Next-Generation Ransomware Defense: High-Performance Computing Strategies for Monitoring Disk I/O and CPU Performance

Khirsagar Rishitha Dept. of CSE(CS)
Institute of Aeronautical Engineering Hyderabad,India
rishithakhirsagar07@gmail.com

Mr.Y.Manohar Reddy
Assistant Professor Dept. of CSE(CS)
Institute of Aeronautical Engineering Hyderabad,India
y.manoharreddy@iare.ac.in

D.Kundhan Rao Dept. of CSE(CS)
Institute of Aeronautical Engineering Hyderabad,India
kundhanrao2003@gmail.com

Manish Kumar Dept. of CSE(CS)
Institute of Aeronautical Engineering Hyderabad,India
manishkumarbanda@gmail.com

Abstract— Ransomware frequently bypasses antivirus tools, encrypting files and making data inaccessible. Traditional detection methods, which involve monitoring processes, system calls, and file activities, have high overhead and can be disrupted by sophisticated ransomware. This Researchintroduces a method for detecting ransomware on a virtual machine by collecting specific processor and disk I/O event data from the host machine and using a machine learning classifier. The random forest model excelled among seven classifiers, achieving 0.98 accuracy within 400 milliseconds across various user loads and 22 ransomware types.

INDEX TERMS Deep learning, disk statistics, hardware performance counters, machine learning, ransomware, virtual machines.

I.INTRODUCTION

Ransomware is a type of malware that encrypts files or locks computers, rendering them unusable to extort money from victims. Nation-state actors may also use ransomware to disrupt critical infrastructure. These attacks frequently involve data theft to coerce victims into paying ransoms or to sell the stolen information on the dark web. In 2022, nearly 70% of organizations experienced ransomware attacks. By 2031, it is estimated that ransomware will strike a business, individual, or device every 2 seconds, a significant increase from every 11 seconds in 2021

Traditional signature-based detection methods rely on antivirus-generated hash values to identify known ransomware. Thus, behavioral methods, which analyze the sequence of actions taken by ransomware, are essential. Recent ransomware variants like LockBit2.0, Darkside, and BlackMatter quickly encrypt parts of files to render them unusable. Ransomware's need to rapidly encrypt user files distinguishes its behavior from benign applications, making it detectable through elevated activity using machine learning (ML) techniques.

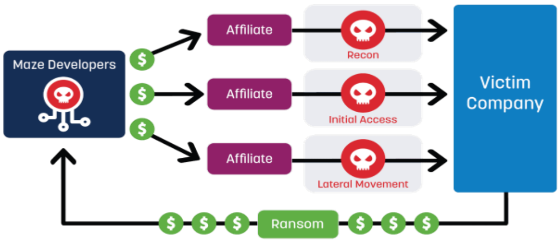


Fig.1: Ransomware Detection

On the target machine, runtime detection entails constant monitoring of numerous programs and components, which is resource-intensive and vulnerable to ransomware disabling it. Special- purpose registers called hardware performance counters (HPCs) track processor and system. incidents and have been investigated to look for malware. But keeping an eye on every process can make the system perform worse. Previous studies have gathered data at the machine level, but they were restricted to a particular workload, which affected the accuracy of identification under different workload

1. Accurate and overhead-free ransomware detection from the host machine using HPC

contamination from monitoring multiple processes on the target machine.
1. Evaluation of detection effectiveness underdifferent user workloads, demonstrating improved accuracy of
3. ML and deep learning (DL) models trained with varying workloads.
4. Combining HPC and disk I/O data to enhance detection accuracy compared to using either data source alone.
5. Achieving high-probability ransomwaredetection within a few hundred milliseconds of execution, even with varying workloads.
6. Providing experimental datasets, codes, and scripts for further research and verification.

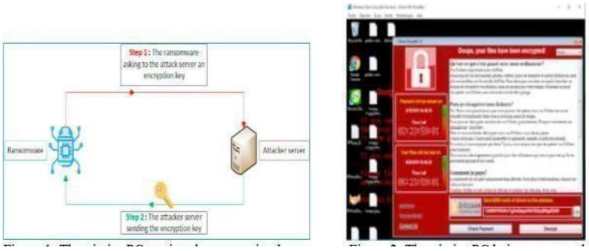

Step 1 : The ransomware asking to the attack server an encryption key
Ransomware
Attacker server
Step 2 : The attacker server sending the encryption key
Ooops, your files have been encrypted!
Check Payment
Decrypt

Fig 2 : The victim PC getting the encryption key and the victim PC being encrypted

The arrangement of the research is as follows: Section
II examines the literature on runtime ransomware detection, Section III describes the tools and experimental setup,
The ML models are described in Section IV, and Section V shows findings of a ransomware detection,

II. LITERATURE REVIEW

[1] SR Division (2022) The number and sophistication of ransomware attacks against businesses have significantly increased, according to the SR Department's report on the ransomware victimization rate for 2022. The report, available via Statista, offers a thorough examination of the proportion of companies impacted by ransomware attacks and identifies significant patterns.

