International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-4, April 2019

Integrated Malware Analysis Using Markov Based Model in Machine Learning

S.S.Subashka Ramesh, Kartik Singh Rathore, Ritik Raj, Kumar Vatsalya, Mridula Vatsa

Abstract*: *In the world full of advanced gadgets and communication rolled out everywhere, in order to overcome the malicious nature of any attacker, it is necessary and an important aspect to analyses and detect malware. The gadgets now a days are used for internal transaction and also in banking sector which enhances their vulnerabilities which a malicious attacker can use up to his advantage. There are several types of analysis mainly static and dynamic analysis which can be used depending on the condition and the nature of attack. Time is an important factor and here, the Markov model surpasses Noriben as it takes a lot lesser time and is more efficient.

Index Terms*: *Malicious attacker, Static analysis, Dynamic analysis, Markov model.

I. INTRODUCTION

Malwares are mostly malicious in nature and are developed with a purpose to harm the systems. Malwares can be further classified as Trojans, adware, spyware etc. The nature and the intensity with which it hits the software determines its types. Some malwares are just capable of distorting the data while some can actually harm the software.The malware has been attacking the countries and uses since ages and according to a survey, it was found that almost 35.91% of Bangladesh users are attacked by mobile malware during the 2018 period. Other top countries which are attacked and affected by the same are Nigeria, Iran, China, India, Pakistan and Algeria.Every year the pool of attacker increases with the pool of civilians. The cybersecurity companies have tried from heaven to earth to bring up new technologies to stop the malicious attackers from doing the same. Using diverse fields as hardware authentication and machine learning we also open new threats and new challenges to overcome with new technologies.

Analysis of software is a crucial part of the malware analysis and can be done using two types:

- Static Analysis
- Dynamic Analysis

Where dynamic analysis analyses the runtime behavior, the static uses the hashes for the identification of binary file.

There are mainly 4 mechanisms to deal with this:

- Anti-VM
- Anti-Debugging
- Obfuscation

II. RELATED WORK

A lot of existing systems are ill-suited for the advancements in technology and malware. [1] Akash Kumar Singh, Aruna Jain performed automatic detection of advanced and unknown malwares using the results of static and dynamic analysis by training the classifier. However, performance of a classifier suffers with addition of more data. Ivan Firdausi, Charles Lim, Alva Erwin and Anto Satriyo Nugroho [6] again make use of classifiers in their work by using automatic malware analysis for detecting the malwares.

RansHunt [2] uses cuckoo sandbox which is now susceptible to the sandbox bypass that allows bypassing of user mode API hooks. The technique makes use of malware wrapped with a Visual Basic custom packer named VBCrypter. The malware families protected by VBCrypter include Fariet, Lokibot, NanoCore, NetWireRec, and Remcos to name a few. Rodrigo Rubira Branco, Udi Shamir [3] proposed an architecture that automates malware analysis by using a which gets the machine's availability and automatically reverts virtual machines. Also, packet sniffers are used to detect the traffic generated by machines. MASS [4] is a setup for malware analysis that had the motive to give power coo-operation between all the researchers of the malwares. The execution time of analysis affects the performance of the MASS. Malware visualization is another field in malware detection that uses visual cues confirm a lot of information about the particular detection of the malwares. Malware behavior and its potential benefit for malware classification are highlighted bySyed Zainudeen Mohd Shaid, Mohd Aizaini Maarof[5] in their work. Their research suggest how malware behavior visualization can be used to identify malware variants with high accuracy.

III. PROBLEM STATEMENT AND PROPOSED METHODOLOGY

The development of zero - day vulnerabilities and the increment in the quantity of malwares requests a proficient and precise location of malwares. This leads to the need of great importance is machine learning arrangement.

