"Malware Detection in Portable Executable Files Using Machine Learning"

A minor project report,
submitted in partial fulfillment of the requirements for Mini Project

by

Dasari Jayanth (2019BCS-016)

Under the Supervison of

Dr. Saumya Bhadauria

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015
2021

Contents

List of Figures

ABBREVIATIONS

ML	Machine Learning
PE	Portable Executable
DOS	Disk Operating System
API	Application Programming Interface
MD5	Message-Digest Algorithm 5
KNN	K Nearest Neighbor
SVM	Support Vector Machine
DLL	Dynamic-Link Library
CNG	Common N-gram
TF	Term frequency
FCG	Function Call Graph
CFG	Control Flow Graph
IDE	Integrated Development Environment

1 Abstract

The number of Malicious files is increasing every day because of existing open-source malware and obfuscation techniques. So, it became essential to use machine learning methods in malware detection. This project aims to classify the unknown files as benign or malware with different machine learning methods by extracting the significant features from PE files using hybrid static malware analysis and training the models with the acquired feature set. It also identifies the best-suited machine learning classifier model for malware detection for these static feature set characteristics.

2 Introduction

Malware is malicious software, script, or program. This intentionally written program has various features, such as stealing, encrypting, or deleting sensitive data, modifying or hijacking essential computer functions, and monitoring computer activity, showing user permission. A program/software is classified as malware if one of the signatures is identified in it.

The number of malware attacks increased enormously due to the growth of technology, elevated sophistication of the malicious code, and the number of available vulnerable machines because users are unaware of security. Damage caused by these malware attacks is also growing. Malware is one of the topmost obstructions to the expansion and growth of digital acceptance among users. So, it has become crucial to develop countermeasures to eradicate malware. It is crucial for businesses and consumers alike.

3 Problem Statement

In the past decade, the malware industry has overgrown. Detecting malware attacks using traditional methods became difficult due to polymorphism, metamorphism, and soon. So, the demand for technologies to detect malware has grown abruptly. Effective and automated malware detection has become an essential requirement to guarantee system safety and user protection. The major part of protecting a device from a malware attack is identifying whether a given piece of file/software is malware.

4 Background and Motivation

4.1 Background

A portable executable (PE) file is the most used file format due to the wide use of the Windows operating system. A PE file is a data structure that contains the information necessary for the Windows OS loader to manage the wrapped executable code. There are no mandatory constraints in many fields of PE files and contain many redundant fields and spaces, creating opportunities for malware propagation and malware attacks. A PE file contains the PE file header, section table, and section data.

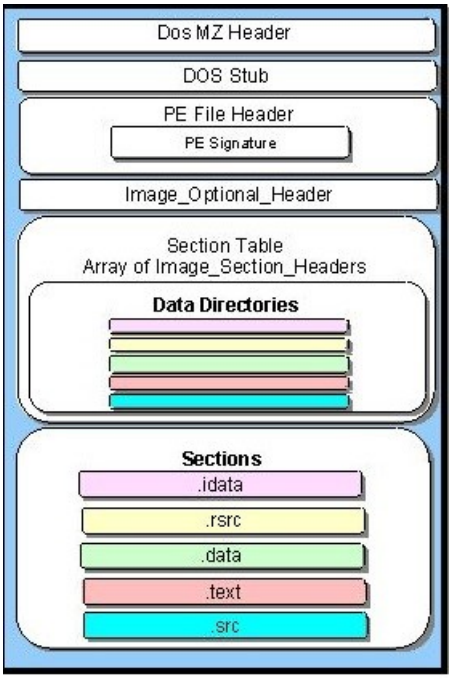

Figure 1: PE file

A malware detection system is a system used to determine whether a program has malicious intent or not. It includes two tasks - analysis and detection. Static analysis and dynamic analysis are the main malware analysis methods used to create a feature set. Malware detection is mainly categorized into signature-based detection, behavior-based detection, which are used to detect the malware and classify files as malware or benign.

In static malware analysis(also called code analysis), the program is analyzed without executing it, and reverse engineering is performed using different tools like debugger, disassemble, decompile, and so on. N-grams byte sequences, Opcode sequences, API calls, PE header, control flow graphs are the different characteristics of static malware analysis.

The behavior of a file is analyzed during its execution in dynamic malware analysis(also called behavioral analysis). In dynamic analysis, the file is detected by monitoring its system interaction, behavior, and effect on the machine while executing it in the real environment(virtual machine environment or sandboxes).

