IJITCE

ISSN 2347- 3657

International Journal of
Information Technology & Computer Engineering

www.ijitce.com

Email : ijitce.editor@gmail.com or editor@ijitce.com

Ransomware Detection Through Processor and Disk Usage Patterns Using Machine Learning

G Venkata Pradeep Kumar
Assistance Professor
Department of Computer Science
Sir C R Reddy College, Eluru.
gpk@sircrreddycollege.ac.in

ABSTRACT

This project addresses the challenge of detecting ransomware by focusing on the limitations of existing methods that rely heavily on process monitoring and traditional data analysis. The goal is to develop a reliable and efficient method for identifying ransomware on virtual machines (VMs) by monitoring specific processor and disk I/O activities across the entire VM from the host machine. The proposed solution employs machine learning (ML), particularly a Random Forest (RF) classifier, to build a robust detection model that minimizes monitoring overhead and reduces the risk of data corruption by ransomware. A key advantage of this approach is its adaptability to varying user workloads, allowing the model to function effectively in diverse scenarios without the need for constant process monitoring on the target machine. The project's effectiveness is tested using 22 ransomware samples and various user workloads, demonstrating its practical application in real-world environments. To further enhance detection accuracy, the project incorporates a Convolutional Neural Network 2D (CNN2D) and an ensemble model with a voting classifier. This ensemble approach, which combines multiple machine learning classifiers, achieved an impressive 99% accuracy, showcasing the effectiveness of integrating various models for robust ransomware detection. This project offers a practical solution to the evolving threat of ransomware, providing efficient detection while maintaining adaptability and low overhead.

Index terms: Deep learning, disk statistics, hardware performance counters, machine learning, ransomware, virtual machines.

INTRODUCTION

The problem of detecting ransomware is the focus of the project, which takes into account the drawbacks of the existing methods that rely on process monitoring and data analysis. The goal is to create a reliable and useful method for detecting ransomware on a virtual machine (VM). Information assortment centers around unambiguous processor and plate I/O occasions for the whole VM from the host machine. The project aims to develop an efficient detection model by making use of machine learning (ML), particularly a random forest (RF) classifier. This approach limits checking above and mitigates the gamble of information defilement by ransomware. A common obstacle in ransomware detection is overcome by the proposed method's adaptability to user workload variations. The model can still be used in a variety of user scenarios because it does not require constant monitoring of each process on the target machine. The project's efficacy is evaluated using 22 ransomware samples and various user workloads. By providing a reliable detection model, this project contributes a practical and effective solution to the ongoing threat of ransomware. The project reduces monitoring overhead, speeds up detection, and ensures adaptability to changing ransomware variants by utilizing specific processor and disk I/O events and incorporating machine learning. Convolutional Neural Network 2D (CNN2D) and an ensemble model with a voting classifier were added to this project to further improve the accuracy of ransomware detection. The voting classifier, made up of multiple machine learning classifiers, produced accurate final predictions with a remarkable 99 percent accuracy, demonstrating the utility of combining various models for robust detection..

LITERATURE SURVEY

On the classification of Microsoft-Windows ransomware using hardware profile: Due to the expeditious inclination of online services usage, the incidents of ransomware proliferation being reported are on the rise. Ransomware is a more hazardous threat than other malware as the victim of ransomware cannot regain access to the hijacked device until some form of compensation is paid. In the literature, several dynamic analysis techniques have been employed for the detection of malware including ransomware; however, to the best of our knowledge, hardware execution profile for ransomware analysis has not been investigated for this purpose, as of today. In this study, we show that the true execution picture obtained via a hardware execution profile is beneficial to identify the obfuscated ransomware too. We evaluate the features obtained from hardware performance counters to classify malicious applications into ransomware and non-ransomware categories using several machine learning algorithms such as Random Forest, Decision Tree, Gradient Boosting, and Extreme Gradient Boosting. The employed data set comprises 80 ransomware and 80 non-ransomware applications, which are collected using the VirusShare platform. The results revealed that extracted hardware features play a substantial part in the identification and detection of ransomware with F-measure score of 0.97 achieved by Random Forest and Extreme Gradient Boosting.

Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence: Emergence of crypto-ransomware has significantly changed the cyber threat landscape. A crypto ransomware removes data custodian access by encrypting valuable data on victims' computers and requests a ransom payment to re-instantiate custodian access by decrypting data. Timely detection of ransomware very much depends on how quickly and accurately system logs can be mined to hunt abnormalities and stop the evil. In this paper we first setup an environment to collect activity logs of 517 Locky ransomware samples, 535 Cerber ransomware samples and 572 samples of TeslaCrypt ransomware. We utilize Sequential Pattern Mining to find Maximal Frequent Patterns (MFP) of activities within different ransomware families as candidate features for classification using J48, Random Forest, Bagging and MLP algorithms. We could achieve 99 percent accuracy in detecting ransomware instances from goodware samples and 96.5 percent accuracy in detecting family of a given ransomware sample. Our results indicate usefulness and practicality of applying pattern mining techniques in detection of good features for ransomware hunting. Moreover, we showed existence of distinctive frequent patterns within different ransomware families which can be used for identification of a ransomware sample family for building intelligence about threat actors and threat profile of a given target.

