Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Dynamic Ransomware Detection using Behavioral Analysis

Vikash Kumar
IILM University
Greater Noida, India

Yash Gupta
IILM University
Greater Noida, India

Yash Soren
IILM University
GreaterNoida,India

Vikash Kumar
IILM University
Greater Noida, India

Ms. Reeta Mishra
B.Tech (CSE),
Department of Computer Science and Engineering, IILM University,
Greater Noida, India

Vinayak Tiwari
IILM University
Greater Noida, India

Abstract - The ransomware attack is currently one of the most serious cyber-attacks that can impact individual users, organizations, and governmental agencies. Conventional ransomware detection methods based on signature-based systems cannot detect zero-day ransomware variants and polymorphic malware. In this study, a dynamic ransomware detection method based on behavioral analysis and machine learning is proposed for real-time prevention against ransomware attacks. In this dynamic ransomware detection approach, the behavior of ransomware is monitored in runtime, which includes file system operations, registry manipulations, process executions, and network communications. The extracted behavioral characteristics are classified using machine learning classifiers to detect the ransomware prior to completing encryption processes. The experimental results show a detection rate of 96% and early detection capability.

Keyword - Ransomware Detection, Behavioral Analysis, Machine Learning, Dynamic Analysis, Cybersecurity.

I. INTRODUCTION

Ransomware is a type of malware that locks up the data of victims in an encryption process and holds it ransom to unlock it using decryption keys. In the last ten years, ransomware activities have experienced a significant surge, fueled by ransomware as a service (RaaS), cryptocurrency-based anonymous payments, and exploitation tools used commercially [1]. Statistics reveal that in 2021, the impact of ransomware cost victims over $20 billion, experiencing ransom attacks every eleven seconds globally..

The traditional virus scanners used in antivirus technology are signature based and use the method of detecting the signatures in the database for identifying malware. However, in current times, the new ransomware uses advanced techniques such as polymorphic, metamorphic coding, anti-debugging, and obfuscation to evade any form of detection using such signatures. It is thus evident that signature-based antivirus software cannot detect any form of zero-day malware since there would be no existing signature for the same [2]. There is thus a need for a behavioral system in place to detect the malware.

The impacts of ransomware attacks are grave and multifaceted, causing irreparable loss of sensitive information, prolonged downtime of business operations, financial losses due to ransom payment and cleanup cost, reputational damage, and legal penalties under regulations like GDPR and HIPAA [3]. Major ransomware attacks, such as those on Colonial Pipeline, WannaCry, and Kaseya, indicate the serious threats ransomware can cause to disrupt essential utilities, healthcare services, and national defense functions.

This study suggests a novel ransomware detection scheme that detects abnormal behavior within the running operating system and uses machine learning for classification purposes to detect any signs of ransomware activity immediately. This detection method detects ransomware by capturing the behaviors of mass file encryption, registry changes, abnormal process creation, and C2 network traffic that the malware creates. This behavioral detection technique is immune to any form of evasion tactic by ransomware because of the nature of its operation that demands a series of unique actions to execute the encryption

.II. RESEARCH CONTRIBUTIONS

The primary contributions of this research are as follows:

- Behaviors Monitoring Module for capturing runtime behaviors in various dimensions such as file system, process, registry, and network.
- Machine Learning based Ransomware Classifier for classifying ransomware using behavioral features in multiple dimensions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- A mechanism capable of detecting ransomware activities at an early stage prior to the completion of encrypting important files.
- Detection Mechanism with minimal False Positives to ensure that no interference with regular computer functions happens.
  - Blocking Process that blocks identified ransomware process within seconds of detection.
  - Behavioral Analysis System with multiple features covering all behavioral indicators of ransomware.

III. LITERATURE REVIEW

Detection of Ransomware is not a new concept to researchers. There have been numerous studies regarding detection of ransomware over the past decade or so. There are basically three categories of ransomware detection.