[1] D. Braue. (2022) According to D. Braue's report on ransomware damage costs, the economic burden of ransomware attacks is set to increase dramatically over the next decade. The report, accessible via Cybersecurity Ventures, provides an in-depth analysis of the financial ramifications of ransomware attacks, highlighting both current and future projections

[2] Logix Consulting. (2020) This technique relies on matching the unique signatures of files against a database of known malware signatures.

The method is praised for its speed and accuracy in detecting known threats, significantly reducing false positives. However, the report highlights significant limitations, such as its inability to detect new or mutated malware that 1 lacks a matching signature in the database.

[3] M. Loman. (2021) M. Loman's 2021 report on Lockfile ransomware details innovative evasion techniques and the use of intermittent encryption to avoid detection. This method encrypts only parts of files, making it harder for traditional security tools to recognize the attack. Lockfile also employs sophisticated evasion strategies, such as terminating security processes and exploiting system vulnerabilities to maintain persistence. These advanced tactics highlight the evolving nature of ransomware and the increasing challenge of defending against such threats.

III. EXISTING METHOD

Different methods have been developed to identify crypto-ransomware by analyzing file input/output operations. Kharraz et al., for instance, unveiled UNVEIL, a dynamic analysis tool that watches file system input/output (I/O) events, like writes and deletes, from both malicious and benign apps in order to detect ransomware. Although their approach is intended to detect ransomware samples that are not actively running, it managed to obtain a high true positive rate (TPR) of 0.96 with no false positives.

In a similar vein, ShieldFS, an add-on driver for Windows developed by Continella et al., collects I/O access patterns from both malicious and benign apps. They trained a proprietary machine learning classifier with these patterns in order to identify malicious activity in real time.

Their model, however, has trouble differentiating between safe and harmful encryption.

RW Guard, a hybrid detection approach that combines decoy tactics and entropy for file change monitoring, was first presented by Mehnaz et al.

The EldeRana framework, developed by Sgandurra et al., uses machine learning classifiers like Regularized Logistic Regression to identify dynamic aspects of ransomware. This framework examines binary strings, file I/O operations, registry operations, and Windows API calls. In a similar vein, Zavarsky et al. demonstrated that anomalies in disk and registry activity can be used to detect ransomware on both the Windows and Android operating systems. Nevertheless, a number of these detection techniques have drawbacks, such as the potential for single points of failure and difficulties distinguishing between benign and malicious activity, particularly when entropy-based identification techniques are employed.

IV. PROBLEM STATEMENT

The problem statement revolves around the inefficiency and limitations of existing ransomware detection methods:

- Traditional ransomware detection methods typically involve continuous monitoring &

High Overhead of Monitoring Processes

- Signature-based detection methods are heavily reliant on known patterns and signatures of ransomware.

V. PROPOSED METHOD

The suggested solution gathers host CPU and disk I/O events, which offers a novel method for identifying malware on virtual machines. framework. This technique makes use of a random forest (RF) classifier in order to generate a successful recognition model, simplifying the identification procedure by lowering the must keep an eye on each action taken toward the goal. decreasing the possibility of ransomware-related data breaches.

It is adaptable to changes in user workloads, ensuring rapid and accurate detection of both new and known malware. Compared to other methods, the RF classifier demonstrates superior effectiveness. This research further enhances malware detection by employing a CNN2D and an ensemble model with a voting predictor. The voting classifier, which integrates multiple machine learning classifiers, achieved 99% accuracy, highlighting the robustness of combining models for malware detection.

This dataset offers a solid basis for training and testing models since it contains samples of known and unknown malware. Data manipulation entails turning unprocessed data into useful insights, which data scientists usually manage. This comprises gathering, arranging, cleaning, and confirming data analysis and format conversion into usable forms. One essential component of feature engineering entails determining the most dependable, practical and unique characteristics for the model constructing. Through a methodical reduction of data sizes, Feature selection improves the predictive model

VI. METHODOLOGY

Hybrid Esemble For Detection:

Ransom attackers will evade antivirus and then enter into victim system to execute malicious script which will encrypt and make entire system data unusable and to decrypt files back they will ask ransom from the victim and world has loss billions of dollars in ransom since the birth of internet network. Many existing techniques are available such as SYSTEM PROCESS MONITORING to identify and prevention of malicious script execution but this monitoring will impact system performance.

