

A Layered Framework for Ransomware Detection and Automated Response

Ms. Bhoomika Naik¹, Ms. Amulya B S², Ms. H P Deeksha³, Ms. H Suvarna⁴, Prof. Chithrakshi⁵
^{1,2,3,4,5}*Department of Computer Science and Engineering, Shree Devi Institute of Technology*

Abstract—Ransomware represents one of the most dangerous phenomena in the space of cyber security where the main idea implies the damage of the normal functioning of files or computer data resources. Given the explosive increase of ransomware families, it is no longer possible to rely only on previously recorded signatures in order to ensure efficient detection and prevention mechanisms. The proposed project deals with the development of a multi-layer ransomware attack detecting and preventing system that is based on the following core principles – static analysis, entropy, PE inspection, YARA scan, machine-learning classification, honeypot or tripwire monitoring, alert design and response. All suspicious files and activities are analyzed using a combination of indicators of compromise. The system is able to detect ransomware using the use of static indicators, entropy analysis in search of high-randomness content, rule-based scanning with YARA patterns in search of known malicious constructs or behaviors, machine-learning classification of suspicious files based on extracted features, behavioral detection with honeypot or tripwire monitoring of sensitive resources. All of this contributes to efficient detection and response to ransomware attacks.

Index Terms—Ransomware, malware detection, machine learning, entropy, YARA, PE analysis, honeypot, tripwire, cyber security, attack prevention.

I. INTRODUCTION

Computer systems, servers, shared storage, and other systems are critical elements for the proper functioning of almost any modern enterprise [1], [2]. Thus, the ransomware attack on such systems can cause severe disturbances to the business due to data being inaccessible [1], [2]. Meanwhile, the ransomware attack can occur in various forms, and the attacker can use different methods to disguise the malicious code [1], [4].

In general, a ransomware attack is when the criminal's software takes over control of the computer system, disrupting its operation [1]. Several common patterns can help identify such an attack, for example, files' encryption, which is the most common one, processes' appearance, files' encryption using special extensions or attributes, cryptographic functions, the use of non-specific paths, or other signs [1], [4]. These patterns can be used as the basis for a program that would automatically detect ransomware attacks [1], [3]. The majority of commercial anti-virus software traditionally utilize signature-based detection, where certain patterns which appear in malicious files are compared to a database of known signatures [1]. Signature based detections are simple and efficient for detecting known threats, as there is a clear indicator of malicious intent [1]. A major disadvantage is that such detections cannot be made towards previously unknown ransomware, as well as code which was deliberately modified to avoid detection [1], [4]. Multiple techniques can be used in combination to achieve better results in ransomware detection and prevention [3], [4], [6].

Static analysis allows for extracting valuable information about a file without executing it [4]. Information such as imported libraries, file structure, presence of strings, permissions, and other attributes can be used to determine if a file warranted closer inspection [4].

Static analysis is especially useful for inspecting suspicious files because it does not require executing them, thus preventing malicious code from running in the process [4].

Another simple detection method is entropy analysis that helps to identify whether a file has been encrypted or not [7]. The level of entropy indicates how much randomness there is in a file's data sequence [7]. Basically, it can be applied to determine if some

encryption or packing algorithms have been used in the creation of the file [7]. In practice, it should not be relied on for detecting ransomware exclusively since files with high entropy can be legitimate too, especially those having comparable compression levels [7]. Machine learning is another efficient method for detecting ransomware [5], [9]. In general, machine learning algorithms can be used to identify distinguishing traits of different file categories based on specific features [5], [9]. For example, one algorithm can be trained to distinguish between normal and malicious files using a sufficient amount of data with corresponding examples [5]. Then, the algorithm will be able to analyze any new file and separate it into a malignant or benign category [5], [9]. It is worth noting that machine learning can be employed as one of the many steps in the overall ransomware attack detection process, which also includes rule-based detection methods and behavioral inspection of suspicious files [3], [5], [6].

The proposed project implements a layered ransomware attack detection and prevention system, which makes use of the following elements: machine learning, static analysis, entropy, rule-based detection with YARA, behavioral inspection with honeypot or tripwire, alerting, logging and response [3], [4], [6], [7], [8]. Basically, the system is designed to inspect suspicious files from multiple angles and collect evidence that can be used to confirm a ransomware attack [3], [4], [8].

The stated purpose of the project is to implement a system, which detects ransomware attacks by analyzing suspicious files [2], [3], [5], [6]. Specifically, the project involves analyzing suspicious executable files, extracting relevant information for further inspection, inspecting files with YARA to detect malicious patterns, using machine learning to classify files, inspecting the behavior of suspicious processes with honeypot or tripwire to identify unauthorized attempts to access protected resources, generating alerts, collecting relevant logs, and implementing a response, which helps mitigate the effects of an ongoing attack [2], [3], [5], [6], [8].

