INTERNATIONAL JOURNAL · ENGINEERING RESEARCH ·

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 1st Jun 2019. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-06

Title: **MALWARE PREDICTION BASED ON SANDBOX GENERATED REPORT USING SYSTEM CALLS**

Volume 08, Issue 06, Pages: 249–251.

Paper Authors

VINAY KUMAR G, SHIVAKUMAR B R, DR. R NAGARAJA

BIT, Bengaluru.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

MALWARE PREDICTION BASED ON SANDBOX GENERATED REPORT USING SYSTEM CALLS

[1]VINAY KUMAR G, [2]SHIVAKUMAR B R, [3]Dr. R Nagaraja
[1] PG Scholar, Department of ISE, BIT, Bengaluru.
[2]Assistant professor, Dept of ISE, BIT, Bengaluru.
[3]Professor, PG and Research Coordinator, Dept of ISE, BIT, Bengaluru.

ABSTRACT

Malware detection is one of the common problems in modern time due to the increase in their evasive techniques. Most of the antivirus products fail to detect the new unknown malware types due to their signature based detection method and they are costly. Static malware detection method like code analysis can be easily obfuscated by encrypting the payload data. The more promising method to detect the malware is through dynamic analysis, where the behavior of malware is studied in a controlled environment. In this paper, the malware detection is done with the help of N-gram analysis method. The behavioral data produced by executing the files in cuckoo sandbox is converted to malware instruction set and from that data, system calls is extracted to make N-grams for analysis. The unique N-grams patterns obtained by this method is converted to their equivqlent binary form and it is applied to the Machine learning algorithms to classify the malware and benign files. In this method we can predict the malware with 94.28% accuracy using Random Forest algorithm which outperforms other N-gram based Malware detection based methods with less overhead and increased efficiency.

KEYWORDS

Sandbox, N-grams, dynamic malware analysis, system calls, feature selection, machine learning.

1. INTRODUCTION

A computer program or software which purposefully infiltrate the legitimate user with a malicious intent is known as malware. There are different types of malware variants based on their functionalities such as viruses, rootkits, trojan horses, ransomwares, etc. which are affecting the genuine users worldwide causing huge damages which are sometime irreversible and posses a potential threat on a regular basis. There are many popular antivirus and antimalware products online which helps their users to protect from these threats. But most of the antivirus or antimalware vendors use an old signature based malware detection method. In this method, they store the information of malware in their databases which is collected over a period of time in the form of md5 hashes. These hashes represent the integrity of a file. Whenever an antivirus product scans a computer, it'll check for the hashes that matches their database. If it detect any hash of a file which matches the malware hash in it's database, that file is considered as malware or malicious and the user will be notified for further processing or it will remove that file from the system. On the other hand, the malware developers create new malware variants which are capable of evading these old detection methods on a regular basis. Hence most of the antivirus products fail when there is an attack from advanced malware variant and keep the users in a high risk.

In the static malware detection method, the code section of the malware is analyzed in detail, if there is any suspicious code in the program the file is treated as the malware. The static code detection method is inefficient for the advanced malware variants as the malicious code can be binary obfuscated or encrypted and the malware analyst is unable to detect the actual code, even the antivirus software treat the file as benign. Therefore this method is unreliable and the malware can easily escape from detection.

In the dynamic malware detection method, the file or Portable Executable (PE) is executed in a controlled environment and it's complete behavior is analyzed. The sandboxes are use to provide this environment with the help of virtual machines. Figure 1 represent the dynamic malware detection method. The analysis report produced by the sandbox contains the complete behavioral data of the file or PE from which an analyst can know the behavioral characteristics and classify it as benign or malware. The behavioral data contains the activity of the file or PE in memory, processing level, networking, etc.. For example, when a malware is executed in the sandbox it may load a file from the main memory of a computer like .dll file from system32 folder of Windows 7 operating system(OS), it may delete, inject or modify a file. It may try to connect to a remote command and control server of an attacker. These data is recorded by the sandbox and reported.

