ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

TOWARDS CYBER SECURITY MACHINE LEARNING METHODS FRO MALWARE DETECTION AND CLASSIFICATION

Mr. K. Ram Mohan, Professor, Department Of ECE SICET, Hyderabad
Mandadi Akshitha,Kethireddy Vanaja,Macha Kumarmani,Kola Tharun
UG Student, Department Of ECE, SICET, Hyderabad

Abstract:
Malicious software, known as malware, continues to grow in complexity. Previous methods for detecting malware foc used on easily hijacked softwarebased detectors. Therefore, recent efforts are introducing hardwareassisted malware d etection. In this work, we propose a new hardwarassisted malware detection framework that leverages machine learni ng to monitor and classify memory access patters. The framework provides enhanced automation and coverage by red ucing dependency on usespecific malware signatures. Our work is based on the simple understanding that for access, malware must change th control and/or data structure to leave a trace in memory. Continuing this understanding, we p resent an online framework for malware detection that uses machine learning to classify malicious behavior based on virtual memory access patterns. The key elements of this framework include the process of writing and recording cont ext memory access to the structure of the function and the call, and the two-level classification of the structure.

Keywords: (Dynamic analysis, machine learning, malware, obfuscation techniques, static analysis, API calls, ransom ware.)

I. Introduction

The continuous growth of malicious software (commonly known as malware) poses a significant security threat that r equires constant research to well define control. The first step in malware detection involves scanning, which can be d one using static or dynamic methods. Often this review is done offline and involves the expertise of human experts. T he result of the analysis is compressed into a document called a "signature". One way to detect malware is to use stati c signatures to examine what happens after the program is installed and before it is executed. However, this approach can be bypassed by malware that uses cloaking techniques to evade detection. A researchbased behavioral approach h as been proposed in response to this challenge. This technology uses the operating system or hypervisor device to mo nitor behavior to detect malicious activity. Both static and dynamic signatures can be derived from deterministic or st atistical methods. Statistical methods use machine learning to discover patterns associated with bad behavior. Signatu re decisions, on the other hand, are often made through careful analysis by human experts.

A. Malware Definition:

The term “malware” is derived from “malicious software” and includes viruses, Trojans, worms, etc. It is a general ter m that refers to various types of threats such as. These programs have many capabilities, including gaining unauthoriz ed access without explicit permission, encrypting or extracting sensitive information, controlling or monitoring comp uter activity, and controlling user activity.

B. Types of Malware

1) Viruses: Computer viruses are generally unauthorized and can be installed without the user's permission, damaging

the computer, its operating system and its hardware. The effects of these viruses can be different and can be dangerou s for the overall performance and security of your computer.

2) Viruses: Computer viruses are programs that infect each other using communication between computers. Diseases and related diseases, ii. Viruses can be created like viruses, but instead of being distributed locally, they are distribute d to other computers. It uses computer networks to spread to other machines.

3) Trojan Horse: Trojan horse is a trick designed to create security vulnerabilities in the system and allow users to acc ess the system without permission. Trojan horses, unlike computer viruses, do not have the ability to replicate themsel ves. These “copycat” programs attempt to use shared information to gain access to systems and compromise their sec urity.

4) Ransomware: Ransomware is a type of malware that restricts access to the victim's computer and demands the pay ment of a ransom. The specific reimbursement price and stated reasons for payment may vary depending on the natur e of the disease.

