DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.6084

Volume 9, No. 3, May-June 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A STUDY ON DIGITAL FORENSICS USING VARIOUS ALGORITHMS FOR MALWARE DETECTION

Dr. Anjana Pandey
Asst. Professor: IT Dept. UIT RGPV
Bhopal, India

Shruti Mujmer
Student: IT Dept. UIT RGPV
Bhopal, India

Poorvi Gyarsiya
Student: IT Dept. UIT RGPV
Bhopal, India

Sarthak Kanungo
Student: IT Dept. UIT RGPV
Bhopal, India

Abstract: Malware Detection is a field of Digital Forensics which involves detection of known and unknown malware by various methods. Detection of real-time malware becomes a big challenge, the research done in the field has shown the advancement achieved in malware detection system designs and implementations. Although each malware is unique, malware has some common behavioral characteristics which can be examined and used for malware detection. This paper has a survey and analysis of various research works on Malware Detection using behavior characteristics and also introduces its problems and issues. Finally, we have compared various machine learning algorithms which can be used for most effective malware detection process. The implementation and the results of the study show that the Random Forest algorithm is a most efficient algorithm for detection of malicious files in any system.

Keywords: malware; malware detection; machine learning; algorithm; malicious files; dataset

I. INTRODUCTION

"Digital Forensics is the field of forensic science which involves the collection, detection, identification, extraction, analysis of digital data". Digital Forensics has various branches such as Malware Detection, Criminal Data Mining, Database Extraction, File Recovery etc. This work is based on Malware Detection. Malware detection has become a challenge for the modern world. Malware threats have increased exponentially due to the continuous use of the internet since the internet is the best way for an attacker to attack any system. The term Malware refers to all the software which have harmful or human undesired goal. Malware may be in any form such as Trojan Horses, Viruses, Spyware, Worms, Rootkits, Backdoors, Key-loggers, etc.
Malware detection; thus, as the name refers to all of the techniques and technologies used for detection of any malicious attack on any system. Malware detection approaches can be classified as below:

- Dictionary-based detection [1]: It is the approach where the Malware is identified on the basis of some particular attributes. The file is identified as a malware if it matches the same attributes as provided.
- Signature-based detection: It is also known as role-based detection. Dictionary-based detection fails when the malware is not known or when malware changes the behavior at a fast rate. The signature-based process can extracts malware signatures more frequently and add them to the main dictionary.
- Behavior-based detection: It uses unique behavior characteristics to detect malware.

These techniques can be further classified as Static and Dynamic: Static method implementation involves detection of malware without running them. Dynamic detection involves detection at runtime.

There are various machine learning algorithms which can be used for the process of detection of any malicious file. Machine learning algorithms could be supervised where the output is provided in the dataset, or they can be unsupervised where the output is not known. The algorithm in this study uses supervised learning.

II. PREVIOUS WORKS

Liu Wu, et al. [2] proposed an algorithm which used the behavior characteristics of malware for detection. The paper described the Malicious Behavior Characteristics, techniques of extraction of malicious behavior, detection algorithm for malware behavior, and implementation of detection of behaviors. Malicious behavior detection is to find the unique characteristics of malware by analysis of the behavior and semantic information. The semantic information about any malicious content can be found using the behavior events in the malware code and thus identify the malware. On the basis of the malicious behavior characteristics, they implemented a malware detection system which had a modular design. It used malware training dataset as input and generated malware detection result.

Priyank Singhal, et al. [3] applied Random Forest Algorithm (RFA) is a supervised machine learning algorithm which is used to construct the classification models for detection of malware.

The Portable Executable file format is used for object code, DDLs, executables. This file format is also used in 32 bit and 64-bit operating system. For managing the wrapped executable code, it encapsulated all the necessary information which are important or necessary for windows operating system.
As most of the antivirus is not able to detect viruses, they start controlling them once the system effects. So they have created a firewall level security in the network. The malicious code first passes through the firewall, where there is a portable executable file which scans the code and creates the report and stores in the data mine.
After experimenting with various executable files which have a combination of various normal and malicious codes, they have an accuracy of 99.5556 % for correctly classified instances and 0.4444 % for incorrectly classified instances.

