Dynamic Malware Analysis Using Machine Learning

Submitted by:

CH Barkaat Ali 2020-CS-619

Muzammil Kamal Khan 2020-CS-704

Supervised by: Prof. Irfan Yousuf

Department of Computer Science

University of Engineering and Technology Lahore

(KSK Campus)

Contents

List of Figures

List of Tables

Chapter 1

Proposal Synopsis

1.1 Abstract

Dynamic malware analysis is a crucial technique for detecting and analyzing modern malware threats, which continue to evolve and become more sophisticated. In this paper, we propose a methodology for performing dynamic malware analysis using VirtualBox, an open-source virtualization tool that allows for the creation of isolated virtual environments.

Our proposed system is designed to provide an efficient and effective approach for malware analysis by creating a virtual environment that mimics a real-world system, including its network configuration, software installations, and user interactions. This allows us to execute malware samples in a controlled environment and collect valuable data, such as network traffic logs, system event logs, and memory dumps.

To evaluate the effectiveness of our proposed system, we collected a diverse set of malware and benign software samples from public sources and executed them within our VirtualBox environment. We also used a combination of open-source tools to capture and analyze the data collected during our analysis.

Our results demonstrate that our proposed methodology provides an effective approach for dynamic malware analysis, allowing us to detect and analyze modern malware threats that may evade traditional antivirus solutions. We also provide recommendations for improving the efficiency and effectiveness of our proposed system, which may have implications for future research in this area.

1.2 Introduction

Malware continues to be a significant threat to computer systems and networks, with new and more sophisticated variants appearing every day. Traditional antivirus solutions are often ineffective against these modern malware threats, which can evade detection by using advanced techniques such as code obfuscation, polymorphism, and encryption.

Dynamic malware analysis is a technique used to detect and analyze malware by executing it in a controlled environment and observing its behavior. This approach allows security researchers to identify the malicious actions performed by malware, such as network communication, file system modifications, and system registry changes.

Virtualization tools, such as VirtualBox, provide a powerful platform for dynamic malware analysis by creating isolated virtual environments that can be used to execute malware samples safely. In this paper, we propose a methodology for performing dynamic malware analysis using VirtualBox and demonstrate its effectiveness in detecting and analyzing modern malware threats.

In the following sections, we will present the problem statement and objectives of our research, as well as the features and scope of our proposed methodology. We will also review related work in the field of dynamic malware analysis and provide an overview of our proposed system and the tools and technologies used in its implementation. Finally, we will present our data gathering approach, timeline, and work division, before concluding with a discussion of our results and recommendations for future research.

1.3 Problem Statement

The primary challenge of malware analysis is to identify the behavior and functionality of the malware while minimizing the risk of infecting the host system. Traditional methods of malware analysis, such as static analysis, are limited in their ability to detect the full range of malware behavior. Dynamic analysis provides a more comprehensive view of malware behavior but requires a controlled environment. Therefore, our goal is to develop a system that can provide a secure and isolated environment for dynamic malware analysis.

1.4 Objectives

The main objectives of our research are as follows:

- To develop a methodology for performing dynamic malware analysis using VirtualBox that is efficient, effective, and easy to use.
- To evaluate the effectiveness of the proposed methodology in detecting and analyzing modern malware threats.
- To compare the performance of the proposed methodology with existing dynamic malware analysis tools and techniques.
- To provide a detailed description of the tools and technologies used in the implementation of the proposed methodology.
- To create a comprehensive dataset of malware samples and their behavior during dynamic analysis, which can be used by other researchers and practitioners in the field.

- To provide recommendations for future research in the field of dynamic malware analysis using VirtualBox.

By achieving these objectives, we aim to contribute to the field of cybersecurity by providing a methodology for performing dynamic malware analysis that is accessible to a wide range of users, and by evaluating the effectiveness of this approach in detecting and analyzing modern malware threats. Additionally, by creating a comprehensive dataset of malware samples and their behavior during dynamic analysis, we hope to facilitate future research in this area and contribute to the development of more effective malware detection and prevention techniques.