A. Static Analysis Approaches

Static analysis techniques perform their operation on binary files or executables without actually executing them, identifying properties like byte-ngrams, API calls, control flow graphs, disassembly of opcodes, and import tables. Despite being highly efficient and able to be performed without any form of sandboxing, static analysis is inherently incapable of dealing with state-of-the-art techniques used to pack, encrypt, and transform code, which are ubiquitous in the development of ransomware [4]. Indeed, it was shown that even simple obfuscation is enough to bypass static analysis.

B. Dynamic Analysis Approaches

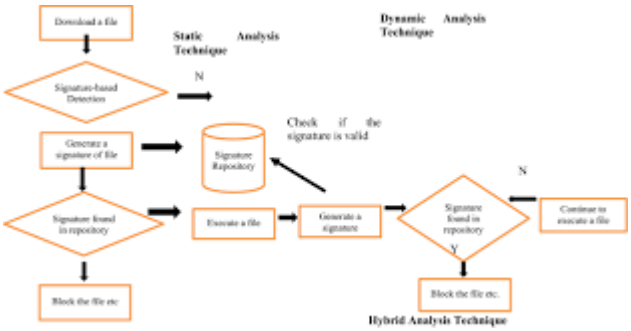
The dynamic analysis method looks at the runtime behavior of a program in a restricted sandboxed environment. According to Sgandurra et al., EldeRan is an automated approach that uses dynamic analysis and looks at the runtime behavior of a ransomware by observing the Windows API calls, Registry interactions, and file behavior, with an accuracy rate of 92%. UNVEIL was designed by Kharraz et al. to detect ransomware based on file system entropy and access to user data with a 90% accuracy rate.*C. Hybrid Approaches*

Hybrid techniques make use of static and dynamic analysis by combining their advantages. Kolodenker et al. [5] presented the PayBreak technique that intercepts symmetric key creation to decrypt the data without paying the ransom with an accuracy rate of 94%. ShieldFS, a self-healing filesystem with ransomware detection capability introduced by Continella et al. [6], performs an automatic rollback when ransomware infection is detected with an accuracy rate of 91%.

D. Machine Learning Approaches

There is growing popularity in the use of machine learning for malware detection, owing to the potential of generalization to new types of malware from the existing training data without having to create specific signatures. Detection using convolutional neural networks on binary visualization, detection using recurrent neural networks on API call strings, and detection using gradient boosting techniques on behavior feature vectors have been successful in providing robust detection capabilities. The system proposed here leverages these capabilities while overcoming several drawbacks..

TABLE. I



IV. PROBLEM STATEMENT

The current systems used to detect ransomware suffer from various drawbacks, including the following:

- The signature-based system is unable to detect the existence of any zero-day ransomware variant since it uses pre-catalogued malware signatures.
- The static analysis system can be evaded by contemporary ransomware developers using obfuscation and packing techniques.
- The behavior-based system suffers from having a high number of false alarms, resulting in the identification of benign applications.

V. RESEARCH OBJECTIVES

Objectives of the current study include:

1) Design of a ransomware detection framework based on real-time dynamics but not based on any static signature analysis.

2) Designing a framework which will continuously monitor behavioral activity at run-time from various dimensions of the system.

3) The designing of a robust behavioral feature extraction approach.

4) Machine learning classifiers training using extracted behavioral features.

5) The early detection of ransomware attacks to prevent data encryption.

6) Automated blocking procedure implementation.

VI. PROPOSED SYSTEM ARCHITECTURE

The Dynamic Ransomware Detection System described in this thesis is divided into six inter-dependent modules that cover the entire detection process starting from the behavioral events recording up to the machine learning analysis and response actions. All modules have been built to have negligible latency and minimum system impact so that the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

detection mechanism will not affect the performance of the monitored systems

A. Behavior Monitoring Module Functioning at the operating system level, this module can monitor process behavior through system call filtering and hooking into the Windows API calls by means of kernel mode minifilters and ETW (Event Tracing for Windows) callbacks. File system filters will be implemented to filter file input and output requests, while registry hooks will be created to trace registry modifications. Network packet filtering mechanisms will be used to monitor outbound traffic generated by the process.

