Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347 - 2812

Volume 14 Issue 03s, 2025

Proactive Ransomware Early Warning System

[1]Tanvika B. Padole, [2]Hridaya S.Thangan, [3]Arya R. Chahankar, [4]Bhumika K. Soor, [5]Nitiket G. Karmore, [6]Prof.Sachin Janbandhu

[1,2,3,4,5,6] Department of Computer Science and Engineering (Cyber Security), St.Vincent Pallotti College of Engineering and Technology
Nagpur ,Maharashtra, India-441108
Email: [1]tanvikapadole28@gmail.com, [2]hridayathangan@gmail.com, [3]aryachahankar@gmail.com,, [4]bhumikasoor16@gmail.com, [5]nitiketkarmore30@gmail.com, [6]sjanbandhu@stvincentngp.edu.in

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

Ransomware Detection, Early Warning System, Entropy Analysis, Process Monitoring, Alert Generation

Abstract

Ransomware has emerged as one of the most severe cybersecurity threats, encrypting sensitive data and demanding ransom for decryption. Traditional antivirus mechanisms often fail to detect new and evolving ransomware variants. This paper presents a Proactive Ransomware Early Warning System designed to detect and prevent ransomware activity during its initial stages. The system continuously monitors file directories and process behaviors to identify signs of encryption, such as abnormal entropy changes and mass file access. Built using Python, the system integrates modules like Watchdog for file tracking, Psutil for process monitoring, and Tkinter with Pystray for user alerts. When suspicious activity is detected, the tool alerts the user and can terminate malicious processes to prevent damage. Experimental results show that this lightweight and real-time system effectively provides early ransomware detection for personal and small office environments.

Introduction

Ransomware is malicious software that encodes user files and asks for money in return for a decryption key. The attacks have progressed very quickly, threatening individuals, businesses and critical infrastructure. The latest attacks such as WannaCry and Petya indicated the disastrous effect of such an attack, which tends to spread across the world in less than a day. Conventional antivirus solutions are mainly based on signature detection, which is ineffective against zero-day and polymorphic variants. Therefore, an intelligent behavior-based detection system that can detect early indicators of ransomware activity prior to the key or significant amount of damage going to or the damage being done is necessary. Traditional detection approaches like signature-based and heuristic-based systems are mostly dependent on known malware signatures or rule-based inspection. These methods fail against zero-day ransomware that changes its code or behavior to avoid detection. Also, most available tools detect ransomware activity only once the encryption process has commenced, giving users little room for recovery.

To counter such threats, the Proactive Ransomware Early Warning System is programmed to identify potential ransomware traffic in its incipient phase before real harm is done. The system scans file directories and process activity constantly to find irregular patterns like abrupt entropy fluctuations, large-scale file access, or unauthorized encryption efforts. Developed using Python, it employs modules like Watchdog for real-time file tracking, Psutil for process monitoring, and Tkinter with Pystray for user alerts. When suspicious behavior is detected, the system immediately

© 2025 The Authors. Published by MRI INDIA.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

alerts the user and can terminate the malicious process to prevent further damage. This proactive mechanism is concerned with real-time monitoring, behavior-based detection, and user notification instead of overdependence on conventional signature-based scanning. The suggested system seeks to offer an effective, lightweight, and feasible defense mechanism for personal computers and small office settings, providing early detection of ransomware threats.

Objective

1.Early Ransomware Threat Detection:
The major objective of the system is early detection of ransomware activity. Through behavior analysis, the tool observes unusual process activities and file operations. Furthermore, entropy-based measures assist in detecting an abrupt change in file randomness, which is a common indicator of file encryption by ransomware. Early detection minimizes potential data loss and enables timely intervention.

2.Real-Time Alert for Users:
The system is designed to alert users in real time when malicious activity is noticed. Users can take preparatory measures, for example, halt operations, save vital files, or check the threat if they receive instantaneous alerts. Timely notifications are needed to prevent damage and ensure system security.

3. Optional Termination of Malicious Processes:
In order to avoid severe damage, the software offers the feature to optionally kill processes that have been marked as malicious. With the early termination of ransomware execution, it helps in protecting important files and maintaining overall system integrity. Users can opt to permit or automate this process based on their preference.

