Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Detecting Zero Day Malware

Ashwin Kumar H R, Deepika S,
G A Priyanka, Neha Bindinganavalle
Students, Dept of ISE
BNMIT

Manjunath G.S
Asst. Professor, Dept. of ISE
BNMIT

***Abstract*:- Computer Networks have always been overwhelmed by self-propagating malware. Malware or malicious software is any file that is harmful to a computer. A malware could either be computer viruses, worms, Trojan horses or spyware. The previously unknown security vulnerabilities are exploited through these malware and cause a Zero-Day Attack. The traditional solutions of malware detection use signatures of malware to detect their presence. But these methods get evaded due to some obfuscation techniques used by malware authors. This paper highlights the existing methodologies used for detecting and analyzing these obfuscated malicious codes. This paper also presents a Survey on the various existing malware detection systems and proposes a novel Zero-Day malware detection model that can efficiently distinguish between a malware and a benign sample. The survey includes the various methodologies used to detect malicious files along with a note on the results achieved.**

I. INTRODUCTION

Malware is any software that is specifically designed to damage, disrupt, or gain unauthorized access to a computer. It is notoriously difficult to combat because it appears and spreads very so quickly. It was estimated that in the year 2009 every fourth computer operating in the United States had been infected with a malware. [7] In 2012 McAfee Labs identified more than 75 million new malware samples, ie on an average 55,000 new samples of malware were identified per day. Similarly, Panda Labs reported more than 60,000 new malware samples being introduced per day in 2013 resulting in an average of more than 73,000 per day in the first quarter of 2014 [9]. Studies show that over 317 million new malware specimens were discovered in the year 2014, which means about 1 million new malware released on a daily basis. This number had increased to a whopping 430 million in the year 2015 [12]. The total number of malwares had been increasing exponentially since 2008 and had reached more than 583 million. [14].

Unfortunately, the problem of uncontrolled growth of malicious code is likely to continue in the future, because writing malware is quickly turning into a profitable business. Malware authors often sell their creations to malefactors, who use these malicious codes to compromise a large number of computers that get linked together to form botnets. These botnets are then used to launch DOS (denial-of-service) attacks or as spam relays.[3]

A major challenge anti-malware researchers are facing in recent days is the sheer number of new malware samples that are appearing every day. The common Security products like virus scanners look for signatures in the sample as most malware evolves from existing malware as a new variant. Signatures are characteristic byte sequence that helps in identifying a malware code. However, Malware has adapted to this approach. Metamorphic worms avoid their detection by changing their appearance or by behaving differently in a controlled environment. For example, flash worms silently observe without infecting vulnerable machines, waiting to pursue a strategic spreading plan so that they can infect thousands of machines at a time within seconds [1]. Besides failing to detect threats with evolving capabilities such as metamorphic and polymorphic malware, Another major limitation of the signature-based approach is its failure to detect zero-day attacks, which are emerging threats that are previously unknown to the malware detector system [8].

The manual technique of examining malware to extract the signature or to determine the intent of the code does not scale in accordance with the ever increasing volume of malware. An automated malware classification system is, therefore, an important aid to anti-malware researchers to speed up the analysis process [4].
The remaining of this survey paper is organized as follows: Section II presents an overview of the various methods of detecting malware. Section III discusses in detail the existing systems. The proposed system is presented in Section IV and the expected Results in Section V. In the end, we summarize the paper in Section VI.

II. MALWARE DETECTION METHODS

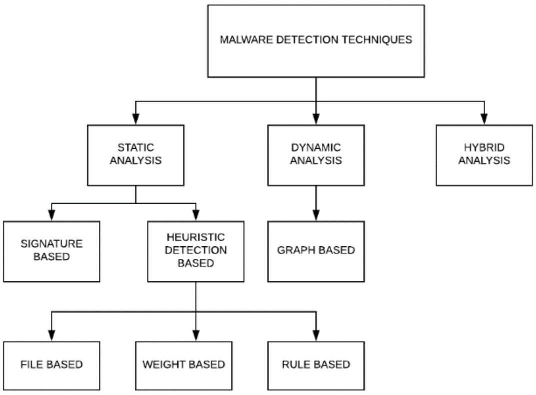

Fig 2.1 Malware detection techniques

1 **Static analysis** is the process of evaluating a software without executing it. Basic static analysis examines a malware without viewing the actual code or instructions. It employs different techniques to determine whether a file is

malicious or not, provides information about its functionality and also collects technical indicators to produce signatures. Technical indicators gathered with basic static analysis can include file name, file size, file version, MD5 checksums or hashes, file type, etc.