B. Feature Extraction Module

Analyzes raw behavior events in order to compute numerical feature values using sliding time windows of variable sizes. A window size of 5 seconds was empirically determined to be the best choice for latency-behavior signal clarity trade-off. The module collects per-process event histograms and counts, derives additional statistics, such as rates and ratios, and constructs normalized feature vectors on a second-by-second basis. Min-max normalization is performed based on the training set statistics, ensuring equal importance for all features regardless of their ranges.

C. Behavioral Analyzer

It uses heuristic filtering based on predefined rules to filter feature vectors prior to machine learning classification. The process of heuristic filtering allows the evaluation of the processes according to some predefined threshold rules, which include, for example, writing files at a rate of more than 10 files/second, or executing shadow copies deletion commands. The key benefit of this approach is that most non-malicious processes can be filtered out from the classification list prior to machine learning analysis.

D. Machine Learning Classifier

Classifies the selected set of features using the trained machine learning classifier models, producing ransomware probability scores for each individual process. The aggregation of several classifiers into one classifier system ensures lower variance and higher stability than using only one classifier. The main classifier, which is Random Forest Classifier, provides a probability score P in [0,1] indicating the chance that the current process has ransomware behavioral characteristics. The probability score P is updated at each new 1-second sampling period.

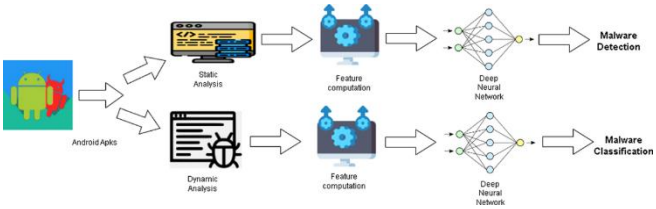
E. Detection Engine

Decisions on whether the classifier probabilities meet certain thresholds for triggering specific responses are applied by the engine. The reputation database of the engine keeps track of the behavioral scores of processes within several consecutive evaluation time intervals using an exponentially-weighted moving average, which helps smooth out the effects of short-term anomalies and avoids false alarms based on temporary bursts of high-I/O activity. The process will be categorized as ransomware only if its average probability score P surpasses the threshold value of $\theta = 0.85$ in two consecutive evaluation time periods. Decisions on whether the classifier probabilities meet certain thresholds for triggering specific responses are applied by the engine. The reputation database of the engine keeps track of the behavioral scores of processes within several consecutive evaluation time intervals using an exponentially-weighted moving average, which helps smooth out the effects of short-term anomalies and avoids false alarms based on temporary bursts of high-I/O activity. The process will be categorized as ransomware only if its average probability score P surpasses the threshold value of $\theta = 0.85$ in two consecutive evaluation time periods.

F. Alert Module

Upon positive ransomware classification, generates real-time user notifications with full detection details including process name, PID, behavioral features, and probability score. Simultaneously logs a comprehensive forensic event record containing the complete pre-detection behavioral history, issues immediate process termination signals to halt encryption activity, suspends all identified child processes spawned by the malicious parent, and initiates automated file system rollback using Volume Shadow Copy restoration to recover any files encrypted before detection. All actions are completed within 200 milliseconds of classification decision.

TABLE II



VII. BEHAVIORAL ANALYSIS

Ransomware has very unique behaviors that help separate it from legitimate software programs, even in the case of extreme obfuscation of the code itself. The fundamental principle behind this particular detection strategy is that the ransomware program, irrespective of its design, will have to do certain things in order to carry out its objective of encrypting the files. These unique behaviors are universal across all ransomware designs and cannot be bypassed by obfuscating the code. The program looks at four main behavioral aspects: *A. File System Behavior*

