique, Malware either shares the property of one or the other[3].s

Malware came into existence since the computer started. Malware increases nowadays with rapid amount. So one of the most prevalent threat faced by the IT community is malware and so steps must be taken by IT professional accordingly. The latest malware i.e. Mirai malware that performed DDOS attack on IOT devices, the wannacry malware that took advantage of eternal blue vulnerability to collect ransom from the victim that left the world shocked[19]. Malware increases as it is available in dark net easily. Only one percent of malware is seen in more than 10 computers, majority of malware are seen in more than one computer. The traditional method for identifying the malware is not effective for zero day malware or the unknown malware whose signature is not known. The latest malware where not identified due to traditional method being used. Again signature making for a malware is a long process; first identify the malware and then forming signature and making it available to customer for protection from the malware identified before. During that period the malware can affect many of the computers and so to protect against the unknown malware and the zero day malware we uses machine learning technique [17].

Malware analysis is an art of understanding malware working, how to identify it and how to remove it [2]. Two different techniques for malware analysis are static and dynamic malware analysis [2,3,4] . In static malware analysis, sample of malware is not executed and hence it is safe analysis. Here the malware analyst reverses the malware code. Disassembling of code gives information about all execution path taken by malware [2,4] . In dynamic malware analysis, the code is been executed and the runtime behavior is being analyzed but it is analyzed within virtual environment [2,4] . Static analysis is the safe analysis and the dynamic analysis require more resources, it requires a virtual environment for analysis and malware after detecting the virtual environment can change its behavior [2,4]. In recent years researcher have applied machine learning for malware analysis for detecting the unknown malwares whose signature is not known beforehand [8]. It uses dataset for the same. The dataset must be divided into training and testing dataset. But in real world the known samples are used to train the model so that it can identify the unknown malware.

In order to train the model we require to first get the features of the samples. Features can either be obtained statically or dynamically [4,8].

Different features used are printable string information, n-gram, op-code sequence, API calls, PE header, format feature, etc.

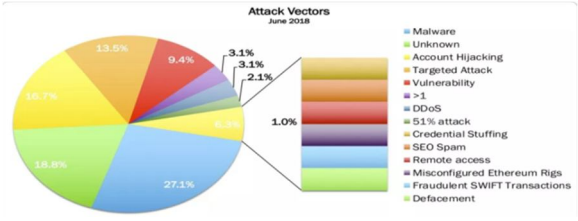



**Figure 1: Attack Vector [12]**

©IJRASET: All Rights are Reserved

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177*
*Volume 7 Issue VI, Jun2019- Available at www.ijraset.com*

## II. LITERATURE SURVEY

In this section we represent the related work done by different researcher.

In [4] the author has done survey on static features such as byte n-gram, opcode-ngram, PE32, API calls and concluded that the result obtained from navie bayes and bayes network is poor. In [8] the author combines different features such as byte n-gram, opcode n-gram and format features to obtain the nature of the file. In [4] the author analysis the malware on the basis of behavior and the features used is based on API call. In [6] the author used string as feature and concluded string as a powerful feature for malware classification. In [5] the author combined static and dynamic features because malware author cannot obfuscate the entire feature but at the end concluded that malware change its behavior after detecting virtual environment. In [9] the author used n-gram as a feature because N-gram implicitly captures the occurrence of the longest substring and features captured are implicit so malware author cannot fool the n-gram analyst. In [9] the author concluded that we must update the training set with both the malicious as well as benign files in order to get efficient result later. In [11] it states that the one must be protected from malware at first sight. In [13] it uses entropy as a feature because entropy of a packed or encrypted file is more, so one can identify the malware if is obfuscated or packed.

## III. PROPOSED SYSTEM

We begin with an overview of the proposed system, followed by a discussion of the generation of our data sets, features we extracted, and finally the set of machine learning classifiers we use to evaluate our methodology.