Muhammad Salman Khan et al. [4] developed a malware testing sandbox using Microsoft native APIs and an AdaBoost Algorithm which is based on the information fractal (cognitive analysis).
The algorithm is a machine learning algorithm. This is a framework classifier that helps in increasing performance of classifiers by using the dynamic voting machine. Yoav Freund and Robert E. Schapire proposed the original AdaBoost machine learning algorithm in 1997.
The main focus of this algorithm is to provide better performance of the classifier after each training iteration and to give proper attention to the wrongly estimated classifier.
The approach resembles the cognitive learning methodology in which major area of working are those examples which are not easily learned and hard.
The AdaBoost algorithm outperforms in detecting the true positives and in reducing false negatives.

Mohammad Akour, et al. [5] presented an ongoing study in the fields of Malware detection algorithms. In this research paper, three experiment was conducted: An application was developed to use APIs from various websites and uses the files and links passed from websites and creates a list of malware scanners. In the list, there are two parameters first one is link number and the second one is infection percentage which is calculated as:

Infection percentage = No. of Infected ÷ Total files scanned

Where No. of Infected indicates the number of scanners that suspected a link is Malicious.
They used Malware scanners to evaluate infection percentage formula by linking normal websites. Some scanners provide different decisions for same evaluated files or links. The result for same file or link reported by the scanner is different. Some scanner reported the same file as “clean” and other reported as “malicious”.
The percentage of unrated links is calculated by the division of the number of unrated links by the total number of the evaluated link. The unrated link percentage is different for every scanner. This shows that there is lack of complete information to identify whether a file is infected or not and that datasets need to be revised for detection using signature-based detection.
Da-Yu Kao et al. [6] the paper proposed a Windows malware forensic toolkit which collects important data from the targeted system to decide if an attack has occurred. The collection of data is done by running commands that produce data in an easily interpretable format. This framework provided an entire view of the Windows malware detection toolkit. The features of this framework suggest major benefits for system administrators, incident response specialists and forensic laboratory managers if the critical reference used is the ISO/IEC 27037:2012 framework. Digital Triage Forensics (DTF), a procedural model for cyber forensic applications is used for the initial assessments of any attack. The study presents a critical review of how the DTF framework can be used and implemented into detection and digital investigations by gathering quick intelligence.

Sudhir Kumar Pandey, et al. [7] performed the analysis and comparison of 17 different malware detection tools from various sources. The study includes four static, six dynamic and seven online malware detection tools. They have used 29 different malware samples. The malware tools provide a virtual environment for various malware detection processes. In the conclusion, Regshot Process Monitor and Process Explorer were the best malware detection tools. The result comparison report indicates that the above tools are the some of the tools for malware detection.

III. MALWARE DETECTION

A. Malware detection techniques:

There are basically two techniques for malware detection:

- Static Malware Analysis (SMA)
- Dynamic Malware Analysis (DMA)

Static Malware Analysis (SMA) is a malware analysis method where only basic analysis is done.
In SMA malware are detected without executing them. The methods used for static analysis are Basic Information Analysis, Structure Analysis, and Control Flow Analysis etc. But malware used different measures of Polymorphism, Metamorphism, ShellCode etc. to make analysis and detection more difficult. The tools for static analysis are PEid, Hex-Rays Compiler, Resource Hacker.
Dynamic Malware Analysis (DMA) is different from SMA. These techniques are used for detection of malware in a file. In this method to avoid the difficulty faced in SMA like Polymorphism, Metamorphism, Shell Code etc. Malware needs to execute at runtime.
Dynamic analysis is done by either Debug Execution or Run-time Execution.

B. Behavior characteristics of malware:***[1]***

The Behavior-based Malware Detection (BMD) method uses the behavior data of malware for detection and classification purposes. The main difference between Behavior-based and Signature-based malware detection is that behavior-based uses behavior characteristics rather than using the characteristics of the malware. This feature is more effective because the malware has unique behaviors rather than characteristics.
Since behaviors do not change by shellcode, Polymorphism or Metamorphism, BMD can more effectively detect new malware.
It could be better understood by the following example [1]:
The table describes 6 behavior events of a malware. The first behavior shows the malware process (malware.exe:316)

releases a copy of itself in the system directory of Windows, the second event shows the addition of a key item (cmdbcs, "C:\Windows\cmdbcs.exe"), the 3rd shows the semantic