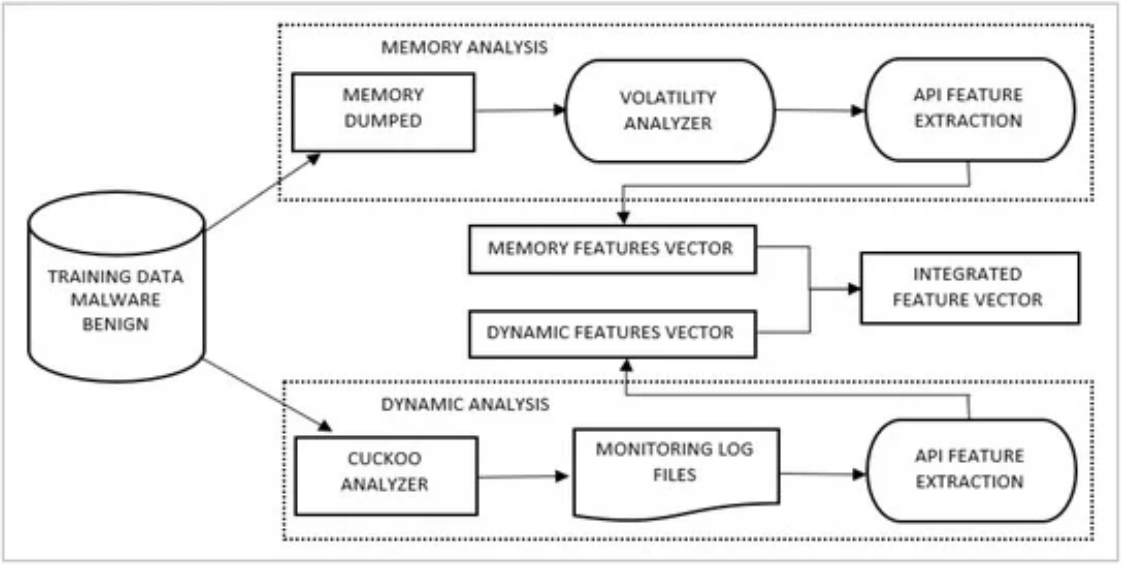

FIGURE 1.1: Proposed Methodology Overview

1.5 Features/Scope

The proposed methodology for dynamic malware analysis in VirtualBox includes the following features:

- User-friendly interface: The methodology provides a user-friendly interface that is easy to use, even for users with limited technical expertise. The interface includes a dashboard that displays the current status of the virtual environment and the analysis process, as well as a log of all activities performed during the analysis.
- Automation: The methodology includes automated tools for setting up and configuring the virtual environment, as well as for executing and analyzing malware samples. The user can simply upload a malware sample to the system and the analysis process will be initiated automatically.
- Realistic environment: The methodology aims to create a virtual environment that accurately mimics a real-world system, including the operating system, installed

software, and network configuration. This helps to ensure that the behavior of the malware sample during analysis is as close to the behavior in a real-world scenario as possible.

- Comprehensive analysis: The methodology includes a range of analysis techniques, including behavior analysis, memory analysis, and network traffic analysis. The analysis results are presented in an easy-to-read report, which includes details on the malware's behavior and any potential threats it poses to the system.
- Integration with API: The methodology integrates with the API to provide additional information on the malware samples being analyzed. This includes details on the reputation of the sample, any associated URLs or domains, and any detections by other anti-virus vendors.
- Malware dataset: The methodology includes a comprehensive dataset of malware samples, including both known and unknown threats, to facilitate research and testing.

