Interdisciplinary Journal of Computing & AI

IJCAI
Interdisciplinary Journal of Computing & AI

IJCAI

CONCEPTUAL

A Lightweight Real-Time Ransomware Detection System Based on File System Behavioral Monitoring

Joel Francis[1], Barnali Gupta Banik[2*]

[1]*Neerja Modi World School, Jaipur, Rajasthan, India; joel.francis1311@gmail.com,*
[2]*Mahatma Gandhi Institute of Technology, Hyderabad, India; barnali.guptabanik@ieee.org*

ARTICLE INFO.

Keywords: Behavioral Analysis, File System Monitoring, I/O Request Packets (IRP), Machine Learning, Shannon's Entropy, Ransomware Detection, Real-Time Security, Zero-Day Attacks
*Corresponding author email address: barnali.guptabanik@ieee.org
Manuscript received: 20 June 2026/
Accepted: 1 July 2026
Published online: 8 July 2026
https://doi.org/10.5281/zenodo.21127413

Cite as: J. Francis, B. Gupta Banik, "A Lightweight Real-Time Ransomware Detection System Based on File System Behavioral Monitoring", Interdisciplinary Journal of Computing & AI (IJCAI), vol. 1, no. 2. Aspiration Publishing Trust (DARPAN ID: WB/2025/0887371), p. 251-262, Jul. 2026. doi:10.5281/zenodo.21127413.

ABSTRACT

Ransomware is a persistent security threat. It is so effective because of the rapidity of encryption and interruption of systems forced into paralyzed operations. Yet static signature-based antivirus systems fail to provide sufficient protection against ransomware, especially zero-day attacks that lack identifiable signatures during their initial stages. Instead, a more efficient and adaptable approach comes from detection systems that actively monitor specific behaviors in real time. This paper presents a lightweight ransomware detection system based on file operations that operates in real time and leverages dynamic behavioral features of real-world attacks, such as abnormal read, write, and rename operations, systematic file modifications, and intensive I/O activities that typify high-frequency encryption. Focusing on these behavioral features allows a system to detect ransomware (and differentiate it from legitimate user activity) without relying on previously established malware signatures.

1 Introduction

One of the most dangerous cybersecurity threats is ransomware. It is particularly effective. With the touch of a button, people's entire systems are shut down. What is more, antivirus programs often cannot mitigate ransomware attacks. They rely on signatures to stop attacks. Signatures are digital fingerprints that have been recorded to identify known threats. This leads to the "zero-day" dilemma. A zero-day attack means that the ransomware software is a new variant that has no existing signature in the security databases yet. Because the software is unknown, the signature has no historical point of reference and the virus bypasses the security defenses entirely [1].

This research suggests that we should hunt for what a file does versus what it is. By hunting for a file's behaviors, an endpoint can detect even unknown zero-day ransomware in action. The best methods of real-time ransomware detection are those based on behavioral characteristics. The main signals of infection are the rising "data entropy" as the file converts from human-readable to encrypted, and a flurry of orderly file renaming. Such insights lead to the ability to detect infection without first requiring the identification of the specific variant [2].

The main technical difference between an attack and normal use is in the "I/O operations". Where a normal user or even an administration will make irregular infrequent changes to the files, there will be an enormous amount of "READ, WRITE and RENAME" calls over a very short period of time by the very act of the attack itself. By monitoring the frequency and order of these events, it is possible for the system to identify a malignant process rather than a working human. Since signature-based systems cannot protect against unknown zero-day attacks, "anomaly-based" models are necessary. These do not look for the signature of a specific file, but rather any changes to the filesystem that are unusual. This paper presents an evaluation of a simplistic behaviour-based prototype, to ascertain whether it can detect zero-day ransomware better than other methods, in a secure testing environment.

2 Literature Survey

The rapid evolution of ransomware has outpaced static analysis to determine its origin, contributing to the focus on dynamic analysis of ransomware. The literature identifies three types of footprints that can indicate ransomware activity: file system I/O, the entropy of the data, and the consumption of system hardware resources.

2.1 File System I/O and Request Packets

The existing work determined that the most powerful footprints of ransomware are found in the I/O stack of the machine. All ransomware activity will generate I/O Request Packets. During ransomware activity, there will be a spike in READ and WRITE requests to perform the encryption, followed by RENAME or DELETE requests to replace the user's original file with the encrypted data. Unlike the sporadic I/O activity that is exhibited by a normal user, ransomware will have high I/O velocity that can be detected by the system [3].

