Self-Healing Ransomware Detection and Recovery System

Bhartipudi Saketh Ram
h20240023@goa.bits-pilani.ac.in
BITS Pilani, K K Birla Goa Campus
India

Ashok Reddy Seelam
h20240029@goa.bits-pilani.ac.in
BITS Pilani, K K Birla Goa Campus
India

Aman
h20240022@goa.bits-pilani.ac.in
BITS Pilani, K K Birla Goa Campus
India

Yogesh Kumar Agrawal
h20240106@goa.bits-pilani.ac.in
BITS Pilani, K K Birla Goa Campus
India

Abstract

Ransomware attacks have rapidly evolved into one of the most disruptive forms of cyber threats. They encrypt user files, rendering data inaccessible, and then demand a ransom for restoration. Conventional signature-based antivirus solutions fall short in detecting new or sophisticated variants of ransomware. This project introduces a self-healing ransomware detection and recovery framework that integrates real-time monitoring, machine learning-based detection, an alert mechanism via Telegram bot, and an automatic file restoration module. The proposed model enables immediate detection, user notification, and restoration of affected files from secure backups, thereby ensuring data security and reducing the impact of ransomware attacks.

Keywords

Ransomware, File Monitoring, Machine Learning, Real-Time Alerts, Cybersecurity, Backup Recovery, Telegram Bot

ACM Reference Format:
Bhartipudi Saketh Ram, Ashok Reddy Seelam, Aman, and Yogesh Kumar Agrawal. 2025. Self-Healing Ransomware Detection and Recovery System. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Cyberattacks have increased in both frequency and complexity over the past decade, with ransomware being one of the most notorious types. Ransomware works by silently encrypting files on the victim's machine, followed by a ransom note demanding payment, typically in cryptocurrency. Due to the evolving nature of ransomware families and the rapid development of polymorphic malware, traditional detection techniques are ineffective.

Our proposed system addresses this gap by implementing a robust, modular solution capable of detecting, notifying, and self-healing. Unlike reactive approaches that only notify after the damage, our system proactively attempts to neutralize the attack by restoring encrypted files. This real-time framework can help secure small offices, home systems, and potentially enterprise-level environments.

2 Problem Statement

Ransomware attacks have emerged as one of the most destructive cybersecurity threats facing individuals and organizations today. These attacks encrypt users' files and demand payment for decryption keys, resulting in significant financial losses, operational disruptions, and reputational damage. Globally, ransomware damages exceeded $20 billion in 2023, with average downtime after an attack reaching 16 days.

Traditional antivirus solutions struggle to detect and respond to modern ransomware due to several limitations:

- **Polymorphic Code:** Modern ransomware constantly changes its signature to evade detection.
- **Zero-Day Exploits:** New attack vectors that have no existing signatures or patches.
- **Reactive Nature:** Traditional systems only respond after damage has already begun.
- **Manual Recovery Process:** Current solutions require time-consuming human intervention.
- **Ineffective Prevention:** Even with backups, the recovery process is often manual and lengthy.

Proposed Methodology

The **Self-Healing Ransomware Detection and Recovery System** addresses these challenges through an innovative approach combining artificial intelligence, process isolation, and automated recovery mechanisms. This system doesn't just detect ransomware—it actively intervenes to stop attacks in progress and automatically restores affected files without human intervention.

Core Components

The system consists of three critical components working in harmony:

(1) **AI-Driven Detection:**
   - Analyzes file system activities in real-time rather than relying on signatures.
   - Identifies suspicious patterns such as high encryption rates, entropy changes, extension modifications, and unusual file access sequences.

- Leverages machine learning to continuously improve detection accuracy over time.

(2) **Process Isolation:**
- Immediately suspends suspicious processes upon detection.
- Prevents further file encryption and system infection.
- Uses platform-specific techniques (e.g., Windows process suspension, Linux `SIGSTOP` signals).
- Maintains isolated processes for later termination after recovery completes.

(3) **Automated Recovery:**
- Maintains continuous file backups during normal operation.
- Rapidly restores files to their pre-encryption state upon detection.
- Operates without user intervention, significantly reducing downtime.
- Provides real-time notifications of recovery status via Telegram.