Another technique is to monitor files which are getting deleted or created to know malicious file but this technique also impact system performance and detection accuracy also not good enough.

To overcome from above issue author of this proposed work employing VMWARE on host system which will read Hardware Performance Counters (HPC) and IO EVENTS data and then applying this data on machine learning models to predict whether executing script is normal (benign) or Ransomware. Extracting HPC and IOEVENTS features using VMware will not affect system performance and machine learning models also able to predict Ransomware with more than 90% accuracy.

In all algorithms Random Forest, XGBOOST is giving accuracy. To train all algorithms author has publish HPC and IOEVENTS from different programs such as 7ZIP, AES and many more and this dataset can be downloaded from below link

HPC dataset IOEVENTS dataset
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/GHJFUT
We combined both datasets, and the following screen shows the details of the integrated dataset. This research examined five machine learning (ML) classifiers and two deep learning (DL) classifiers to determine their effectiveness in ransomware detection using data from hardware performance counters (HPC) and disk I/O. Three models were created: one that used only HPC data, another that relied solely on disk I/O data, and an integrated model combining both types of data.

In the integrated model, each instance included both HPC and disk I/O data. We evaluated the performance of these three models using the seven classifiers. The ML models were built using the default hyperparameters available in the scikit-learn library, a popular machine learning library for Python. The DL models required more careful hyperparameter tuning. Since DNN models are highly sensitive to these settings, we utilized autokeras, an AutoML library, to automatically optimize them.

The LSTM model was structured with four layers. To handle discrepancies in the length of I/O and HPC data, padding was applied to the shorter data, and the raw data were fed directly into the model. A masking layer was used to ignore the padded sections as the data moved through the network.

A global average pooling layer was added to the 64 memory units that made up the LSTM layer in order to lower the possibility of overfitting. The output layer used a sigmoid activation function and was composed of a dense layer with a single neuron. Scikit-learn v1.1.2 was used to implement SVM, k-NN, Decision Trees, and Random Forest; XGBoost v1.6.2 was used for the XGBoost method; autokeras v1.0.20 was used for deep neural networks (DNN); and Keras v2.9.0 was used for the LSTM model, utilizing TensorFlow as the backend.

A.USER WORKLOADS

The research explored the multivariate connection between HPC metrics and I/O data, focusing on how these factors varied with the type of application (ransomware or benign) under different user workloads. The analysis included four benign applications: 7zip, aesCrypt, s Delete, and dry Run. The normalized average values of the features that were recorded mid-execution of an application are represented by the vertical axis. Every polyline represents how well an application performs under a specific workload; ransomware-associated red lines correspond to malicious programs, while innocuous blue lines represent the same. For improved visual representation, lines are utilized to connect the data points from the same application's execution even when there is no interpolation between the individual events plotted along the x-axis.

B.ALGORITHMS

a. Long Short Term Memory (LSTM): it is a recurrent neural network (RNN) that was created to fix the problem of disappearing gradients that happens in regular RNNs. The new memory cell lets the model understand long-term relationships in sequential data. This makes it perfect for jobs that involve time series or sequential patterns. [15] LSTMs are likely used in the Research because they can model and understand how events and behaviors depend on time. This is very important for finding ransomware because the order of events and behaviors plays a big part. [45] Over time, LSTMs can pick up on subtle trends, which makes the model better at finding malicious actions.

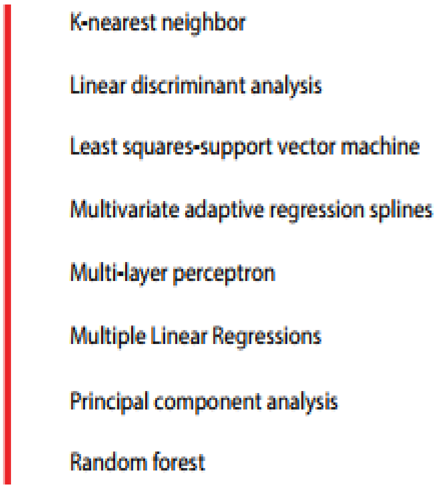
b. Deep Neural Network (DNN): it is a fancy name for an artificial neural network that has many buried layers between theinput and output levels. Because these networks can learn complex hierarchical representations of data, they can be used for hard tasks that need to abstract and describe features. DNNs could be used in the because they can learn complex traits and connections in the data that is collected.

c. XGBoost: it is a machine learning method from the gradient boosting family, designed to create an ensemble of weak learners, commonly decision trees, in a sequential manner. Each successive tree aims to correct the mistakes made by its predecessor, improving the model's performance with each iteration. This iterative process helps develop a highly accurate and robust model. it is often selected for its ability to efficiently manage large datasets and perform well in both classification and regression tasks. Its efficiency and scalability make it a popular choice for complex problems.