2.2 Shannon's Entropy as a Detection Metric

Another essential metric to measure the behavior of ransomware is Shannon's Entropy. Shannon's Entropy measures the randomness of data. Data that a user creates normally will have low entropy. However, encryption will raise the entropy of the data that is being encrypted to extreme values, to a maximum of 8.0. Detecting this "Entropy Spike" can help differentiate between a legitimate file save and a ransomware infection.

2.2.1 Mathematical Framework of Shannon's Entropy

In information theory, Shannon's Entropy is a metric for determining the randomness of a dataset. In case of ransomware detection, it allows one to distinguish between data from a text file and encrypted data [4].

The entropy H of a random variable X is given in the following formula:

$$H = -\sum p(x_i) \log_2 p(x_i)$$

Where:
n is the number of possible outcomes of the represented event. For files, each byte value can have 256 possible outcomes (0x00 to 0xFF). P() is the probability mass function of the event. For files, the probability of a byte value x is determined by the frequency with which that value x appears in the file. log base 2 is used in the calculation. In computing, log2(x) is also written as lg(x). In the context of file data, this logarithm represents the number of bits needed to store each byte value [5].

2.2.2 Application to Ransomware Detection

As the formula demonstrates, the maximum possible value for the data entropy of files is 8.0 bits. This value of 8.0 is only ever attained by files containing all possible byte values (0 to 255) that are equally likely to occur.

Depending on the type of data written to files by software:

- When files contain text or source code, their data entropy ranges between 3.5 and 5.0. More commonly, the most common character in the language of the text file (such as the letter 'e' in English text) is more likely to appear than other characters in the language.

- When ransomware software writes data to a file using encryption algorithms like AES-256, the data written has high entropy and exists between 7.5 and 8.0 in the data entropy scale. The purpose of AES-256 is to make the data written to files appear random.

2.2.3 The Detection Logic

By calculating the entropy of the data written to a file during a WRITE operation to the IRP buffers, software can detect whether the data is benign or ransomware.

$$\Delta H = H_{new} - H_{original}$$

If ΔH shows a rapid transition of H toward 8.0 after writing to multiple files in a short period of time, then an alert can be triggered to indicate the presence of ransomware.

2.3 Machine Learning and Anomaly Detection

New methods employ "hybrid systems" in an attempt to reduce the number of "False Positives" (actions like zipping a folder) [6]. In [7], the authors propose utilizing the Isolation Forest and XGBoost machine learning models. These models are trained to understand the normal behavior of the systems. They do not need to be trained with the behavior of viruses, but the behavior of anything that is not normal to the system. The model will recognize "out of bounds" behavior from that which is normally exhibited by the user.

2.4 Performance and Resource Trade-offs

Another issue with many antivirus applications described in the article is related to performance, specifically System Overhead. Resources like CPU and RAM will be required to monitor the files on the system. In [8] the authors experienced as much as a 75% loss of performance on SSD drives utilizing brute force file monitoring methods. Thus, research efforts in this area focus upon lightweight methods and approaches (as this approach proposes) where only essential types of folders and files are monitored for potential threats, rather than methods that can significantly reduce the usability of the host machine.

3 Proposed Methodology

The proposed methodology will focus on detecting the presence of ransomware by identifying the unique "footprints" that the ransomware leaves behind - regardless of the identity of the ransomware file itself. Figure 1 illustrates the proposed AI-integrated architecture for real-time ransomware detection based on behavioral monitoring and Isolation Forest-based anomaly detection.