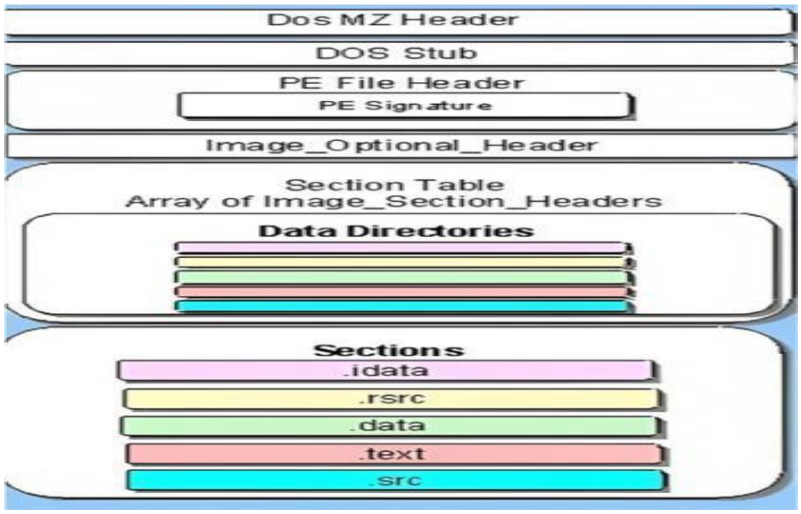
2. Purpose

The purpose of this research is to investigate the use of machine learning in malware detection to detect unknown mal ware. The goal is to develop software solutions that use machine learning to better identify unknown malware. The m ain goal is to verify the effectiveness of machine learning-
based malware detection by achieving high accuracy while minimizing false positives.

III. Methodology

A. Feature Extraction

There are many types of features in PE files, but most of them do not help distinguish malware from benign software. Based on our research on reliability and indepth analysis of PE file type features, we extracted 54 features from PE fil es that can distinguish benign software from malware. These features are documented. In the following discussion, w e provide a brief description of the extracted features.



INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

B. Algorithm:

1) Random Forest: Random Forest is another good method that uses trees with stable structure and random features to classify tasks. This algorithm creates a set of independent decision trees that can be easily parallelized. Each tree is constructed as a complete binary tree with randomly selected features (usually modified) used for branching. The leaves of the tree are processed according to the training data. The results are divided by each tree's predictions from the survey. This method works well when all features are relevant because only a small number of features are selected per tree. The beauty of random forest is that some trees can make predictions because they are not necessary, while other trees require important features and do so accurately based on training data.

2) Ada Boost: AdaBoost, also known as Adaptive Boosting, was proposed by Freund and Schapire in 1996 as a combination of boost. The goal is to increase accuracy by combining multiple classifications. AdaBoost uses an iterative approach to create a robust distribution by combining weak ones. During each iteration, the weights of the classifier and training samples are tested to ensure accuracy. The training process involves selecting a set of training data based on the accuracy of the previous iteration and giving more weight to incorrect observations. The weight of each classifier is proportional to its accuracy. This iterative process continues until the training data is correctly classified or the maximum number of iterations is reached.

3) Gradient Descent: Gradient descent is a technique that strengthens a weak learner or learning algorithm by changing it. It is based on the concept of Probabilistic Approximate Learning (PAC) and learns the complexity of machine learning problems. Gradient boosting classifiers rely on variable losses, such as log loss for classification or squared error for regression.

4) Decision tree: Decision tree is a hierarchical structure in which features are represented by muscles, decision rules are represented by branches, and results are represented by branches. leaves The top node is called the root node and the tree is split recursively based on the attribute value. Visualizing the decision tree is similar to a flowchart, making it easier to understand and interpret, reflecting people's emotions. Decision trees are nonparametric and do not require probabilistic assumptions. Decision trees have high accuracy and can perform well on large amounts of data.