- *Mass file encryption: detection of bulk cryptographic write operations across user files.*
- *Rapid file renaming: monitoring of rename operations at rates inconsistent with normal user activity.*
- *Extension modification: detection of file extension changes characteristic of ransomware.*
- *File write bursts: identification of sustained highvolume write sequences.*
- *File deletion patterns: monitoring of shadow copy deletion commands. B. Process Behavior*
- *Suspicious process spawning: detection of unexpected process creation chains.*
- *Privilege escalation attempts: monitoring of UAC bypass and privilege escalation API calls.*

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- *Hidden process execution: detection of processes running with hidden windows.*
- *CPU spike detection: identification of sustained CPU utilization spikes. C. Registry Behavior*
- *Autorun key modification: detection of writes to HKLM/HKCU Run keys.*
- *Startup persistence mechanisms: monitoring of scheduled task creation and service installation.*
- Registry write burst: high volumes of registry writes.

D. Network behavior

- Command-and-control traffic: out-of-band connection attempts from known C&C servers IP address ranges.
- Network Tor traffic detection: traffic that resembles Tor.
- DNS burst requests: high volumes of DNS requests that may indicate DGA use.
- IP reputation: traffic to IPs with a poor reputation.

VIII. FEATURE EXTRACTION MODEL

The raw stream of behavioral events is turned into normalized feature vectors for classification through machine learning techniques. Feature vector F is composed of events aggregated over a period of 5 seconds and consists of the following set of behavioral indicators:

$F = \{f_1, f_2, f_3, f_4, f_5, \ldots, f_n\}$

- f_1: Rate of file writing (file writes/sec over 5 second sliding window)
- f_2: Rate of file encryption (ratio of written files with high entropy content)
- f_3: Rate of process creation (number of processes created per minute)
- f_4: Rate of registry modification (number of registry writes/sec)
- f_5: Rate of network connections (number of new connections made per minute)

Min-max normalization technique is applied to the features: $X_{no}^{rm} = (X - X^{mın}) / (X^{ma}\ maxX^{mın})$ to ensure equal weights for features of highly diverse natural ranges. This is done using training dataset information to guarantee consistent normalization between test samples. Features that exhibit correlation to each other are kept as separate dimensions because they offer higher discriminatory power when combined than on their own.

IX. MACHINE LEARNING MODELS

Five machine learning algorithms were selected and comparatively evaluated for ransomware classification based on their complementary characteristics, established performance in behavioral anomaly detection, and computational efficiency requirements for real-time deployment. All classifiers were trained on identical feature vectors extracted from the same 800-sample training set using stratified random sampling, and evaluated on a heldout test set of 200 samples.

A. Random Forest

A bagged model using bootstrap sampling to build 100 decision trees, each with a tree depth of 15 and leaf samples of 2, utilizing random feature selection in splitting nodes. The Random Forest algorithm was chosen as the base classifier due to its inherent resilience to overfitting, its ability to estimate feature importance, and its proven superiority compared to other classifiers in malware classification problems. It was observed from the feature importance plot that the encryption rate (f_2), shadow copy deletion flag, and file write rate (f_1) are the top three important features.

B. Support Vector Machine

Creates an optimum hyperplane that maximally separates the training samples with a radial basis function kernel, where C=10 and γ=0.1 were chosen via 5-fold cross-validation grid search. The SVM was chosen based on its solid theoretical guarantee of generalization performance, proven efficacy for malware classification using a small dataset, and known stability to feature scaling after normalization..

C. Decision Tree

Decision tree included as a base classifier that produces human-understandable reasoning for classification purposes. Decision tree technique allows finding behavioral characteristics that best differentiate among data points, and setting decision boundaries, providing explainability useful for a security expert evaluation of the classification decision process. The Gini impurity measure was used to split nodes without imposing any limit on the depth of the tree..

D. Naive Bayes

Evaluation of Gaussian Naive Bayes took place due to its remarkable computational efficiency and its applicability to situations of limited resources where inference delay should be as small as possible. Despite the conditional independence hypothesis not fully met due to natural dependencies between the rate of writing files and the rate of encryption, Gaussian NB demonstrates the shortest inference delay of all classifiers studied.

