International Journal of Recent Trends in Engineering & Research
ISSN(ONLINE) : 2455-1457
IMPACT FACTOR : 3.344

An Approach to Malware Analysis and Reporting by Machine Learning

O. Gireesha[1], L.V. Reddy[2]
[1,2]Sree Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India.

Abstract- Now-a-days cyber security is major concern information and related security areas to protect data from threat. In this context, we frame a methodology by using machine learning to detect various types of Malwares by using machine learning methods. In the present work, we use machine learning approach to the various static and dynamic analysis techniques to discuss different algorithms in cyber security. For this, we use a dataset. The various malware details will be analyzed in the present context by using clustering algorithm.

Keywords: Machine Learning, Static and Dynamic Analysis, Classification, Clustering.

I. INTRODUCTION

With the rapid raise of malware, it is significant for malware analysis to categorize unknown malware files into malware families. By doing so, the behavior and uniqueness of malware will be recognized precisely. Malware is a general utterance used for various types of malicious software. Malware 5 includes Worm, Virus, Back door, Trojan house, and further malicious software which be characterize by malicious code. Due to extensive usage of Internet, computer users face numerous hazardous propagations of malware. The modern malware's use is generally illicit profit. For example, a hefty number of computers are grimy by key logger and $24.3 billion is leveraged by e-payment system losing. Malware analysis is defined as the process of analyzing the idea and functionally of a malware. Malware analysis purpose understands of characteristics that all viruses in a family have in frequent and generate a set of signatures so as to detect malwares.

Commonly, dynamic and static malware analysis has been applied. When new malware is detected, dynamic malware analysis technique executes malware in the Virtual Machine using ProcMon, RegShot, and other tools. These tools are used to identify the general behavioral analysis techniques such as network traffic analysis, file system, and other Window features such as service, process, and the registry. However, the dynamic techniques are susceptible to a variety of anti-monitoring defenses, as well as time bombs or logic bombs and can be slow and tedious to identify and disable code analysis techniques to unpack the code for examination. Furthermore, it takes large amount of time to prepare malware analyzing environment to analyze malware such as virtual machine environment. However, some malware cannot be executed in that kind of environment. With the static malware analysis technique, researchers231213 perform reverse engineering using IDA Pro and Ollydbg tool to analyze malware based on its structure in order to discover its purpose and functionality but it takes a huge time to observe the malware structure.

Malware analysis 1213 is necessary to understand the behavior of malware. Consequently, malware signature is shaped to efficiently perceive malware. Nevertheless, it wastes a lot of time to observe the behavior and feature of malware.

1.1. Malware Types

Over the years multiple malware types have been seen executing different malicious actions. From simply presenting the user unwanted content to completely taking over the machine and restricting access to it. The known and most commonly seen malware types are:

Trojan horse is presented as software that the user might find useful, just like any other legitimate program. By opening the package, this malware releases other types of malware that will infect the machine; including key-loggers, account stealers etc. Compared with Viruses and Worms, Trojans do not replicate on their own but instead they require user interaction to do so. For this reason, this type is one of the most dangerous out there as it is usually detected when it has already infected the machine.

Virus represents a malware type that can exhibit actions ranging from just showing random errors to taking the system in a Denial of Service (DoS) state. The main difference between a Trojan and a Virus represents the ability to self-replicate by becoming part of other legitimate software. These types are commonly spread by sharing files, disks or e-mails to which the virus has attached on.

Adware represents one of the least dangerous types as its only purpose is to display ads to the user. In order to provide the infected machine with ads that the user might be interested in, it logs information like browser history, search engines history or history of installed programs. Depending of the severity of the logging, Adware may be labeled by AV vendors as Spyware.

Spyware represents a type of malware which installs itself without the permission of the user. Used to collect browsing history along with tracking information it ordinarily bundle with free software. AV vendors also name this type, PUP, just because of the bundling with freeware.

Worm represents a similar type as a Virus, being able to do the same amount of damage to an infected machine. The main difference is represented by its independence from other software as it does not require a host program to attach itself to. A worm usually infects its target via exploits or vulnerabilities and it uses different transport protocols to spread and infect other machines.

Bot represents a malware type that grants access of the infected machine to its master. This type can spread using Backdoors opened on the target by a Virus or a Worm and it is mostly known for using Internet Relay Chat (IRC) to communicate with its master. With multiple bots, Distributed Denial of Service (DDoS) attacks can be initiated that could block the services of the target by overwhelming it with requests.

Ransomware represents a more sparse type of malware that takes control of the graphical interface and blocks the user from accessing its machine until a certain amount of money is paid. Most commonly, these types infect their targets via Trojan horse.