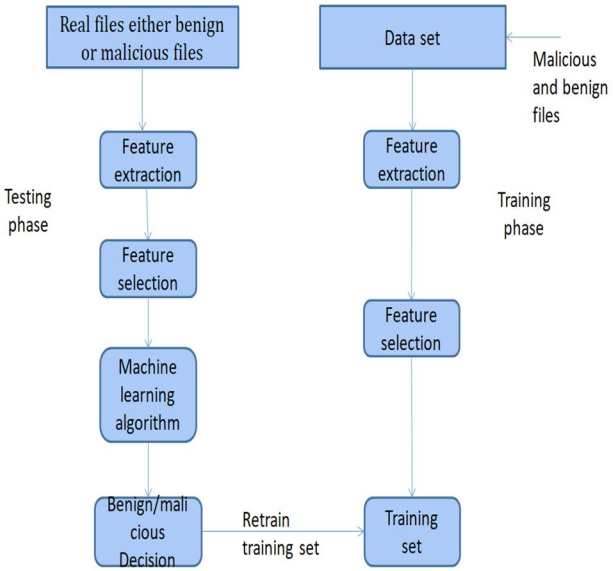



**Figure 2: proposed system**

The system has two phase the training phase and the testing phase. In training phase, we will have datasets both the benign as well as new malwares to get the effective results.

Features extracted from files are static features i.e. n-gram, string, PE format features, function length frequency, entropy. We will combine these features and will have multi view of features for efficiency. Output of the feature extracted is the input to classification algorithm. We use the classification algorithm which has pre-defined label of benign or malware files. Thus we make the data set from the feature extracted to supply it to the classification algorithm. Thus we obtain a model to classify a file as malware or benign on later stage. The second phase is the testing phase where input supplied is unknown for training set. The features here extracted are obtained same as obtained in training set. Now the model is supplied here to classify the file as malware or benign file.

We uses different features such as printable string information, PE format features and entropy. For machine learning algorithm we uses WEKA tool. It is used for finding the accuracy of the dataset obtained by doing cross validation testing.

©IJRASET: All Rights are Reserved

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177*
*Volume 7 Issue VI, Jun2019- Available at www.ijraset.com*

## IV. IMPLEMENTATION

We perform experiment on 997 WinEXE malware samples taken from VirusShare and VXHeaven and benign samples taken from windows OS manually.

We experimented on different features such as n-gram, printable string information (PSI), PE format features, entropy. Table 1 shows combination of different features for data set and the accuracy and false positive rate.

*A. Algorithm for n-Gram Feature*

*1) For Calculating n-gram of each File:*
- *a)* For loop starts, for each file.
- *b)* Open file in hexadecimal form.
- *c)* Calculate the 4-gram as n-gram of the file.
- *d)* Convert the n-gram obtain in hexadecimal form.
- *e)* Store all this in a csv file with the name of a file and n-grams obtain of each file.

*2) Find Occurrences of each n-gram from each File:*
- *a)* find unique n-gram from each file.
- *b)* Calculate the total occurrences of n-gram from all files.
- *c)* Sort with highest occurred n-gram.

*3) Filter the n-gram obtained:*
- *a)* Count total number of n-gram obtained.
- *b)* Accordingly get the most frequent n-gram and name it as global list.

*4) Obtain the Dataset for n-gram:*
- *a)* Get the global list.
- *b)* For each n-gram in global list check whether it is present in a file or not.
- *c)* For malware file in classification attribute write malware and for non-malware files write benign.

We uses 4-gram as n-gram because result obtain from 4-gram is better [8].

*B. Algorithm for Entropy*

*1)* Start of for loop, for each file.
*2)* Size of each file.
*3)* Frequency of each byte value of a file.
*4)* Calculate Shannon entropy, for each byte frequency, ent = ent + freq * math.log(freq, 2)
*5)* If ent >5 then append 1 for the attribute entropy in dataset.
*6)* Else append 0 for the attribute entropy in dataset.
*7)* For attribute classification in dataset append benign for goodware files else malware.
*8)* Thus we will obtain the dataset of the entropy calculation.
*9)* End of for loop.