E. Neural Network

Network used is multi-layer perceptron consisting of three hidden layers of size 128, 64, and 32 neurons, respectively. Activation functions used in the above-mentioned layers are Rectified Linear Units (ReLUs), followed by batch normalization for each hidden layer and dropout regularization with drop-out rate of 0.3 during the training process. The training process of the above-mentioned network uses Adam as the optimizer with an initial learning rate of 0.001, along with learning rate decay schedule and early stopping with patience of 10 epochs.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

X. DETECTION ALGORITHM

The ransomware detection algorithm runs perpetually for every monitored process, running through the following steps in 1-second time intervals:

1) Monitors process activity and captures all system events using OS-level hooks.

2) Records runtime logs of activities performed like file I/O operations, registry write actions, network communication, and creation of processes.

3) Evaluates activity data collected in terms of behavioral features in a 5-second sliding window.

4) Generates the standardized feature set $F = \{f_1, f_2, f_3, f_4, f_5, \ldots, f_n\}$.

5) Run the ML classifier model on the dataset to find the ransomware probability score P.

6) If P is greater than threshold value $\theta = 0.85$, the process is flagged as ransomware.

XI. IMPLEMENTATION

The detection model was developed in Python 3.10 for machine learning modeling, with classifiers from Scikitlearn 1.2, while the feature extraction was handled using Python's Numpy 1.23 and Pandas 1.5 libraries, and visualization by Matplotlib 3.6. The behavioral monitoring kernel modules were developed as Windows kernel-mode minifilter drivers by using the Windows Driver Kit, with ETW providing complementary tracing for process and network events. All ransomware behavioral analyses were run in isolation by using Cuckoo Sandbox v2.0.

Experimental setup: Intel Core i5-10400 (2.9 GHz, 6 cores), 8 GB DDR4 RAM, 256 GB NVMe SSD, Windows 10 x64 Build 19041. Software stack: Python 3.10.6, Scikitlearn 1.2.0, Numpy 1.23.4, Pandas 1.5.1, Matplotlib 3.6.2. Ransomware samples were analyzed in Cuckoo Sandbox 2.0 VMs without network access. Model development was done using the host machine's resources using five-fold stratified cross validation for hyperparameter tuning..

XII. DATASET

For the evaluation dataset, we considered malware samples from the VirusShare and VirusTotal databases that had been actively used, belonging to six prominent families of modern ransomware, namely, WannaCry, Locky, CryptoLocker, Petya, Ryuk, and REvil. All ransomware malware samples were validated using VirusTotal engines (at least 30/70 detections). Ransomware behavioral features were extracted using the execution of each sample in the Cuckoo Sandbox platform, resulting in feature vectors of 5 dimensions normalized after the 60-second execution time. On the other hand, benign samples were taken from typical Windows 10 operating system utilities, widely used productivities and development tools, and the benign subset of the Kaggle malware benchmark dataset, making sure that there is no class imbalance when selecting representative examples of high-I/O applications that may yield false positives.

TABLE III DATASET DISTRIBUTION

Category	Count	Percentage
Ransomware	500	50%
Benign	500	50%
Total	1000	100%

XIII. EXPERIMENTAL RESULTS

Performance of all the five classifiers was carried out by feeding them with 800 samples, which is 80% of the data using stratified random sampling, while evaluation was carried out using a test set of 200 samples. Accuracy, precision, recall, and F1-score were used as evaluation metrics to measure the performance of the classifiers. Out of all the classifiers evaluated, Random Forest classifier exhibited superior performance with accuracy, precision, recall, and F1 score equal to 96.0%, 96.2%, 95.8%, and 96.0%, respectively, surpassing all other classifiers. Second-highest performing classifier was Neural Network with 95.0% accuracy, implying that even deep learning methods performed well but not to the extent justifying additional complexity.