II. LITERATURE REVIEW

The existing research literature explores various techniques and ideas for detecting and preventing ransomware attacks [3], [4], [5], [6], [8]. In particular,

methods such as signature-based detection, static and dynamic analysis, machine learning classification, deep learning approaches, deception-based prevention, and others were proposed [3], [4], [5], [6], [8]. The most promising ideas and technologies were combined in the current project to develop a complete ransomware attack detection and prevention solution [3], [4], [5], [6], [8].

Signature-based malware detection refers to comparing the contents of a file with known patterns detected in malware [1]. Overall, this method is typically simple and efficient since it only requires an exact match to be detected [1]. The main challenge with signature-based malware detection is the ability to stop unknown malware, including polymorphic ones, since they can significantly change their code to avoid existing signatures [1], [4].

Static malware analysis aims to detect suspicious patterns in the content of the executable file [4]. Particular file header, imported libraries, section names and sizes, file strings, permissions, and other characteristics can be used to identify potential threats [4]. On the whole, it is typically safe to perform a static analysis of an unknown file to retrieve general information about the program and its potential risks [4].

Ransomware behavioral detection focuses on the specific behavior of a process initiated by malware [3], [4]. In particular, the key aspects include rapid file modification, frequent access to suspicious file extensions, excessive file creation, interaction with other processes, and others [3]. Behavioral analysis typically involves monitoring a process initiated by a suspicious file [3], [4]. However, in some cases, particular behaviors can be associated with ransomware, allowing detection without initiating an executable [3], [4].

Machine learning can be used to train a model that can distinguish between normal and malicious files, thus detecting ransomware [5], [9]. In particular, a machine learning algorithm can be trained with examples of files that include ransomware and those that do not [5]. Typically, such algorithms use various features extracted from files for processing [5], [9]. The efficiency of machine learning algorithms can be estimated using precision, recall, F-score, and other metrics [5], [9]. It is essential to note that the effectiveness of machine learning algorithms depends highly on the quality of data used to train them [5].

Entropy can be used when analyzing the content of a file to estimate the likelihood of ransomware attack [7]. In particular, encrypted data typically has a higher entropy, thus being easily identified [7]. However, it is essential to note that entropy alone is not enough to identify ransomware since other data can also have a high entropy value due to compression [7].

YARA-based detection involves using regular expressions to mark files that include certain patterns as malicious [8]. Particular patterns can be used to detect known ransomware variants by scanning files with YARA rules [8]. In most cases, the rules define specific strings that can appear in a file, but regular expressions can also be used [8]. Both individual strings and complex combinations of different elements can be used for detection [8]. The use of YARA rules alone is not enough for detecting ransomware [8]. Hence, it is typically used as a part of the broader detection system [8].

Finally, honeypots and tripwire can be used to monitor unauthorized access to protected data, which can be used to infer potential ransomware attack attempts [3]. In general, honeypots attract attackers by simulating vulnerable resources that can be compromised [3]. A tripwire can monitor certain files or other data and is triggered by any suspicious activity [3]. This means that it can be used to monitor certain files for unauthorized changes, which indicates that a ransomware attack is happening [3].

In general, the reviewed research literature suggests that signature-based detection, rule-based pattern matching, machine learning classification, entropy analysis, behavioral analysis, and other techniques can be used for ransomware detection [3], [4], [5], [7], [8], [9]. However, in most cases, a combination of methods is required to achieve satisfactory results [3], [4], [6]. In isolation, each of the methods described is inefficient due to the complexity of the ransomware attack and the constant development and improvement of malicious software [3], [4], [5].

III. METHODOLOGY

The method of the described solution may be divided into several distinct stages: data gathering and input, preprocessing and feature extraction, entropy calculation, PE and YARA analysis, machine-learning classification, honeypot and tripwire surveillance, event logging and alerting, and preventive and

responsive measures. In general, the system is designed to process inputs and analyze them with different algorithms in order to detect ransomware attack indicators.