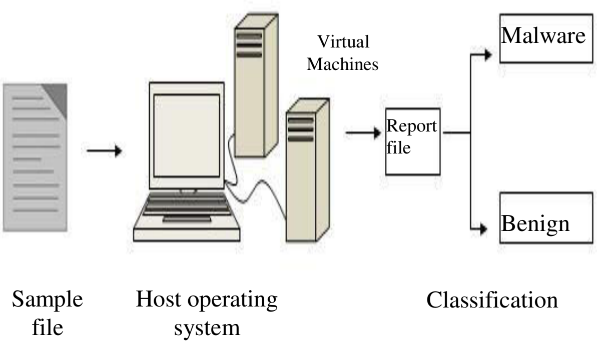


Figure 1. Dynamic malware detection by sandbox

The rest of the paper is arranged as follows: Section II covering the related work. Section III presents the methodology and the proposed work. Section IV provides the performance analysis. And finally concluded paper in Section V.

2. RELATED WORK

A. LITERATURE SURVEY

There are various approaches in malware detection using the machine learning algorithms. There are three phases in this method. i.e., file representation (datasets collection), feature selection and using Classification algorithm.

The file representation methods are in one of the following format: N-gram, Strings, PE features and function based. In the N-gram method, the file can be represented as either byte level n-gram or character n-gram. Here, the byte level N-gram is used. In Strings representation method, the strings are obtained from the code section of the PE and processed. In PE features method, the top features of the malware samples are selected like file size, packing type, etc. And they are used as the feature to predict malware. The feature selection methods used by machine learning algorithms are gain ratio, fisher score, document frequency and hierarchical feature selection. The machine learning Classification algorithms commonly used in malware detection are Artificial Neural Network, Naive Bayes, K-NN(K – Nearest Neighbor), Decision Tree, Support Vector Machine, J48, etc.

Ekta Gandotra et al. [1], proposed Zero-Day Malware Detection which uses the integrated feature set from both static and dynamic analysis of malware. This model holds good for the selected features. The selected feature set is limited in number. It has around 18 attributes. The information gain is calculated from these set of attributes only. The other important features like file-creation, memory, etc.. are not addressed in this approach.

Sachin Jain et al. [2], proposed Byte Level n-Gram Analysis for Malware Detection. It is non-signature based malware detection using machine learning algorithm It uses n-gram and 'Class-wise Document frequency' method which reduces the entropy of the system. The n-grams are obtained from raw byte patterns targeting the code section of the executables which can be easily obfuscated.

The most recent malware use binary obfuscation techniques to hide the original source code that posses malicious activity. This method is not suitable for the malware which are packed with encryption techniques.

Shiva Darshan S.L. et al. [3], proposed Windows Malware Detection Based on Cuckoo Sandbox Generated Report Using Machine Learning Algorithm. It uses the relevant N-grams term frequency, Information gain approach to detect the malware. The N-grams are obtained from the dynamic analysis report. The N-grams patterns are unique for Malware and Benign executables. It uses the overall information gain approach which in turn reduces the accuracy of Malware detection.

Pengtao Zhang and Ying Tan [4] work on Class-wise Information gain (IG) of the N-grams justify the use of IG approach for malware and benign class separately instead of total IG calculation. The Class-wise Information approach increases the accuracy of the malware Detection. The main drawback of this method is that it don't classify the N-gram patterns which is present in both the malware and Benign files.

B. EXISTING MALWARE DETECTION BY N-GRAM METHOD

The existing system of N-gram malware detection method is mainly based on the document frequency and information gain approach.