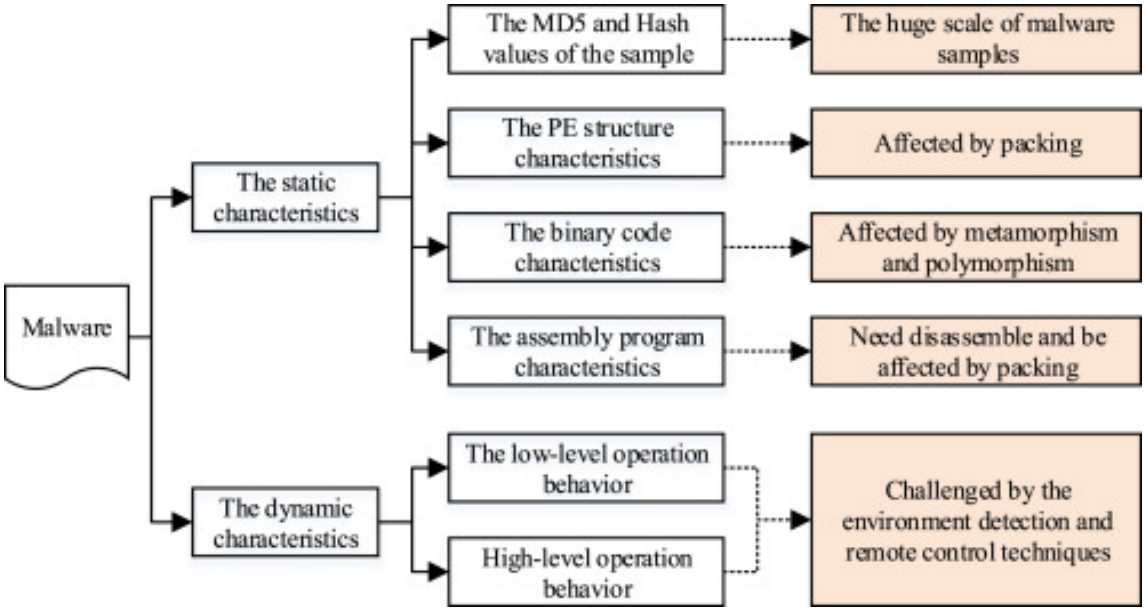

Figure 2: Malware Analysis types and it's challenges

Signature-based detection maintains the data of signature and detects malware by comparing patterns against that data. A signature is a unique feature set that distinguishes executables like a fingerprint. Behavior-based detection(heuristic-based or anomaly-based) analyzes behavior and file characteristics to determine the malware.

Machine Learning is a category of algorithms that allow software applications to predict much better results without being specifically programmed. Different ML classifiers like logistic regression, random forest, K-NN, decision trees, SVM are trained to classify the files into malware or benign. Their performance is measured using different performance metrics like f1 score, confusion matrix.

4.2 Motivation

Attackers are targeting the Windows operating system as it is most widely used. The portable executable(PE) format is the file format for executables, object code, DLLs, FON Font files, and others used in 32-bit and 64-bit versions of the Windows operating system. So, PE files are used in this project.

Dynamic analysis may be accurate in analyzing the file, but it is expensive and very slow as it requires execution and time to prepare the environment for malware analysis. So, it is vulnerable, whereas static analysis is fast and safe as it executes the code before execution, so we can prevent the attack. In this project, static malware analysis will be used.

Machine learning is a data analytics tool used to perform specific tasks without explicit instructions effectively. It is difficult to detect malicious attacks with traditional methods due to its enormous growth and the number of attacks per day. Many malware programs are derived from another with just slight changes. So, it will be easy and efficient to use ML methods due to the vast data to verify. So, in this project, Machine learning is used in detecting malware.

There are many studies regarding both static and dynamic malware analysis and classification of malicious code. Most of the works have used only one type of feature construction methods like either byte n-gram or opcode sequence or PE headers. Due to the growth of malware in recent days, it became difficult to detect malware using one characteristic simply. By building a feature set combining more than one characteristic and extracting significant features, we can create a better feature set that gives greater accuracy. So, this project aims to build a feature set with more than one characteristic and train different classifiers using this feature set to improve accuracy in detecting malicious files.