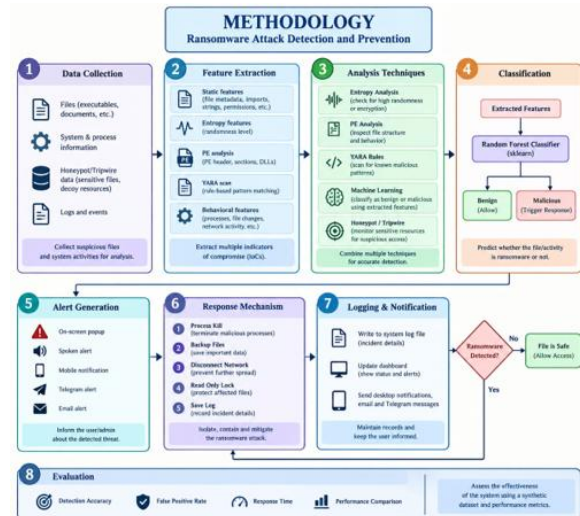


Fig 1: Methodology

1. Data Collection and Input

The first step involves collecting the data and input files that will be used for analysis. It is important to note that the input can contain both benign and malicious files. The collected data can be used for static feature extraction and further processing.

2. Preprocessing and Feature Extraction

The input file is subjected to a series of analyses that provide relevant features for ransomware detection algorithms. The features are extracted and prepared in a manner that is conducive to further processing and classification.

3. Entropy Computation

The entropy value is computed for the file or for its sections, if applicable. The resulting value is relevant for further processing, but it is not sufficient for detecting ransomware on its own.

4. PE and YARA Analysis

The file is analyzed to look for malicious patterns and extract relevant information. Analysis techniques that can be used to detect ransomware include inspection of Portable Executable (PE) file headers and checking file content against known patterns via YARA rules.

5. Machine-Learning Classification

The features extracted from analysis are input into the machine-learning component, where the file is classified as either malicious or benign. The classification results can help the other components focus on suspicious files.

6. Honeypot and Tripwire Monitoring

Files are processed to monitor and detect unauthorized access. In practice, legitimate applications tend to access only a limited set of resources, and other activity can be used to detect ransomware attacks.

7. Alert Generation and Logging

Alerts are generated when appropriate, allowing the user to respond to a ransomware attack. The relevant information can be logged to help determine the nature and extent of the attack.

8. Prevention and Response

Prevention and response are implemented in accordance with the detected threat. In general, the process involves configuring rules that allow the system to respond to a ransomware attack by preventing further damage and ensuring that relevant information is available for review.

9. Evaluation

The performance of the machine-learning component can be estimated using precision, recall, F-score, and similar metrics. The performance of the system as a whole can be evaluated using detection time, false-positive rates and other relevant factors.

IV. IMPLEMENTATION

The ransomware detection system is implemented in the form of separate modules that can be combined together to perform multi-stage data analysis. In general, the input file is passed through these modules to extract information and detect ransomware attack indicators.

A. System Input and File Analysis

The system analyzes the input file in order to obtain the basic information of it. The obtained data can be used for further processing of the file with the remaining modules. Separately processing the input and the detection modules allows for easier maintenance and updates.

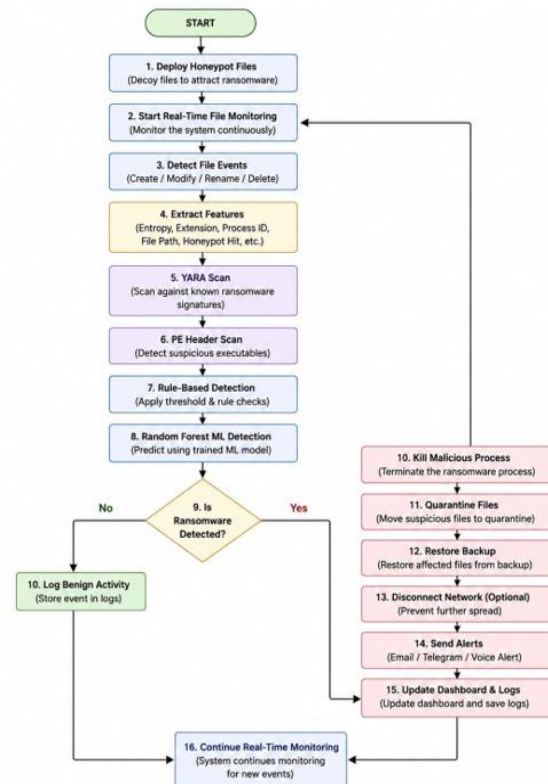


Fig 2: Implementation

B. Static Analysis

Static analysis extracts the relevant features without executing the input file. The information that can be extracted would include PE-specific elements and other relevant attributes that can be used for ransomware detection.

C. Entropy-Based Module

The entropy-based module estimates the entropy of the file. The resulting value is useful information that can be used for further processing, but it is not enough to confirm ransomware on its own.

D. Rule-Based Detection with YARA

The rule-based detection module checks the file against known patterns with YARA rules. Each rule represents a certain pattern that can be used to identify ransomware; the information described by the rules is useful for further processing.

