International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 2, Mar-Apr 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Memory Forensics for Malware Detection in Virtual Environments

Saurav Bagul[1], Kunal Gilbile[2], Arti Thombare[3], Prof.Mrs. Amruta Sakhare[4]
Student, Department of Cyber and Digital Science, MIT Collage of Arts, Commerce, and Science

Abstract:

Every day, the complexity and quantum of vicious exercise increase and change. Conventional stationary law analysis is pointless when confronted with disparate variations. It is not unusual for there to be a daily prolog of new malware samples, and the malware created by bushwhackers can evolve as it spreads. As a result, automated dynamic malware analysis emerges as a highly fevered method for identifying unknown infections.

This research presents a dynamic malware analysis approach-based automated malware detection system. The way malware behaves is seen in the regulated setting of the well-known malware analysis program. It detects the existence of dangerous activity by classifying and clustering embedded malware behavior reports. Comparing the suggested system to the conventional hierarchical classification approach, it is clear from the evaluation and experimentation that it can achieve better F-measures, FPR, FNR, TPR, and TNR values, leading to accurate classification and more effective detection of unknown malware.

Key Words:- Clustering, categorization, malware, dynamic malware, and static code analysis

1. Introduction:-

Malware is a type of code created with harmful intentions. Its primary function is to threaten the privacy of users and their data, prompting a continuous evolution of malicious activities. Nonetheless, a significant threat comes from security researchers. Traditional methods widely used, such as signature-based and static malware analysis, necessitate that human analysts manually review the malicious code. Unfortunately, the rapid increase in the creation of malicious software has led antivirus companies to encounter thousands of new malware samples daily. Analyzing a large volume of files using this method has been hindered by techniques like obfuscation and polymorphism. Thus, having a dependable and automated analysis process is essential to effectively address this threat. The rise in the increasing availability of advanced tools has led to a rise in sophisticated cyber threats and attacks that are more targeted, persistent, and often unknown. Today's advanced malware is personalized, stealthy, and zero-day, in contrast to traditional malware, which is more generic, known, and one-off. Once infiltrated, these malware types conceal themselves, replicate, and disable protective measures on the host system. After installation, they connect to their command-and-control servers to receive instructions, which may involve data theft, spreading to other devices, or conducting reconnaissance.

Antivirus (AV) software was designed to identify, prevent, and eliminate malicious software. Its approach primarily relied on content-based signatures for the automatic classification and analysis of malware samples. Unfortunately, these methods are vulnerable to inaccuracies due to the use of polymorphic and metamorphic techniques. As a result, both Intrusion Detection Systems and AV solutions have struggled to achieve comprehensive success in malware analysis. Consequently, dynamic malware analysis tools are increasingly employed to automate the classification of malware and legitimate samples. Dynamic malware analysis entails executing binaries in a controlled setting to gather necessary information for identifying malicious activities.

The trend toward cloud infrastructure has been on the rise for years, offering opportunities for scalability, flexibility, and efficiency. However, despite its growth in contemporary contexts, critical attacks such as VM holes and cloud-specific threats remain concerning for the environment. Ensuring security and reliability is

essential in a virtualized setup. Numerous studies are currently exploring the virtualized environment, focusing on elements such as networks, Virtual Machine Managers, guest virtual machines, and Operating Systems (OS).

Cloud data centers support a variety of continuous services across private, public, and commercial sectors. These services must be secure and robust against cyber threats, component malfunctions, and configuration errors. However, the unique characteristics and inherent operational environments of cloud systems have complicated conventional detection mechanisms. The constraints related to prior attack signatures and payload data are the challenge that can be addressed by utilizing dynamic features such as per-flow meta-statistics derived from packet headers and volumetric data (i.e., counts of packets, bytes, etc.). In this context, dynamic analysis is crucial for cloud infrastructure.

In this paper, a framework for analyzing malware behavior through virtualization techniques is presented. The detection of malware files within cloud computing environments using machine learning methods is accomplished. The primary contribution is the reduction of the False Positive rate, which misclassifies harmful files.

The organization of the paper is as follows: Section II discusses related works, outlining system vulnerabilities and malware issues in virtualized environments, followed by the System Architecture in Section III. Section IV illustrates the implementation of the system with comprehensive details regarding each module using clustering and classification algorithms. Section V showcases the experiments, results, and evaluations, concluding with Section VI.