5 Related Work

Walenstein [1] extracted PE header and body information base on static analysis. They evaluated the effect of information to distinguish benign and malicious files in their work. They extracted all PE header information, DLL names, and API function names called within those DLLs contained in a PE file.

Ajit Kumar et al. [4] proposed an ML-based model. They concentrated on creating an integrated feature set by combining raw features and derived features of PE files so that the model will have low computation overhead and gives better accuracy. For feature importance, they used Extra Trees Classifier and trained various ML classification algorithms like decision tree, K-NN, random forest, logistic regression and compared their results.

Tina Rezaei et al.(2020) [9] proposed a model which can be used in real-time malware detection systems. With only nine features with a significant difference in PE header and PE file structure, they created a model to detect malware with high speed and good accuracy.

In 2009, Shabtai et al.[2] provided a taxonomy for malware detection using ML algorithms, some feature types, and feature selection techniques used in the literature. They focussed on the feature selection techniques like document frequency, gain ratio, and classification algorithms like Artificial Neural Networks, Bayesian Networks, Naïve Bayes, K-NN. They also experimented on how ensemble algorithms can be used to combine a set of classifiers.

In the model proposed by Fuyong et al.(2017)[3], they selected k n-grams as features based on information gain. To classify malware and benign files, they took the average of the feature vector's attributes and classified them based on similarity in feature vectors of unknown samples and their average vectors.

In 2012 Shabtai et al.[5] proposed a framework for malware detection using features of opcode n-gram where n ranges from 1 to 6. They experimented to identify the best term representation(Term Frequency (TF) or Term Frequency-Inverse Document Frequency), to determine the n-gram size, and find the optimal K top n-grams and feature selection method. They evaluated the performance of various machine learning algorithms.

In 2010, Sami et al.[6] proposed a three-step framework to classify PE files based on API call usage. They extracted the list of imported API calls by analyzing PE files. They reduced the feature vector using the Clospan algorithm, and the Random Forest model is trained with the feature set.

Hassen and Chan (2017)[8] proposed a method to extract a vector representation of the linear time function call graph based on function clustering. They used the minhash hashing technique and improved the speed of measuring similarity between functions. Using function clustering from minhash signatures, they converted graph representation into a vector representation to compute similarity. Due to this, non-graph features can be used combinedly with graph features.

In 2011, Eskandari and Hashemi [7] presented an approach to detect metamorphic malware through their Control Flow Graphs. They disassembled the PE files and applied a preprocessing algorithm to assembly files and API calls, and the vector representation is created from the resulted sparse graph. Their model achieved 97% accuracy using the Random Forest classifier.

6 Objective

In this project, We create a malware detection model using hybrid malware static analysis on PE files. We train a classifier model to classify malicious and benign files using hybrid static analysis of the extracted Portable executable files(PE files). Instead of using a particular characteristic in creating a feature set, we combine different characteristics like n-gram sequence, PE header, opcode sequence to create a feature set for achieving better accuracy and efficiency. We train different classifier models like K-NN, random forest, logistic regression, decision trees, support vector machines(SVM) on this feature set and compare them to find which model gives better accuracy.

7 Salient Features of this Project

- We create a feature set with significant features from hybrid static malware analysis of PE files.
- We will detect malicious files more accurately and fast from this model before executing the file, as the static analysis does not require execution. (i.e., classifying files as malware and benign).
- We will find the best suitable machine learning classifier model for malware detection for the characteristics used in the feature set.

8 Workflow

- Data Collection. Download PE files from websites like virusshare and kaggle.
- Data Analysis and Feature Extraction using hybrid static analysis (like n-grams, PE header, soon), and all the features will be merged.
- Normalization, Feature reduction as there will be so many features.
- Creating a feature set with significant features.
- Dividing feature set into two sets: train feature set and test feature set.
- Training different classifier models and testing them.
- Model validation.

