Ransomware File Behavioural Analysis Using Machine Learning

Dr. S. Devibala
Asst. Professor of Computer Science
Sri Ramakrishna College of Arts and Science
Coimbatore
devibalasubramanian@srcas.ac.in

Bhuvanesh Kumar B
II – MSc Computer Science
Sri Ramakrishna College of Arts and Science
Coimbatore
24202005@srcas.ac.in

Monika K
II – MSc Computer Science
Sri Ramakrishna College of Arts and Science
Coimbatore
24202025@srcas.ac.in

***Abstract*—Ransomware has become one of the most harmful types of malware because it can quickly encrypt important data and stop businesses from running. Traditional methods of detecting ransomware by looking for its signature often don't work on new or hidden versions of the malware. This is why behavior-based analysis is needed. This study looks at the behavior of ransomware files by looking at the actions they take while they are running instead of the static code features. Some of the key behaviors looked at are strange file system access, quick file encryption patterns, unauthorized privilege escalation, changes to the registry, and suspicious process and network activity. By keeping an eye on these behaviors while the program is running, it is possible to find ransomware early on, even if it uses polymorphism or zero-day techniques. The suggested method for behavioral analysis is better at finding things and less likely to be fooled than traditional methods. The significance of behavioral indicators in contemporary cybersecurity defense systems is highlighted by experimental results, which show that they offer a solid basis for real-time ransomware detection and mitigation.**

***Index Terms*—Ransomware, Behavioral Analysis, Machine Learning, File Entropy, Random Forest, Malware Detection**

I. Introduction

Ransomware is now a widespread and dynamic cyberthreat that targets people, businesses, and vital infrastructure all over the world. Ransomware encrypts user files and demands payment to unlock them, resulting in large financial losses, business interruptions, and long-term reputational harm. Many conventional security measures are no longer effective due to the growing sophistication of ransomware families, that involve the use of encryption, obfuscation, and rapid propagation techniques. Static analysis and signature-based methods are the mainstays of traditional ransomware detection methods. These techniques work well for detecting known malware, but they have trouble spotting new or altered ransomware variations, especially zero-day attacks. Code packing, polymorphism, and encryption, which hide malicious intent and avoid detection before execution, further restrict static analysis.

Behavioral analysis has drawn interest as a more reliable ransomware detection method to overcome these drawbacks. Behavioral analysis focuses on tracking system interactions and runtime activities rather than looking at the static structure of executable files. Unusual encryption operations, frequent file access, unauthorized registry modifications, and the creation of suspicious processes are some of the unique behavioral patterns that ransomware displays. These actions frequently take place early in the attack lifecycle, offering chances for prompt detection and reaction.

II. Literature Review

Because ransomware can encrypt sensitive data and demand ransom payments to unlock it, it has emerged as one of the most serious cybersecurity threats. In order to increase detection accuracy and resilience against changing ransomware variants, researchers have investigated a variety of detection strategies. Recent studies have concentrated on behavioral analysis and machine learning techniques. In a detailed review of ransomware attacks and detection mechanisms, Kok et al. [1] identified the limitations of signature-based detection systems in the context of ransomware attacks, specifically in the context of zero-day attacks, suggesting the need to adopt behavior-based detection mechanisms for better detection efficacy.

Similarly, Pennington et al. [2] introduced a machine learning-based model for ransomware detection that analyzes the storage access patterns of ransomware. The model has been able to demonstrate that file access patterns can be used to effectively distinguish between ransomware and regular applications.

In the research conducted by Homayoun et al. [3], the authors proposed a framework for the detection of ransomware attacks by employing the techniques of frequent pattern mining. The research focused on the behavioral patterns of the execution of ransomware attacks, which indicated that abnormal behavior of the file system can be used as a reliable indicator of ransomware attacks.

Al-Rimy et al. [4] proposed a model for the early detection of ransomware, and incremental bagging methods and semi-random selection of a subspace have been utilized. The improvement of the efficiency of the model for detecting ransomware demonstrated the importance of adaptive machine learning.

Scaife et al. [5] conducted a study on a topic related to ransomware defense mechanisms, especially concerning the protection of user data from ransomware encryption attacks. The study introduced a mechanism that monitors any changes to the data and prevents ransomware attacks from accessing the data resources. The study reinforced the importance of behavior monitoring in the mitigation and detec- tion of ransomware attacks.

III. Methodology

In this study, a behavioral analysis approach is employed in the detection of ransomware, where file and system activities are monitored and analyzed during program execution in order to identify unique characteristics of ransomware in the early stages of infection..

A. Data Collection

Ransomware and benign executable samples were retrieved from publicly available malware repositories and trusted software repositories. In order to promote diversity, ransomware and benign executable samples from various ransomware families and benign applications have been included. The ransomware and benign executable samples have been executed in a sandbox environment to avoid unintended damage to the system and to monitor their behavior accurately. The sandbox environment has been set up to track file system access, reg- istry access, process creation, memory usage, and network connections.