2. Related Works:-

Malware developers have devised numerous methods to identify the presence of malware analysis systems, but dynamic malware analysis effectively counters many of the hackers' tactics such as evasion, obfuscation, and polymorphism. Today, various online tools leverage dynamic analysis techniques that produce reports in a format easily understood by humans. The analysis system must have a suitable representation for malware, which is then utilized for classification based on similarity measures or feature vectors. However, the continuous influx of new malware samples to antivirus vendors daily necessitates an automated approach to minimize the number of samples requiring detailed human examination. Several artificial intelligence methods, especially those based on machine learning, have been utilized in existing literature for automated malware analysis and classification. The various studies conducted by researchers are outlined below. Watson et al. [1] highlighted an online cloud-based anomaly detection system using a one-class support vector machine

Bayer et al. [2] introduced a scalable method for clustering that detects and groups malware. The enhanced ANUBIS system utilizes taint tracking for behavioral analysis. One drawback of this method is its reliance on trace dependency. Additionally, there is a concern with dynamic data tainting, which allows for the injection of malicious binaries. Certain malware activates only under specific actions or events, such as setting a timer, which presents a significant limitation for this framework. Evasion techniques are also a challenge that cannot be circumvented in this approach, making it another disadvantage. Syarif et al. [3] examined both static and dynamic malware analysis techniques. The static method is performed without executing the system, while the dynamic method involves running the system. This paper emphasizes the main advantages and disadvantages of each technique along with their different applications. The static method relies on manual inspection, which is ineffective against evasion, obfuscation, and polymorphism techniques. Conversely, the advanced static method yields better results by providing more specific details concerning the malicious programs. Basic dynamic analysis can lead to DLL infections, which offer insight into the malware's actions. In contrast, the advanced dynamic analysis approach reveals more severe impacts caused by the malicious code within the system, such as disabling the firewall. The combination of these two strategies leads to a more effective solution for detecting malware activities.

Lorenzo et al. [4] introduced a framework for analyzing malware based on behavior. This framework enhances behavioral analysis for suspicious applications, enabling end users to operate within a secure

environment that mimics their original setting. System calls are utilized between the security lab and potentially harmful programs, allowing for the analysis of malware behaviors.

Lakshmanan et al. [5] developed a framework that employs a texture analysis method, which is resistant to packing strategies and effectively classifies a large collection of malware. This method offers flexibility against packing services and enables robust malware classification. The texture analysis relies on images generated during the execution of malicious code, which are used to represent the malware's behavior in similarity measures that aid in the classification process. When compared to other dynamic malware analysis techniques, texture analysis yields the best results, although its limitations arise from the differences in images within a single family and variations of the same malware.

Dilung et al. [6] introduced a comparative framework featuring the BareCloud concept, which enhances the detection rate of evasive malware, although further improvements can be made by adding more comprehensive filesystem-level event traces. The BareCloud utilizes hierarchical clustering techniques that effectively categorize malware into various clusters based on selected features. The experimental outcomes demonstrate superior performance when compared to alternative methods and approaches. The analysis of a vast number of malware samples facilitated the comparison of performance and accuracy among various available tools. Dilung et al. [7] introduced an automated method named MALGENE for extracting evasion signatures. MALGENE employs bioinformatics algorithms to automatically identify evasive behaviors in the form of system calls. Data mining techniques are utilized to analyze data events and call events to characterize malware behavior. These findings assist in creating evasion signatures, which are valuable for malware analysis. The primary techniques employed in this approach include system call alignment and parameter selection. The experiments are conducted at various levels, such as hypervisor and emulation.

Philip O'Kane et al. [8] suggested a method for malware detection utilizing N-gram analysis techniques. This approach examines the structure of programs by analyzing characters, strings, and bytes. The experiments detailed in this paper utilized an operational code (opcode) density histogram acquired during dynamic analysis. A reference model is established using a support vector machine to assess feature reduction methods. However, the relationships among features are intricate, thus making the use of straightforward filtering methods impractical. To yield improved results, the experiment employs the Eigen subspace analysis technique.

Konrad et al. [9] introduced an automated analysis framework that utilizes prototype-based feature vectors for clustering and classifying malware samples. The primary contribution from the authors is the ability to identify unknown malware while minimizing runtime. The proposed framework incorporates clustering, classification, and incremental analysis to execute dynamic malware assessment. As previously noted, many papers faced challenges with larger datasets. In this case, the implementation of incremental analysis has decreased the runtime when dealing with substantial amounts of data.

Vulnerabilities

This section provides an overview of vulnerabilities that are frequently targeted by malware. Malware primarily takes advantage of weaknesses in operating system architecture, applications, or specific versions of browser plugins. Flaws in the system's design can create vulnerabilities that enable hackers to inject their code with ease. Shared resources and services offered by the server to clients within the network present additional opportunities for attacks, including denial of service attacks, among others.