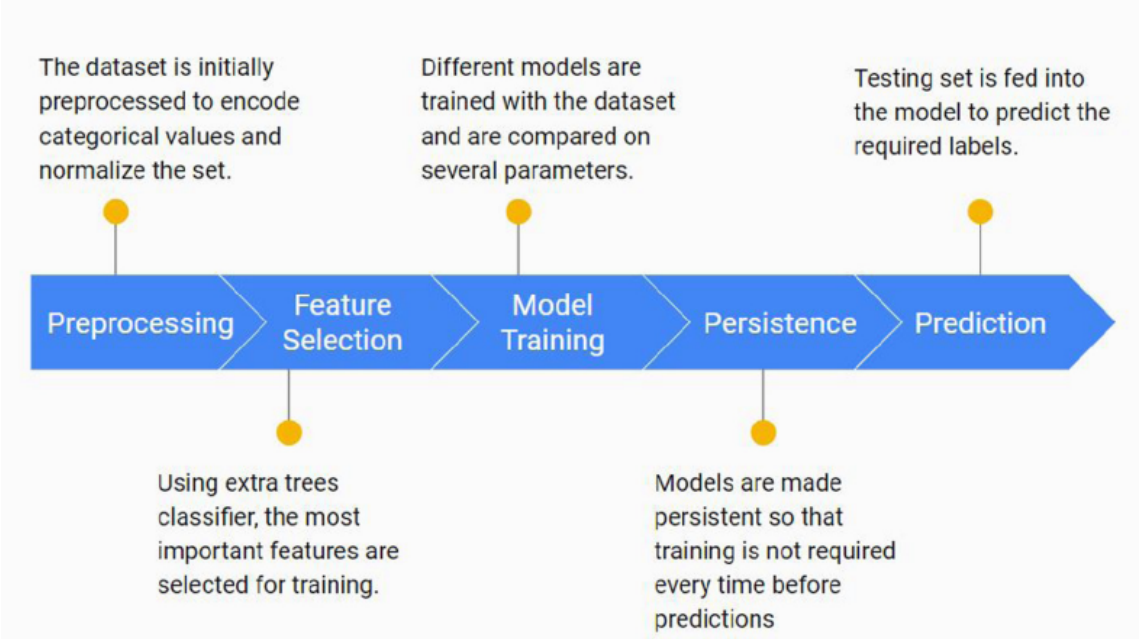

Figure 3: Overview of Workflow

9 Gantt Chart

	18 May,2021	7 June,2021	15 June,2021	25 June,2021	5 July,2021	15 July,2021	25 July,2021	30 July,2021
Project Finalization								
Literature Survey on Malware Detection								
Data Collection and Analysis								
Feature extraction and Feature reduction								
Normalization								
Merging features and Creating Integrated feature set								
Training and Testing Classifier Models								
Model Validation								

Figure 4: Project Timeline

10 References

1. Andrew Walenstein, Daniel J Hefner, Jeffery Wichers. 'Header information in malware families and impact on automated classifiers.' 5th Int Conf on Malicious and Unwanted Software Malware 2010, Nancy, France, 2010:15–22.

2. A. Shabtai, R. Moskovitch, Y. Elovici, C. Glezer. 'Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey.' Inf. Sec. Tech. Rep., 14 (1) (2009), pp. 16-29.

3. Z. Fuyong, Z. Tiezhu. "Malware detection and classification based on n-grams attribute similarity." 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1 (July 2017), pp. 793-796.

4. Ajit Kumar, K.S. Kuppusamy, G. Aghila. 'A learning model to detect maliciousness of portable executable using integrated feature set.' Journal of King Saud University - Computer and Information Sciences Volume 31, Issue 2, April 2019, Pages 252-265.

5. A. Shabtai, R. Moskovitch, C. Feher, S. Dolev, Y. Elovici, Feb. 'Detecting unknown malicious code by applying classification techniques on opcode patterns.' Security Informatics, 1 (1) (2012)

6. A. Sami, B. Yadegari, H. Rahimi, N. Peiravian, S. Hashemi, A. Hamze. 'Malware detection based on mining API calls.' Proceedings of the 2010 ACM Symposium on Applied Computing. SAC 10, ACM, New York, NY, USA (2010), pp. 1020-1025

7. M. Eskandari, S. Hashemi. 'Metamorphic malware detection using control flow graph mining.' 06 International Journal of Computer Science and Network Security, 11 (12) (2011)

8. M. Hassen, P.K. Chan. "Scalable function call graph-based malware classification." CODASPY 17 Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, ACM, New York, NY, USA (2017), pp. 239-248.

9. Tina Rezaei, Ali Hamze. "An Efficient Approach For Malware Detection Using PE Header Specifications." 2020 6th International Conference on Web Research (ICWR).

10. https://link.springer.com/chapter/10.1007/978-3-030-62223-7_10

11. https://www.ijert.org/research/malware-and-malware-detection-techniques-a-survey-IJERTV2IS120163.pdf