Regarding finding ransomware, XGBoost can be very good at making predictions, able to accurately capture the different ways that ransomware acts and help make a good detection model [8, 13, 14].

d. Random Forest: During the training phase, an ensemble learning approach produces several decision trees. It takes the average of the predictions for regression tasks and the mode of the tree outputs for classification tasks to forecast the class.

e.Decision Tree: It is a model thatlooks like a tree, and each point is a choice that is made based on the traits that are given. In a looping process, it divides the information into smaller groups, ending with nodes that represent the final guess or choice. Decision trees are used to show how decisions are made because they are easy to understand and use.



K-nearest neighbor

Linear discriminant analysis

Least squares-support vector machine

Multivariate adaptive regression splines

Multi-layer perceptron

Multiple Linear Regressions

Principal component analysis

Random forest

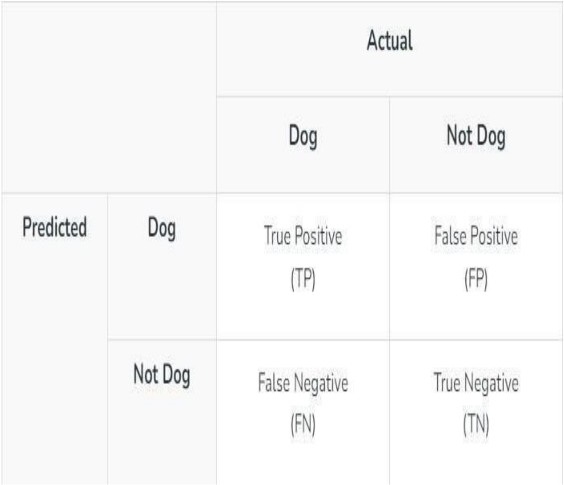
Fig.3 : Machine learning algorithms

learning that is used for jobs like regression and classification. It finds a hyperplane that best divides data into groups or guesses a continuous result, with the most space between groups.

SVM is utilized because it can identify the optimal choice limits and handle data with multiple dimensions. SVM can be a dependable method for classifying files in malware detection, when feature spaces can be complex, by providing distinct decision lines.

An instrument for assessing a machine learning model's performance on test data is a confusion matrix, which summarizes the quantity of accurate and inaccurate forecasts. It is especially helpful for evaluating categorization models that forecast classification labels for examples of input.The matrix provides a detailed breakdown of how the model performed:

This matrix helps in understanding the model's accuracy by showing the distribution of prediction outcomes across these categories.



		Actual	
		Dog	Not Dog
Predicted	Dog	True Positive (TP)	False Positive (FP)
	Not Dog	False Negative (FN)	True Negative (TN)

Fig.4: Confusion Matrix

VII. IMPLEMENTATION

The ransomware detection system is implemented within a virtualized environment to leverage High-Performance Counters (HPC) and disk Input/Output (I/O) data for accurate monitoring and detection. The process begins with setting up multiple Virtual Machines (VMs) with diverse operating systems and configurations to simulate a realistic IT environment.

Tools such as `perf` and `virsh` are installed to facilitate continuous data collection of HPC and disk I/O metrics from these VMs. This initial setup ensures a comprehensive baseline dataset representing normal, benign application behavior

Tools such as `perf` and `virsh` are installed to facilitate continuous data collection of HPC and disk I/O metrics from these VMs. This initial setup ensures a comprehensive baseline dataset representing normal, benign application behavior.

Next, the focus shifts to data preprocessing and machine learning model training. The collected raw data undergoes feature extraction, where statistical measures are computed to capture significant patterns. A Random Forest classifier is then trained using this processed data, incorporating both benign and ransomware scenarios to create a labeled dataset. The training involves splitting the data into training and testing subsets, performing cross- validation, and fine-tuning the model's hyperparameters to enhance detection accuracy and robustness.

Finally, the trained classifier is deployed for real-time monitoring and alerting. The system continuously analyzes incoming HPC and disk I/O data, detecting anomalies indicative of ransomware activities. Upon detection, alerts are generated and communicated to system administrators, while automated response mechanisms are activated to mitigate the threat. These responses may include isolating the affected VM and halting suspicious processes.

Regular updates to the training dataset, incorporating new benign and ransomware samples, ensure the system remains adaptive and effective against evolving threats. Continuous feedback integration and performance optimization further enhance the system's reliability and efficiency in protecting infrastructure.

