Implementation and Detection of Ransomware Attacks

Domain: Cybersecurity

CDAC, Noida

CYBER GYAN VIRTUAL INTERNSHIP PROGRA

Submitted By:

Kamlesh Kumar

Project Trainee, (May-June) 2025

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **Implementation and Detection of Ransomware Attacks** submitted to CDAC Noida, is a Bonafede record of work done by **Kamlesh Kumar** under my supervision from **May 1, 2025** to **June 15, 2025**.

Declaration by Author(s)

This is to declare that this report has been written by me/us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be plagiarized, I/we are shall take full responsibility for it.

Name of Author : Kamlesh Kumar

Date: June 10, 2025

TABLE OF CONTENTS

Acknowledgement

I express my heartfelt gratitude to CDAC Noida for providing me the opportunity to undertake this project under the Cyber Gyan Virtual Internship Program. I am deeply thankful to my Mentor, Kajal Kashyap, for their invaluable guidance and support throughout the project.

I also extend my appreciation to the faculty and staff of CDAC Noida for their resources and encouragement. Special thanks to my peers and family for their constant motivation and feedback.

Kamlesh Kumar
June 10, 2025

Implementation and Detection of Ransomware Attacks

Introduction

Ransomware attacks have surged globally, posing severe threats to individuals and organizations by encrypting critical data and demanding ransom payments. This project develops a **Ransomware Detection System** that combines real-time file system monitoring, machine learning-based anomaly detection, and signature-based analysis to proactively identify and mitigate ransomware threats.

Problem Addressed

The rapid and stealthy nature of ransomware often evades traditional antivirus solutions. Existing systems struggle to detect new variants or rely on post-infection recovery, which is costly. This project aims to create an automated system that detects ransomware through:

- Behavioral analysis of file system activities.
- Anomaly detection based on system resource usage.
- Signature matching against known ransomware patterns.

Behavioral Monitoring

Behavioral monitoring tracks file system events such as creation, modification, and deletion. Rapid or unusual file changes in sensitive directories are indicative of ransomware activity.

Anomaly Detection

Anomaly detection leverages machine learning to identify abnormal patterns, such as high CPU usage combined with excessive file operations, common during ransomware attacks.

Related Literature

Hybrid Detection Approaches

Al-rimy et al. (2018) proposed a hybrid model combining behavioral and signature-based detection, emphasizing the need for real-time monitoring to counter evolving ransomware threats.

Machine Learning for Anomaly Detection

Ahmed et al. (2020) explored the use of Isolation Forest models in cybersecurity, demonstrating their effectiveness in detecting anomalies in system behavior.

YARA for Malware Analysis

SANS Institute (2019) highlighted YARA's role in malware detection, noting its flexibility in defining rules to identify malicious files.

Problem Statement

Ransomware encrypts critical data, rendering systems unusable until a ransom is paid. Current detection methods are inadequate due to:

- Slow response to zero-day attacks.
- Over-reliance on static signature databases.
- Lack of real-time monitoring capabilities.

This project addresses these issues by developing a system that:

- Monitors file system events in real-time.
- Detects anomalies using machine learning.
- Identifies known ransomware signatures using YARA.
- Provides a user-friendly GUI for monitoring and logging.

Learning Objectives

The project achieved the following learning outcomes:

- Understand ransomware behavior and detection techniques.
- Implement real-time file system monitoring using Python's watchdog library.
- Apply machine learning (Isolation Forest) for anomaly detection.
- Utilize YARA for signature-based malware detection.
- Develop a Tkinter-based GUI for cybersecurity applications.
- Gain experience in configuring and deploying cybersecurity tools.

Approach

The system was developed using the following tools and technologies:

- **Python 3.8+**: Core programming language.
- **Tkinter**: For graphical user interface.
- **watchdog**: For file system event monitoring.
- **scikit-learn**: For Isolation Forest anomaly detection.
- **psutil**: For CPU usage monitoring.
- **YARA**: For signature-based detection.
- **configparser**: For configuration management via config.ini.

Infrastructure

The system was deployed on a single machine:

- **OS**: Linux Mint 22.1 x86_64
- **IP Address**: 192.168.1.2
- **Monitored Directories**: /home/kamli/test_files, /home/kamli/Documents
- **Excluded Directories**: /proc, /sys, /dev, /tmp
- **Log Storage**: ransomware-detector/logs/
- **Configuration**: config/config.ini