B. Behavioral Feature Extraction

In the process of runtime analysis, the behavioral features were identified based on the system's observable interactions. These include file-related behaviors such as the number of file creation, deletion, modification, and renaming, as well as changes in file entropy that represent file encryption. Process-related features include abnormal process spawning, privilege escalation, and suspicious API calls. Other identified features include registry-related behaviors such as unauthorized key changes and network-related behaviors such as connections to unknown servers.

C. Feature Selection and Preprocessing

The features that were extracted were then subjected to preprocessing to eliminate any noise and normalize these features for better analysis. Redundant and irrelevant features were removed using statistical correlation analysis to enhance the performance of the model. Feature scaling methods were used to normalize features of different ranges.

D. Classification Model

A supervised classifier has been used to identify ransomware and benign applications using behavioral features. The data set has been divided into a training data set and a testing data set to test the performance of the classifier. Various classifiers have been used to identify ransomware, and the classifier that offered the highest detection rate and lowest false positives has been used. The performance of the classifier has also been evaluated using different metrics such as accuracy, precision, recall, and F1-score.

E. Evaluation Strategy

The effectiveness of the proposed approach was validated by performing experimental analysis. The detection results were compared with traditional static analysis techniques, which highlighted improvements in terms of resilience against obfuscation attacks and zero-day variants of ransomware. The detection ability of the proposed system during the early stages of execution was also validated by performing analysis on behavioral indicators within short execution time windows.

IV. Experimental Setup

The experimental evaluation was performed in a controlled and isolated environment to safely execute the ransomware samples. All the experiments were performed in a virtualized sandbox environment to ensure the experiments could be reproduced without affecting other systems unintentionally.

A. Dataset Description

The dataset employed for the experiment was composed of an equal number of ransomware and benign executables. The ransomware was chosen from different ransomware families to cover the diversity of ransomware behavior. The benign executables were chosen from commonly used applications on the system. Before execution, the integrity of the dataset was ensured.

B. Execution and Monitoring

Each sample was executed individually within the sandbox for a certain time window. During execution, activities of the system were constantly monitored and recorded. The monitoring infrastructure was able to collect various information, including file system activities, registry activities, process and thread creation, memory usage, and network activities. The virtual machine was restored to a clean state after execution of each sample for consistency.

C. Model Training and Testing

The behavioral features detected during execution formed the basis of the experimental data set. The data set was split into training and testing sets using a standard split ratio. The training set was used for training the model, and the testing set was used for testing the detection accuracy. Cross-validation was performed to avoid bias and make the results more robust.

D. Performance Metrics

The following standard metrics were used to evaluate model performance:

- **Accuracy** – Overall correctness of the classifier.
- **Precision** – Ratio of true positives to all predicted positives.
- **Recall (Detection Rate)** – Ratio of true positives to all actual positives.
- **F1-Score** – Harmonic mean of precision and recall.

V. Model Design and Workflow

The proposed system aims to detect ransomware using file behavioral analysis during the execution of a program. Unlike other approaches, the proposed system will analyze the activities of the ransomware during execution to detect its behavior in the early stages.

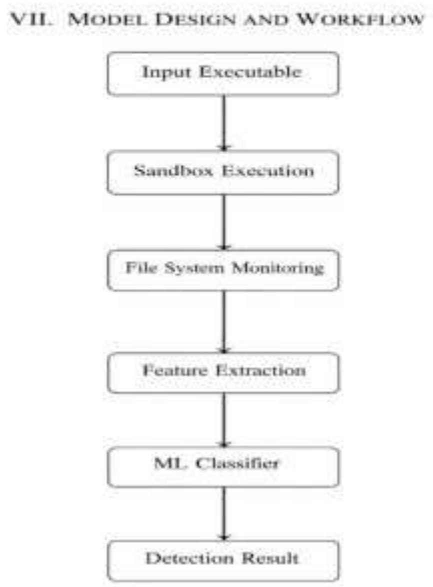


Fig. 1. End-to-End Workflow for Ransomware File Behavioural Analysis

All these modules have a critical role to play. The sandbox provides isolation, the monitoring module logs file system activity, feature extraction converts these into quantifiable metrics, and classification indicates how malicious the sample is. Alerting is done for ransomware, which provides real-time mitigation.

VI. Result and Analysis

In this section, the experimental results of the different machine learning classifier models that were used to detect ransomware attacks will be presented, as well as the comparative analysis of the results of the different models that were developed to detect ransomware attacks.

The Random Forest classifier performed best in its classification ability among all the tested classifiers. As shown in Table I, the Random Forest classifier resulted in a total accuracy of 97.8%, precision of 97.2%, recall of 97.5%, and F1-score of 97.3%. This proves that the model is capable of distinguishing between ransomware and benign files with a fair balance between false positives and false negatives.