Malware in a virtualized machine

Virtualization has been gaining momentum for several years, providing avenues for scalability, efficiency, and adaptability. While end users reap the benefits of cloud services, there are also various threats, including vulnerabilities in virtual machines (such as rootkit attacks); and cloud-specific internet-based threats designed to infiltrate cloud networks (like malware and DDoS attacks on cloud services). As a result, security and resilience have become increasingly crucial. Numerous studies have explored different aspects of cloud security [16] [17] [18] at various levels, including the network, hypervisor, guest virtual machine, and operating system (OS).

3. System architecture:-

The primary objective of the proposed system is to automatically identify malware while minimizing the occurrence of false malware detections. Dynamic malware analysis is utilized to recognize any malicious behavior. Initially, the system captures the execution traces to pinpoint the harmful traces. The architectural representation of the proposed system is illustrated in Fig. 1, which primarily comprises an analysis module and a classification module. The analysis module focuses on preprocessing tasks, such as producing data that is suitable for the classifier tool. On the other hand, the classification module processes the preprocessed data to carry out accurate classification, thus distinguishing each sample according to its relevant neighbors.

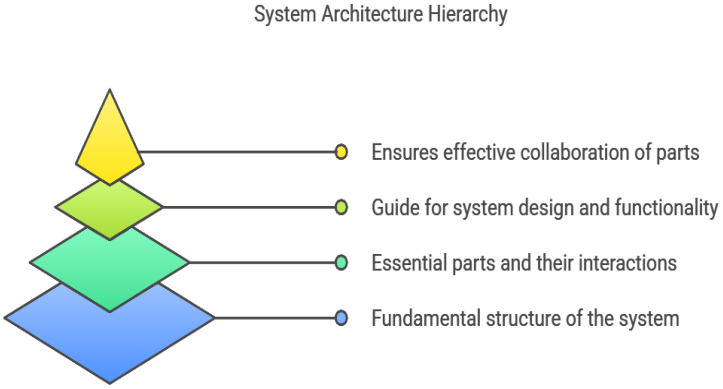

fig 1: Basic System Architecture

The primary objective of the proposed system is to automate the detection of malware while minimizing the occurrence of false positives. Dynamic malware analysis is utilized to identify malicious behaviors. The system begins by capturing execution traces, which helps in recognizing the harmful trace. Figure 1 illustrates the architectural overview of the proposed system, which primarily comprises an analysis module and a classification module. The analysis module focuses on preprocessing tasks, such as generating data that is appropriate for the classifier tool. In contrast, the classification module handles the preprocessed data to carry out accurate classifications, enabling the differentiation of each sample into its respective neighbors.

A. Analysis module

The initial step in malware analysis involves executing malicious traces within an instrumented environment. In this context, the frameworks Anubis and Cuckoo Sandbox are utilized to produce behavioral reports for both malware and benign samples. These frameworks operate atop the KVM hypervisor to prevent adverse effects on the host machines. This preprocessing phase aims to identify the most intriguing samples likely to exhibit harmful behavior. The malware samples that have undergone preprocessing are then provided to dynamic malware analysis tools to detect any malicious actions.

B. Classification Module

Malware identification is accomplished by classifying malware with the use of the Malheur tool. Accurate and dependable classification of malware leads to effective detection through the innovative techniques outlined in the subsequent sections.

4. Implementation:-

The role of protection and resilience is crucial for utilizing dynamic malware analysis within a cloud computing environment. To detect malicious code, the sample must be executed and analyzed to identify features that correspond to those of malware samples. The system employs the following methodologies to experiment.

A. Collection of samples

A collection of malware binaries is obtained from a major malware repository called Virusshare [10]. The framework's input is divided into a training dataset and a testing dataset. Several known samples are included in the training dataset, which is utilized to train the machine to recognize the characteristics of malicious code. The testing dataset comprises unknown samples that need to be classified based on malware features.

B. Generated report

After the sample is tagged using an antivirus provider like Virus Total, known samples are run in a sandbox environment to analyze the malware's behavior. The execution trace of the malicious software is recorded regarding system, network, file, and registry actions. The resulting report containing all this data will aid in the classification and identification of malware.