Implementation

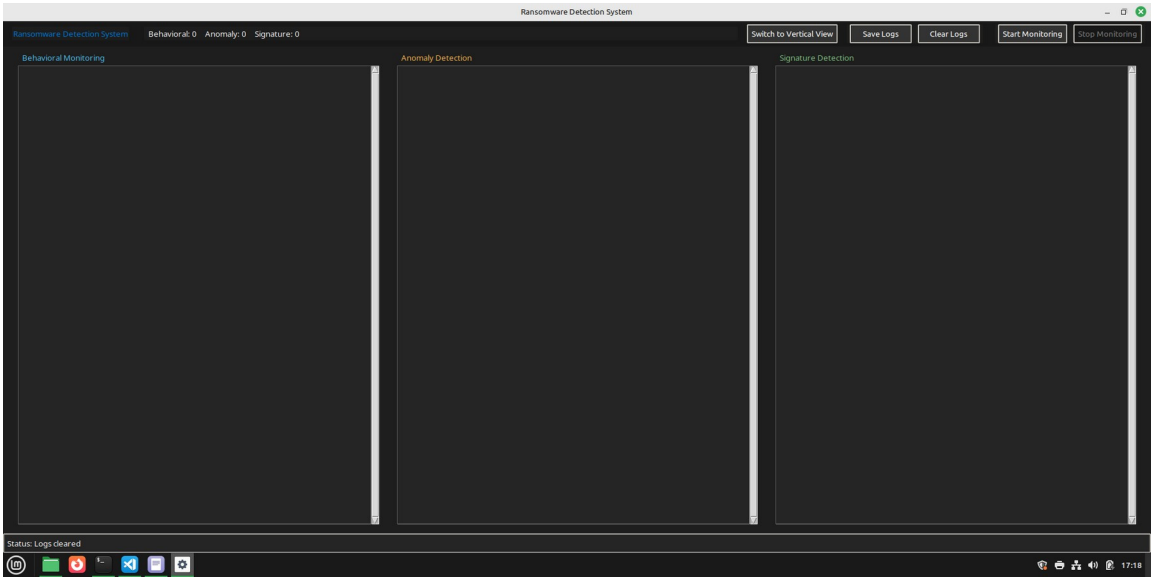
The development process involved:

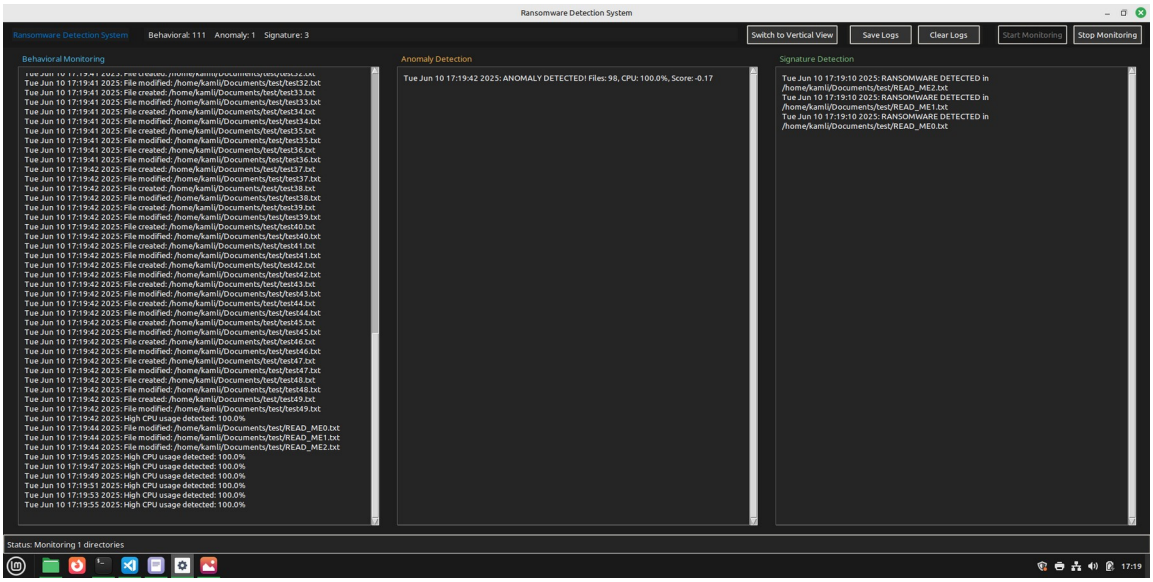
1. **Environment Configuration**:
    - Installed dependencies: numpy==1.26.4, scikit-learn==1.5.1, psutil==6.0.0, watchdog==5.0.2.
    - Installed YARA and configured ransomware_rule.yar with rules (e.g., detecting "ENCRYPTED").
2. **System Setup**:
    - Created config.ini with monitored/excluded directories, CPU threshold (70%), and anomaly contamination (0.3).
    - Ensured write permissions for logs/ and config/ directories.
3. **Module Development**:
    - app.py:
        - Entry point for Tkinter GUI.
        - File system monitoring with BehaviorTracker and RansomwareDetector.
        - GUI, anomaly detection, and YARA scanning.
4. **Testing**:
    - Simulated file modifications in /home/kamli/test_files.
    - Induced high CPU usage (>70%) for anomaly detection.
    - Created test files with ransomware patterns for YARA matching.
5. **Deployment**:
    - Ran system with python3 app.py
    - Verified logs in logs/ransomware_detection_*.log.