4. User-Friendly Desktop Interface:
The project is aimed at providing a user-friendly and intuitive desktop interface. Users are able to observe the system status, read alerts, and operate the tool without difficulty. An intuitive interface provides maximum accessibility even for non-expert users, improving usability and productivity.

5. Lightweight and Efficient Design:
The software is made to be light and effective, with minimal use of system resources. This makes it perfect for personal computers and small office settings, where low system usage and high performance are required. The system may be left running in the background continuously without interfering with normal activities.

Literature Survey

Ransomware has matured from basic locker- and scareware to advanced crypto-ransomware that attacks user files by silently encrypting them and requiring payment for redemption. Extensive forensic and behavioural examinations have determined that contemporary ransomware often follows a pattern of reconnaissance, credential abuse, lateral movement and ultimately bulk file encryption — a cycle that makes early discovery prior to complete encryption essential. Seminal reverse-engineering and fieldwork form the basis of understanding these attack patterns and indicative compromise indicators. Kharraz et al. offer a thorough analysis of real-world ransomware samples and their behaviors, pointing out shared primitives (file enumeration, quick file changes, and utilization of encryption routines) that detection mechanisms can attack. [1]

Detection methods tend to divide into signature-based methods and behavioural (anomaly) methods. Signature methods are accurate for familiar families but do not work for new or polymorphic variants; this shortfall inspires behaviour-based, pre-encryption detection mechanisms that search for unusual filesystem and process activity instead of recognizable byte patterns. The MITRE ATT&CK framework collates seen tactics and techniques employed by ransomware and other connected threats and is an applied guide to mapping detection telemetry to adversary behaviours, allowing better monitoring strategies and rule design. [4]

Filesystem-level protection and proactive containment solutions have been suggested to limit damage even when malicious action starts. Continella et al. presented ShieldFS, a self-healing filesystem that is able to roll back malicious changes and offer an automated means of recovery, exhibiting the usefulness of staying at the filesystem level in order to counteract encryption effects. That paper emphasizes the importance of pairing detection with automated containment or remediation to prevent losses in the early phases of an attack. [2] Reinforcing this, CryptoDrop targets ransomware prevention through watching for user data for indications of encryption activity (e.g., drastic file entropy changes and file writing patterns) and initiating defensive actions upon threshold breach; CryptoDrop's approach directly leads to the abstract's emphasis on entropy and large-scale file access as prime indicators. [3]

Pre-encryption or early warning detection studies have recently advanced to more formal

taxonomies and newer algorithmic methods. The PERD taxonomy organizes pre-encryption detection approaches (statistical, machine-learning, rule-based, and hybrid), providing an organized method to choose and explain detection features and trade-offs (false positives, latency, and resource utilization). [6] Solutions like Zero-Ran Sniff use contemporary machine-learning methods (specifically zero-shot learning) to identify heretofore unknown ransomware activity without labelled training data and exhibit promising avenues for identifying zero-day variants while demarcating the tradeoffs between model complexity and runtime expense on endpoint machines. These works point to feasible avenues for enhancing detection sensitivity to new attacks without being too heavy for home and small office settings. [5][6]

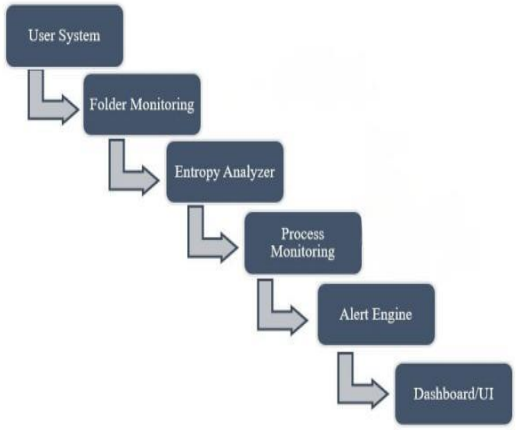
Process and system monitoring are just as vital to file monitoring. Runtime monitoring libraries and tools (as employed in most prototype systems) provide the ability to observe process trees, CPU and I/O patterns, and anomalous process behavior such as spawning encryption threads or widespread file handle usage. Combining process telemetry with filesystem data enhances detection confidence and allows for targeted response (for example, killing a particular malicious process as opposed to quarantining the entire host), which is an ability discussed in the project abstract and enabled by existing work on behaviour-based remediation tactics. [1][3][4]