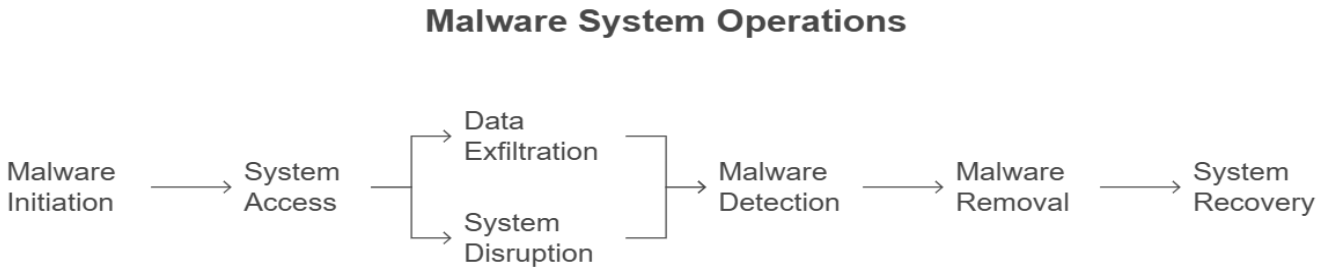

fig 2: Flow diagram showing overall activities of the system

C. Embedding Behaviour

The unprocessed behavioral report is not suitable for the classification process; therefore, it needs to be transformed into a more effective format that lessens the burden on the algorithms. First, system calls are converted into the MIST [19] format, which includes system calls along with their associated attribute values. MIST provides a substantial characterization of behavior.
However, this representation is not ideal for effectively analyzing malware samples, as these operate on a vectorial representation. The conversion of a report into vector format using instruction q-grams involves a set of instructions with a predetermined length of instruction patterns. This can be clarified with (1).

D. Distance Matrix

The embedded reports are analyzed against one another to create a distance matrix utilizing the Euclidean Distance formula. This approach assesses the behaviors of the reports, with values ranging from 0 to 4 upward, where 0 signifies identical behavior.

E. Clustering and Classification

The generated report is isolated and applied to subsequent processes, such as clustering strategies. Machine learning techniques are beneficial for analysing behavioral reports, which include methods like “behavior clustering” and “behavior classification.” Clustering and classification are employed to examine malicious behavior utilizing a hierarchical clustering method and an SVM approach. A report is initially selected as a prototype, either randomly or by a fixed method. Using this prototype, the reports containing malware are grouped into clusters. The clustering of these embedded reports organizes the data into significant groupings known as clusters. Each cluster embodies a shared characteristic among its objects while differing from other clusters. This process aids in identifying characteristics of unknown malware data across a larger sample size. The clustering algorithm described here illustrates the control flow between the creation of a

prototype and its comparison with the embedded reports. The process starts by selecting one report as a prototype and comparing it with other reports, leading to the nearest neighbors being grouped based on similar values. This ultimately facilitates the formation of clusters.

Algorithm for Clustering:

1. **Initialize** by considering each prototype as a cluster
2. Iteratively find and merge the close pair of clusters
3. **While** (min(distance) < d_c) (i.e., the Maximum distance between Individuals)
   do
   Cluster the nearest prototypes
   Update distance using complete linkage
4. **for** (x ∈ reports)
   do
   Assign nearest prototypes to x
   Reject Clusters with fewer
   than minimum members

The concept of machine learning enables systems to identify the characteristics of malware during the clustering phase. The identified traits are stored and utilized for the classification technique. This process is subsequently employed to detect unfamiliar malware categories by utilizing the prototypes generated in the earlier stages. Effectively classifying unknown malware necessitates accurately identifying the features in the behavioral patterns, which can be accomplished through clustering and classification methods, although it is constrained since the process operates in batches. Consequently, incremental analysis is necessary.

This approach utilizes previously classified behavioral reports along with stored prototypes to analyze a new behavioral report of unidentified malware. With the rise in malware generation, this method aids in classification by decreasing runtime. The classification algorithm outlined below is applied to categorize embedded reports into distinct clusters. The outcome of the classification distinctly separates malware based on their similar behaviors. This differentiation is beneficial for identifying traces of malware.

Algorithm for Classification:

1. **for** x ∈ reports **do**
2. **Determine** the nearest prototype in the training data
3. **If** (a nearest prototype is within the radius d_r) (i.e., the Radius of a cluster),

Assign reports to the particular cluster
Else
Reject as unknown

5. Result Analysis:-

The study consists of two sets of data: training and testing. Initially, the training samples are provided to the algorithms to learn the various features of the malware samples. This process is conducted in a controlled setting, specifically a sandbox. The execution traces are gathered in the form of a written report. However, the dynamic analysis of these malware samples within the sandbox offers detailed insights into the actions that occurred during the analysis, clearly illustrating different elements such as network, file system, registry, and other activities.