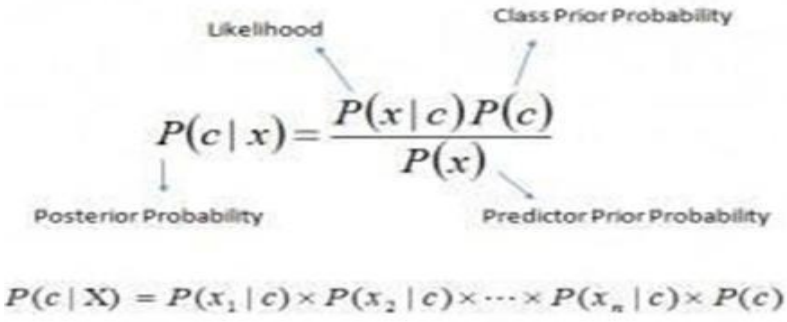
5) Naive Bayes: Naive Bayes is a simple but powerful classification algorithm based on Bayes' theorem. Naive Bayes calculates the probability that the data point belongs to each category and selects the category with the highest probability. Naive Bayes requires very little information to predict the appropriate action. It computes well and performs well on highdimensional datasets. However, the concept of freedom can limit its reality in some difficult situations. Bayes' theorem provides a way to calculate the posterior probability P(c−x) using the prior probability P(c), P(x), and probability P(x−c).The posterior probability P(c−x) represents the probability of the category (c, target) given the predicted variable (x, behavior). The prior probability P(c) is the initial probability of the category. Probability P(x−c) represents the probability of a predictor for a class. Finally, the prior probability P(x) refers to the initial probability of the predicted variable.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024



IV. FRAMEWORK and ARCHITECTURE

A. System Design

Signaturebased technology and behaviorbased technology both have advantages and limitations in malwar e detection. To take advantage of the advantages of both methods and overcome their shortcomings, many researchers have proposed a hybrid method that combines both static and dynamic methods for malware d etection. This section discusses various hybrid malware detection techniques and compares them based on various factors. Labek et al. (2003) proposed a method to detect data anomalies. They perform static check s to collect information about calls, including job titles, addresses, and return addresses. It combines static data with dynamic features by processing malware data in a dynamic control environment. An executable i s classified as malicious if it calls the same system call as the repository call (representing known malware). However, this will fail if the developer's malware injections do not block the call in the code. Collins et al. (2008) proposed a graph detector technique to detect errors in networks. They form a network represent ation where hosts are nodes and links are edges. This method simulates the network to observe the worm's behavior. It specifically targets viruses and does not target other types of malware such as Trojans or virus es. Mangiarado et al. (2015) introduced the FAMA framework to overcome the weaknesses of traditional a nd passive analysis methods and reduce negative feedback. Use IDA Pro to extract static features and Cuc koo Sandbox to capture dynamic behavior. The extracted features are then fed into the random forest and C5.0 algorithm to train the classifier. Experimental results show that the accuracy of separating bad data fr om bad data is as high as 95.75%. Shijo et al. (2015) proposed an integrated malware detection system. To find the code, they disassembled the binary file and removed the printed lines, accounting for the addition of unwanted printed strings. Overall, this work discusses various hybrid malware detection strategies that a im to increase the benefits of static and dynamic detection while also addressing their limitations.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

