IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING IS USED FOR STATIC AND DYNAMIC MALWARE ANALYSIS

K.DEEPA
Department of Computer Science and Engineering,
Tamilnadu College of Engineering,
Coimbatore, Tamilnadu, India- 641 659.

Dr. A. S. SHANTHI
Department of Computer Science and Engineering,
Tamilnadu College of Engineering,
Coimbatore, Tamilnadu, India- 641 659.

G.KOKILA
Department of Computer Science and Engineering,
Tamilnadu College of Engineering,
Coimbatore, Tamilnadu, India- 641 659.

J.SIKKANDHAR BATCHA
Department of Computer Science and Engineering,
KGISL Institute of Technology,
Coimbatore, Tamilnadu, India- 641 035.

ABSTRACT

Malware detection is a very important factor in security of internet oriented machines. The combination of different features is used for dynamic malware analysis. The different combinations are generated from APIs, Summary Information, DLL and Registry Keys Changed. Cuckoo sandbox is used for dynamic malware analysis, which is customizable, and provides good accuracy. More than 2300 features are extracted from dynamic analysis of malware and 92 features are extracted statically from binary malware using PEFILE. We used machine learning to discover different types of windows malwares. The dataset used to train the model has static and dynamic analysis of different programs. According to the data every one program was labeled as humanely or malware. There are 6 types of malwares in total: Backdoor, Trojan, Trojan Downloader, Trojan Dropper, Virus, and Worm. Static features are extracting from 39000 hateful binaries and 10000 benign files. Dynamically 800 humanely files and 2200 malware files are analyzed in Cuckoo Sandbox and 2300 features are extracted. The precision of dynamic malware analysis is 94.64% while static analysis precision is 99.36%. The dynamic malware analysis is not effective due to tricky and intelligent behaviors of malwares. The dynamic analysis has some restrictions due to controlled network behavior and it cannot be analyzed completely due to limited access of network.

Keyword: API, DLL, Registry Key, Static and Dynamic analysis

1. INTRODUCTION

Malware is the shortest term used for malicious software which is a harmful malicious piece of code. Malware's intention is to harm computer or steal information from system by exploiting vulnerabilities in existing security infrastructure. Malwares are rapidly increasing with the passage of time and we can categorize malware in to different categories according to their behaviors. The malware can be a script, executable binary or any other piece of code, which have malicious intention. The main aims of malware are to gain access of system, disrupt system services, denial of service, and steal confidential information and destruction of resources. Sometime malware is not defective software but some legitimate software can have malware inside it. Legitimate software often acts as wrapper for malware.

Malwares are not only executable codes but sometimes they act as downloader for malware e.g. PDF and PHP link which gains control of system and download more malicious software to execute on system. Some software gain control of system and do some legitimate work so we cannot classify them malicious. According to Virus Total 47.80% of malwares are executable files, so aim of this project is to analyze the executable binaries. There are many types of malwares, which can be classified into Virus, Trojan horse, Adware, Worm and Backdoor. Some of malwares cannot be classified into one category, because malwares have multiple characteristics which organize them in multiple categories and sometime we called them generalize malware.

Malwares are analyzed on basis of static as well as dynamic features. More than 2300 features are extracted from dynamic analysis and 92 features are extracted statically from binary file using PEFILE. Different dynamic features combinations are used for analysis. Four types of dynamic features are used for malware analyses which are Registry, DLLs, APIs and summary information. Machine learning is applied on these dynamic features combinations. The scope of this project is to present a malware detection advance using machine learning. In this project we will focus on windows executable files. Because of the abnormal growth of this malicious software's we need to use different automated approaches to find theses infected files. In this project we are going to study and realize a script used for data extraction from the PE - files to create a data set with infected and clean files, on which we are gonna train our machine learning algorithms: K-NN, XG Boost and Random Forest. The last chapter of this project the algorithms are tested with all the data set features. The precision of all algorithms is over 90%. Once applying a feature selection algorithm over the data set, the precision has been improved for all the learning algorithms. Therefore, malware protection of computer systems is one of the most important cyber security tasks for single users and businesses, since even a single attack can result in compromised data and sufficient losses. Substantial losses and recurrent attacks dictate the need for accurate and timely detection methods. Present static and dynamic methods do not provide efficient detection, especially when dealing with zero-day attacks. For this reason, machine learning-based techniques can be used. This paper discusses the main points and concerns of machine learning-based malware detection, as well as looks for the best feature representation and classification methods. The goal of this project is to develop the proof of concept for the machine learning based malware classification based on Cuckoo Sandbox. This sandbox will be utilized for the extraction of the behavior of the malware samples, which will be used as an input to