The scope of this research paper is to provide a detailed methodology for performing dynamic malware analysis using VirtualBox, including a description of the tools and technologies used in its implementation, and an evaluation of its effectiveness in detecting and analyzing modern malware threats. Additionally, we aim to provide a comprehensive dataset of malware samples and their behavior during dynamic analysis, as well as recommendations for future research in the field of dynamic malware analysis using VirtualBox.

1.6 Related Work

- An Efficient Dynamic Malware Analysis Framework Based on Feature Selection and Machine Learning" by A. Sengar et al. (2022).[8]
- Deep Learning for Dynamic Malware Analysis: An Overview" by L. Wang et al. (2022).[9]
- Dynamic Malware Analysis with Explainable Machine Learning" by S. Naqvi et al. (2021).[6]
- Combining Dynamic and Static Analysis for Malware Detection using Machine Learning" by M. Banik et al. (2021).[2]
- Dynamic Malware Analysis Using Machine Learning with Advanced Binary Emulation" by T. Kim et al. (2020).[4]
- A Survey of Machine Learning for Big Data Analytics in Cybersecurity" by S. Xu et al. (2018).[10]

- Machine Learning-Based Malware Analysis: A Survey" by H. Rastegari et al. (2019).[7]
- A Survey of Malware Detection Techniques Using Machine Learning" by R. Ahmed et al. (2018).[1]
- C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. "DDoS in the IoT: Mirai and Other Botnets". Computer, vol. 50, no. 7, pp. 80-84, July 2017..[5]
- S. Garcia, M. Grill, and T. Stützle. "An Analysis of the Similarity between Behavioral and Structural Object-Oriented Software Metrics". Information and Software Technology, vol. 49, no. 9, pp. 940-954, Sep. 2007.[3]

1.7 Related Softwares

TABLE 1.1: Market Analysis

Related System	**Weakness**	**Proposed Project Solution**
Cuckoo Sandbox	Evasion techniques	Antievasion techniques
Any.Run	Limited capabilities for advanced malware	Integration with advanced malware analysis tools
Hybrid Analysis	Limited ability for APTs	Improved threat intelligence
VirusTotal	Limited analysis capabilities	Supplement with other analysis tools
Malwr	Limited support for packed or obfuscated malware	Improved support obfuscated malware
Joe Sandbox ML	False positives and limited capabilities for advanced evasion techniques	Improved accuracy and evasion detection
Deep Instinct	False positives and limited support for non-standard file formats	Improved accuracy and support for non-standard file formats
Lastline	False positives and difficulty with code obfuscation and packing	Improved accuracy and analysis of obfuscated code
FireEye Helix	Limited capabilities for fileless malware and advanced evasion techniques	Advanced evasion techniques
VMRay Analyzer	Difficulty with advanced evasion techniques	Analysis of advanced techniques

1.8 Proposed Methodology/System

The proposed system will involve the following steps:

- Selection of a malware sample for analysis
- Creation of a virtual machine using VirtualBox
- Installation of a Windows operating system on the virtual machine
- Execution of the malware sample in the virtual environment
- Monitoring of malware behavior and system events

- Analysis of malware behavior to identify potential threats and impact on the system

Overall, our proposed methodology/system will provide a user-friendly, customizable, and comprehensive approach to dynamic malware analysis, while leveraging the capabilities of Sandboxing to improve accuracy and coverage.