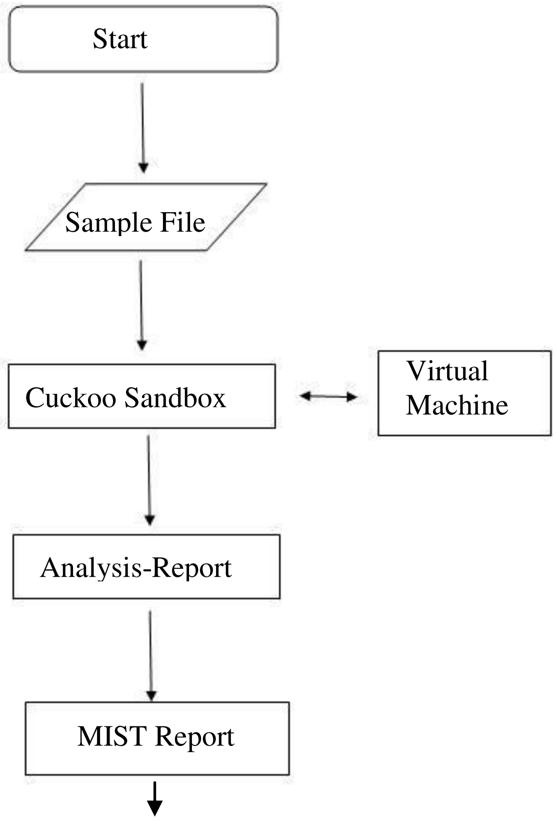
The N-grams are generated either by Dynamic or Static malware analysis method. The unique N-grams are extracted for the entire malware or benign class and they are compared with the other N-gram files [3] in order to obtain the document frequency table and a feature contingency table is made based on the document frequency table by comparing the absence or presence of N-gram counts within the document frequency table. Then the information gain approach is applied to the feature contingency table in order to obtain the data suitable for machine learning algorithm. The information gain (IG) is calculated[6] using the formula:

$$\mathrm{IG}(Ng) = \sum_{v_{Ng}\in\{0,1\}} \sum_{C\in\{C_i\}} P(v_{Ng}, C) \log \frac{P(v_{Ng}, C)}{P(v_{Ng})P(C)} \quad .. (1)$$

Where, C is the class i.e., malware or benign, vNg is the frequency of N-gram count that belongs to present category or absent and P( ) represent the probability. The main drawback of this method is that it don't specify about the N-gram patterns that is present in both the malware and benign class. Most of the malware files also have legitimate behavior as in the benign category. There is no measure to separate the common features between the given class instead it extract only malware and benign features by information gain approach.

3. PROPOSED WORK

The overview of the proposed work is as shown in the flowchart below.