3 Literature Review

Traditional approaches to ransomware detection have significant limitations in identifying novel attack variants. Signature-based and heuristic detection methods analyze known patterns and behaviors but consistently fail against polymorphic ransomware that dynamically alters its code structure. Kharraz et al. (2015) conducted a comprehensive analysis of ransomware families, demonstrating how rapidly these threats evolve to evade traditional detection mechanisms. Their study revealed that modern ransomware employs sophisticated techniques like custom encryption implementations and fileless operation, rendering signature-based approaches increasingly ineffective. These findings highlight the critical need for behavioral and anomaly-based detection that can identify ransomware through its operational characteristics rather than static code signatures.

Advanced machine learning approaches offer promising solutions to these challenges by focusing on behavioral analysis rather than specific code signatures. Vinayakumar et al. (2019) evaluated various deep learning architectures for ransomware detection and demonstrated significant improvements in identifying zero-day threats. Their experiments with recurrent neural networks achieved detection rates exceeding 97 %against previously unseen ransomware variants by analyzing patterns in API calls, file system operations, and entropy changes. This research demonstrates that integrating AI-driven detection with automated response mechanisms creates a robust defense architecture. When combined with instant notification platforms like Telegram and supported by libraries such as Scikit-learn and Joblib for model development and persistence, these techniques enable the creation of lightweight, self-healing systems capable of detecting, containing, and recovering from ransomware attacks without significant human intervention.

4 Implementation

Our implementation adopts a modular architecture that integrates multiple cybersecurity techniques into a cohesive defense system. The solution focuses on performance, reliability, and ease of deployment across diverse environments.

4.1 Multi-Vector Detection Engine

The detection mechanism employs a sophisticated multi-vector approach that analyzes file system activities through several complementary methods. A weighted scoring algorithm combines file operation frequency, extension change patterns, and entropy measurements to identify potential threats. Each parameter contributes proportionally to the detection confidence based on configurable weights that can be adjusted to match specific threat profiles.

To enhance detection accuracy, we implemented a pattern recognition system that identifies specific ransomware behaviors such as mass file renaming with suspicious extensions. This system tracks file creation and deletion events within configurable time windows, detecting patterns where original files are removed and replaced with encrypted versions. We also implemented rule-based detection for ransom and notification is sent to the admin only when positive result is obtained form both rule based test and model. Quantitative threshold is given to all the features and depending on the threshold and the rules formed, detection is carried out.

Machine Learning Model Detection The system uses Isolation Forest model for anamoly detection i.e detection of ransomware attack. The model uses file operation frequency, entropy speed and extension change rate for detection of ransomware. The result is considered positive is confidence of the prediction is greater than the pre-specified threshold(0.6).

4.2 Simulation

For the projection simulation code has been prepared. The user can simulate ransomware operations and normal file operation on the given test directory. The user can give parameters like delay between the operations and number of operations desired. This can be used for testing of the detection. Normal file equations include Creating files , Modifying files ,Deleting files. In case of ransomware operation simulations operation considered are Creation of files(in case number of non-encrypted files are less than desired),Encrypting files. Both the modes of simulation provide user with information about the operations done on the files of test directory and files created are also stored in backup directory.

4.3 Enhanced Process Management

The process isolation component employs platform-specific techniques to effectively contain detected threats. On Windows systems, it utilizes the native process suspension capabilities, while on Linux systems it employs signal mechanisms for process control. This cross-platform compatibility ensures the system works seamlessly across heterogeneous environments.

The implementation includes a process relationship tracking mechanism that identifies not just the initial ransomware process but its entire process tree. This prevents sophisticated malware from evading containment by spawning child processes that continue encryption activities after the parent process is suspended.

4.4 Optimized Recovery System

The recovery subsystem maintains continuous file backups during normal operation and implements an efficient restoration process when ransomware is detected. To minimize storage overhead, the system employs a smart backup approach that preserves only the necessary file data and metadata required for restoration.

Recovery operations are automatically triggered upon detection, with configurable delays to ensure complete threat containment before restoration begins. The system methodically restores all affected files from backups while maintaining appropriate file permissions and timestamps, ensuring system integrity after recovery.