Practical, real-time detector implementation decisions focus on lightweight, ubiquitous tooling and cross-platform compatibility. Some research and prototype systems utilize high-level languages and readily available monitoring libraries to decrease development time and resource expense. The combination of Python packages you cite (Watchdog for file system events, Psutil for process and system information, Tkinter/Pystray for user notification and minimal GUIs) are appropriate for implementing a real-time early warning system for desktop and small office computers; their documentation and existing projects show feasibility for continuous monitoring with acceptable overhead. [7]

Even with advances, gaps exist. Most research prototypes are tested on artificial or small datasets that lack production diversity of file systems, workloads, and high-churn benign behaviors; this can overstate detection metrics or amplify false positives in the wild. There is also a conflict between usability (not triggering nuisance alarms) and sensitivity (sensing early, subtle changes); good systems tend to blend multi-signal heuristics (entropy + mass accesses + process context + observed ATT&CK telemetry) to increase confidence before taking disruptive measures such as killing processes or reversing files. The literature suggests cautious threshold tuning, user-informed response choices for the event, and (where feasible) an automatic but conservative containment approach like quarantining or snapshot rollback instead of direct process destruction. [1][2][3][4][6]

System Architecture

The proposed Proactive Ransomware Early Warning System is designed to monitor and detect malicious encryption activities in real time. The architecture consists of six main components that work sequentially to ensure early detection and user notification.



Methodology

1. Data Collection:

- Sample ransomware behaviors and benign activities were simulated to test detection accuracy.
- Data sources include ransomware case studies, MITRE ATT&CK framework, and system log patterns.
- Public documentation from Python libraries such as Watchdog and Psutil was referenced.

2. Experimentation:

- Implemented in Python using a modular design.
- Simulated ransomware-like behaviors (rapid encryption, mass file modification).
- System response measured for detection time, alert accuracy, and prevention success.
- Compared with normal operations to assess false positive rates.

3. Analysis Techniques:

- **Entropy Analysis:** Measures randomness in file content to detect encryption.
- **Process Behavior Analysis:** Analyse, Monitor and Tracks CPU usage and file access rates.
- **Anomaly Detection:** Flags irregularities over thresholds.
- **User Interaction**: Alerts via GUI pop-ups, allowing user decision.

4. Experimental Setup:

- Operating System: Windows 10
- Programming Language: Python 3.12
- Libraries: Watchdog, Psutil, Tkinter, Pystray, OS, Math.
- Test Environment: Local system with simulated ransomware and normal tasks.
- Evaluation Metrics: Detection Time, Accuracy, Resource Utilization.

5. Ethical Considerations and Limitations

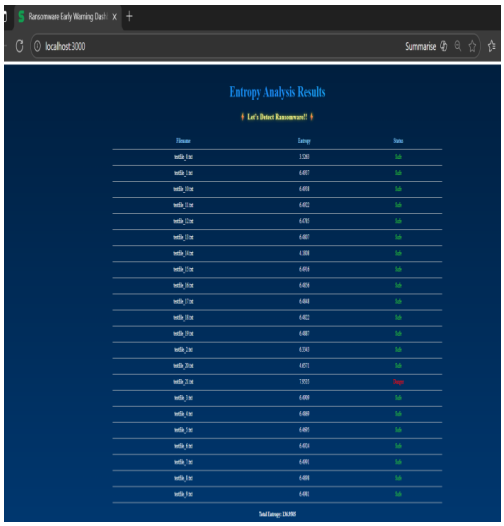
Ethical Considerations:

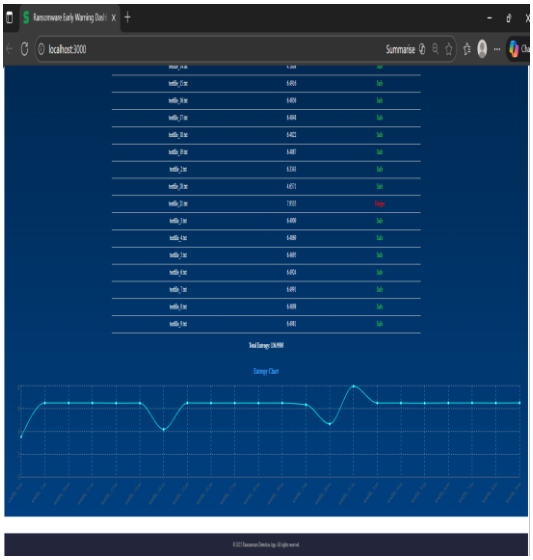
- Experiments conducted on controlled and isolated environments.
- System intended for defensive and educational use only.
- No real ransomware deployed on production systems.
- Logs collected by the tool should be securely stored.