Table I. Behavior characteristics of malware

Malware name	Behavior	Destination	Attributes
Malware.exe:316	Create	C:\windows\cmdbcs.exe	Attributes: A Options: overwrite
Malware.exe:316	Write	C:\windows\cmdbcs.exe	Offset: 0 Length:19297
Malware.exe:316	SetValue	HKLM\Software\Microsoft\Windows\CurrentVersion\RUN\cmdbcs	"C:\windows\cmdbcs.exe"
Malware.exe:316	Create	C:\windows\system32cmdbcs.exe	Attributes:N Options: OverwriteIf
Malware.exe:316	Write	C:\windows\system32cmdbcs.exe	Offset: 0 Length:32256
Explorer.exe:1484	LoadImage	C:\windows\system32cmdbcs.exe	ImageSize: 45056 Properties: 3

information about the above 3 events the execution of the malware it is set to run automatically after the start of Windows. In the last 3 behavior events, the malware first releases a Dynamic Link Library file (cmdbcs.dll) in the system32 directory, which is then automatically loaded by the process Explorer.exe:1484 and after the execution of the malware, a process-injection behavior to the system process explorer.exe to hide the malware itself is observed.

IV. WORK ISSUES AND PROBLEMS

Evolution of Malware detection processes occurs from several different perspectives. APIs are offered by various websites which allow users to create their own interfaces or applications to use the available detection methods and services. These websites either have their own databases or use public Malware datasets. Problems in these detection systems can be surmised as:

1: Multi-faceted detection decision [5]: The different malware repositories provide varied classifications for the same malware (i.e. whether the suspected file/program is Malicious or not).

2: Removal decisions [5]: Malware removal involves malware detection as a pre-process. But due to problems of false positives and false negatives, detection process can be dangerous for the system especially if the decisions about malware detection are made automatically without a second check. Any decision made without supervision can cause damage to important system files and databases, especially if there are no provisions for rollback or backups.

The following reasons cause the failure of traditional machine learning algorithms and detection tools for detecting the ever-evolving malware patterns and threats [4]:

1. There can be frequent changes in the features and inter-dependencies among the various features of any malware in the data set in order to avoid being detected.

2. As the datasets continuously change, the statistical distribution of the malicious traffic over the internet is difficult to estimate and data needs to be constant for traditional algorithms to work effectively.

3) Since the internet services use protocols like HTTP and DNS Servers, malicious codes use these technologies and destinations, they are difficult to block.

Moreover, hardware limitations of the device also limit the work in machine learning.

The methodology for this study aims to minimize these issues and problems.

V. METHODOLOGY

To perform malware detection, the algorithm applied on the dataset also needs to be accurate and efficient so as to detect the changes in the features or the behavior of the malicious code. The algorithm needs to perform with the same efficiency on each dataset so as to provide state-of-the-art detection of the malware. For this purpose it is necessary to select the correct algorithm which would provide the highest accuracy and efficiency. For this study we have selected five machine learning algorithms namely, AdaBoost [3], Random Forest, Gradient Boosting, Decision Tree and GNB [8], which are trained and tested on a malware dataset to determine which of these provide the maximum accuracy.

To determine whether which of the five testing algorithms; AdaBoost, Random Forest, Gradient Boosting, Decision Tree or GNB; is the best algorithm to predict any malicious content in the portable executable files and hence find the malicious threats to the system, the comparison is made between these five algorithms over a training data set. After the training [12] of the program, the classifiers develop a set of rules for decision making, which are then tested over test dataset. The output would provide the accuracies of each algorithm.

Algorithm:

```
BEGIN /*study of various detection algorithms */
   Divide the dataset into training and test dataset;
   Select the parameters which would be used for classification;
   Create a dictionary structure of five classifiers;
   FOR all five classifiers TEST
      BEGIN
            Train all the classifiers on training dataset;
            Test all the classifiers on the test dataset;
```

Calculate the percentage accuracies and the false negative and false positive rate;

END

END

A flowchart for the process for study for each algorithm is given below:

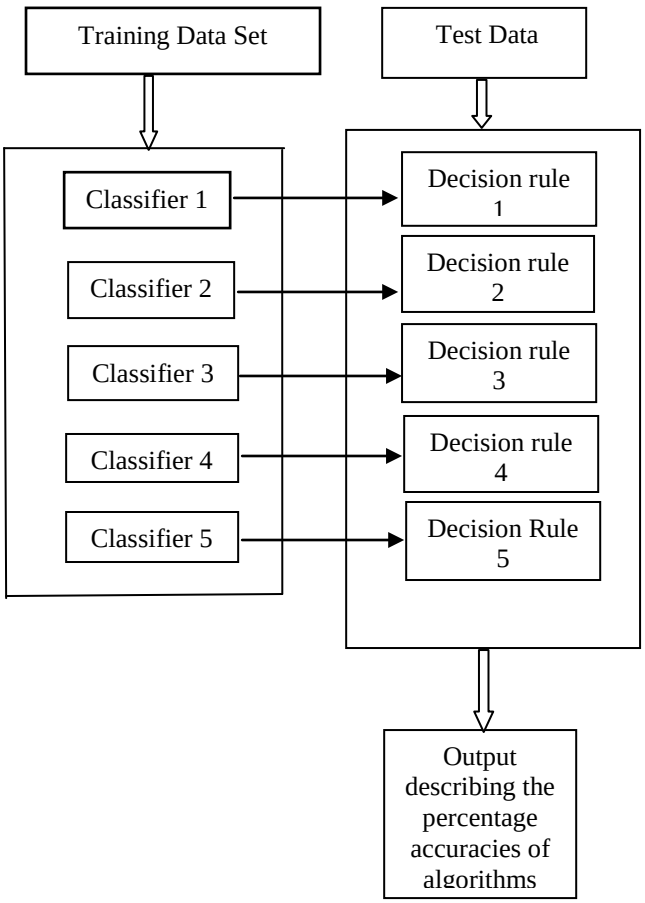

Fig. 1 Flowchart for the detection process

VI. IMPLEMENTATION

The system used is a Windows 10 operating system, with implementation done using Anaconda ide [9] and the documentation done with the help of Jupyter Notebook [10]. The dataset consists of exe files each of which has predefined 54 characteristic features described in the dataset. Some of which are malware and others are clean. The dataset is broken into a training dataset and test data set; 80% of the dataset is made to be the part of the training dataset.

	A	B	C	D	E	F	G	H	I	J
1	Name|md5|Machine|SizeOfOptionalHeader|Characteristics|MajorLinkerVersion|MinorLinkerVersion|									
2	memtest.exe|631ea355665f28d4707448e442fbf5b8|332|224|258|9|0|361984|115712|0|6135|4096|3727									
3	ose.exe|9d10f99a6712e28f8acd5641e3a7ea6b|332|224|3330|9|0|130560|19968|0|81778|4096|143360|									
4	setup.exe|4d92f518527353c0db88a70fddcfd390|332|224|3330|9|0|517120|621568|0|350896|4096|8110									
5	DW20.EXE|a41e524f8d45f0074fd07805ff0c9b12|332|224|258|9|0|585728|369152|0|451258|4096|79872(									
6	dwtrig20.exe|c87e561258f2f8650cef999bf643a731|332|224|258|9|0|294912|247296|0|217381|4096|536									
7	airappinstaller.exe|e6e5a0ab3b1a27127c5c4a29b237d823|332|224|258|9|0|512|46592|0|4488|4096|81									
8	AcroBroker.exe|dd7d901720f71e7e4f5fb13ec973d8e9|332|224|290|9|0|222720|67072|0|219331|4096|:									
9	AcroRd32.exe|540c61844ccd78c121c3ef48f3a34f0e|332|224|290|9|0|823808|650240|0|587663|4096|83:									

Fig. 2 a screenshot of the dataset

After dividing the dataset, features are selected which would be used for classification purposes.

The program for detection purposes consists of a dictionary structure which involves the classifiers of the five algorithms which will all train and test on the provided data set. Each classifier has features of an algorithm it represents and uses those features to train the classifiers on the dataset. After training, the classifiers create decision rules or hypothesis which is then tested on the test dataset.

The result of the program provides a list of the features that were used for classification purposes along with the percentage accuracies of each algorithm and the false positive percentage and false negatives percentage of the best algorithm.

```
Researching important feature based on 54 total features

12 features identified as important:
1. feature DllCharacteristics (0.149182)
2. feature Machine (0.137001)
3. feature VersionInformationSize (0.084721)
4. feature Characteristics (0.078240)
5. feature MajorSubsystemVersion (0.069004)
6. feature ResourcesMinEntropy (0.065149)
7. feature ImageBase (0.058166)
8. feature SectionsMaxEntropy (0.052607)
9. feature ResourcesMaxEntropy (0.051133)
10. feature Subsystem (0.035358)
11. feature MajorOperatingSystemVersion (0.021402)
12. feature SizeOfOptionalHeader (0.021216)
```

Fig. 3 the list of features used for detection

The false positives represent the files that are not malicious and identified as malicious, and false negatives represent the files that aren't malicious and are classified as clean.

VII. RESULTS

The output of the program to detect the best algorithm found out the Random Forest Algorithm to be the most effective algorithm with 99.492% of accuracy.