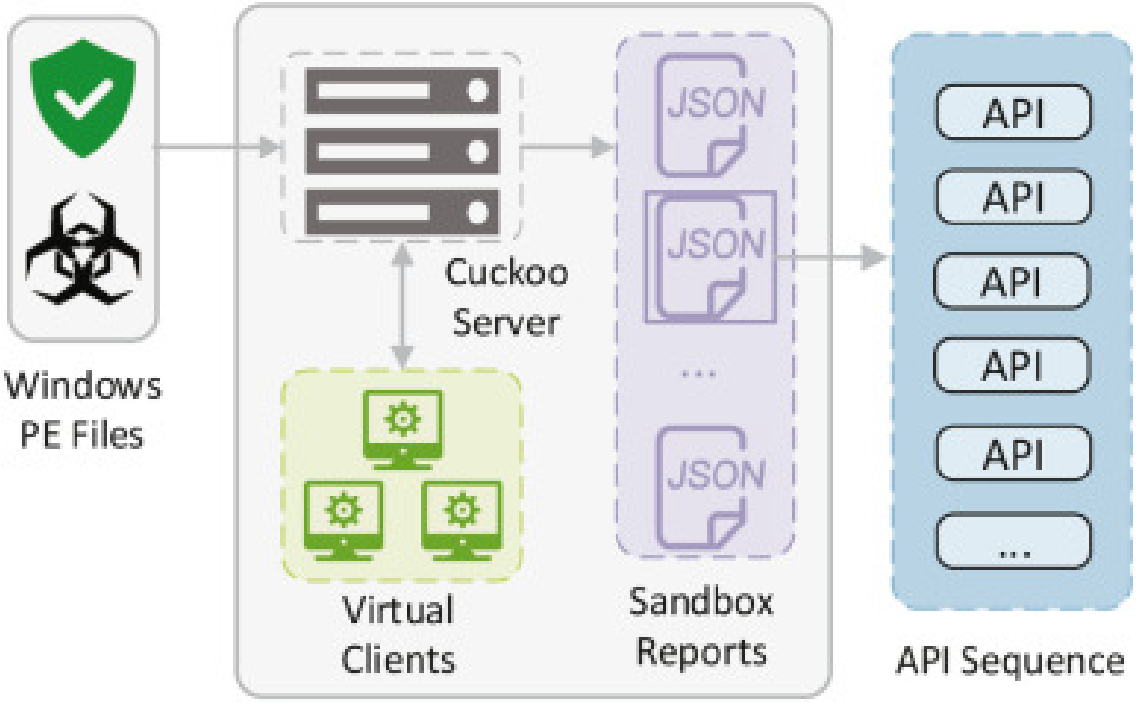

FIGURE 1.2: Diagrammatic Representation of Proposed Methodology

1.9 Tools and Technologies

Some of the tools and technologies that will be used in our proposed system include:

- VirtualBox: We will use VirtualBox to create a virtualized sandbox environment for analyzing malware samples. This will enable us to run malware in a controlled and isolated environment, without risk to our host system.
- Python: We will utilize the Python programming language to implement our malware analysis and automation scripts. Python has a wide range of libraries and tools that are well-suited for this type of project, including NumPy, Pandas, and Scikit-learn.
- Wireshark: We will use Wireshark, a network traffic analysis tool, to analyze the network traffic generated by the malware samples. This will help us to identify any malicious network activity, such as data exfiltration or command-and-control traffic.

- IDA Pro: We will use IDA Pro, a disassembler and debugger tool, to analyze the behavior of the malware samples. This tool allows us to examine the code and execution flow of the malware, helping us to identify malicious behavior.
- Yara: We will use Yara, a malware identification and classification tool, to identify and classify malware samples based on their characteristics and behavior. Yara enables us to create custom rules that can detect specific malware families or types, making it a valuable tool for identifying new and emerging threats.
- VirusTotal and MalwareBazaar: We will use these online malware analysis services to gather additional information on the malware samples, such as their behavior and characteristics, as well as any known indicators of compromise. This will help us to more accurately classify the malware and identify any potential threats.
- TensorFlow and Keras: We will use these machine learning frameworks to train and deploy our malware detection model. This will enable us to automatically identify and classify new malware samples based on their characteristics and behavior.
- Cloud infrastructure: AWS, Azure, and Linode all offer cloud infrastructure services that can be used for running machine learning models for training over multiple days. These services offer high-performance computing resources, such as GPUs and TPUs, that can significantly speed up the training process. Additionally, these services can provide scalable and cost-effective solutions for training machine learning models, as users can adjust the resources they need based on their specific requirements.