VIII. RESULTS

The provided Jupyter Notebook output demonstrates the effectiveness of a ransomware detection model. The model processes various feature sets from test data, predicting whether each instance is "Ransomware" or "Benign." The results indicate a high accuracy in predictions, correctly identifying all instances in the provided examples. This accuracy suggests the model is well-trained and capable of distinguishing between ransomware and benign data based on the given features, making it a valuable tool for cybersecurity applications.

IX. CONCLUSION

This Proposed Work introduces an approach aimed at swiftly and accurately detecting ransomware running on virtual machines (VMs) by leveraging processor and disk I/O activity data collected from the host machine. The method utilizes machine learning techniques for analysis, specifically focusing on a subset of processor events and disk I/O events selected through feature elimination techniques.

Processor-event data is gathered using the perf tool and consists of five important events that are selected from a pool of more than 40 using recursive feature elimination.HPCs, or hardware performance counters. In a similar vein, diskEight events total in the I/O event data are collected.virsh domblkstats are used.

The study evaluates five classifiers for machine learning (ML) and two deep three distinct learning (DL) models, each of which was used to distinct models: one that is only based on HPC data, and another only using disk input/output data, and a third hybrid model combining the two kinds of data.
Every time, the integrated model performs better than the others.

X. FUTURE SCOPE

The directions include exploring detection of data exfiltration activities by analyzing network traffic alongside HPC and I/O data. Additionally, efforts will focus on improving data labeling to distinguish between ransomware scouting phases and active encryption phases, aiming to enhance detection accuracy during ransomware's destructive activities. Further enhancements will involve adapting the model for standalone machines and investigating its scalability across different hardware configurations.

REFERENCES

[1] A. Tang, S. Sethumadhavan, and S. J. Stolfo, "Unsupervised anomaly-based malware detection using hardware features," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, Cham, Switzerland: Springer, 2014, pp. 109–129.

[2] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and drop it): Stopping ransomware attacks on user data," in *Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2016, pp. 303–312.

[3] M. Shukla, S. Mondal, and S. Lodha, "POSTER: Locally virtualized environment for mitigating ransomware threat," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2016, pp. 1784–
1786.

[4] A. Continella, A. Guagnelli, G. Zingaro, G. De Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "ShieldFS: A self-healing, ransomware-aware filesystem," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, Dec. 2016, pp. 336–347.

[5] F. Mbol, J.-M. Robert, and A. Sadighian, "An efficient approach to detect torrentlocker ransomware in computer systems," in *Proceedings of the International Conference on Cryptology and Network Security*, Cham, Switzerland: Springer, 2016, pp. 532–541.

[6] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection," arXiv preprint arXiv:1609.03020, 2016.

[7] A. Kharraz and E. Kirda, "Redemption: Real- time protection against ransomware at end-hosts," in
Proceedings of the International Symposium on Research in Attacks, Intrusions, Defenses, Cham, Switzerland: Springer, 2017, pp. 98–119.

[8] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "UNVEIL: A large-scale, automated approach to detecting ransomware (keynote)," in
Proceedings of the IEEE 24th International Conference on Software Analysis, Evolution, and Reengineering (SANER), Feb. 2017, pp. 757–772.

[9] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics," *Computer and Electrical Engineering*, vol. 66, pp. 353–368, Feb. 2018.

[10] R. Moussaileb, B. Bouget, A. Palisse, H. Le Bouder, N. Cuppens, and J.-L. Lanet, " International Conference on Availability, Reliability and Security*, 2018, pp. 1– 10.

[11] Z. A. Genc, G. Lenzini, and P. Y. Ryan, "No random, no ransom: A key to stop cryptographic ransomware," in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Cham, Switzerland: Springer, 2018, pp. 234–255.

[12] S. Mehnaz, A. Mudgerikar, and E. Bertino, "RWGuard: A real-time detection system against cryptographic ransomware," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, Defenses*, Cham, Switzerland: Springer, 2018, pp. 114–136.

[13] B. Zhou, A. Gupta, R. Jahanshahi, M. Egele, and A. Joshi, "Hardware performance counters can: roceedings of the Asia Conference on Computer and Communications Security*, May 2018, pp. 457–468.

[14] S. Das, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, "SoK: The challenges, pitfalls, and perils of using hardware performance counters for security," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 20–38.

[15] K. Lee, S. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

[16] C. J. Chew and V. Kumar, "Behaviour based ransomware detection," in *Proceedings of the International Conference on Computer and Their Applications*, in EPiC Series in Computing, vol. 58, 2019, pp. 127–136.