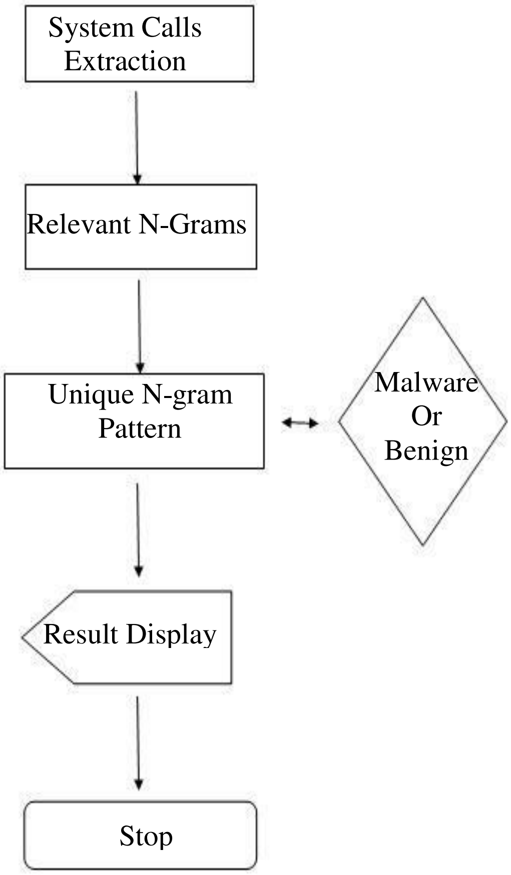


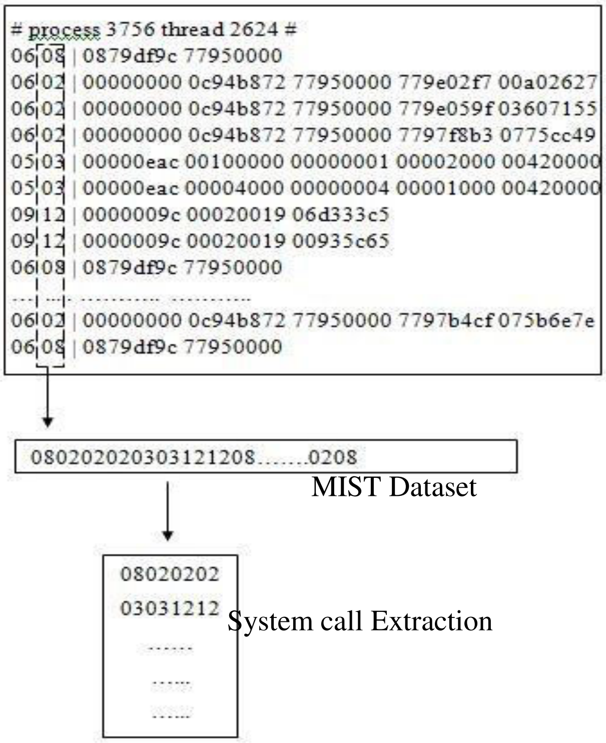
Figure 2. Flow Chart of the Proposed Work

As shown in Figure 2., the files are executed in the controlled sandbox environment to obtain the complete behavioral report of the sample during the runtime. The behavioral report is then converted into their equivalent malware instruction set (.mist) format. The .mist file contain the complete behavioral data of the sample which includes it's memory utilization, processes in the kernel level, file handling, network activity, etc. The system calls are extracted from these file and converted into the N-gram format. These N-grams are then processed in order to obtain the unique N-gram patterns for both the malware and benign samples separately. It is then applied to a machine learning algorithm and the results are predicted. The random forest algorithm predicts the malware with more accuracy than the other algorithms in this case.

Dataset Description

The behavioral data obtained by executing a file samples inside the cuckoo sandbox is in the JSON (Java Script Object Notation) format and it is converted to the malware instruction set (MIST) dataset format which is used in the proposed work.

The project is carried out by collecting around four hundred .MIST datasets of the various malware samples from the pubic source Malheur*. The malware dataset has four different families which are combined as classified as malware class. The benign files samples are obtained by executing genuine Windows 7 portable executable files inside the cuckoo sandbox and they are converted into the .mist format. The dataset of the .mist format is as shown in Figure 3. From the MIST behavioral report, for each process of the executable the dataset has three blocks. i.e., file handling, system calls and argumentation block. Only the system call data is extracted from the file and the data is divided into relevant N-gram format for further processing.



```
# process 3756 thread 2624 #
06 08 | 0879df9c 77950000
06 02 | 00000000 0c94b872 77950000 779e02f7 00a02627
06 02 | 00000000 0c94b872 77950000 779e059f 03607155
06 02 | 00000000 0c94b872 77950000 7797f8b3 0775cc49
05 03 | 00000eac 00100000 00000001 00002000 00420000
05 03 | 00000eac 00004000 00000004 00001000 00420000
09 12 | 0000009c 00020019 06d333c5
09 12 | 0000009c 00020019 00935c65
06 08 | 0879df9c 77950000
... ... .......... ..........
06 02 | 00000000 0c94b872 77950000 7797b4cf 075b6e7e
06 08 | 0879df9c 77950000
```

```
08020202030312 08.......0208
```

Figure 3. N-Gram Generation from MIST file

N-Gram Generation

*MIST Dataset Source: https://www.sec.cs.tu-bs.de/data/malheur/.