the machine learning algorithms. The goal is to determine the best feature representation method and how the features should be extracted, the most accurate algorithm that can distinguish the malware families with the lowest error rate. The accuracy will be measured both for the case of detection of whether the file is malicious and for the case of classification of the file to the malware family. The accuracy of the obtained results will also be assessed in relation to current scoring implemented in Cuckoo Sandbox, and the decision of which method performs better will be made. The study conducted will allow building an additional detection module to Cuckoo Sandbox. However, the implementation of this module is beyond the scope of this project and will not be discussed in this project.

2. COMPARATIVE CASE STUDY TABLE

S.no	Title	Methodology	Advantages	Disadvantages
1	A Hybrid Static Tool to Increase the Usability and Scalability of Dynamic Detection of Malware	Static analysis, dynamic analysis, and hybrids.	Malware detection is a fundamental tool necessary to prevent attacks on information and security.	Malware detection is a paramount priority in today's world in order to prevent malware attacks.
2	A Static and Dynamic Visual Debugger for Malware Analysis	Static and dynamic debugger a technique incorporates with graph visualization provides a comprehensive reverse engineering environment to the security expert in malware tracing process.	The research involves with the reverse engineering of binary executable by transforming a stream of bytes that constitutes the program into a corresponding sequence of machine instructions.	In order to provide immediate security solutions and reduce the amount of time on understanding malicious portion consisted in viruses, Trojans and other general security flow, a comprehensive design of visual debugger is introduced in this paper.
3	Malware behavior analysis using Binary code Tracking	BFS algorithms, KNN algorithms	We proposed a method to track execution flow on the binary code and detect malicious behavior.	The rapidly increasing malware goes beyond personal security threats and has a negative effect on criminal society.

4	Integrating Static and Dynamic Malware Analysis Using Machine Learning	Malware Analysis and Classification Systems use static and dynamic techniques	We propose the unification of static and dynamic analysis, as a method of collecting data from malware that decreases the chance of success for such evasion techniques.	Both techniques have weaknesses that allow the use of analysis evasion techniques, hampering the identification of malwares.
5	Forensic Malware Identification Using Naive Bayes Method	The process of identifying malware and benign files using the Naive Bayes machine learning method.	This research proposes an automatic malware detection system.	The malicious actions might be data theft, system failure, or denial of service.
6	Advance Malware Analysis Using Static and Dynamic Methodology	To focus on malware analysis using the static and the dynamic method which will help us to access damage,	The indicators of compromise and to determine the sophistication level of an intruder and to catch the creator of the malware.	We are witnessing the increasing potential of a cyber-attack on the critical infrastructure.
7	Combining Static Permissions and Dynamic Packet Analysis to Improve Android Malware Detection	KNN algorithms, Naive algorithms	Android application classification system that combines static permissions and dynamic packet analysis.	The main target of malware developers, so detecting and preventing Android malware has become an important issue of information security.

Table no.1.1 Comparative Case Study

3. EXISTING SYSTEM

- Malware detection by using window API sequence and machine learning.
- Detecting unknown malicious code by applying classification techniques on oppose patterns.
- Detecting scare ware by mining variable length instructions sequence.

- Accurate adware detection using opposes sequence extraction.
- Detection of spyware by mining executable files.
- Detection by using neural networks on the malware.