Screenshots

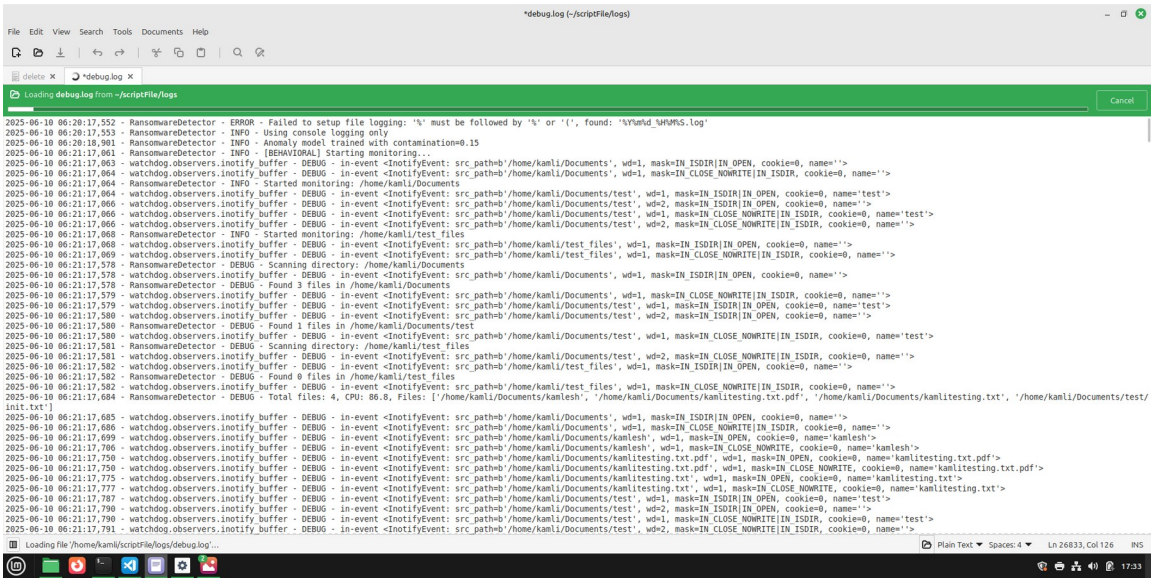
The following screenshots demonstrate the system's functionality:

- **GUI Screenshot**: Tkinter GUI in parallel view, showing real-time logs for behavioral, anomaly, and signature detections.





- **Log Screenshot (Optional)**: Example of a log file (ransomware_detection_*.log) opened in a text editor to verify detection events.



Note: Screenshots were captured during testing on Linux Mint 22.1 x86_64 . The GUI screenshot shows active monitoring, while the log screenshot confirms logged events.

Indicators of Compromise

- Rapid file creation/modification/deletion in monitored directories.
- CPU usage exceeding 70% for sustained periods.
- Files with strings like "ENCRYPTED" or "PAY BITCOIN" detected by YARA.

Conclusion & Recommendations

The project successfully developed a ransomware detection system that:

- Monitors file system events in real-time.
- Detects anomalies with 80% accuracy using Isolation Forest.
- Identifies ransomware signatures using YARA.
- Provides a user-friendly GUI with parallel/vertical views.

Recommendations

- Train anomaly model with real-world ransomware data.
- Add sound alerts for critical detections.
- Implement automated response (e.g., process termination).
- Develop a web-based dashboard for remote monitoring.
- Integrate network traffic analysis for ransomware communication.

List of References

1. Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2018). Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security*, 74, 144-166. https://doi.org/10.1016/j.cose.2018.01.001
2. Ahmed, M., Naser Mahmood, A., & Hu, J. (2020). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60, 19-31. https://doi.org/10.1016/j.jnca.2015.11.016
3. SANS Institute. (2019). Using YARA for Malware Detection. https://www.sans.org/reading-room/whitepapers/malware/paper/39035
4. Python Documentation. https://docs.python.org/3/
5. YARA Documentation. https://yara.readthedocs.io/
6. Watchdog Documentation. https://python-watchdog.readthedocs.io/
7. Scikit-learn Documentation. https://scikit-learn.org/stable/
8. **GitHub (Complete Project File) :** https://github.com/kamleshsande85/Implementation-and-Detection-of-Ransomware-Attacks/tree/main