*C. Algorithm for PE Format Feature*

*1)* For each file obtain the value of attributes such as e_magic, etc
*2)* Store this values obtained in a csv format.
*3)* If it is malware in classification field write malware else benign.

*D. Algorithm*

*1)* Obtain printable string from each file.
- *a)* String is a command line tool to obtain printable string and so obtain string by executing command on terminal.
- *b)* Decode the result obtain.
- *c)* Store each file values in csv file.

*2)* Obtain the global list by filtering the required string such as function call, string with length greater than 8, URL, etc.
*3)* Find the most frequent and obtain a global list.

©IJRASET: All Rights are Reserved

From the table-1 we can conclude that the features combination of string, PE format feature and entropy increases the accuracy as well decreases the false positive rate.

We uses WEKA tool for classification algorithm. For testing it has been used cross fold validation. From the table it is also clear that overall performance of the random forest classification algorithm is good, so we use the random forest algorithm to train the model.

Figure displays the result of the trained model using random forest algorithm.

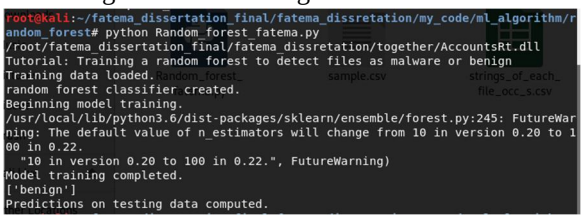



**Figure 3: sample output of trained model**

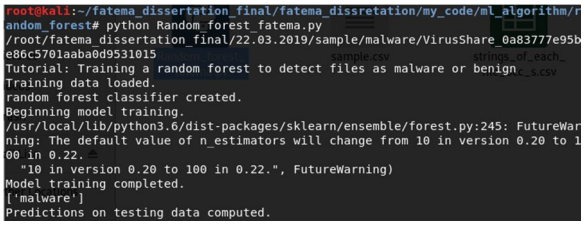



**Figure 4: sample output of trained model**

## V. CONCLUSION AND FUTURE WORK

We can combine different features for better accuracy and decreasing the false positive rate. Random forest overall perform good for malware detection. Features combined such as PSI, format feature and entropy gives more accuracy and false positive rate increases upto some extent. To balance both the accuracy and false positive rate the different static features must be used.

| Sr. No. | DATASET (Features) | Random forest | | SVM | | J48 | | Bayes Net | |
|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy (%) | False positive Rate | Accuracy (%) | False positive Rate | Accuracy (%) | False positive Rate | Accuracy (%) | False positive Rate |
| 1 | n-gram | 79.646 | 0.215 | 78.761 | 0.226 | 77.4336 | 0.243 | 70.354 | 0.300 |
| 2 | n-gram and PE | 98.7952 | 0.033 | 97.5904 | 0.038 | 99.3976 | 0.017 | 100 | 0.000 |
| 3 | PE | 99.3916 | 0.017 | 98.7952 | 0.019 | 99.3976 | 0.017 | 100 | 0.000 |
| 4 | String | 97.2118 | 0.068 | 91.2118 | 0.062 | 95.4032 | 0.108 | 78.2216 | 0.1 |
| 5 | String and PE | 99.7438 | 0.015 | 99.573 | 0.015 | 99.4876 | 0.020 | 95.3032 | 0.018 |
| 6 | String, n-gram and PE | 97.5904 | 0.067 | 98.7952 | 0.033 | 99.3976 | 0.017 | 99.3976 | 0.017 |
| 7 | n-gram and entropy | 72.9592 | 0.274 | 74.4898 | 0.247 | 72.9592 | 0.256 | 69.3878 | 0.303 |
| 8 | N-gram, string, PE, entropy | 97.0588 | 0.061 | 98.5294 | 0.031 | 99.2647 | 0.015 | 99.2647 | 0.015 |
| 9 | String, PE, entropy | 99.7076 | 0.009 | 99.7076 | 0.009 | 99.5614 | 0.013 | 95.1754 | 0.028 |