Manuscript published on 30 June 2019.
***** Correspondence Author (s)
S.S. Subashka Ramesh*, Computer Science & Engineering, SRM Institute of Science& Technology, Assistant Professor, Ramapuram, Chennai 600089, India.
Kartik Singh Rathore, Computer Science & Engineering, SRM Institute of Science & Technology, Student, Ramapuram, Chennai 600089, India.
Ritik Raj, Computer Science & Engineering, SRM Institute of Science& Technology, Student, Ramapuram, Chennai 600089, India.
Kumar Vatsalya, Computer Science & Engineering, SRM Institute of Science& Technology, Student, Ramapuram, Chennai 600089, India.
Mridula Vatsa, Computer Science & Engineering, SRM Institute of Science & Technology, Student, Ramapuram, Chennai 600089, ndia.

The primary issue looked by the majority of the malware examiners are the counter investigation systems utilized by the malware designers to upset examination and avoid recognition. None of the inspected papers referenced about identifying those malwares which utilize hostile to examination methods. Static examination along these lines flops in identifying new noxious executables for which it is hard to discover a predefined mark or personality. Investigating just PE32 headers won't give enough data about the executable. The Printable String Data (PSI) and capacity length recurrence include extraction must be joined with different highlights of hostile to - examination. There must be a distinct guideline to choose the limit for every PSI above which the document is delegated malware. Therefore, we propose to take a shot at an answer where we will incorporate the highlights of against investigation procedures amid our static examination stage

The dynamic examination of malware is a significant time adopting strategy. Aside from this, the virtual condition and devices utilized for dynamic component extraction can without much of a stretch be identified by the malwares which can disturb their investigation work. For dynamic examination, we are utilizing a lesser realized condition called Markov Model Based Detection is a method which is a virtual condition which will take less time than most regularly utilized sandbox, for example, CUCKOO and NORIBEN. The principle goals of our work are:

- To utilize Markov Model machine learning method to perform malware recognition utilizing Hidden Markov Model.
- To remove social highlights of malware utilizing a Hidden Markov Model to demonstrate a framework that is thought to be a Markov procedure with shrouded states. In shrouded Markov Model, since the state is concealed the client can just watch the yield.
- To make a perception arrangement that is gotten from the malware the Hidden Markov Model can be prepared. Subsequent to preparing we can test the perception arrangement against a prepared Hidden Markov Model to decide the probability of the succession being a piece of the model. A high likelihood shows a decent match.

A. Architecture of The Proposed Method

We have developed a framework for Markov Model which includes the static and dynamic analysis of execution. In static execution it is performed by extracting the features of static analysis and passing the extraction to the Static Markov Model. In the Dynamic analysis, it is performed by the extracting the features of the dynamic analysis and passing the extraction to the Dynamic Markov Model.

The results from both the Static and Dynamic Markov Model goes through the integrated analysis where a single report is generated which gives us the result about the malware.

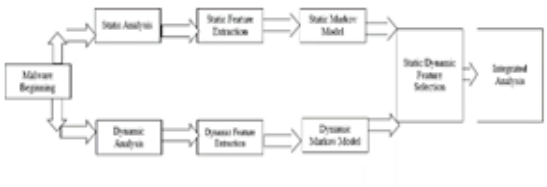

Fig.1.Proposed Methodology Architecture

B. Data Set

The informational index utilized in the examination are based for windows Portable Executable document group that incorporates test for malware of different classes for the most part comprises of adware, spywares, packed malwares, and remote organization trojans. The principle purpose behind choosing the malware of these classes is because of the way that the majority of the counter investigation procedures are commonly found on the malware type. The complete two-fold executable have utilized is 107.

IV. APPROACH TO MALWARE ANALYSIS

In this section, we can discuss the methods for extracting anti-analysis features the use of static analysis.

A. Suspicious Module

Before we discuss about the methods to fetch the suspicious API and calls, we first talk about the Portable Executable (PE) area which is utilized by the Windows executable files. The Portable Executables contains various kinds of information in its suspicious capacity and API'S extricated from the test of the malware.

Fig.2. Lists of function

B. Anti-VM Module

The malware designers are aware of the way
area. Field and information uncovered use the versatile executables design.