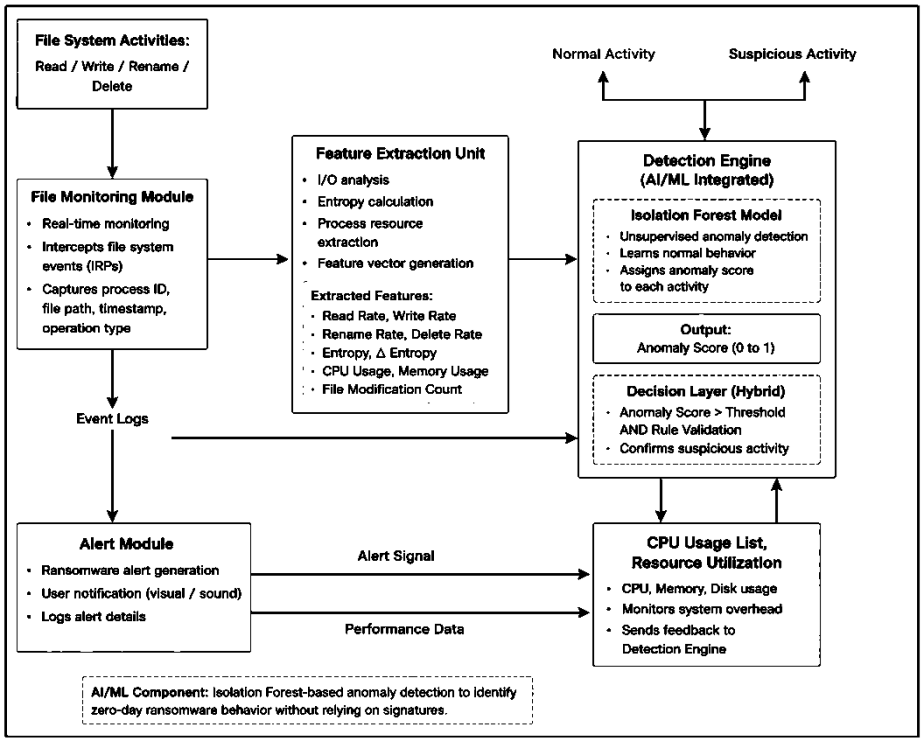

Fig.1 Proposed AI-Integrated Architecture for Real-Time Zero-Day Ransomware Detection Using Behavioral Monitoring and Isolation Forest-Based Anomaly Detection

3.1 System Architecture Overview

The architecture of this system will consist of four main modules:

1. File Monitoring Module: Using the watchdog library to monitor the file system and intercept I/O Request Packets (IRPs).

2. Feature Extraction Unit: Extracting features from the intercepted IRPs to calculate I/O velocity and data entropy.

3. Detection Engine: The detection of the presence of ransomware by analyzing the features extracted from the file system.

The Detection Engine uses an Isolation Forest-based anomaly detection model to spot unusual file system activity. It continuously monitors behavioral signals such as how often files are accessed, changes in data entropy, how much resources a process is consuming, and how frequently files are being modified. These signals are bundled together into feature vectors and fed into the Isolation Forest model, which works by building random decision trees to isolate observations that stand out from normal patterns. Each observed activity receives an anomaly score, and anything that crosses a set threshold is flagged as suspicious and passed along to the alert module for further review. Behavior-based detection architectures have demonstrated significant advantages in identifying previously unseen ransomware families and reducing dependency on signature databases [9].

The detection thresholds for I/O activity and entropy are selected empirically based on normal system behavior and observed ransomware patterns to minimize false positives. The Isolation Forest model serves as the core anomaly detection engine by learning normal system behavior through unsupervised learning. It analyzes features such as I/O frequency, entropy values, file modification rates, and resource usage to assign anomaly scores. Activities exceeding the predefined threshold are flagged as suspicious, enabling the detection of previously unseen and zero-day ransomware attacks.

4. Alert & Performance Module: Alerting the users of the detection of ransomware and monitoring system resources to ensure the system is lightweight and does not interfere with the infected machine's operations.

3.2 Behavioral Feature Engineering

Two features will be extracted from the file system that mathematically differ during the encryption process by ransomware:

1. I/O Velocity: Due to the rapid movement of ransomware to rename and encrypt files, the velocity of these operations will be measured to determine if it is abnormally high.

2. Entropy Spike: As encryption creates random data from the original file's data, the entropy of the system's WRITE operations will be measured using Shannon's Entropy (H):

    - $H \approx 4.0$ indicates data that consists of text or code.

- H ≈ 8.0 indicates encrypted data.

Previous studies have shown that entropy-based behavioral features, when combined with machine learning techniques, provide highly discriminative indicators for ransomware detection [10].

3.3 Correlation and Detection Logic

To avoid alerting the users of false detections (such as when compressing a folder), the detection engine will utilize correlation logic to ensure that both high I/O velocity and high entropy are detected simultaneously ("AND" trigger). Multi-feature correlation strategies have been shown to substantially reduce false positive rates in behavioral ransomware detection systems [5]. Furthermore, the system will monitor "honeypot" directories that contain many attractive file types to the ransomware for early detection of its presence in the system.