1.1. **Signature-based detection technique**: This technique is similar to fingerprinting technique or pattern matching mask. To recognize a malware, the malware detector searches for a formerly specified signature in the code. Commercial antivirus scanners too look for signatures.
The implication is that there are certain unique factors that defines a piece of code. While this may be true only in the case of a certain sample, given the variety of obfuscation techniques, it is unlikely to be true for a general family; there may be several features in a piece of code which together indicates its purpose, but separately do not definitively reveal this information.[4]

1.2.**Heuristic detection technique**: Heuristic detection is quite similar to signature-based detection, except that instead of searching for a specific signature, heuristic detection searches for a certain set instructions or commands within a program that are usually not found in application programs. Thus a heuristic engine is able to detect potentially malicious functionality(such as replication mechanism of the virus) in a new, previously unknown sample.

- **File-based heuristic analysis** technique is also called file analysis. This method involves the software taking an in-depth look at the file and trying to understand its destination and purpose and determine its intent. If the file contains commands to delete or damage another file, then it is noted as malicious.
- **Weight-based heuristic analysis** is a quite old styled approach. It rates every functionality it detects with a certain weight based on the degree of danger it may pose. If the sum of those weights reach or exceed a certain threshold, it is noted as a malicious sample
- **Rule-based heuristic analysis** the rule-based analyzer extracts certain rules from a file and these rules will then be compared against a set of pre-determined rules to identify a code as malicious. If a rule match is found, an alarm can be triggered.

2. **Dynamic analysis detection techniques:** Dynamic analysis involves running the malware in a controlled environment like a sandbox to observe and understand its functionality and behavior, identify technical indicators that can be used in detection signatures. Technical indicators revealed during dynamic analysis include file path locations, domain names, IP addresses, register keys and additional files located on the system. In addition to this, it will also identify any communication with an external server controlled by an attacker for command and control purposes or if an attempt is being made to download additional malware files.

2.1 **Graph-Based analysis techniques:** This method uses a combination of graph kernels to create a similarity matrix between the instruction trace graphs. The resulting graph kernel measures the similarity between graphs on local and global levels. Finally, the similarity matrix is sent to a classifier to perform classification

3 **Hybrid analysis detection techniques**: This technique is a combination of both static analysis and dynamic analysis. Hybrid Analysis saves a fine-grained memory dump snapshots of the monitored runtime processes as well as symbol information to perform static analysis.

III. SURVEY ON EXISTING SYSTEMS

Paper	**Authors**	**Methodology**	**Conclusion**	**Release year**
Data Mining methods for detection of new malicious executables	Matthew G Schultz, Eleazar Eskin, Erez Zadok.	Uses a data-mining framework that detects new but previously unseen malicious executables accurately and automatically. Comparing the detection methods with a traditional signature-based method, this method more than doubles the current detection rates for new malicious executables.	The Multi-Naive Bayes algorithm has the highest accuracy and detection rate of any algorithm over unknown malware, 97.76%, over double the detection rates of signature-based methods. The methods of this paper were being implemented as a network mail filter.	IEEE 2001
Toward automated dynamic malware analysis Using CWSandbox	Carste Willems, Thorsten Holz, Felix Freiling,	Combines dynamic malware analysis, API interruptions, and DLL injection in codes within the CWSandbox and lets analysts trace and monitor all relevant system calls and generates an automatic, machine-readable report that describes if the file is malware	Assembles the techniques of API hooking and dynamic linked library (DLL) injection in a unique combination that provides a fully functional, simple, and arguably powerful automated malware analysis tool.	IEEE 2007
Exploring Multiple Execution Paths for Malware Analysis	Andreas Moser, Christopher Kruegel, Engin Kirda.	This paper proposes a system that explores multiple execution paths and identifies malicious actions that are executed only when certain conditions are met. This helps to automatically	Presented a system to explore multiple execution paths of Windows executables. The goal is to obtain a more comprehensive overview of the actions that an	IEEE 2007