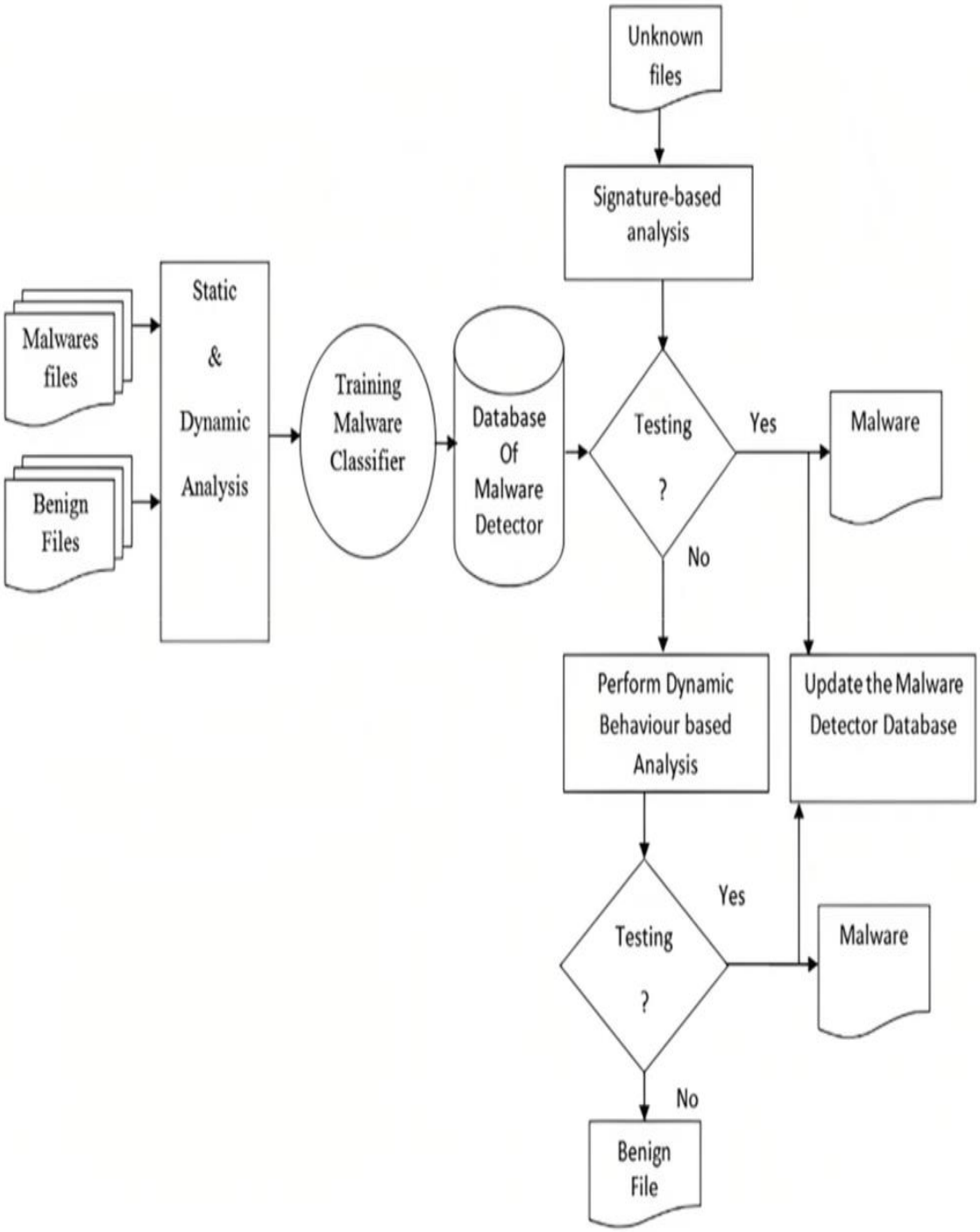


Fig. 3 Schematic Architecture of the proposed Hybrid Malware Detection Technique

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

B. Hypervisors

Hypervisors play an important role in malware detection when using machine learning for a variety of reasons. First of all, the hypervisor provides a secure and controlled environment from which malware can potentially at tack. When run in an administratorcreated virtual machine (VM) environment, malware runs in a standalone sa ndbox, isolated from the host operating system and other applications. This isolation helps prevent malware fro m infecting the root of the system, ensuring the security and integrity of the host. Excluding the middleman als o serves another important purpose: evading the malware itself. Malware often uses techniques to determine wh ether it is running in a virtual environment or being monitored by security tools. By leveraging the virtual mach ine hypervisor to provide malware with an environment that simulates the real operating system, malware will be less likely to be detected by existing monitoring and analysis tools. This increases the risk of detecting malw are behavior and discovering its malicious intent. Hypervisors allow snapshots or checkpoints to be taken durin g malware attacks. Snapshots capture the state of the virtual machine at different points in time, thus preserving the actual state of the malware at every stage of its execution. These snapshots help identify malware behavior, allowing researchers to investigate system changes, monitor network interactions, and determine the impact of potential vulnerabilities. Additionally, these snapshots can be used to create training data for machine learning models and classify across various malware samples. allows training. Another advantage of using Hypervisor is the redundancy it provides. Researchers can easily recreate a successful environment by going back to previou s captures, ensuring that experiments and analyzes can be repeated without regularity. This repeatability is nece ssary to conduct rigorous testing, compare different detection methods, and prove the effectiveness of machine learning algorithms in malware detection. In summary, hypervisors are an important part of machine learning f or malware detection. Its ability to isolate malware, run detection, provide quality control, facilitate monitoring and analysis, enable snapshotbased analysis, and enable replication make it an important tool for developing po werful and accurate malware.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940