RanStop: A Hardware-assisted Runtime Crypto-Ransomware Detection Technique: Among many prevailing malware, crypto-ransomware poses a significant threat as it financially extorts affected users by creating denial of access via unauthorized encryption of their documents as well as holding their documents hostage and financially extorting them. This results in millions of dollars of annual losses worldwide. Multiple variants of ransomware are growing in number with capabilities of evasion from many anti-viruses and software-only malware detection schemes that rely on static execution signatures. In this paper, we propose a hardware-assisted scheme, called RanStop, for early detection of crypto-ransomware infection in commodity processors. RanStop leverages the information of hardware performance counters embedded in the performance monitoring unit in modern processors to observe micro-architectural event sets and detects known and unknown crypto-ransomware variants. In this paper, we train a recurrent neural network-based machine learning architecture using long short-term memory (LSTM) model for analyzing micro-architectural events in the hardware domain when executing multiple variants of ransomware as well as benign programs. We create timeseries to develop intrinsic statistical features using the information of related HPCs and improve the detection accuracy of RanStop and reduce noise by via LSTM and global average pooling. As an early detection scheme, RanStop can accurately and quickly identify ransomware within 2ms from the start of the program execution by analyzing HPC information collected for 20 timestamps each 100us apart. This detection time is too early for a ransomware to make any significant damage, if none. Moreover, validation against benign programs with behavioral (sub-routine-centric) similarity with that of a crypto-ransomware shows that RanStop can detect ransomware with an average of 97% accuracy for fifty random trials.

METHODOLOGY

Proposed Work:

The proposed system introduces a novel approach to ransomware detection on virtual machines (VMs). It collects specific processor and disk I/O events for the entire VM from the host machine. Machine learning, particularly a random forest (RF) classifier [52], is employed to develop a robust detection model. This method aims to avoid the monitoring overhead associated with continuous monitoring of every process on the target machine, reducing the risk of data contamination by ransomware. It also demonstrates resilience to variations in user workloads. The proposed system achieves fast detection with high accuracy for both known and unknown ransomware, with the [52] RF classifier outperforming other tested classifiers. In this paper additional enhancements were introduced, incorporating Convolutional Neural Network 2D (CNN2D) and an ensemble model with a voting classifier to further improve ransomware detection accuracy. The voting classifier, comprising multiple machine learning classifiers, demonstrated a remarkable 99% accuracy in making final predictions, showcasing the effectiveness of combining different models for robust detection.

System Architecture:

This paper investigates the fast detection of ransomware in execution on a Windows 10 virtual machine (VM). We collect both the HPC and disk I/O data at the host-machine level. The target (VM) is ignorant of the monitoring and data collection; also, there is little or no impact on its performance. [24] We use machine learning (ML)-based models to analyze the data and detect ransomware in execution [52]. Our method is particularly suited for protecting users of VMs in a cloud environment. We present an approach to detect ransomware accurately using HPC and disk I/O data observed from the host machine. Our approach avoids the overhead of monitoring many processes on the target machine and prevents data contamination by the ransomware designed to thwart such monitoring activities.

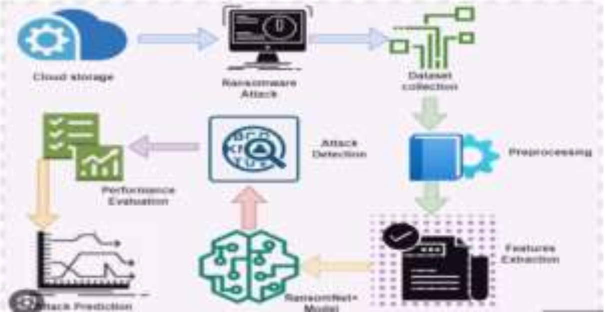
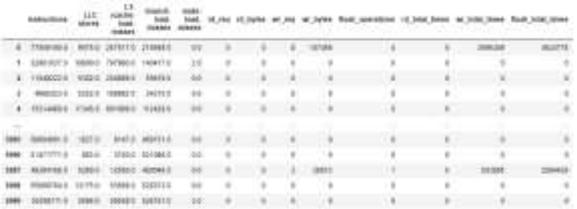


Fig 1 Proposed architecture

Dataset collection:

The HPC dataset utilized in the project consists of records capturing processor and disk I/O events during the execution of virtual machines. This dataset is carefully curated to represent a diverse range of system activities, providing a comprehensive foundation for training and testing ransomware detection models. With both known ransomware samples for model calibration and unknown samples for robustness testing, the HPC dataset facilitates a realistic simulation of potential ransomware behaviors in real-world computing environments [16], [17], [18], [19], [20].



	instructions	LLC stores	L1-icache-load-misses	branch-load-misses	node-load-misses	rd_req	rd_bytes	wr_req	wr_bytes	flush_operations	rd_total_times	wr_total_times	flush_total_times
0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	[illegible]	0	0	[illegible]	[illegible]
1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
...													
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	0	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0	0

Fig 2 Dataset

Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection, one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:
Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

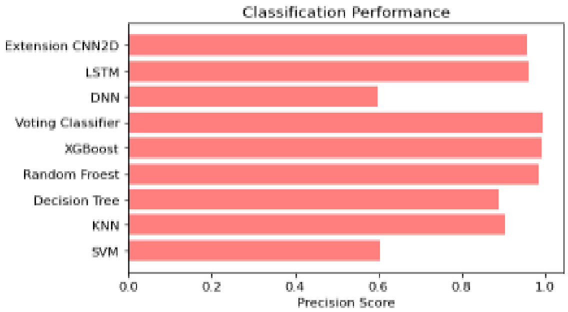


Fig 3 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP + FN}$$

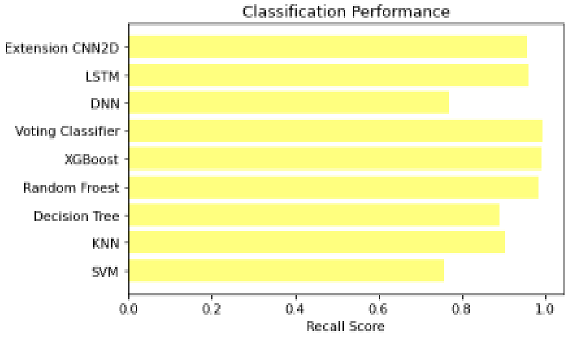


Fig 4 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 12, Issue 3, 2024

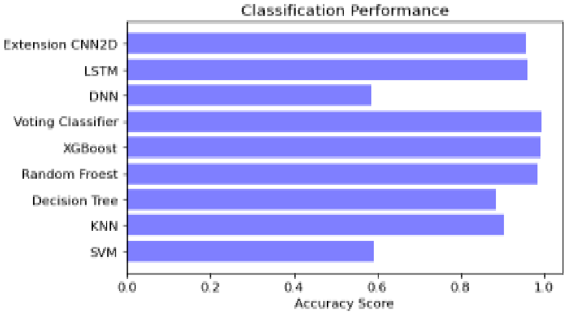


Fig 5 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

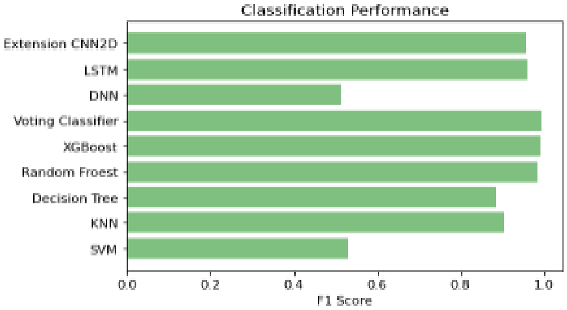
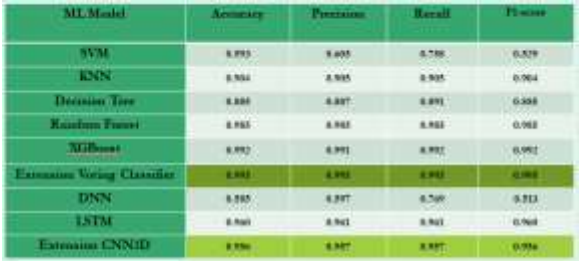


Fig 6 F1Score



ML Model	Accuracy	Precision	Recall	F1-score
SVM	0.593	0.605	0.788	0.529
KNN	0.904	0.905	0.905	0.904
Decision Tree	0.888	0.887	0.891	0.888
Random Forest	0.983	0.983	0.983	0.983
XGBoost	0.992	0.991	0.992	0.992
Extension Voting Classifier	0.993	0.993	0.993	0.993
DNN	0.585	0.597	0.769	0.513
LSTM	0.960	0.961	0.961	0.960
Extension CNN1D	0.956	0.957	0.957	0.956