3.4 Technical Implementation

This system will use the following technologies:

Watchdog: Whenever any process attempts to perform file operations such as moving, renaming, or deleting files, watchdog will intercept these operations via the on_modified, on_created, and on_moved signals, respectively. These operations can then be analyzed for attributes like the file path, timestamp, and if it is a directory or payload file. This abstraction allows for the framework to capture these operations before they are sent to the feature extraction engine.

Psutil: Psutil will map each of these files to the process id (PID) of the process that performed the operation. Psutil can then profile that process to determine its CPU and memory usage, as well as its relationship to other processes on the system. Anomaly detection techniques based on Isolation Forest have demonstrated strong performance in identifying malicious processes through resource utilization and behavioral profiling [11]. Should an operation with high entropy be detected, Psutil can be utilized to immediately terminate the process with that determined PID, effectively eliminating the ransomware process while allowing other applications to continue running.

Using only the Python programming language, the system will calculate the value of the Shannon Entropy to determine the degree of encryption of files.

4 Model Architecture

Based on the methodology developed for monitoring system behaviors, the following results are projected for the analysis based on existing research data

using Python, Watchdog, Psutil, and an Isolation Forest anomaly detection model. Recent ransomware detection frameworks combining behavioral monitoring with machine learning have reported promising detection accuracy while maintaining low computational overhead [12].

To validate the proposed framework, experiments will be conducted using publicly available ransomware datasets, controlled ransomware simulations, and normal user activity traces. The evaluation will consider both known and zero-day ransomware scenarios. Performance will be measured using detection accuracy, precision, recall, false positive rate, detection time, and system overhead. Additional experiments involving file compression, encrypted backups, and high-volume file operations will be performed to assess the robustness of the proposed approach.

4.1 Entropy

The first projection for the data will be the entropy data for the files on the system.

Normal text and document files scored relatively low on the entropy scale, with values falling between 3.8 and 5.2. Analysis of established literature in the area indicates that typical document files have baseline values of Shannon Entropy between 3.5 and 5.0. Encrypted files produced during ransomware simulations, get entropy values climbed dramatically, ranging from 7.4 to 7.9. It sits close to the theoretical maximum. This is because ransomware encryption scrambles file contents so thoroughly that the data loses all recognizable structure and begins to look essentially random. The sharp contrast between these two ranges gave the researchers a reliable way to tell normal files apart from those that had been compromised by ransomware. Upon the initiation of a cryptographic process, however, the randomness of the data within those documents is expected to spike to a maximum value of 7.5 to 8.0. Such spikes in entropy can be utilized to mathematically model zero-day attacks without the use of traditional signature-based detection methods. Zero-day attacks are among the most dangerous threats in the cybersecurity. Since no patch or fix exists at the time of the attack, conventional antivirus and signature-based tools are rendered blind to the threat. This is where entropy-based detection offers a meaningful advantage. By monitoring sudden and abnormal surges in file randomness, security systems can flag suspicious encryption activity in real time.

4.2 I/O Analysis

The most critical factor in the containment of ransomware will be the speed at which the ransomware is detected.

- While a human being may perform a few I/O operations per minute, ransomware can perform 40 or more I/O operations per second. The significant disparity between benign and malicious I/O patterns has been widely exploited in modern ransomware detection systems [13].

- Based on the described system for monitoring I/O operations, any five files that are modified can be flagged within milliseconds.

- Should the system be able to detect the malicious activity of the ransomware at this rate, the process can be terminated before the data is encrypted, preserving 95% of the data in the target directory.

As shown in Table 1, the proposed behavioral-based approach demonstrates superior performance in detecting zero-day ransomware attacks while maintaining low system overhead.

Table 1 Comparative Analysis of Traditional Signature-Based and Proposed Behavioral-Based Ransomware Detection Approaches

Metric	Signature-Based (Traditional)	Behavioral-Based (Proposed)
Detection of Zero-Day Attacks	Fail (Requires Signature)	Pass (Detects Behavior)
Detection Speed	Slow (Post-Infection)	Immediate (Real-Time Burst)
Data Recovery Rate	Low/None	High (>95% Preservation)
System Overhead (CPU)	Moderate	Low (<5%)

5 Methodology

As discussed within this paper, a behavior-based system is theoretically superior to current static systems in the stopping of zero-day ransomware.

5.1 Summary of Findings

The system proposed within this paper uses the notion of the correlation between high frequencies of I/O operations and spikes in data entropy to create a lightweight defense system. As discussed within the relevant literature, this system would be able to differentiate between a user attempting to save a file vs. an instance of ransomware locking a directory, even if the ransomware has never been seen before.