II. RELATED WORK

Schultz et al introduced the concept of machine learning for malware detection. The three diverse static features that group the malwares are: Portable Executable (PE), byte sequences and strings where byte sequence is for extracting a chain of n sequences and strings were obtain from the text strings determined in the program files. When an input string is passed to the Naïve Bayes algorithm the classification achieves the 97.11% of program files.
Patterns in the DLL data were established using a rule based induction algorithm called Ripper [7]. According to the author's conclusion machine learning ascertained to be twice as efficient as signature based method.

Kolter et al. introduced an approach on n-gram yields better results. The analysis part is done by multiple classifiers and decision tree afford the best results.

Tian et al. classify Trojans based on the number of bytes in the code. Their results showed that function length when used in conjunction with the frequency of occurrence can be very useful for ascertaining malware class. It is united with other features for paced and growing malware classification. They also observed that for obfuscated files strings of word or sentences were hidden [10]. They applied machine learning algorithms present in WEKA [11] library for classifying malwares.

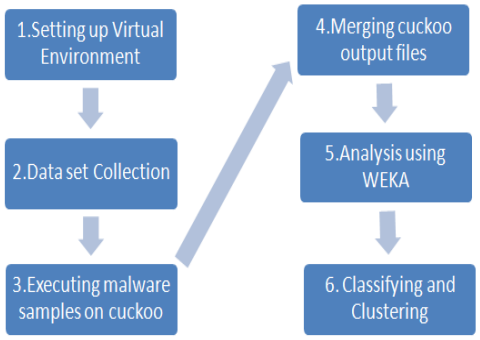
Rieck et al. premeditated a layout for automatic analysis of malware behavior using data mining. The framework so designed accordingly identifies distinct classes of malware with identical behavior (clustering) and assigns undiscovered malware to these already discovered classes (classification).

Christodorescu et al. advised a technique in which malware is inspected by the execution of the malware against a collection of benign programs for any dissimilarity. Any dissimilarity thus found, can be used by other anti-malware writers to identify malware variants.

III. Methodology

Malware be the Swiss-army scalpel of cybercriminals and every other challenger to any conglomerate or organization. Malware Analysis is the cram of a malware by dissecting its various components and studying its actions on the host computer's operating system [1]. Different malware analysis techniques allow the analyst to quickly and in detail understand the risk and intention of a given sample. The steps involved in the malware analysis (as shown in the figure) are as follows:

1. Setting up Virtual Environment
2. Data set Collection
3. Executing Malware Samples on Cuckoo
4. Merging Cuckoo Output Files
5. Analysis using WEKA
6. Classifying and Clustering



I. Setting up Virtual Environment

Within malware analysis, the function of virtual environment is to form source environment so as to diminish damage to the definite computer resources once the malware model executed. It is to let alone damage to the real operating system and computer resources if the malware executed. There are some malware samples that try to avert against malware analysis that used virtual environment tool. In our purpose structure, we determined to develop secure environment infrastructure that make use of Windows operating system and Linux, Backtrack as guest operating System.

II. Dataset Collection

The types of datasets have been flexibly updated to remain abreast of threat transitions in the wild. The technical obstacles of data collection such as developing and operating honey pots. The simple procedures for accessing these datasets as much as possible in order to make the datasets available to any examiner wish to perform anti-malware research using datasets.
Malware activities: collected by sandbox and forensic analysis are

- PRACTICE Dataset (’13): long-term packet traces collected from the dynamic malware analysis system operated through the PRACTICE project.
- FFRI Dataset (’13~’14): logs collected from the dynamic malware analysis system Cuckoo sandbox and yarai analyzer Proficient.
- MARS for MWS (’08~’10): memory dump and forensic data collected from the vigorous malware analysis system by means of not-virtualized machine, MARS.

The malware samples are made available in different formats like HTML document format, jpeg, executable zipped, PE format and many more.

III. Executing Malware Samples on Cuckoo

The malware samples are executed on Cuckoo [3]. Within the context of malware analysis (and computer security in general), a sandbox runs a program in a secure environment (e.g. a virtual machine.). Cuckoo, an open source system provided by Cuckoo Foundation. Cuckoo at times provide analysts by means of enough information they require to get the job finished. Cuckoo display its output in different formats like MongoDB interface, HTML report and HPfeeds interface.

IV. Merging Cuckoo Output Files

Cuckoo is accustomed to automatically run and examine files and accumulate comprehensive analysis outcome that delineate what the malware do while running inside an inaccessible Windows operating system. The cuckoo output file (in HTML format) is altered into CSV format. The input to CSV file include File size, File Name, File type, SHA1, CRC32, SHA256, MD5, SHA512, Size of raw data, Virtual address, Entropy, Virtual size, Imports, Registry keys, and IP address in the equivalent order. All CSV files are pooled into a single CSV file.