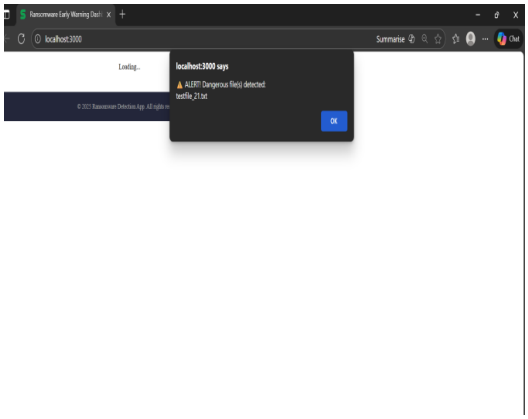
Limitations:

- Limited to early detection; cannot decrypt encrypted data.
- High entropy from legitimate compression tools may cause false alerts.
- May not detect highly stealthy or fileless ransomware variants.
- Currently desktop-based; lacks network-wide monitoring.

RESULT







Future Scope

- The platform can be improved with machine learning-based algorithms to enhance detection quality and automatically label suspicious

ransomware activity.

- Integrating with cloud-based threat intelligence platforms may allow real-time comparison against known ransomware patterns of suspicious files and processes.
- Network traffic monitoring can be introduced to detect command-and-control communications or attempts at data exfiltration.
- An automated recovery and backup process can be added in order to safeguard important files immediately upon detection of initial ransomware activity.
- The system can be made multi-os capable with the addition of support for Linux and macOS for broader use and effective utilization of tools in the systems.
- A centralized dashboard may be created for the monitoring and management of multiple systems on enterprise networks.
- The graphical user interface could be enriched with detailed logs, activity charts, and real-time visualizations of threats aids for constant monitoring.
- Machine learning can be applied to learn and update normal file entropy levels, enhancing the accuracy of anomaly detection.
- The system could be enhanced with current antivirus.
- Endpoint Detection and Response (EDR) solutions for multi-layer protection.
- Later versions can be equipped with a sandbox environment to analyze and verify suspicious processes safely before killing them.

Acknowledgement

We would like to extend our heartfelt thanks to Prof. Sachin Janbandhu for his expert mentorship, useful suggestions, and constant encouragement throughout this work. His timely remarks constructive feedback immensely improved the quality and intensity of our work. We also thank the St. Vincent Pallotti College of Engineering & Technology, Nagpur Department of Computer Science and Engineering (Cyber Security) for the facilities and support extended to us.

References

Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks.

Continella, A., et al. (2016). ShieldFS: Self-healing Filesystem Against Ransomware.

Scaife, N., Carter, H., Traynor, P., & Butler, K. (2016). CryptoDrop: Stopping Ransomware Attacks on User Data.

MITRE ATT&CK Framework: Ransomware Techniques and Behaviors.

Sunkara, S. P. (2025). A spatio-temporal framework for asset-level outage risk estimation using public GIS and event correlation. International Journal of Computer Engineering and Technology (IJCET), 16(1), 4211–4227. https://doi.org/10.34218/IJCET_16_01_286

Hazarika, I. (2014). Performance metrics versus wealth metrics of Dubai telecommunication sector. *In* Proceedings of the International Business Information Management Association Conference–IBIMA (Vol. 23). Valencia, Spain.

Sharma, B. (2025). Liability and virtual spaces: Examining legal responsibilities in Metaverse. National Journal of Cyber Security Law, 8(2).

Zero-Ran Sniff (2023): A Zero-Day Ransomware Early Detection Method Based on Zero-Shot Learning.

PERD Taxonomy (2023/24) – Categorizes methods for pre-encryption ransomware detection

Python Documentation (2023) – Watchdog, Psutil, Tkinter, Pystray Libraries.