# Algorithmic Analysis of Malware Detection Techniques

Course: CSC 350- Analysis of Algorithms

Instructor: Dr. Md Ali

Group Members: Akacia Howell, Ricardo Joasilus, Subhan Alvi

Date: March 22, 2026

## Project Overview

This project focuses on developing a behavior-based malware detection system using graph modeling. Traditional antivirus systems rely on static analysis, which examines program code without executing it. Modern malware often avoids detection through code polymorphism. To address this, our project uses runtime behavior analysis to monitor actions such as process creation, file modifications, registry changes, and network communication. The goal is to analyze how graph algorithms can be used to detect malicious behavior efficiently while maintaining accuracy and scalability.

## Progress Summary

So far, we have made progress toward completing the technical portion of the project. First, we set up a controlled virtual environment using a virtual machine to safely execute programs. This ensured that all experiments were isolated and did not affect the host system. A sandbox environment was configured to monitor and record runtime behavior, allowing us to capture detailed logs of system activity.

Next, the collected data was obtained in JSON format, containing information about system behavior such as process activity, file interactions, and network communication. These logs served as the foundation for further analysis. At this stage, most of the technical implementation has been completed. The remaining work involves refining detection methods and evaluating system performance. The technical portion is expected to be fully completed by the upcoming deadline.

## Challenges and Adjustments

During the development of this project, several challenges were encountered related to implementation, coordination, and system design. These challenges require ongoing adjustments to ensure the project remained on track and the system functioned correctly. The challenges and adjustments were:

- Choosing the right data structure- determining the most effective way to represent behavior as graphs requires careful consideration.
- Debugging- identifying and fixing errors in parsing, graph construction, and logic took additional time.
- Inconsistent coding styles and logic errors
- Communication gaps
- Unexpected inputs
- Changing approach midway- adjustments were needed after realizing certain methods were not as effective as expected.

## Preliminary Results

Preliminary results suggest that behavior-based analysis using graph modeling is an effective approach for understanding program activity. The graph structures clearly illustrate relationships between system components and provide insight into how programs interact with the system during execution.

Although the detection system is still being refined, the current results demonstrate that the approach is capable of identifying meaningful patterns in runtime data. Further evaluation will be conducted to improve accuracy and validate the effectiveness of the detection methods.

# Updated Timeline

The project timeline has been adjusted to reflect the current stage of progress and remaining tasks. Since most of the initial implementation has been completed, the focus is now on final development, evaluation, and project completion.

Sprint 3- Detection and Evaluation (March 22- April 5)

- Finalize detection logic using extracted graph features
- Improve rule-based classification methods
- Test the system with multiple program samples
- Evaluate detection accuracy and consistency

Sprint 4- Performance Analysis (April 6- April 15)

- Analyze runtime performance as graph size increases
- Measure efficiency of graph construction and feature extraction
- Compare results across different behavior patterns
- Identify strengths and limitations of the approach

Sprint 5- Finalization (April 16- April 29)

- Complete final experiments and testing
- Prepare final report
- Develop and finalize presentation materials

References

B. Yu, Y. Fang, Q. Yang, Y. Tang and L. Liu, "A Survey of Malware Behavior Description and Analysis," in Frontiers of Information Technology & Electronic Engineering, vol. 19, no. 5, pp. 583-603, May 2018, doi: 10.1631/FITEE.1601745.

L. Liu, B. -S. Wang, B. Yu and Q. -X. Zhong, "Automatic Malware Classification and New Malware Detection Using Machine Learning," in Frontiers of Information Technology & Electronic Engineering, vol. 18, no. 9, pp. 1336-1347, September 2017, doi: 10.1631/FITEE.1601325.

Venugopal, D., & Hu, G. (2008). Efficient signature-based malware detection on mobile devices. Mobile Information Systems, 4(1), 33–49. https://doi.org/10.1155/2008/712353

Kim, H., Kim, D., & Kim, J. (2018). Runtime detection framework for Android malware. Mobile Information Systems, 2018, Article 8094314. https://doi.org/10.1155/2018/8094314