- Imports: The external library function is being used by the malware imports and the outer library is being used by the malware.
- Sections: The names of areas and the size of their circle are in the same memory.
- Resources: The strings, symbols, menus, and other data incorporated into record.
- Time Date & Stamp: This gives the details of the time and date when malware was accumulated.
- Exports: The functions that are in the malware are called by different project or library.
- Subsystem: This indicates if the program is a direction line or GUI application.

Utilizing the information, we parsed the executables utilizing an outside module accessible in python called PEFILE. Subsequent to parsing, we acquired the imports utilized by the executables. For removing suspicious capacities and we have officially utilized most recent predefined marks. For instance, if the program starts importing the capacity it is very evident that it is endeavoring to begin another string in a remote procedure which can be utilized by the malware to begin another procedure inside the current procedure.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

The Executable is corrected of a rundown about the predefine suspicious API'S, if the rundown matches with the imported table of executables, then we take the API or capacity for highlighting vector. Fig.2 demonstrates arrangements ofthat the vast majority of the malware examination are completed in a virtual situation. As the malware can taint the primary machine amid its investigation so malware experts can utilize virtual machines. Knowing the reality, the malware designers utilize Anti-VM strategies to find the nearness of virtual machines. On off chance that the malware recognizes the machine, this will either just not run or pulverize itself so as to disturb examination or discovery. Figure 3 demonstrates the Anti-VM system recognition of an example.

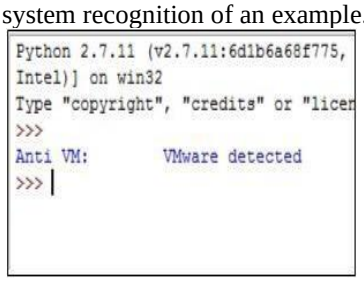

Fig.3.Anti-Vm Technique Detection

C. Packer Module

The packers are commonly utilized as a product defender to keep saltines from breaking the owner programming. Be that as it may, packers are utilized by the malware designers nowadays to pack their vindictive records to stay away from antivirus discovery. At whatever point the malware examiner open a pressed malware, he approaches just to the packer, so as to unload the malignant program the expert must fix the techniques performed by them to pack the program, which is very hard to perform. In our Markov Model Based Detection, we have utilized to identify the sort of packer utilized by the malware to stay away from discovery, parse the executables into different headers and codeareas which helps in investigating the pressed document in a lot simpler manner. This program comprises of client dB which comprises of the dB of different packers which have been available till now. The python program utilizes customary articulation to coordinate with the packers living in client db. Fig.4 indicates packer identification for the test of malware.

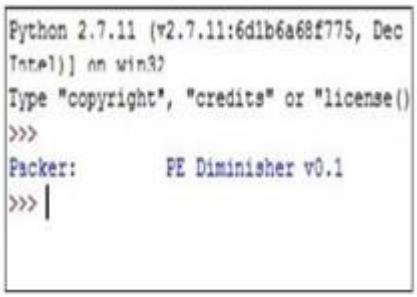

Fig.4. PackerDetection

V. RESULTS AND DISCUSSION

There are three different units found using static analysis, dynamic analysis and integrated analysis approach on forest, solicit systems and support vector machine. The table shows TPR and FPR for the following mentioned above.

When it comes to static analysis given in table 5, it can be seen that the precision is the most in support vector machine. When dynamic analysis table is taken into consideration, which is given in table 6, the precision again is the most in support vector machine. It all changes when integrated Markov analysis is taken into consideration where the precision is more in solicit systems. Due to the anti- analysis technique, there is a deflection for support vector machine from 63.3% in dynamic analysis to 59% in static analysis.