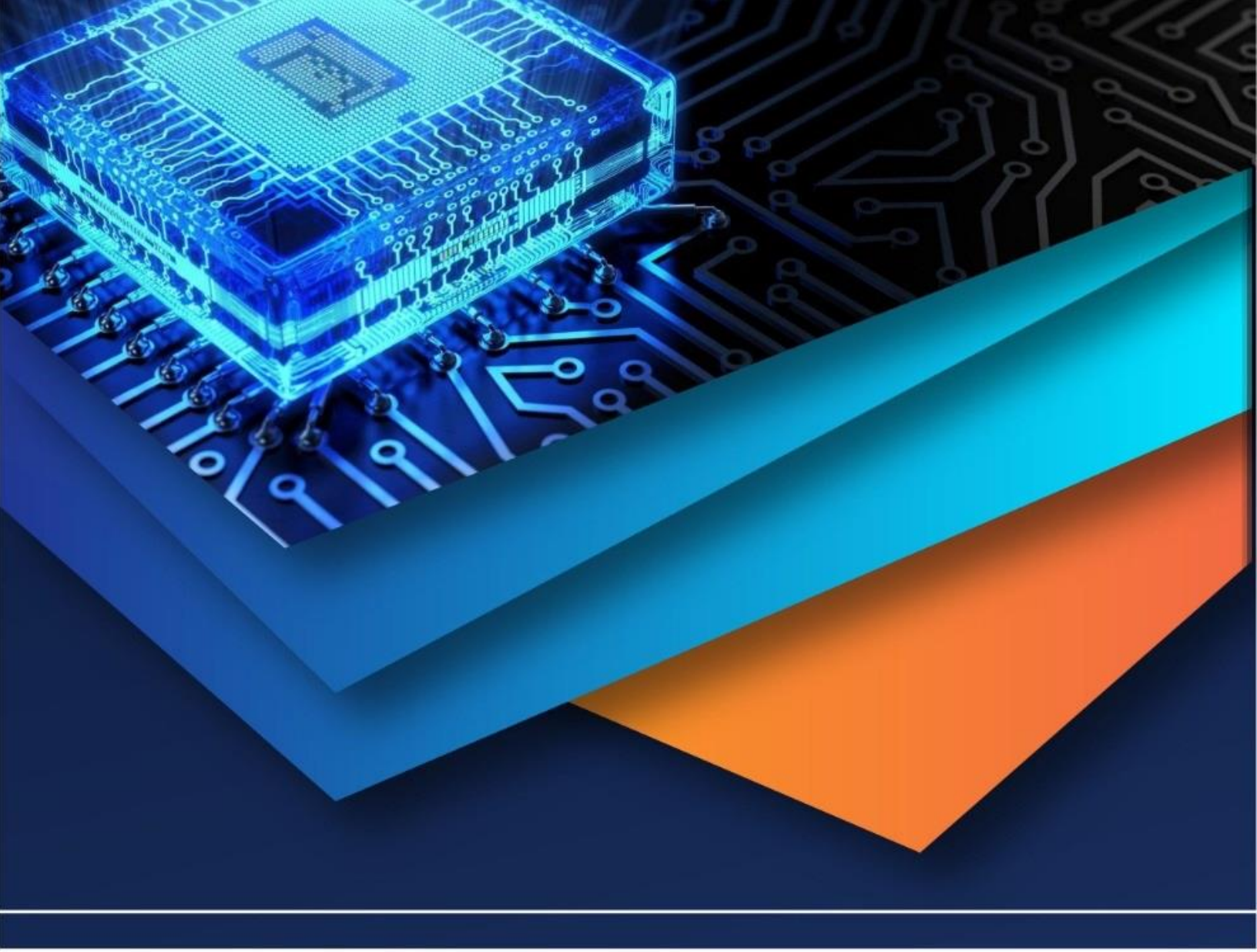
**Table 1: comparision of combining different features for data set**