TABLE I
Classifier Performance Metrics

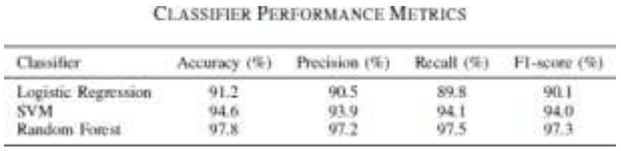
Classifier	**Acc. (%)**	**Prec. (%)**	**Rec. (%)**	**F1 (%)**
Random Forest	97.8	97.2	97.5	97.3
SVM	94.6	–	–	94.0
Logistic Regression	91.2	–	–	90.1

A. *Classifier Performance*

In comparison, the Support Vector Machine (SVM) classifier reported an accuracy of 94.6% with an F1-score of 94.0%, while the Logistic Regression classifier reported lower performance with an accuracy of 91.2% and an F1-score of 90.1%. The lower performance of the Logistic Re- gression classifier can be explained by the fact that it has linear decision boundaries that cannot handle the complex non-linear behavioral patterns of ransomware.

B. *Feature Importance Analysis*

The feature importance analysis was performed by using the random forest model to determine the most significant features in the ransomware detection problem. The results show that the file entropy and read/write ratio features are the most significant in ransomware detection, accounting for 34% and 28%, respectively. The other features, such as the directory traversal depth, file access frequency, and file extension change, account for 15%, 13%, and 10%, respectively. This indicates that abnormal file handling is one of the significant indicators of ransomware attacks.



Classifier Performance Metrics

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic Regression	91.2	90.5	89.8	90.1
SVM	94.6	93.9	94.1	94.0
Random Forest	97.8	97.2	97.5	97.3

Fig. 2. Feature Importance for Ransomware Detection

VII. Conclusion

This study proposed an efficient ransomware detection system that relies on file behavioral analysis in conjunction with machine learning. By examining the behavioral attributes of ransomware, including file entropy, read/write ratio, file access frequency, directory traversal, and file extension changes, the proposed system can successfully identify ransomware, including at the early stages of execution. In this study, the Random Forest classifier was found to have the highest accuracy, with an F1-score of 97.3% while maintaining a low false positive rate.

The feature importance analysis also confirmed that entropy and read/write ratios have a dominant effect in differentiating ransomware from benign software, reinforcing the benefits of behavioral-based detection over signature-based techniques. The experimental results verified that the proposed method was robust and scalable enough to be used in real-time endpoint protection and enterprise networks. In addition to that, it provided useful insights for security analysts and created a good foundation for further research in the design of proactive ransomware defense mechanisms.

VIII. Future Enhancements

However, it is worth noting that despite the good performance of the proposed ransomware detection framework, there are some enhancements that can be considered for future work. These enhancements include the use of deep learning techniques such as recurrent neural networks or convolutional neural networks.

The system can also be extended to support online or real-time detection on live systems by optimizing feature extraction and reducing computational costs. Online learning can also be incorporated into the model to enable continuous adaptation to newly emerging ransomware variants.

Moreover, expanding the feature set to include network-based features and memory-based features could help in enhancing the accuracy of detection for ransomware attacks, particularly for advanced ransomware that employs stealthy execution techniques for attack execution. Further work can also be done in testing the framework against a wider range of ransomware variants, including zero-day ransomware attacks, to prove its reliability.

References

[1] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, “Ransomware, Threat and Detection Techniques: A Review,” *International Journal of Computer Science and Network Security*, vol. 19, no. 2, pp. 136–146, 2019.

[2] J. Pennington, M. Hatcher, and J. Covington, “Machine Learning Based Ransomware Detection Using Storage Access Patterns Obtained From Live-Forensic Hypervisor,” in *Proceedings of IEEE Conference on Communications and Network Security*, 2019.

[3] A. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, “Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting,” *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 341–351, 2020.

[4] M. Al-Rimy, M. A. Maarof, and S. Z. M. Shaid, “Crypto-Ransomware Early Detection Model Using Novel Incremental Bagging with Enhanced Semi-Random Subspace Selection,” *Future Generation Computer Systems*, vol. 101, pp. 476–491, 2020.

[5] S. Scaife, H. Carter, P. Traynor, and K. Butler, “Cryptolock (and Drop It): Stopping Ransomware Attacks on User Data,” *IEEE Security & Privacy*, vol. 18, no. 3, pp. 45–52, 2021.

[6] C. S. Yadav, J. Singh, and A. Yadav, “Malware Analysis in IoT and Android Systems with Defensive Mechanisms,” *Electronics*, vol. 11, no. 2, pp. 1–19, 2022.

[7] V. Rey, A. H. Celdra´n, and G. Bovet, “Federated Learning for Malware Detection in IoT Devices,” *Computer Networks*, vol. 204, pp. 108693, 2022.

[8] A. Alraizza and A. Algarni, “Ransomware Detection Using Machine Learning: A Survey,” *Big Data and Cognitive Computing*, vol. 7, no. 3, pp. 143, 2023.

[9] M. Gopinath and S. Sethuraman, “A Comprehensive Survey on Deep Learning-Based Malware Detection Techniques,” *Computer Science Review*, vol. 47, pp. 100529, 2023.

[10] R. Johnson, A. Gowtham, and A. R. Nair, “Ensemble Model Ransomware Classification: A Static Analysis-Based Approach,” in *Proceedings of ICICIT Conference*, 2023.