As shown in Figure 3. the system calls data from the second column of every MIST file is extracted separately. Here, The extracted dataset has four N-grams i.e., the length of the N-gram, N=4. The N-gram can also be of different lengths but from the evaluation results [2][3] it is found that the N-grams of length N=3 and N=4 provides better feature representation and by using that data in the machine learning algorithms gives more accurate predictions. There are around twenty system calls in the MIST dataset and all the system calls are hexadecimal in number.

In the Unique N-gram file generation phase, the N-grams of length 'N' are divided into chunks as shown in Figure 3. for each MIST report file of malware class separately, they are sorted in the descending order and the repeated N-grams are removed, the obtained data is stored to another file for each MIST report file. Then unique N-grams of all the files is extracted and stored into the unique malware N-gram file as shown in Figure 4. The same process is repeated for the benign class.

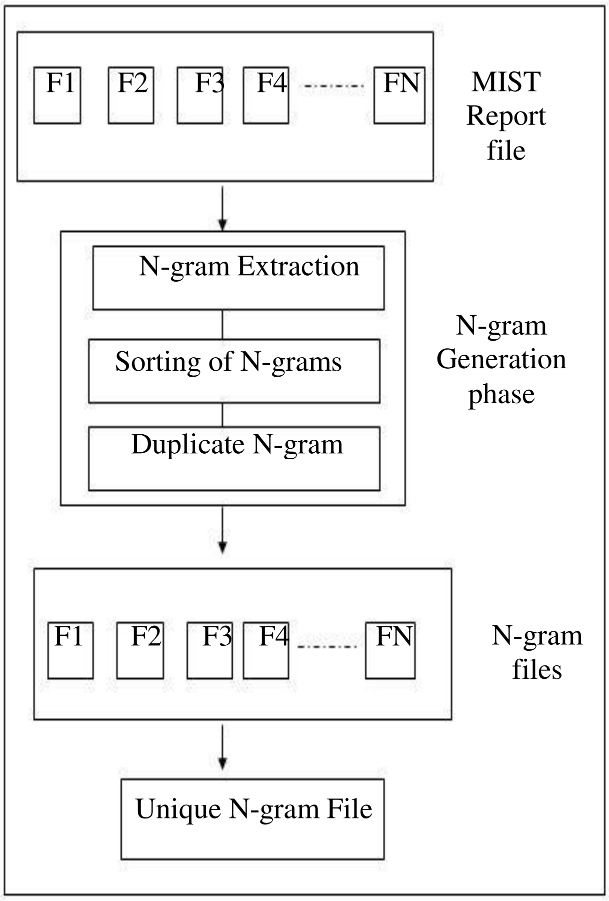


Figure 4. Unique N-gram file Generation

Then the Unique malware N-gram file and Benign N-gram file is compared and the repeated or matching N-grams in both the files are removed as shown in Figure 5. Thus the N-gram pattern obtained for malware and benign category is unique.

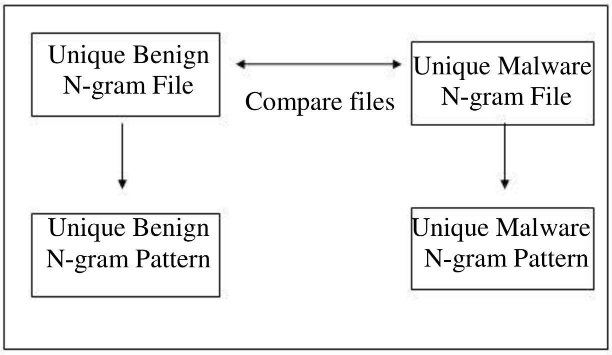

Unique Benign N-gram File
Compare files
Unique Malware N-gram File
Unique Benign N-gram Pattern
Unique Malware N-gram Pattern

Figure 5. Unique Patterns for malware and benign class

The obtained unique patterns are then converted into the binary format to feed the data to the machine learning classification algorithm. Here each N-gram has two digits. Therefore there are 4*2*4 = 32 binary digits for each individual N-gram when they are converted from their hexadecimal values to the binary form.