V. Analysis using WEKA

WEKA [16] is a free S/W and developed at the University of Waikato, New Zealand. This S/W generally accepts the merged output file and it include a set of algorithms for classification and clustering that can be used for analysis.

VI. Classifying and Clustering

The depiction can facilitate classifiers to efficiently and effectively associate data across abundance of objects. Malicious software is classified into families, each family originate from a solitary source base and exhibit a set of reliable behaviors [16]. The IBk classifier shows the accuracy of 80.1370%. It attires the k Nearest Neighbor algorithm; a user defined variable depends on kind of data preferred. Precision rate is 76%, TP Rate is 80.36%, FP rate is 18.31% and ROC area being 90.5%.

IV. CONCLUSION

The result of our execution shows IBk algorithm and simple k-Means algorithm give finest outcome for classification and clustering. These results can be used by anti-malware writers for detecting whether a particular malware is malicious or benign as behavior is the most important factor for deciding whether a file is malicious or benign. Some more insights can also be drawn on the behavior of the malware including the file being affected, Registry keys being used; IP addresses contacted, import functions, etc.

In our future, we may select different dataset in order to that assessment prepare at a larger scale to provide high accuracy and efficiency. For this purpose, we use a combination of one or more algorithms are applied to speed up analysis process. Further, we analyze Zero day's attacks in a much more efficient way.

REFERENCES

1. (2013) The Need for Speed: 2013 Incident Response Survey, Fire Eye. http://www.inforisktoday.in/surveys/2013-incident response-survey-s-18.
2. (2013) Next Generation Threats. http://www.fireeye.com/threat-protection/.
3. (2012) Addressing Big Data Security Challenges: The Right Tools for Smart Protection http://www.trendmicro.com/cloudcontent/us/pdfs/business/white-papers/wp_addressing-big-data-security-challenges.
4. Al-Assad (2011, April).Syrian Malware Samples [online]. Available:http://www.syrianmalware.com/, drafted on November 20, 2015 at 11:57 pm.
5. You, I. and Yim, K.(2010), Malware Obfuscation Techniques: A Brief Survey. *Proceedings of International conferenceon Broadband, Wireless Computing, Communication and Applications*, Fukuoka,pp 297-300.
6. Tian, R., Batten, L., Islam, R. and Versteeg, S. (2009) An Automated Classification System Based on the Strings of Trojan and Virus Families.*Proceedings of the 4th International Conference on Malicious and Unwanted Software*, Montréal, 13-14 October 2009, 23-30.
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. (2009) The WEKA Data Mining Software: An Update.*ACM SIGKDD Explorations Newsletter*, 10-18.
8. K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic Analysis of Malware Behavior using Machine Learning", 2009.
9. Tian, R., Batten, L. and Versteeg, S. (2008) Function Length as a Tool for Malware Classification. *Proceedings of the 3rd International Conference on Malicious and Unwanted Software*, Fairfax, 7-8 October 2008, 57-64.
10. K. Rieck, T. Holz, C. Willems, P. Duessel, and P. Laskov, "Learning and Classification of Malware Behavior", DIMVA, LNCS 5137, pp. 108–125, Berlin Heidelberg: Springer-Verlag, 2008.
11. M. Christodorescu, S. Jha, and C. Kruegel, "Mining Specifications of Malicious Behavior", Proceedings of the 6th joint meeting of the ESEC and the ACM SIGSOFT Symposium on the FSE, September 3–7, Dubrovnik, Croatia, ACM, 2007.
12. Claudio Guarnieri(2007), Cuckoo[online].Available:http://cuckoosandbox.org/about.html, drafted on November 20, 2015 at 11:46 pm
13. Lenny Zelster(2007). Malware Sample Sources for Researchers[online].Available: https://zeltser.com/malware-sample-sources.
14. Bayer, U., Moser, A., Kruegel, C. and Kirda, E. (2006) Dynamic Analysis of Malicious Code. *Journal in Computer Virology*, **2**, 67-77. http://dx.doi.org/10.1007/s11416-006-0012-2.
15. Kolter, J. and Maloof, M. (2004) Learning to Detect Malicious Executables in the Wild. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*,pp 470-478.
16. Cohen, W. (1995) Fast Effective Rule Induction. Proceedings of 12th International Conference on Machine Learning, San Francisco,pp 115-123.
17. WEKA, the University of Waikato, Available at: http://www.cs.waikato.ac.nz/ml/weka/, drafted on November 20, 2015 at 11:47 pm.
18. MatteoCantoni. Malware archives download [online].Available:http://www.nothink.org/ honeypots/ malware-archives/, drafted on November 20, 2015 at 11:52 pm.