		extract a more complete view of the program under analysis and identify under which circumstances, suspicious actions are carried out	unknown sample can perform. In addition, the tool automatically provides the information under which circumstances a malicious action is triggered.	
Malware Detection using Statistical Analysis of Byte-Level File Content	S. Momina Tabish, M. Zubair Shafiq, Muddassar Farooq	The novelty of this approach, compared with existing content based mining schemes, is that it doesn't memorize specific byte-sequences or strings shown in the actual file content. The technique implemented is non-signature based and therefore has the potential to detect previously unknown and zero-day malware.	Results of this method showed that the proposed non-signature-based technique surpasses the existing techniques and achieves more than 90% detection accuracy. Also performed a comparison with existing data mining based malware detection techniques.	CSI-KDD 2009
An Automated Classification System Based on the Strings of Trojan and Virus	Ronghua Tian, Lynn Batten, Rafiqul Islam, Steve Versteeg.	The processing phases include: preprocessing of the string data extracted from the software, feature extraction and selection, classification itself, and finally the evaluation of the classification result. Used tree-based classifiers, nearest neighbor statistical algorithm, and AdaBoost.	Five algorithms used, representative of five different generic approaches to classification and used each of them separately and then each in conjunction with a boosting technique AdaBoost. It was observed that the Random Forest method of classification was the most efficient way.	IEEE 2009
Analysis of machine learning techniques used in behavior-based malware detection	Ivan Firdausi, Charles Lim, Alva Erwin, Anto Satriyo Nugroho.	Uses automated behavior-based malware detection using machine learning techniques. The behavior of each malware on a sandbox will be automatically analyzed and reports are generated. These reports will be preprocessed into sparse vector models for further classification processes.	The feature selection process was presented using Best First search algorithm. This best performance was achieved by J48 using the frequency-weight without feature selection data set, with a recall of 95.9%, a false positive rate of 2.4%, a precision of 97.3%, and an accuracy of 96.8%.	IEEE 2010
Graph-based malware detection using dynamic analysis	Daniel Quist, Joshua Neil.	Uses a novel malware detection algorithm by forming Markov chains and the combination of graph kernels to create a similarity matrix between the instruction trace graphs. Finally, the similarity matrix is sent to a support vector machine to perform classification.	Demonstrated the performance of the algorithm on two classification problems: benign software versus malware, and the Netbull virus with different packers versus other classes of viruses. The result shows a statistically significant improvement over signature-based and other machine learning- based detection methods.	Journal in Computer Virology . November 2011
Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection	Prakash Mandayam Comar, Lei Liu, Sabyasachi Saha, Pang-Ning Tan, Antonio Nucci.	This paper presents a unique machine learning primarily based framework to detect known and newly emerging malware at a high precision using layer 3 and layer 4 network traffic features. The framework leverages the accuracy of supervised classification in detecting known classes with the adaptability of unsupervised learning in detecting new classes.	The proposed approach addresses the challenges and identifies flows of existing and novel malware with very high precision. For this, a tree-based feature transformation approach is developed to handle the data imperfection issues. Finally, we present a novel adaptation of 1-class SVM to identify new types of malware.	IEEE 2013
Detecting and Analyzing Zero-day Attacks using Honeypots	Constantin Musca, Emma Mirica, Razvan Deaconescu.	This paper presents methods for analyzing malicious traffic by using a honeypot system and analyzing it in order to automatically generate attack signatures for the Snort intrusion detection/prevention system. The honeypot is deployed as a virtual machine and its job is to log as much information as it can about the attacks.	Implemented the processing logic for both types of honeypots: a high-interaction honeypot and a low-interaction honeypot. Both solutions are promising and give relevant output. It is easier to use and implement a detecting method for Honeyd as it offers logging capabilities.	IEEE 2013
Integrated static and dynamic analysis for malware detection	P. V. Shijo, A. Salim.	Uses cuckoo framework over VMWare on Ubuntu 10.04. The dynamic analysis searches for n-grams for API calls. Uses the weka library for classification using random forest and support vector machine Used virus share for the database. N-gram: is a continuous sequence of n items. In this paper, the sequence are the API calls	Presented an integrated approach that uses both static and dynamic features for malware detection. They have proven the thesis that combined static and dynamic features will increase the detection accuracy than stand-alone static and dynamic methods. The results show that the support vector machine learning technique is best equipped to classify our data	ICICT 2014
Towards probabilistic identification of Zero-day attack Paths	Xiaoyan Sun, Jun Dai, Peng Liu,	Implements a probabilistic approach to identify zero-day attack paths and implement a prototype system named	Uses Bayesian networks to identify the zero-day attack paths. By incorporating the intrusion	IEEE 2016