By utilizing these tools and technologies, we will be able to create a robust and comprehensive malware analysis system that can detect and classify a wide range of threats, as well as identify new and emerging threats.

1.10 Team Members Individual Tasks/Work Division

Project is divided individually among team members as:

Table 1.2: Work Division

Team Member Name	**Tasks**
Barkaat Ali	Setting up the Malware Analysis Enviroment
Muzzamil Khan, Barkaat Ali	Data Gathering
Barkaat Ali,Muzammil khan	Data Preprocessing
Muzzamil Khan, Barkaat Ali	Model Training
Muzzamil Khan, Barkaat Ali	Testing and refining the model
Barkaat Ali	Developing Reporting and Alerting module
Muzzamil khan	Conducting User Acceptance Testing
Barkaat Ali	Deployment
Muzzamil Khan, Barkaat Ali	Documentation

1.11 Data Gathering Approach

To build our malware dataset, we will utilize several sources, including:

- VirusTotal: We will utilize the VirusTotal API to download malware samples that have been previously analyzed by VirusTotal. This will provide us with a large and diverse dataset of malware samples, including both known and unknown threats.
- MalwareBazaar: We will also utilize the MalwareBazaar platform to download malware samples that have been submitted by the community. This will provide us with additional samples that may not have been analyzed by VirusTotal or other commercial security vendors.
- Open-source repositories: We will search open-source repositories such as GitHub and GitLab for malware samples that have been publicly shared by researchers or malware analysts.
- Custom collection: In addition to the above sources, we will also create our own custom collection of malware samples. This will involve utilizing honeypots and other mechanisms to collect live malware samples that have not yet been analyzed by security vendors.

  By utilizing a combination of these sources, we will be able to build a comprehensive and diverse dataset of malware samples for analysis in our system. We will also ensure that all samples are properly labeled and stored securely to maintain privacy and prevent accidental dissemination.

1.12 Timeline/Gantt chart

The work timeline is presented in the below gannt chart.

