# Project Report
# Enhancing Malware Analysis Using AI

Supervised by: Mahmoud Ellouh


Written by:

1st Mohammed Alqurashi

2nd Mohammed Almalki

3rd Husam Haddad

4th Abdulrahman Al Mutairi

5th Hussain Babongi

6th Abdullah Alghamdi


# Abstract:


Malware is a growing problem that threatens the safety of our digital world. This paper describes a system that uses AI to analyze and classify malware. The system combines two methods: dynamic analysis, where the malware runs in a safe environment to see how it behaves, and static analysis, where the malware's code and structure are examined without running it. Both methods use machine learning to help determine whether the malware is harmful or not. The system gives users a detailed report of the findings, helping them understand and deal with potential threats effectively. This approach aims to improve how we detect and manage malware.


Date:

August 31, 2024

## I. Introduction and Problem Statement

In the ever-evolving landscape of cybersecurity, malware detection remains a critical concern. Malicious software, or malware, poses a significant threat to both individuals and organizations, often resulting in data breaches, financial losses, and compromised systems. Traditional malware detection methods, while effective to a certain extent, struggle to keep up with the rapidly changing techniques employed by the attackers.

Static analysis, which examines the code structure without executing it, often fails to detect malware that employs advanced techniques such as encryption and code obfuscation. On the other hand, dynamic analysis, which involves executing the malware in a controlled environment, can be easily circumvented by malware that detects and avoids debugging environments. These limitations leave systems vulnerable to undetected threats, underscoring the need for more advanced detection mechanisms.

To address these challenges, this project presents an approach that integrates static and dynamic analysis with machine learning (ML) models and YARA rules. By combining these methodologies, the project aims to develop a malware detection system capable of accurately identifying and classifying even the most evasive forms of malware. This integrated approach not only enhances detection rates but also reduces false positives, thereby providing a more reliable defense against cyber threats.

### A. Motivation Scenario

Imagine a large financial institution responsible for safeguarding sensitive customer data. This institution has implemented traditional malware detection systems that rely heavily on signature-based methods and basic static analysis. While these systems have been effective in the past, they are increasingly failing to detect newer, more sophisticated malware variants. Attackers are now deploying advanced techniques like polymorphism

and code obfuscation, making it nearly impossible for the institution's current detection systems to identify these threats. As a result, the institution faces a growing risk of undetected malware infiltrating their systems, potentially leading to data breaches and financial losses.

Recognizing the limitations of their existing systems, the institution seeks a more advanced solution. They require a detection system that can go beyond static analysis and signature-based detection, one that can also analyze the behavior of potential threats in real-time while avoiding evasion tactics used by modern malware. This is where the proposed project comes into play, offering an integrated approach that leverages static and dynamic analysis, machine learning models, and YARA rules to provide comprehensive protection against even the most elusive forms of malware. By adopting this system, the institution can significantly enhance its cybersecurity defenses, ensuring the safety of its operations and the security of its customers' data.

## II. Design

The project aims to enhance the process of analyzing malware both statically and dynamically using artificial intelligence to determine whether the software is malicious or benign. The project is structured around four main themes to achieve this goal: static analysis, dynamic analysis, machine learning, and a user interactive interface.

### A. System Model ( Static Analysis )

Static analysis: To analyze the program without execution and identify anomalies or suspicious lines of code, special YARA rules are utilized. These rules help determine if the malware is packed or has activated anti-debug techniques. Additionally, the digital signature of each malware is matched and verified. Finally, the extracted data is prepared to be fed into the machine learning model.

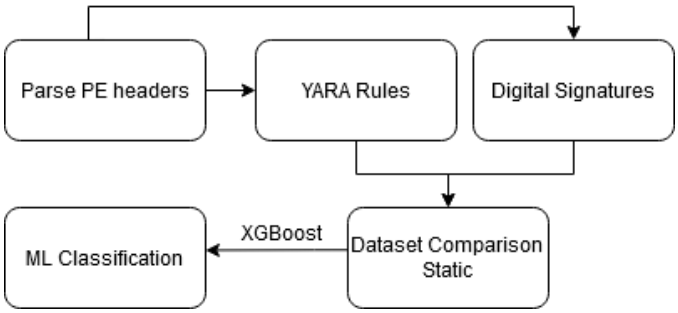



Fig. 1. Static Analysis

### B. System Model ( Dynamic Analysis )

Dynamic analysis: To analyze the program during execution and identify its behavior, an isolated Windows sandbox is used. This sandbox isolates the malware from the guest system, network, and overall file system, ensuring it cannot interact with these components. The sandbox captures all new activities performed by the malware, such as API calls, creating new users, and registry keys. Once the analysis is complete, the results are exported in JSON format to be imported into the machine learning model..