	Anoop Singhal, John Yen.	ZePro. An object instance graph is 1st designed from system calls to capture the intrusion propagation.	evidence and computing the probabilities of objects being infected, the implemented system ZePro can successfully reveal the zero-day attack paths.	
Zero-day malware detection	Ekta Gandotra, Divya Bansal, Sanjeev Sofat.	Proposes a model that uses an integration of both static and dynamic analysis features of malware binaries incorporated with the machine learning process for detecting zero-day malware. This is model is tested and validated on a real-world corpus of malware.	The experimental results show that the integrated feature set provides a very good accuracy. Further, demonstrating that the inclusion of the filter approach for relevant feature selection can improve the model building time without compromising the accuracy of the system.	IEEE 2016
A Framework for Zero Day Exploit Detection and Containment	Richard Ciancioso, Danvers Budhwa, Thaier Hayajneh.	This paper designed a framework utilizing the most efficient methods to detect and contain zero-day exploits. Analyzing the ability of multiple AMTs to detect zero-day malware will assist cybersecurity professionals in selecting the most compatible for their respective environments and defend against these attacks	The results showed that while most Anti-malware tools were able to detect malware created with evasive techniques, it was not successful in detecting zero-day malware. This further confirms the proposal to have a unique anti-malware detection tool combined with sandboxing.	IEEE 2017
A Novel Malware Analysis Framework for Malware Detection and Classification using Machine Learning Approach	Kamalakanta Sethi, Shankar Kumar Chaudhary.	Uses Cuckoo Sandbox for generating static and dynamic analysis report by executing the sample files in the virtual environment. In addition, a novel feature extraction module has been developed which functions based on static, behavioral and network analysis using the reports generated by the Cuckoo Sandbox. Weka Framework is used to develop machine learning models by using training datasets.	In this paper, a novel intelligent malware analysis framework has been developed for dynamic and static analysis of malware samples based on their behavior. Experimental results demonstrate acceptable performance of the proposed procedures in detecting and classifying malicious files using machine learning models in Weka. J48 Decision tree showed the best performance in terms of accuracy and precision.	ICDCN 2018

IV. PROPOSED SYSTEM

This section proposes a system that performs hybrid analysis and carries out both static and dynamic analysis simultaneously. As discussed in section 3 hybrid analysis on the malware sample provided a best and efficient classification system that effectively distinguished between benign and malware samples. Static analysis should be performed initially to identify any signatures if present. Static analysis helps to extract other information such as the header and packing information of the sample. Dynamic analysis is to be performed in a controlled environment such as a sandbox in order to obtain a detailed report on the behavior of the sample. From the two analysis and the several features obtained, the most important features that help in effectively differentiating between a malware and a benign sample are to be extracted. A classification model is to be built using these features with the help of classification algorithms from the WEKA library. The system can be designed as seen below.