www.ijasem.org

Vol 18, Issue 1, 2024

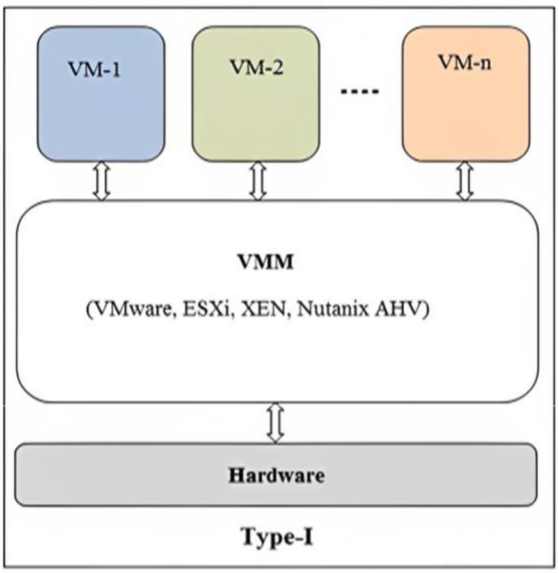


Fig. 1. Architecture of Hypervisor Type-I.

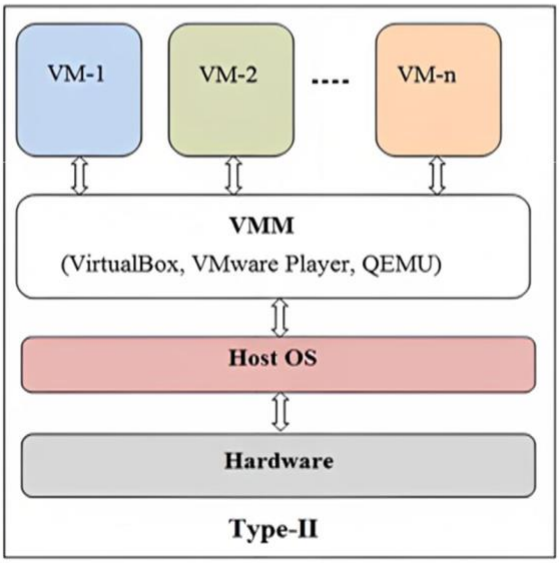


Fig. 4 Type II Hypervisor Architecture

C. Architecture

Behavior-
based malware detection methods rely on identifying the malicious activity that occurs when malware is e xecuted. This approach includes various features (APIs, browser events, event events, network events, etc.) to define behavior. These parameters fall into three main categories: profile operation, access operation, and network operation. The main idea behind malware detection is stealth detection, which is the unusual behavior of malware. During the negative detection process, the malware detector is trained by analyzing only positive data. Through static or weak analysis, benign data is analyzed and normal functions are used to train the classifier. In contrast, anomaly and benignbased methods involve analysis of benign files as w

ell as malicious files, making them the best method to distinguish between malicious and active bad. This method can detect both regular and malware. However, training researchers with this method is more time consuming than traditional methods. Heuristics are an extension of behaviorbased malware detection. Co mpared to traditional malware detection methods, machine learning plays an important role in detecting c omplex malware.