Fig 7 Performance Evaluation

Welcome to Dashboard

About

Fig 8 Home page

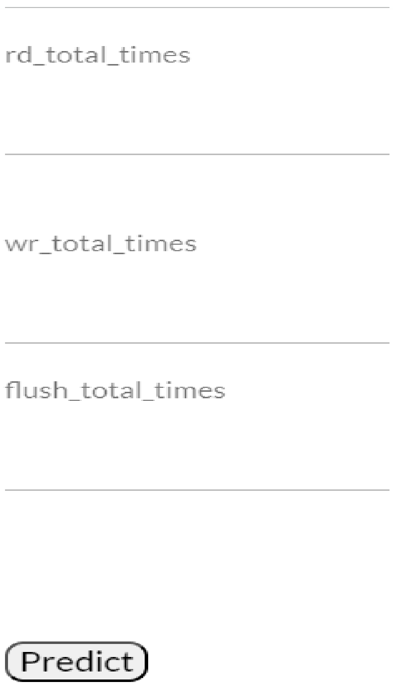


Fig 9 User input

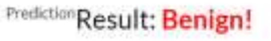

PredictionResult: Benign!

Fig 10 Predict result for given input

CONCLUSION

The project successfully introduces a novel strategy for detecting ransomware [9, 10, 11, 12, 13, 14,] using virtualization innovation, equipment execution counters, and IO occasions information to upgrade precision while limiting framework execution influence. The project examines a variety of machine learning algorithms, including SVM, KNN, Decision Tree, Random Forest, XGBOOST, DNN, and LSTM. Extensive experimentation reveals that Random Forest and XGBOOST consistently exhibit high accuracy in predicting ransomware activities [52]. The project investigates the efficacy of deep learning models, specifically DNN and LSTM. It provides useful insights into how well these models perform in comparison to standard machine learning algorithms and broadens the range of predictive methods. By making a dataset that comes from a variety of programs and is made available to the public, the project helps the cybersecurity community by encouraging collaboration and giving researchers a way to compare and contrast their models for detecting ransomware. A user-friendly interface where users can input data, have it preprocessed, and obtain predictions from the trained model is provided by the project, which seamlessly integrates Flask for the web framework and SQLite for user registration and authentication, enhancing practical applicability.

REFERENCES

[1] SR Department. (2022). Ransomware victimization rate 2022. Accessed: Apr. 6, 2022. [Online]. Available: https://www.statista. com/statistics/204457/businesses-ransomware-attack-rate/

[2] D. Braue. (2022). Ransomware Damage Costs. Accessed: Sep. 16, 2022. [Online]. Available: https://cybersecurityventures.com/globalransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/

[3] Logix Consulting. (2020). What is Signature Based Malware Detection. Accessed: Apr. 3, 2023. [Online]. Available: https://www.logixconsulting. com/2020/12/15/what-is-signature-based-malware-detection/

[4] W. Liu, P. Ren, K. Liu, and H.-X. Duan, ''Behavior-based malware analysis and detection,'' in Proc. 1st Int. Workshop Complex. Data Mining, Sep. 2011, pp. 39–42.

[5] (2021). Polymorphic Malware. Accessed: Apr. 3, 2023. [Online]. Available: https://www.thesslstore.com/blog/polymorphic-malware-andmetamorphic-malware-what-you-need-to-know/

[6] M. Loman. (2021). Lockfile Ransomware's Box of Tricks: Intermittent Encryption and Evasion. Accessed: Nov. 16, 2021. [Online]. Available: https://news.sophos.com/en-us/2021/08/27/lockfile-ransomwares-box-oftricks-intermittent-encryption-and-evasion/

[7] N. Pundir, M. Tehranipoor, and F. Rahman, ''RanStop: A hardwareassisted runtime crypto-ransomware detection technique,'' 2020, arXiv:2011.12248.

[8] S. Mehnaz, A. Mudgerikar, and E. Bertino, ''RWGuard: A real-time detection system against cryptographic ransomware,'' in Proc. Int. Symp. Res. Attacks, Intrusions, Defenses. Cham, Switzerland: Springer, 2018, pp. 114–136.

[9] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo, ''On the feasibility of online malware detection with performance counters,'' ACM SIGARCH Comput. Archit. News, vol. 41, no. 3, pp. 559–570, Jun. 2013.

[10] A. Tang, S. Sethumadhavan, and S. J. Stolfo, ''Unsupervised anomalybased malware detection using hardware features,'' in Proc. Int. Workshop Recent Adv. Intrusion Detection. Cham, Switzerland: Springer, 2014, pp. 109–129.

[11] 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 Proc. IEEE Symp. Secur. Privacy (SP), May 2019, pp. 20–38.

[12] S. P. Kadiyala, P. Jadhav, S.-K. Lam, and T. Srikanthan, ''Hardware performance counter-based fine-grained malware detection,'' ACM Trans. Embedded Comput. Syst., vol. 19, no. 5, pp. 1–17, Sep. 2020.