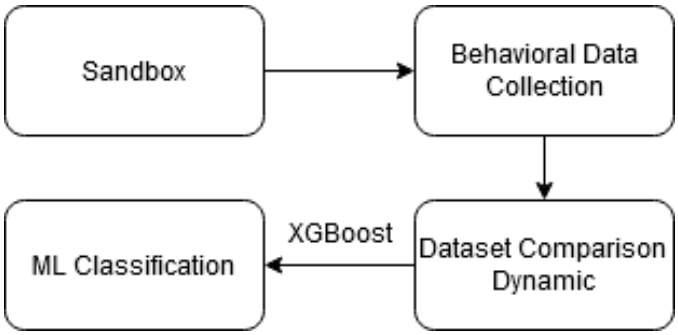



Fig. 2. Dynamic Analysis

### C. System Model ( Machine Learning )

Machine learning: involves utilizing a dataset containing over 40,000 samples. The dataset is trained using an XGBoost classifier. Dynamic and static analysis results are fed into the model, and the prediction outcome determines whether the sample is classified as malware or benign. [1]

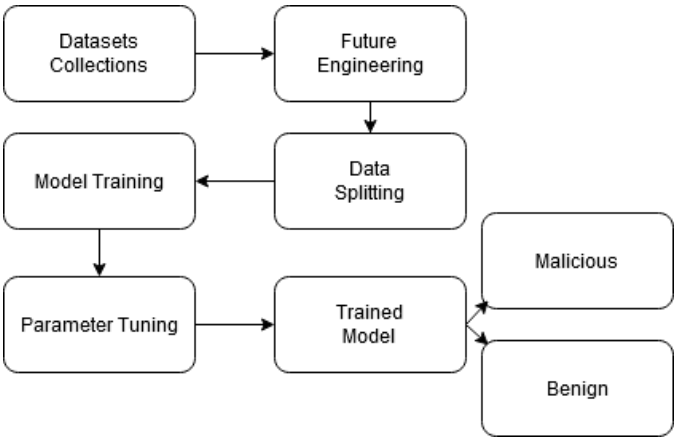



Fig. 3. Machine Learning

### D. System Model ( User Interface )

User Interface: A graphical user interface web page that show the results of both dynamic and static analysis with the help of Next.js and React.js. It very handy to be used by nontechnical person and also it show awareness to users educate them how to handle and response to this type of malware.

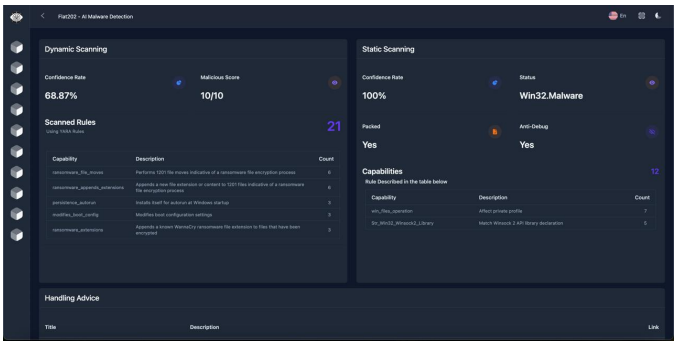



Fig. 4. User Interface

## III. Implementation

Python is used to code the entire project, from the machine learning model to displaying the final results. Static analysis is performed using YARA rules for signature matching, while dynamic analysis is conducted through the Cuckoo sandbox, isolated within VirtualBox.

## IV. Results and Discussion

- **Classification:** Clear indication of whether the sample is Malware or Benign, based on over 40,000 data points.
- **Behavioral Report:** Detailed insights from Dynamic Analysis, including API calls and system interactions.
- **Static Report:** Identification of PE header anomalies, YARA rule matches, and digital signature status.
- **Refining Detection:** Incorporating deep learning models to better identify sophisticated malware variants.
- **Expanding Dataset:** Enhancing detection rates by broadening the training dataset.
- **Ongoing Improvement:** Strengthening the system to counter evolving threats.

## V. Conclusion

This project demonstrates that combining traditional signature-based detection with modern machine learning techniques significantly enhances malware detection capabilities. The integration of static and dynamic analysis with machine learning models and YARA rules provides a comprehensive approach to detecting and classifying malware, even in the face of advanced evasion techniques. This methodology offers a powerful tool for strengthening cybersecurity defenses and protecting systems from evolving threats.

## VI. Acknowledgements

We would like to express our deepest gratitude to Mr. Mahmoud Ellouh for his unwavering support and invaluable guidance throughout the duration of this project.

We are also immensely thankful to Dr. Sultan and the KAUST Academy staff for providing us with this incredible opportunity.

Finally, we extend our sincere appreciation to the National Cybersecurity Authority (NCA) for their continued support.

## References

[1] Suspicious API-Based Malware Detection System, OCSLab. [Online]. Available: https://ocslab.hksecurity.net/sapimmds. Accessed: Aug. 31, 2024.

[2] J.-w. Jang and H. K. Kim, "Function-Oriented Malware Analysis as First Aid," *Journal of Information Security and Cybersecurity Research*, 2016. [Online]. Available: https://journals.nauss.edu.sa/index.php/JISCR/article/view/2975/1353/23351. Accessed: Aug. 31, 2024.