4. PROPOSED SYSTEM

- Machine learning can easily identify the malware in the data and datasets.
- Static and dynamic analysis of malware using machine learning.
- Train a model that takes static and dynamic analysis data, extracts features and classifies the input as Malware.
- **Data collection:** Collect Static and Dynamic Analysis Data for Malware samples provided.
- **Feature Extraction:** Remove features from the collected dataset using a script.
- **Feature Selection:** Select only important features so that prediction time will be reduced.
- **Classification:** Use machine learning classifiers to train the classifiers using extracted features.
- Different types of machine learning algorithms are applied such as :
    1. KNN
    2. SVM
    3. Naive Bayes
    4. Random forest
    5. Decision Tree

5. PROBLEM IDENTIFIED

Detecting unknown malicious code by applying classifications techniques on oppose pattern:

Evaluated number of experiments and found that setting of 2 grams, TF, using 300 features selected by DF measured outperform the perform lacks ML specific techniques.

Detecting scare ware by mining variable length instructions sequence:

This paper present the static analysis method based on data mining which extends the general heuristic detection techniques using a variable length instructions sequence mining approach for purpose of scare ware detection but metrics specific and unsupervised techniques in included can be broken.

6. MACHINE LEARNING AND ALGORITHMS

This chapter gives a theoretical background on machine learning methods, needed for understanding the practical implementation. First, the overview of the machine learning field is discussed, followed by the description of methods relevant to this study. These methods include K-Nearest Neighbors, Decision Trees, Random Forests, Support Vector Machines and Naive Bayes. The rapid development of data mining techniques and methods resulted in Machine Learning forming a separate field of Computer Science. It can be viewed as a subclass of the Artificial Intelligence field, where the main idea is the ability of a system (computer program, algorithm, etc.) to learn from its own actions. The basic idea of any machine learning task is to train the model, based on some algorithm, to perform a certain task:

classification, clusterization, regression, etc. Training is done based on the input dataset, and the model that is built is subsequently used to make predictions. The output of such model depends on the initial task and the implementation. Possible applications are: given data about house attributes, such as room number, size, and price, predict the price of the previously unknown house; based on two datasets with healthy medical images and the ones with tumor, classify a pool of new images; cluster pictures of animals to several clusters from an unsorted pool.

To develop a deeper understanding, it is worth going through the general workflow of the machine learning process, which is shown in Figure 1.

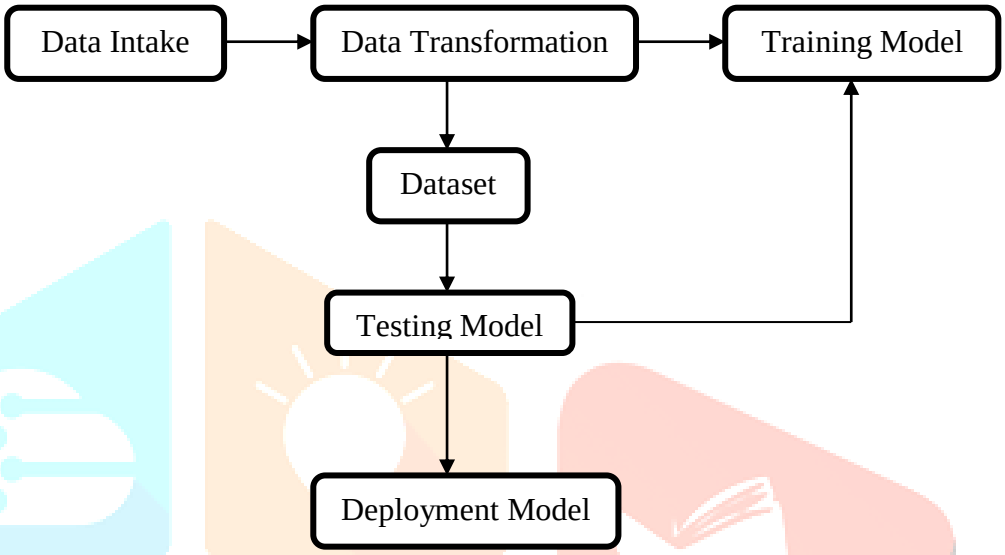

Figure 1.1. General Workflow Process

Data Intake: At first, the dataset is loaded from the file and is saved in memory.

Data Transformation: At this point, the data that was loaded at step 1 is transformed, cleared, and normalized to be suitable for the algorithm. Data is converted so that it lies in the same range, has the same format, etc. At this point feature extraction and selection, which are discussed further, are performed as well. In addition to that, the data is separated into sets – 'training set' and 'test set'. Data from the training set is used to build the model, which is later evaluated using the test set.