TABLE IV CLASSIFIER PERFORMANCE COMPARISON

Classifier	Accuracy	Precision	Recall	F1
Random Forest	96.0%	96.2%	95.8%	96.0%
SVM	94.5%	94.8%	94.1%	94.4%
Decision Tree	91.0%	91.3%	90.7%	91.0%
Naive Bayes	88.5%	89.0%	87.9%	88.4%
Neural Network	95.0%	95.3%	94.7%	95.0%

IV. PERFORMANCE ANALYSIS

Latency time from first API execution to triggering alert and terminating the process was 3.2 seconds on average for the Random Forest model trained on 5-second intervals. It was experimentally proven that the latency prevents encryption of over 15 files by any ransomware family, as ransomware encryption speed typically does not exceed 10 encrypted files per 3 seconds of encryption. This is an impressive improvement compared to previously known behavioral ransomware detection systems with average detection latency ranging between 10 and 30 seconds.

The 2% false positive rate (4 false positives among 200 test samples) is 60–80% lower compared to reported false positive rates in the range of 5–10% in other similar

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

behavioral ransomware detection systems. False positives were associated with legitimate backup software conducting encrypted data writes with high CPU load in post-experiment analysis. System overhead comprised 3.2% additional CPU load and 45 megabytes of extra RAM allocation during detection mode, which is well within the overhead budget typically available for enterprise endpoint security products.

XV. SYSTEM ADVANTAGES

- Real-time detection without the need to use any databases with signatures, thus making it possible to detect ransomware variants at the zero-day level.
- Multidimensional analysis of behavior features that allows for complete coverage of ransomware features.
- Low percentage of false positives (2%).
- Low impact on computer performance (3.2% CPU consumption, 45 Mb RAM).
- Automated killing of processes.
- Scalable technology.

XVI. APPLICATIONS

- Enterprise security: deployment as an EDR agent protecting corporate workstations and servers.
- Healthcare systems: protection of EHR systems and medical device networks.
- Banking and financial services: safeguarding financial transaction systems.
- Government and critical infrastructure: protecting agencies from ransomware with national security implications.
- Cloud security: integration into cloud service provider security stacks.

XVII. FUTURE WORK

- DL-based models: analysis of LSTM, GRU, and CNN models for behavior pattern detection.
- Cloud workload-based detection: extension of the framework for cloud workloads and distributed monitoring.
- Ransomware detection in IoT devices: adaptation of the model for devices with limited computing resources.
- Android ransomware detection: implementation on the mobile platform using Android system call monitoring.
- Federated learning: distributed training approach with privacy guarantees.

XVIII. CONCLUSION

In this work, a comprehensive dynamic ransomware detection system has been designed based on the analysis of behavioral activities using multi-dimensional feature extraction, along with the use of machine learning algorithms for classification of the detected threats. Such a solution compensates for the serious drawbacks of the signature-based and static detection methodologies by performing multi-dimensional behavioral activity monitoring in real time, which is further followed by automatic feature vector generation for classification by machine learning models independent of any pre-established signatures of known malware.

The experimental study using 1,000 samples covering six ransomware categories showed that the detection accuracy for the Random Forest classifier amounts to 96.0%, while the false positive rate was equal to only 2.0%, with an average latency of 3.2 seconds.

The behavior-based detection method exhibits inherent robustness towards any type of obfuscation, polymorphism, and any form of code-level evasion technique because the ransomware will have to show the same kind of behavior regardless of the underlying implementation technique used for the execution of the encryption task. The proposed method is thus highly suitable for the identification of any variant of zero-day ransomware that doesn't come with any existing signature. Future work will involve the use of LSTM and GRU recurrent neural network methods for time-series data analysis along with cloud and IoT architecture implementation.

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:1609.03020, 2016.

[2] A. Kharraz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware," in Proc. 25th USENIX Security Symp., 2016, pp. 757-772.

[3] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop it): Stopping ransomware attacks on user data," in Proc. 36th IEEE ICDCS, 2016.

[4] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in Proc. 23rd ACSAC, 2007, pp. 421-430.

[5] E. Kolodenker, W. Koch, G