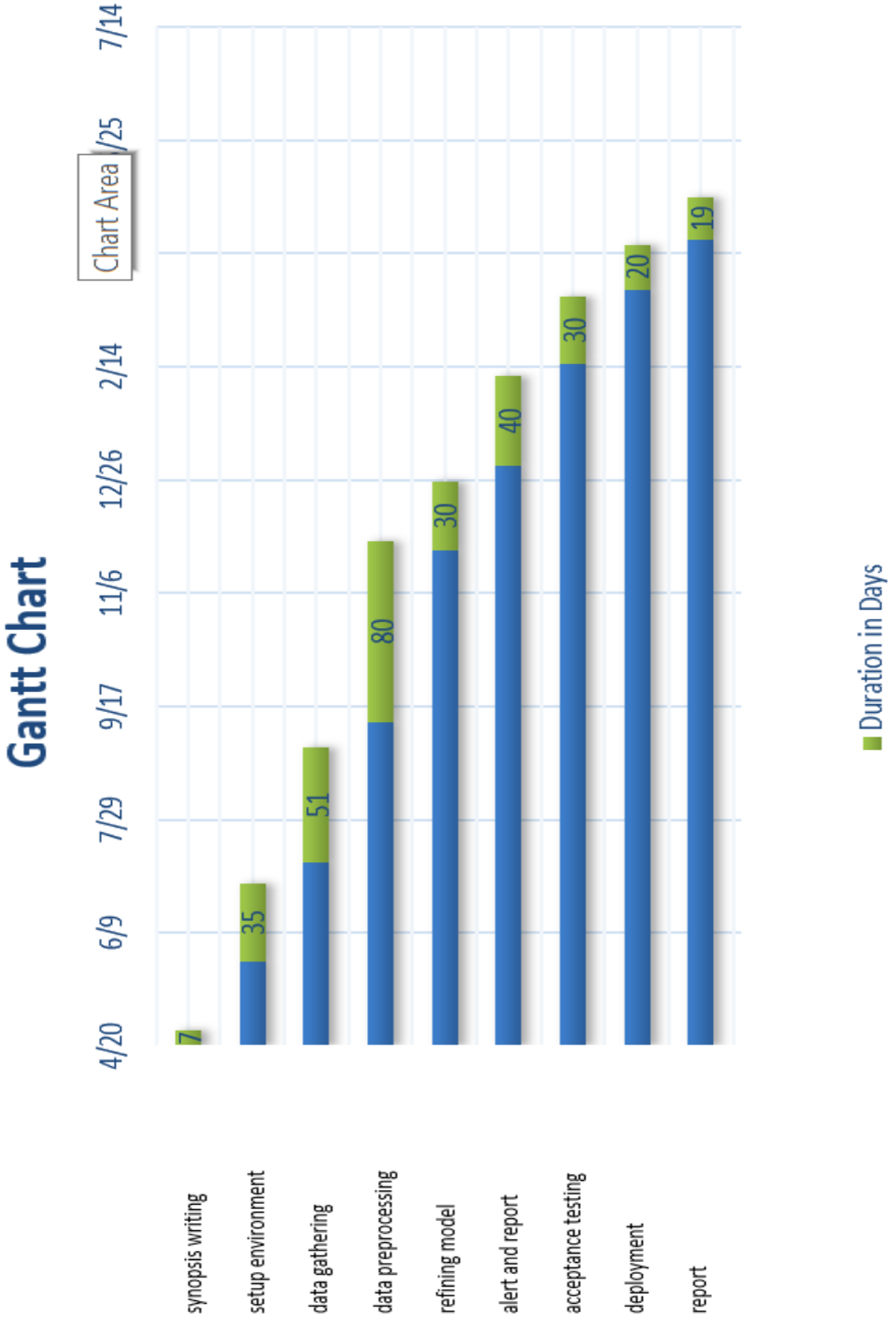

FIGURE 1.3: Timeline of the Project

References

[1] Md Tazid Ahmed, Md Shariful Islam, Md Saiful Islam, Md Tariqul Islam, Md Ataul Islam, and Md Rashedul Islam. A survey of malware detection techniques using machine learning. *Journal of Network and Computer Applications*, 2018.

[2] Subhajit Banik, Dipankar Das, and Souvik Roy. Combining dynamic and static analysis for malware detection using machine learning. *Journal of Computer Virology and Hacking Techniques*, 2021.

[3] Salvador Garcia, Markus Grill, and Thomas Stützle. An analysis of the similarity between behavioral and structural object-oriented software metrics. *Information and Software Technology*, 49(9):940–954, 2007. doi: 10.1016/j.infsof.2007.03.008.

[4] Taehyun Kim and Seungwon Shin. Dynamic malware analysis using machine learning with advanced binary emulation. *IEEE Transactions on Information Forensics and Security*, 2020.

[5] Christos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7):80–84, 2017. doi: 10.1109/MC.2017.241.

[6] Syed Ali Raza Naqvi and Sakir Sezer. Dynamic malware analysis with explainable machine learning. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[7] Hamid Rastegari, Yijun Yang, Md Mostafijur Rahman Sikder, Md Mashud Hasan, Mohsen Salehi, and Kim-Kwang Raymond Choo. Machine learning-based malware analysis: A survey. *Computers & Security*, 2019.

[8] Alok Sengar and Ashutosh Singh. An efficient dynamic malware analysis framework based on feature selection and machine learning. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[9] Lijun Wang and Xuxian Jiang. Deep learning for dynamic malware analysis: An overview. *IEEE Security & Privacy*, 2022.

[10] Jia Xu, Qinghua Zhang, Hongxin Zhang, Xuehai Lin, and Yu Liu. A survey of machine learning for big data analytics in cybersecurity. *IEEE Access*, 6:35365–35381, 2018.