Training Model: At this stage, a model is built using the selected algorithm.

Testing Model: The model that was built or trained during step 3 is tested using the test data set, and the produced result is used for building a new model, that would consider previous models, i.e. "learn" from them.

Deployment Model: At this stage, the best model is selected (either after the defined number of iteration or as soon as the needed result is achieved).

Supervised Learning: Learning is based on labeled data. In this case, we have an initial dataset, where data samples are mapped to the correct outcome. Examples of supervised learning are regression and classification problems:

1. **Regression:** Predict the value based on previous observations, i.e. values of the samples from the training set. Usually, we can say that if the output is a real number/is continuous, then it is a regression problem.

2. **Classification:** Based on the set of labeled data, where each label defines a class, that the sample belongs to, we want to predict the class for the previously unknown sample. The set of possible outputs is finite and usually small. Generally, we can say that if the output is a discrete/categorical variable, then it is a classification problem.

Unsupervised Learning: There is no initial labeling of data. Here the goal is to find some pattern in the set of unsorted data, instead of predicting some value. A common sub class of Unsupervised Learning is Clustering.

3. **Clustering:** Find the hidden patterns in the unlabeled data and separate it into clusters according to similarity. An example can be the discovery of different customer groups inside the customer base of the online shop.

CONCLUSION:

A Malware is critical threat to user computer system in terms of stealing confidential information or disabling security. This project present some of the existing machine learning algorithms directly applied on the data or datasets of malware. It explains the how the algorithms will play a role in detecting malware wit high accuracy and predictions. We are also using data science and data mining techniques to overcome the drawbacks of existing system. Malwares have intelligent and tricky nature and they can detect dynamic malware analysis very quickly, therefore we need a dynamic analysis controlled environment, which is not detectable. In current era, malwares are packed in nature and it is often able to analyze statically. We need a system, which is initially dynamic and when malware is unpacked then apply the static features extraction on it. Machine learning has some limitations like massive store of data, labelling of trained data, bias in train data and algorithm does not collaborate with them. These limitations will be overridden by DNN. Due to obfuscated and packed nature of malware, the static analysis is not so effective. When malware executes in dynamic environment, it changes its behaviours, so static features can be extracted easily and correctly. The static features extraction in dynamic environment will be very efficient to detect malware.

REFERENCES

[1] Kolbitsch, C., Comparetti, P. M., Kruegel, C., Kirda, E., Zhou, X. Y., & Wang, X. (2009, August). Effective and Efficient Malware Detection at the End Host. In USENIX security symposium (Vol. 4, No. 1, pp. 351-366).

[2] David, B., Filiol, E., & Gallienne, K. (2017). Structural analysis of binary executable headers for malware detection optimization. Journal of Computer Virology and Hacking Techniques, 13(2), 87-93.

[3] Wang, T. Y., Wu, C. H., & Hsieh, C. C. (2009, August). Detecting unknown malicious executables using portable executable headers. In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on (pp. 278-284). IEEE.

[4] Aman, W. (2014). A framework for analysis and comparison of dynamic malware analysis tools. arXiv preprint arXiv:1410.2131.

[5] Chumachenko, K. (2017). Machine Learning Methods for Malware Detection and Classification.

[6] Santos, I., Devesa, J., Brezo, F., Nieves, J., & Bringas, P. G. (2013). Opem: A static-dynamic approach for machine-learning-based malware detection. In International Joint Conference CISIS'12- ICEUTE 12-SOCO 12 Special Sessions (pp. 271-280). Springer, Berlin, Heidelberg.

[7] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer, Cham.

[8] Alazab, M., Venkatraman, S., Watters, P., & Alazab, M. (2011, December). Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121(pp. 171- 182).

[9] Chowdhury, M., Rahman, A., & Islam, R. (2017, June). Malware analysis and detection using data mining and machine learning classification. In International Conference on Applications and Techniques in Cyber Security and Intelligence (pp. 266-274). Edizioni della Normale, Cham.

[10] Jain, A., & Singh, A. K. (2017, August). Integrated Malware analysis using machine learning. In 2017 2nd International Conference on Telecommunication and Networks (TEL-NET)(pp. 1-8). IEEE.