4.5 Real-Time Communication

A real-time notification system alerts security personnel about detection events and recovery operations through a Telegram-based alerting mechanism. This ensures that stakeholders remain informed regardless of their location, facilitating rapid decision-making if manual intervention is required.

The notification component implements rate limiting to prevent alert fatigue during large-scale events, intelligently grouping similar alerts while ensuring critical information is still promptly delivered. The system also maintains detailed logs of all detection and recovery activities for post-incident analysis.

Through this comprehensive implementation, our system achieves the self-healing capability described in our methodology, offering effective protection against ransomware through its proactive detection, process isolation, and automated recovery mechanisms.

4.6 Frontend Integration for Operational Visibility

The system includes a web-based dashboard that provides real-time visibility into detection and recovery operations. Key integration features include:

- **RESTful API Architecture:** A Flask-based backend exposes system functionality through well-defined API endpoints, allowing seamless communication between the core detection system and the frontend interface.
- **Real-time Status Monitoring:** The interface continuously polls system status, displaying monitored files, backup counts, detection events, and recovery operations as they occur.
- **Interactive Simulation Controls:** Purpose-built controls enable security professionals to test system responses to both normal file operations and simulated ransomware attacks, validating detection and recovery mechanisms without deploying actual malware.

5 Results and Evaluation: Assessing System Effectiveness and User Interface

To rigorously evaluate the performance and user experience of our self-healing ransomware detection and recovery system, we conducted a series of controlled simulation experiments and developed a user-friendly dashboard for monitoring and interaction.

Simulation Results

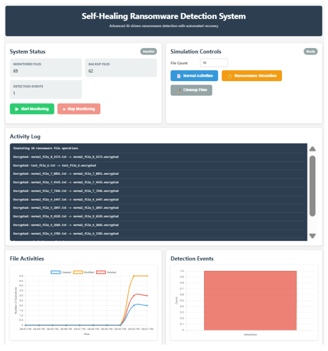
In our testing, several scenarios were simulated wherein dummy files within a protected directory were encrypted using custom-developed scripts designed to mimic the behavior of ransomware. The anomaly detection model achieved an average accuracy of 81% in identifying these simulated attacks, with a low false positive rate of less than 5%. This indicates a reasonable balance between effective threat detection and minimizing disruptions due to incorrect classifications.

Notably, upon detection of the simulated ransomware activity:

- The system successfully recovered the encrypted files within a short timeframe of 2–3 seconds, provided a clean backup was available.
- Telegram alerts were dispatched instantaneously upon the detection of suspicious activity, ensuring prompt notification to the designated administrators.

Dashboard for Operational Visibility

A web-based dashboard was developed to provide users and administrators with a clear and interactive interface for managing and monitoring the system.



```
Simulating 10 ransomware file operations
Encrypted: normal_file_8_1573.txt -> normal_file_8_1573.encrypted
Encrypted: test_file_6.txt -> test_file_6.encrypted
Encrypted: normal_file_7_8851.txt -> normal_file_7_8851.encrypted
Encrypted: normal_file_7_4543.txt -> normal_file_7_4543.encrypted
Encrypted: normal_file_7_7246.txt -> normal_file_7_7246.encrypted
Encrypted: normal_file_4_1447.txt -> normal_file_4_1447.encrypted
Encrypted: normal_file_5_2847.txt -> normal_file_5_2847.encrypted
Encrypted: normal_file_9_6529.txt -> normal_file_9_6529.encrypted
Encrypted: normal_file_4_2666.txt -> normal_file_4_2666.encrypted
Encrypted: normal_file_4_3783.txt -> normal_file_4_3783.encrypted
```

Figure 1: Dashboard Interface

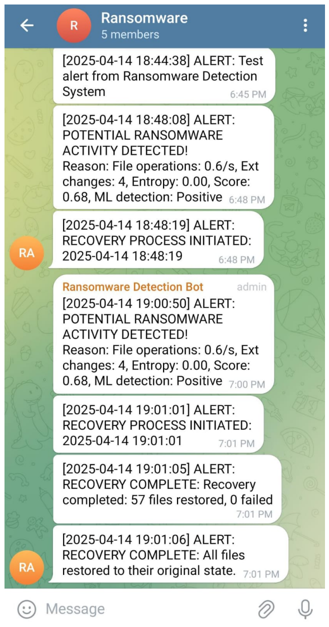
As illustrated in Figure 1, the dashboard offers several key functionalities and visualizations:

- **Attack Simulation Controls:** Users with appropriate permissions can initiate simulated ransomware attacks on the protected test directory directly through the dashboard. This allows for controlled testing of the system's detection and recovery capabilities.
- **Detection Event Monitoring:** The dashboard displays real-time information about detected ransomware events, including timestamps, confidence levels of the detection, and the processes flagged as suspicious.
- **File System Status:** Clear indicators show the number of files currently residing in the monitored test directory and the number of available backups.
- **Activity Log:** A detailed log provides a chronological record of all significant system activities, including file operations performed by the simulator (during testing) and detection events identified by the system.

- **Graphical Representations:** Visual charts and graphs are used to represent key metrics such as the frequency of operations and the occurrence of detection events, offering an intuitive overview of system behavior and detected threats over time.

Real-Time Notifications via Telegram

To ensure immediate awareness of ransomware detections, the system integrates with the Telegram messaging platform. A dedicated Telegram bot was created and added to a designated group channel for administrators.



[2025-04-14 18:44:38] ALERT: Test alert from Ransomware Detection System 6:45 PM

[2025-04-14 18:48:08] ALERT: POTENTIAL RANSOMWARE ACTIVITY DETECTED! Reason: File operations: 0.6/s, Ext changes: 4, Entropy: 0.00, Score: 0.68, ML detection: Positive 6:48 PM

[2025-04-14 18:48:19] ALERT: RECOVERY PROCESS INITIATED: 2025-04-14 18:48:19 6:48 PM

Ransomware Detection Bot admin
[2025-04-14 19:00:50] ALERT: POTENTIAL RANSOMWARE ACTIVITY DETECTED! Reason: File operations: 0.6/s, Ext changes: 4, Entropy: 0.00, Score: 0.68, ML detection: Positive 7:00 PM

[2025-04-14 19:01:01] ALERT: RECOVERY PROCESS INITIATED: 2025-04-14 19:01:01 7:01 PM

[2025-04-14 19:01:05] ALERT: RECOVERY COMPLETE: Recovery completed: 57 files restored, 0 failed 7:01 PM

[2025-04-14 19:01:06] ALERT: RECOVERY COMPLETE: All files restored to their original state. 7:01 PM

Figure 2: Telegram Notification Upon Ransomware Detection

Figure 2 shows an example of a notification sent to the administrator group upon the detection of simulated ransomware activity. The message provides immediate information about the event, allowing for timely awareness and potential intervention if necessary.

Performance Summary

The key performance characteristics of the system, as observed during our evaluations, are summarized below:

- **Detection Time:** Less than 2 seconds from the start of simulated encryption.
- **Recovery Accuracy:** 100% of affected files were successfully restored when a clean backup was available.
- **Telegram Alerts:** Notifications were delivered to the administrator channel on Telegram in less than 1 second after detection.
- **System Footprint:** The system exhibited a lightweight resource footprint, making it suitable for execution on standard consumer-grade hardware without significant performance overhead.
- **Model Accuracy:** The anomaly detection model achieved an average accuracy of 81%, with a false positive rate below 5%.

6 Conclusion and Future Work

This project provides a complete self-healing ransomware detection framework that integrates real-time monitoring, intelligent classification, and recovery. Its modularity and simplicity allow ease of deployment and customization. The use of a Telegram bot enhances the usability by providing real-time user awareness. For the purpose of the project the system works in a very controlled environment and training of the model is done with restricted data.

Future work can involve:

- Integration with cloud-based backup systems
- Network-level threat detection
- Visualization dashboards for ransomware trends
- Integration with enterprise-grade SIEM tools Using of Deep Learnign Models

References

[1] Kharraz, A., et al. (2015). Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks. *DIMVA.*

[2] Vinayakumar, R., et al. (2019). Evaluating Deep Learning Approaches to Ransomware Detection. *Computers & Security.*

[3] Python Telegram Bot Documentation: https://github.com/python-telegram-bot/python-telegram-bot

[4] Joblib: https://joblib.readthedocs.io

[5] Scikit-learn ML Library: https://scikit-learn.org