A variety of input samples are processed within the sandbox environment, and the results are documented. The time taken for executing various sample volumes is analyzed by measuring the time required.

6. Conclusion and Future Scope:-

Malicious actions in virtualized environments are increasingly worsening. The Internet presents vulnerabilities that attackers exploit to gain access to confidential information. Traditional and static methodologies have fallen short in this domain due to the tactics employed by hackers and the inability to

recognize new and unknown malware signatures. Out of the various techniques explored in automated malware analysis, dynamic malware analysis has demonstrated superior performance.

In this proposed method, the processes of extracting malware behavior, selecting the most effective features, clustering related prototypes, and classifying them into corresponding categories are executed, which aids in detecting malware samples within virtualized environments. This research primarily focuses on enhancing the clustering and classification of malware samples by identifying them as either malicious or benign. It has achieved a lower false positive rate (FPR) and false negative rate (FNR) compared to existing techniques. This solution can help address the challenges posed by virtual machine vulnerabilities and malicious actions by hackers. As virtual machines are interconnected with the host, they are safeguarded at the hypervisor level, thus ensuring a high.

Reference

[1] *Ulrich Bayer, Imam Habibi, Davide Balzarotti, Engin Kirda, Christopher Kruegel, "A View on Current Malware Behaviors", In USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET), 2009*

[2] *Michael R. Watson, Noor-ul-Hassan Shirazi, Angelos K. Marnerides, Andreas Mauthe, David Hutchison," Malware Detection in Cloud Computing Infrastructures", IEEE Transactions on Dependable and Secure Computing, DOI. 10.1109/TDSC.2015.2457918,2015*

[3] *Ulrich Bayer, Paolo Milani Comparetti_, Clemens Hlauschek_, Christopher Kruegel§, Engin Kirda," Scalable, Behavior-Based Malware Clustering", In Proceedings of the Network and Distributed System Security Symposium, 2009*

[4] *Syarif Yusirwan S, Yudi Prayudi, Imam Riadi, " Implementation of Malware Analysis using Static and Dynamic Analysis Method", International Journal of Computer Applications, Vol 117 – No. 6, May 2015*

[5] *Lorenzo Martignoni, Roberto Paleari, Danilo Bruschi, "A framework for behavior-based malware analysis in the cloud", 2009*

[6] *D. Kirat, G. Vigna, and C. Kruegel, "BareCloud: BareMetal Analysis based Evasive Malware Detection," in 23rd USENIX Security Symposium (USENIX Security 14). USENIX Association, 2014*

[7] *Dhilung Kirat, Giovanni Vigna, "MalGene: Automatic Extraction of Malware Analysis Evasion Signature", CCS'15, October 12–16, 2015*

[8] *Philip O'Kane, Sakir Sezer, Kieran McLaughlin, "SVM Training Phase Reduction Using Dataset Feature Filtering for Malware Detection", IEEE Transactions on Information Forensics and Security, Vol. 8, No. 3, March 2013*

[9] *Konrad Rieck, Philipp Trinius, Carsten Willems, Thorsten Holz," Automatic Analysis of Malware Behavior using Machine Learning", Journal of Computer Security, Vol. 19, pp. 639-668, 2011*

[10] *Virusshare. https://virusshare.com/*

[11] *ANUBIS. https://anubis.iseclab.org/*

[12] *CWSandbox. http://cwsandbox.org/*

[13] *Ether. http://ether.gtisc.gatech.edu/*

[14] *Cuckoo https://www.cuckoosandbox.org/*

[15] *Malheur http://www.mlsec.org/malheur/*

[16] *A. K. Marnerides, M. R. Watson, N. Shirazi, A. Mauthe, and D. Hutchison, "Malware analysis in cloud computing: Network and system characteristics," IEEE Globecom 2013, 2013.*

[17] *A. K. Marnerides, P. Spachos, P. Chatzimisios, and A Mauthe, "Malware detection in the cloud under ensemble empirical model decomposition," in Proceedings of the 6th IEEE International Conference on Networking and Computing, 2015.*

[18] *M. Christodorescu, R. Sailer, D. L. Schales, D. Gandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: A short paper," in Proceedings of the 2009 ACM Workshop on Cloud Computing Security, ser. CCSW '09. New York, NY, USA: ACM, pp. 97–102. 2009*

[19] *P. Trinius, C.Willems, T. Holz, and K. Rieck. A malware instruction set for behavior-based analysis. In Proceedings of 5th GI Conference "Sicherheit, Schutz und Zuverl¨assigkeit", Berlin, Germany, 2010*