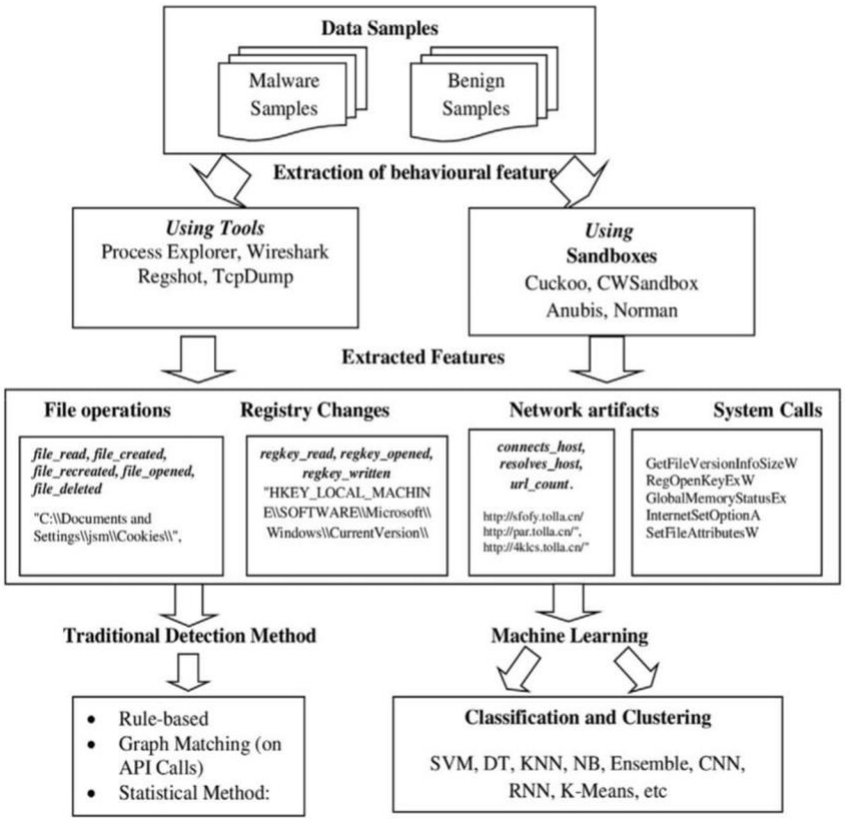


Fig. 5 Behavioural and machine learning based Architecture

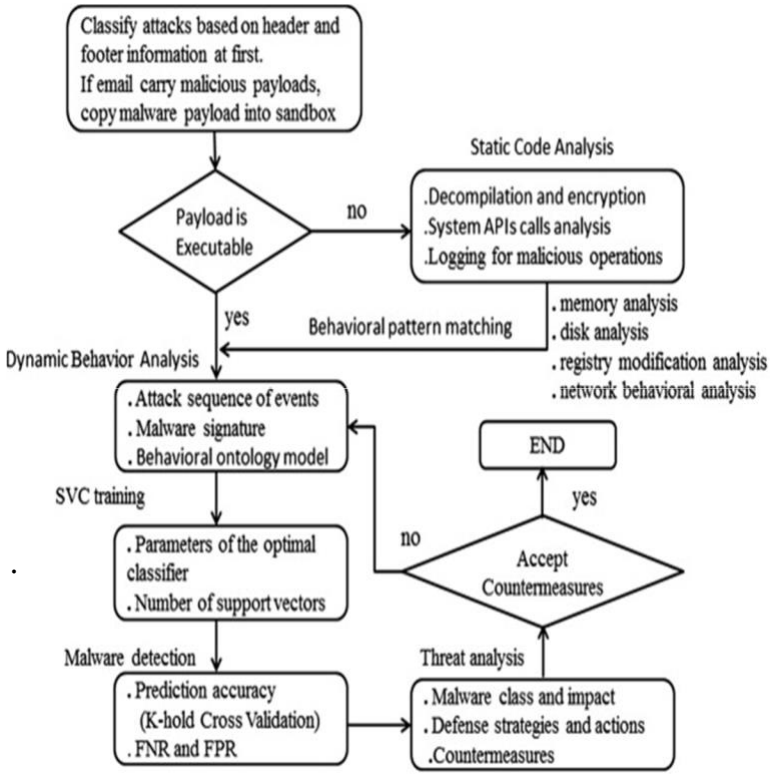


Fig. 6 Use Case Diagram

CONCLUSION

The project aims to improve malware detection by leveraging runtime features. Malware is known for its intelligence and rapid evolution. Malware analysis involves extracting important information from malwa re to identify and classify the malware. Two main techniques are used for malware analysis: static analysi s and dynamic analysis. Signature-based (antivirus) and behavior-
based malware protection are based on this technology. However, signature methods faced two major pro blems: They cannot detect new or unknown malware, and they can be easily bypassed by malware varian ts. On the other hand, behavior-
based techniques can identify new and unusual viruses, and dynamic techniques are more resistant to mal ware obfuscation than signature techniques. However, using dynamic methods can be unpredictable and t ime-consuming; signature methods, on the other hand, can detect known malware quickly and efficiently.