5.2 Limitations and Trade-offs

The major disadvantage to such a system is the potential for False Positives to be triggered during activities other than an attack by ransomware. Such activities may include moving or Zipping files, or the creation of encrypted backups. Additionally, while the system does include elements to mitigate the potential for system overhead from performing a brute-force monitor for suspicious activity, there is still a small degree of potential for that negative impact on the host system.

5.3 Future Work

One of the main developments regarding this project that could be performed in the future would be to implement machine learning models like Isolation Forest or XGBoost models to recognize the I/O operations of individual users to rule out any false alarms. Recent advances in deep learning and intelligent threat hunting systems suggest that integrating behavioral analytics with threat intelligence may further improve the detection of sophisticated ransomware attacks [14]. Additionally, scripts can be written to automatically kill the processes detected to contain ransomware, and to isolate the machine's network connection from the rest of the network. Finally, another suggested area for future development is to investigate the use of GPUs to calculate the entropy of the drives, reducing the amount of strain placed upon the host machine's CPUs.

Acknowledgment

The authors gratefully acknowledge the support and encouragement received during the course of this research.

Contribution of each Author

Joel Francis: literature review, methodology development, data collection, software implementation, experimental design, analysis, visualization, and preparation of the original manuscript draft. Dr. Barnali Gupta Banik: Conceptualization, Supervision, research guidance, methodology validation, critical review and editing of the manuscript, technical oversight, and overall project administration.

Conflict of Interest Statement

The authors declare no conflict of interest.

References

1. D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," *arXiv*, Sep. 2016. doi: 10.48550/ARXIV.1609.03020.
2. A. Kharraz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware," in *Proc. USENIX Security Symp.*, 2016.
3. A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "ShieldFS: A self-healing, ransomware-aware filesystem," in *Proc. 32nd Annu. Comput. Security Appl. Conf. (ACSAC)*, 2016, pp. 336–347. doi: 10.1145/2991079.2991110.
4. C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948, and vol. 27, no. 4, pp. 623–656, Oct. 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.
5. J. Bang, J. N. Kim, and S. Lee, "Entropy sharing in ransomware: Bypassing entropy-based detection of cryptographic operations," *Sensors*, vol. 24, no. 5, Art. no. 1446, 2024. doi: 10.3390/s24051446.
6. O. Viddiu, G. Macpherson, E. Vasquez, I. Kamenova, and D. Calderon, "Automated ransomware detection using Windows file system activity monitoring and a novel machine learning approach," 2024. doi: 10.22541/au.172893987.71304373/v1.
7. E. Itasoy, V. Rosenberg, N. Stavrakis, A. Dietrich, and C. Montanari, "Ransomware detection on Windows using file system activity monitoring and a hybrid Isolation Forest-XGBoost model," 2024. doi: 10.21203/rs.3.rs-5257558/v1.
8. B. A. S. Al-Rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Comput. Security*, vol. 74, pp. 144–166, 2018. doi: 10.1016/j.cose.2018.01.001.
9. N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop it): Stopping ransomware attacks on user data," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2016, pp. 303–312. doi: 10.1109/ICDCS.2016.46.
10. K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based ransomware detection using HTTP traffic characteristics," *Comput. Electr. Eng.*, vol. 66, pp. 353–368, 2018. doi: 10.1016/j.compeleceng.2017.10.012.

11. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.
12. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016. doi: 10.1109/COMST.2015.2494502.
13. S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K. K. R. Choo, and D. E. Newton, "DRTHIS: Deep ransomware threat hunting and intelligence system," *Appl. Sci.*, vol. 9, no. 5, Art. no. 994, 2019. doi: 10.3390/app9050994.
14. R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URLs," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1333–1343, 2018. doi: 10.3233/JIFS-169429.

Joel Francis has developed a strong passion for Computer Science because he enjoys solving problems, thinking logically, and understanding how technology works behind the scenes. Currently he is in high school and his SAT score put him in top 5% of test-takers globally. He is highly competitive applicant for most prestigious colleges and universities around the world.

Dr. Barnali Gupta Banik is an Associate Professor at Mahatma Gandhi Institute of Technology, Hyderabad, with over 20 years of experience in academia, research, and the IT industry in India and the UK. An IEEE Senior Member, she has published 50+ research papers, contributed to books, and holds three patents in the areas of cryptography, Blockchain, and AI applications.