The random forest algorithm fits the given data and provides more accuracy in predicting the malware, since there are only 32 attributes and 2 class i.e., malware and benign.

4. PERFORMANCE ANALYSIS

The experiment is conducted with the 400 MIST malware samples collected from the public source Malheur and some of the malware samples are executed in the cuckoo sandbox installed in Ubuntu, these samples are collected from virusbay. Around hundred benign samples are executed in the cuckoo sandbox and analysis reports are generated. They are then converted into their equivalent MIST dataset format. The MIST datasets are then processed by the N-gram analysis method as mentioned in the previous sections and unique patterns are generated for the malware and benign files. The data is then converted into their equivalent binary format and feed to the machine learning algorithm.

From the experimental evaluation it is found that Random Forest algorithm has done better prediction with 94.28% accuracy. The model is given 696 instances as input with malware and benign attributes. The data is then divided for training and testing on the percentage split basis i.e., the model is trained with 70% of the given input and the remaining 30% of the data is given for testing. The training and testing data is randomized with the help of randomization filter for better classification. During the testing phase, for the total 209 Number of testing instances, 197 instances are classified correctly and 12 instances are classified incorrectly with 94.28% accuracy.

The analysis results for the Random forest algorithm is summarized in the Table I. The confusion matrix is created for the same data as shown in Table II.

Table I. Evaluation results of Random Forest Algorithm

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Malware	0.973	0.136	0.948	0.973	0.961	0.957
Benign	0.864	0.027	0.927	0.864	0.895	0.957
Weighted Average.	0.943	0.105	0.942	0.943	0.942	0.957

Table II. Confusion Matrix for Random Forest algorithm

a	b
146	4
8	51

a = Malware
b = Benign

From the confusion matrix, it can be seen that, during the testing phase, out of 209 instances, 146 malware instances are classified as malware, 4 instances are wrongly classified as benign. Similarly 51 benign instances are classified as benign but 8 instances are wrongly classified as malware. Making the overall accuracy of model as 94.28% for random forest algorithm.

5. CONCLUSION

There are various malware classification methods. The N-gram method of detecting the Malware and Benign files is more efficient than signature based Method. The Unknown malware variants sharing the similar N-gram patterns can be easily detected. In this paper a new approach of malware detection using the relevant N-gram analysis is proposed. This work efficiently classifies the malware and benign files based on their unique N-gram patterns with less computation overhead and predicts the malware during evaluation with increase in accuracy compared to other N-gram analysis method.

In the Future work, The N-gram analysis method will be applied for different Malware families instead of entire malware class and the data is also tested for different length of N-grams to improve accuracy.

6. REFERENCE

[1] Ekta Gandotra, Divya Bansal and Sanjeev Sofat "Zero-Day Malware Detection" . IEEE 2016 Sixth International Symposium on Embedded Computing and System Design (ISED).

[2] Sachin Jain and Yogesh Kumar Meena on "Byte Level n–Gram Analysis for Malware Detection" , ICIP-2011, Springer.

[3] Shiva Darshan S.L, Ajay Kumara M.A., and Jaidhar C.D "Windows Malware Detection Based on Cuckoo Sandbox Generated Report Using Machine Learning Algorithm", IEEE 2016 Sixth International Symposium on Embedded Computing and System Design (ISED).

[4] Pengtao Zhang and Ying Tan "Class-wise Information Gain" Third International

conference on Information science and technology 2013, IEEE.

[5] Asaf Shabtai, Robert Moskovitch, Yuval Elovici, Chanan Glezer on “Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey” information security technical report 14 (2009) 16-29.

[6] D Krishna Sandeep Reddy • Arun K Pujari, on] “N-gram analysis for computer virus detection”, J Comput Virol (2006), Springer.