VI. Acknowledgments

We are grateful to D.Y Patil College of Engineering and Technology, Ambi, Pune for giving us the oppor tunity to undertake this study as part of completion of Bachelor of Engineering course. We want to thank the professor. We thank Madhavi Patil and all the faculty members of the Department of Computer Scien ce, BE for their support and guidance in carrying out this project. We thank D.Y Patil Institute of Engine ering and Technology, Pune for allowing us to work on this project and showing them good working met hods that helped us successfully design and build malware using machine learning. Finally, sincere thank s to our members, mentors and all our supporters for their guidance, support, valuable advice and constru ctive criticism.

References

[1] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, MalDAE: Malware detection and interpret ation based on correlation and combination of static and dynamic features, Computer. Trustw orthy. (2019) 208–233, http://dx.doi.org/10.1016/j.cose.2019.02.007.

[2] P. Burnap, R. French, F. Turner, K. Jones, Malware classifications using self-organizing feature maps and machine activity data, Comput. Secur.73 (2017) 399–410, http://dx.doi.org/10.1016/j.cose.2017.11.016, http://linkinghub.elsevier.com/retrieve/pii/S01674048173
br> . [3 A. Damodaran, F.D. Troy Visagio, California, T.H. Austin, M. Stamp, Comparison o f static, dynamic, and hybrid analysis for malware detection, J. Comput. Disease liability. Ha cking machine. 13 (1) (2017) 1-24, http://dx.doi.org/10.1007/s11416-015-0261-z.
[4] E.M. Dovom, A. Azmoodeh, A. Dehghantanha, D.E. Newton, R.M. Parizi, H. Karimipour , Fuzzy pattern tree for IoT edge malware detection and classification, J. Syst. Architect. 97 (March) (2019) 1-7, http://dx.doi.org/10.1016/j.sysarc.2019.01.017
[5] M. Ficco, F. Palmieri, Pages: opening high -
A Space cybersecurity training platform for real-
world edge IoT scenarios, J. Syst. Architect. 97 (September 2018) (2019) 107 129, http://dx.d oi.org/10.1016/j.sysarc.2019.04.004.
[6] K. Khan, A. Mehmood, S. Khan, M.A. Han, Z. Iqbal, W.K. Mashwani, A study on detecti on and protection in wireless ad-
hoc networks, J. Syst. architectural. (2019) 101701, http://dx.doi.org/10.1016/j.sysarc.2019.1 01701.
[7] A. Bushby, F. Cybersecurity: How is fraud changing cybersecurity defenses? A safe lie. c attle. 2019 (1) (2019) 12-14, http://dx.doi.org/10.1016/ S1361-3723(19)30008-9.
[8] E. Gandotra, D. Bansal, S. Sofat, Malware Analysis and Classification: A Survey, J. Cov ntaub ntawv. Secure.05 (02) (2014) 56–64, http://dx.doi.org/10.4236/jis.2014.52006.
[9] Raff, E., Barker, J., Sylvester, J. , Brandon, T., Catanzaro, B., Nicholas, C.K., … and Bra ndon, T. (2019). Check for malware by consuming the entire exe. arXiv preprint arXiv:1806. 04687.
[10] Chen, J., Jia, K., Chen, X., Cao, H., and Lu, Y. (2022). MalDroid: Android malware dete ction using convolutional neural networks. IEEE Transactions on Trustworthy and Secure Co mputing, 19(2), 405-418
[11] Zhang H., Shen Y., Li X., Yu S., and Lai X. (2021). Malware detection based on deep le arning and multivariate learning. IEEE Access, 9, 58207-58217.