The false negatives and the false positives for the Random Forest Algorithm came out as 0.716% and 0.417% respectively.

```
Now testing algorithms
GNB : 70.188338 %
DecisionTree : 99.105397 %
RandomForest : 99.492937 %
AdaBoost : 98.453459 %
GradientBoosting : 98.801159 %

Winner algorithm is RandomForest with a 99.492937 % success
Saving algorithm and feature list in classifier directory...
Saved
False positive rate : 0.417978 %
False negative rate : 0.716802 %
```

Fig. 4 the output of the program

A graph representing the accuracies and the time taken by the algorithms is given below:

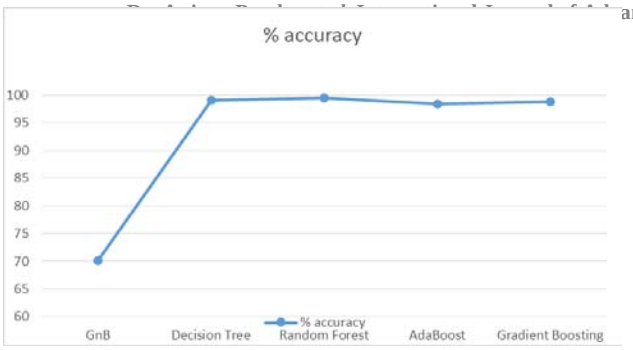

Fig. 5 graph depicting the percentage accuracies of the algorithms

VIII. CONCLUSIONS

Malware detection is imperative in today's scenario so as to make sure that the information is accessed by the intended person only. As detection of malware became successful, new challenges also grew and have to be dealt with. As Malware detection is evolving, it is moving towards becoming online and real time, which requires more enhanced and accurate malware detection systems and methods.

The need to detect any unauthorized access to the user system is necessary and can be done by analyzing various scenarios when such attacks happen. The existing tools analyzed for this purpose provide an insight on how the detection can take place and what more modifications can be made to further safeguard the user system.

For this purpose, characteristics based malware detection can be applied to make malware detection more accurate and effective.

The Random Forest algorithm can be used for effective decision making and to develop a system that would provide not only optimized static malware detection but also can be modified for dynamic malware detections to provide much better security against any unknown or new malicious code that could attack the system.

IX. REFERENCES

[1] Liu Wu, Ren Ping, Lui Kie, Wu Jian Ping, Liu Ke."Analysis and Forensics for Behavior Characteristics of Malware on the Internet", 2016 IEEE International conference on digital signal processing, 2016.

[2] Wu, Ke Liu, Ping Ren, Donghong Sun, Jian Ping Wu, Ke Liu. "Analysis and forensics for Behavior Characteristics of Malware on Internet", 2016 14th Annual Conference on Privacy, Security and Trust (PST), 2016

[3] . Priyank Singhal, Natasha Raul, 2012. Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks in International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.1, January 2012

[4] Muhammad Salman Khan, Sana Siddiqui, Robert D. McLeod, Ken Ferens, Witold Kinsner."Fractal-based adaptive boosting algorithm for cognitive detection of computer malware", 2016 IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), 2016.

[5] . Mohammad Akour, Izzat Alsmadi, Mamoun Alazab: The Malware Detection Challenge of Accuracy, Student Paper submitted to University of Balamand.

[6] Da-Yu Kao, Guan-Jie Wu:"A Digital Triage Forensics Framework of Window malware forensic toolkit: Based on ISO/IEC 27037:2012", 2015 International Carnahan Conference on Security Technology (ICCST), 2015

[7] . Sudhir Kumar Pandey, B.M.Mehtre: Performance of Malware Detection Tools: A Comparison 2014 IEEE International Conference on Advanced Communications Control and Computing Technologies, 2014

[8] Kwong Sak Leung. "Data Mining on DNA Sequences of Hepatitis B Virus", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2009

[9] Anaconda download and installation **https://www.anaconda.com/download/** documentation **https://www.anaconda.com/what-is-anaconda/** and **https://enterprise-docs.anaconda.com/en/latest/** ,and cheat sheets **https://conda.io/docs/_downloads/conda-cheatsheet.pdf**
Jupyter notebook download and installation **https://www.anaconda.com/download/**

[10] . Anaconda information **https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)**

[11] C.C.C. Pang, A.R.M. Upton, G. Shine, M.V. Kamath. "A comparison of algorithms for detection of spikes in the electroencephalogram", IEEE Transactions on Biomedical Engineering, 2003