©IJRASET: All Rights are Reserved

## REFERENCES

[1] D. Gavrilut, M. Cimpoesu, D. Anton, and L. Ciortuz, "Malware detection using machine learning," *2009 Int. Multiconference Computer Science Information Technology*, 2009.
[2] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," *Journal of Computer Virology and Hacking Techniques 2016*, 12(2), 59–67.
[3] Dhammi, Arshi, and Maninder Singh. "Behavior analysis of malware using machine learning." *2015 Eighth International Conference on Contemporary Computing (IC3)*. IEEE, 2015.
*[4]* A. Shalaginov, S. Banin, A. Dehghantanha, and K. Franke, "*Machine learning aided static malware analysis: A survey and tutorial", Cyber Threat Intelligence*. 2018, 70, 7-45.
[5] 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*. 2013, 36(2), 646–656.
[6] R. Tian, L. Batten, R. Islam, and S. Versteeg, "An automated classification system based on the strings of trojan and virus families," *2009 4th Int. Conf. Malicious Unwanted Software, MALWARE 2009*, 23–30.
[7] S. Joshi, H. Upadhyay, L. Lagos, N. S. Akkipeddi, and V. Guerra, "Machine Learning Approach for Malware Detection Using Random Forest Classifier on Process List Data Structure," *Proceedings of the 2nd International Conference on Information System Data Minning 2018*, 98–102.
[8] Bai, Jinrong, and Junfeng Wang. "Improving malware detection using multi-view ensemble learning." *Security and Communication Networks* 2016, 9(17), 4227-4241.
[9] T. Abou-Assaleh, N. Cercone, V. Keselj, and R. Sweidan, "N-gram-based detection of new malicious code," *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004.*, 2(1), 41–42.
[10] K. E. Cybersecurity, "Machine Learning for Malware Detection.", 2018
[11] W. D. Antivirus, "Evolution of malware prevention." 2018
[12] "attack vectors", accessed on 10-12-2018, https://www.hackmageddon.com/2018/07/23/june-2018-cyber-attacks-statistics/
[13] Bat-Erdene, M., Park, H., Li, H., Lee, H., & Choi, M. S., "Entropy analysis to classify unknown packing algorithms for malware detection." *International Journal of Information Security*.**2017,** *16*(3), 227-248.
[14] "Basic Malware Analysis tools" accessed on 25 april, 2019, https://www.hackingtutorials.org/malware-analysis-tutorials/basic-malware-analysis-tools/
[15] Sikorski, M. and Honig, A. (2012). *Practical malware analysis*; San Francisco (California, EEUU); William Pollock.
[16] "traditional static analysis approach" accessed on 25th april,2019, https://www.secpod.com/blog/malware-analysis-by-reverse-engineering/
[17] Kateryna Chumachenko, Bachelor's Thesis, "machine learning methods for malware detection and classification" University of Applied Sciences, 2017
[18] Jakub Ács, Bachelor's Thesis, "Static detection of malicious PE files" Czech Technical University in Prague Faculty of Information Technology, 2018
[19] "structure of PE file" accessed on 27th april 2019, https://docs.microsoft.com/en-us/windows/desktop/debug/pe-format#general-concepts
[20] naiyarah hussain, Bachelor's Thesis, "Malware analysis & detection using machine learning classifiers" heriot-watt university, April 2016
[21] Edward Raff., Richard Zak., Russell Cox., Jared Sylvester., Paul Yacci., Rebecca Ward., Anna Tracy., Mark McLean., Charles Nicholas, "An investigation of byte n-gram features for malware classification" *J Comput Virol Hack Tech* **2016**, 14(1), 1-20.
[22] "calculate file entropy", accessed on 19 march 2019, https://kennethghartman.com/calculate-file-entropy/
[23] "weka", accessed on 5 may 2019, https://wekatutorial.com/

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)