E. PE Analysis

PE analysis helps extract relevant information for ransomware detection and classification.

F. Machine-Learning Classification

The machine-learning module utilizes the extracted features to classify the file.

G. Honeypot and Tripwire Module

The honeypot and tripwire module monitors the ransomware file and checks for unusual behavior. In general, such behavioral patterns can be indicative of a ransomware attack trying to modify unauthorized files.

H. Alert and Dashboard

The alert and dashboard module helps to monitor and visualize the results of processing the input file. In general, it presents the relevant information in a convenient manner so that it can be reviewed by the user.

I. Response Module

The response module helps to respond to a detected ransomware attack. In general, the module follows the response rules to address the threat.

J. Processing Flow and Output

The processing flow involves using the input file to extract features, estimate its entropy, analyze its content, check it against known patterns, classify it, monitor the behavior of suspicious files, and respond to the threat accordingly. In essence, the module processes the suspect file to detect ransomware and gather evidence about the attack.

V. CONCLUSION

The ransomware attack detecting and preventing system proposed in the current project involves a multi-layered approach, which thoroughly analyzes the suspicious files using various algorithms in order to identify ransomware attacks. Overall, the system includes such steps as applying static analysis and entropy-based processing, scanning the file with detection rules using YARA, utilizing machine learning classification, analyzing honeypot and tripwire behavioral patterns, and responding accordingly with alerts, logging, and other actions. The key benefit of the proposed approach is that it combines multiple techniques and methods in order to detect ransomware attacks from various aspects,

which increases the accuracy and reliability of processing.

The implemented system can be considered as a proof of concept for a ransomware detection application. It can be further developed by including more information and rules, which would allow improving its processing and detection capabilities. Specifically, the variables included in the machine-learning step could be extended in order to account for more data, which would further improve the accuracy of classification. Moreover, additional steps could be added in order to make the system more efficient at detecting and responding to ransomware attacks, including utilizing behavioral patterns of file execution. Finally, adding real-time monitoring procedures and making the alerting mechanism more effective and efficient would increase the efficiency of processing. Overall, the project serves as a great basis for further research and development of ransomware attack detection and prevention systems.

REFERENCES

- [1] S. Mohurle and M. Patil, "A brief study of WannaCry threat: Ransomware attack 2017," *International Journal of Advanced Research in Computer Science*, 2017.
- [2] A. K. Upadhyay, S. Gandhi, P. Dubey, and S. Jain, "Ransomware detection and data recovery," in *Proc. 2024 Int. Conf. Electrical Electronics and Computing Technologies (ICEECT)*, 2024, doi: 10.1109/ICEECT61758.2024.10738908.
- [3] D. M. R. Dileep, H. M. Naik, and S. D. Vidya Sagar, "Hybrid ransomware detection and prevention system using machine learning and behavioral analysis," in *Proc. 4th IEEE North Karnataka Subsection Flagship Int. Conf. (NKCon)*, 2025.
- [4] U. Urooj, B. A. S. Al-Rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware detection using the dynamic analysis and machine learning: A survey and research directions," *Applied Sciences*, 2021.
- [5] K. Kunku, A. Zaman, and K. Roy, "Ransomware detection and classification using machine learning," in *Proc. 2023 IEEE Symp. Series on Computational Intelligence (SSCI)*, 2023, doi: 10.1109/SSCI52147.2023.10371924.

- [6] M. J. H. Faruk, M. Masum, H. Shahriar, K. Qian, and D. Lo, “Authentic learning of machine learning to ransomware detection and prevention,” in *Proc. 2022 IEEE 46th Annual Computers, Software, and Applications Conf. (COMPSAC)*, pp. 442–443, 2022, doi: 10.1109/COMPSAC54236.2022.00081.
- [7] K. Lee, S.-Y. Lee, and K. Yim, “Machine learning-based file entropy analysis for ransomware detection in backup systems,” *IEEE Access*, vol. 7, pp. 110205–110215, 2019, doi: 10.1109/ACCESS.2019.2931136.
- [8] M. N. Alam, P. Agrawal, A. Singh, P. Dubey, M. Kumari, and A. Kumar, “Detection and prevention of malware and ransomware threats using malicious string analysis,” in *Proc. 2023 Int. Conf. Sustainable Emerging Innovations in Engineering and Technology (ICSEIET)*, 2023.
- [9] B. M. Khammas, “Ransomware detection using random forest technique,” *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020, doi: 10.1016/j.icte.2020.11.001.
- [10] A. V. Akchayaa, L. V. R., and P. Ramadoss, “AI-driven malware detection and prevention framework for mobile security,” in *Proc. 8th Int. Conf. Intelligent Sustainable Systems (ICISS)*, 2026.