As with the reference to other papers too, it is obvious that the precision and detection rate is a lot better in integrated analysis compared to other analysis techniques. However, a high probability results in a great match, hidden Markov based model using integrated technique in machine learning has proved to be efficient and faster compared to static and dynamic techniques

Table V. Static Analysis Classification

Classifier	TPR	FPR	Precision	Recall	Accuracy
Solicit systems	67.4	8.3	68.29	68.2	68.2315
support vector machine	62.9	8	71	71.04	71.0131
Forest	69.4	7.1	68.18	69.7	69.72

Table VI. Dynamic Analysis Classification

Classifier	TPR	FPR	Precision	Recall	Accuracy
Solicit systems	54.1	10.7	57.2	54.1	54.12
Support vector machine	60.6	9.3	59.4	60.6	60.55
Forest	63.3	9.5	59.01	63.3	63.302

Table VII. Hidden Markov Based Model

Classifier	TPR	FPR	Precision	Recall	Accuracy
Solicit systems	70.6	6.8	69.5	70.6	70.642
Support vector machine	66.1	7.8	64.7	66.1	60.055
forest	73.3	7.4	73.1	73.5	73.47

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

VI. CONCLUSION AND FUTURE SCOPE

In this experiment, we have used hidden Markov based model for integrated detection and analysis of malware techniques. The hmm can be trained using machine learning. The result shows that Markov model is way more effective and faster compared to the regular static or dynamic analysis.

In the future, we plan to expand and figure out solutions to more important static and dynamic techniques to increase the accuracy of malware analysis. Hence the main objective would be to reduce time to do the same in future scope.

REFERENCES

1. Akash Kumar Singh, Aruna Jain "Integrated Malware Analysis Using Machine Learning" 2017 2nd International Conference on Telecommunication and Networks (TEL-NET 2017).
2. Md Mahbub Hasan, Md. Mahbubur Rahman "RansHunt: A Support Vector Machines Based Ransomware Analysis Framework with Integrated Feature Set" 2017 20th International Conference of Computer and Information Technology (ICCIT), 22- 24 December 2017.
3. Rodrigo Rubira Branco, Udi Shamir, "Architecture for Automation of Malware Analysis", 2010 5th International Conference on Malicious and Unwanted Software.
4. Fabian Rump, Timm Behner, Raphael Ernst, "Distributed and Collaborative Malware Analysis with MASS", 2017 IEEE 42nd Conference on Local Computer Networks.
5. Syed Zainudeen Mohd Shaid, Mohd Aizaini Maarof, "Malware Behavior Image for Malware Variant Identification", 2014 International Symposium on Biometric and Security Technologies (ISBAST).
6. Ivan Firdausi, Charles Lim, Alva Erwin, Anto Satriyo Nugroho," Analysis of machine learning techniques used in behavior-based malware detection", 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies.

AUTHORS PROFILE

S.S. Subashkareceived her Bachelor of Engineering in Computer Science and Engineering from in 2006. After graduating in her bachelor's degree, she pursued Master of Engineering in the year 2008. She has a Ph.D. in the same.She is Assistant Professor at SRM Institute of Science and Technology, Chennai, Tamil Nadu. Herresearch interest is in Data Mining and Taxonomy.

Kartik Singh Rathoreis currently pursuing his Bachelor ofEngineering and Technology in Computer Science and Engineering at SRMInstitute of Science and Technology in 2019. He is passionate about DBMS, SQL and Networking. He hasdeveloped projects based on cloud computing, IoT and Data Analysis.

Ritik Rajis currently pursuing his Bachelor of Engineering and Technology in Computer Science and Engineering at SRM Institute of Science and Technology in 2019. He is passionate about the Devops, Machine Learning, Cloud Computing. He has developed projects based on Cloud Computing, IoT and Data Analysis.

Kumar Vatsalyais currently pursuing his Bachelor of Engineering and Technology in Computer Science and Engineering at SRM Institute of Science and Technology in 2019. He is passionate about the DBMS and Cloud Computing. He has developed projects based on Python and Web Development.

Mridula Vatsais currently pursuing her Bachelor of Engineering and Technology in Computer Science and Engineering at SRM Institute of Science and Technology in 2019. She is passionate about the DBMS, SQL, Machine Learning, Data Analytics. She has developed projects based onData Analytics.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)