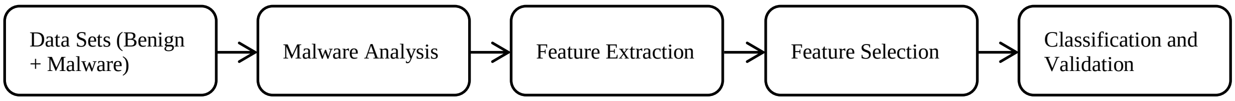

Fig 4.1 Proposed system design

V. EXPECTED RESULTS

The results of the system would be the classification models built on the features that are extracted by performing static and dynamic analysis. The performance of the algorithms used for building the classification can be measured using metrics such as recall, precision, accuracy, and area under the curve. The metrics can be calculated by using the number of true positive, false negative, true negative and false negative classification of the samples. The time required to build the classification model can also be used to compare the working of the different classification algorithms.

VI. CONCLUSION

Advanced malware is a severe threat to the internet and the user's computer systems. Traditional Antivirus products are only able to detect those malware that have been previously caused damage and are registered as malware. This survey paper explains about the different malware analysis techniques namely static, dynamic and hybrid malware analysis techniques. The static analysis gives information about the sample without actually running or executing it. The dynamic analysis runs the code on a sandboxed environment and gives a detailed analysis of its behavior. The hybrid analysis uses both these techniques to provide a better classification model to contradistinguish between a malware sample and a benign sample. The existing Machine Learning algorithms need to be transformed so that their full potential can be leveraged to address the challenges and threats being posed in cybersecurity.

REFERENCES

[1] Matthew G Schultz, Eleazar Eskin, Erez Zadok. “Data Mining methods for detection of new malicious executables”, IEEE 2001.

[2] Carste Willems, Thorsten Holz, Felix Freiling, “Toward automated dynamic malware analysis Using CWSandbox”, IEEE 2007.

[3] Andreas Moser, Christophr Kruegel, Engin Kirda, “Exploring Multiple Execution Paths for Malware Analysis”, IEEE 2007.

[4] S. Momina Tabish, M. Zubair Shafiq, Muddassar Farooq, “Malware Detection using Statistical Analysis of Byte-Level File Content”, CSI-KDD 2009.

[5] Ronghua Tian, Lynn Batten, Rafiqul Islam, Steve Versteeg, “An Automated Classification System Based on the Strings of Trojan and Virus”, IEEE 2009.

[6] Ivan Firdausi, Charles Lim, Alva Erwin, Anto Satriyo Nugroho., “Analysis of machine learning techniques used in behavior-based malware detection”, IEEE2010.

[7] Daniel Quist, Joshua Neil, “Graph-based malware detection using dynamic analysis”, Journal in Computer Virology · November 2011.

[8] Prakash Mandayam Comar, Lei Liu, Sabyasachi Saha, Pang-Ning Tan, Antonio Nucci, “Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection”, IEEE 2013.

[9] Constantin Musca, Emma Mirica, Razvan Deaconescu, “Detecting and Analyzing Zero-day Attacks using Honeypots”, IEEE 2013.

[10] P. V. Shijo, A. Salim, “Integrated static and dynamic analysis for malware detection”, ICICT 2014.

[11] Xiaoyan Sun, Jun Dai, Peng Liu, Anoop Singhal, John Yen, “Towards probabilistic identification of Zero-day attack Paths”, IEEE 2016.

[12] Ekta Gandotra, Divya Bansal, Sanjeev Sofat, “Zero-day malware detection”, IEEE 2016.

[13] Richard Ciancioso, Danvers Budhwa, Thaier Hayajneh, “A Framework for Zero Day Exploit Detection and Containment”, IEEE 2017.

[14] Kamalakanta Sethi, Shankar Kumar Chaudhary, “A Novel Malware Analysis Framework for Malware Detection and Classification using Machine Learning Approach”, ICDCN 2018.