Grenze International Journal of Engineering and Technology, June Issue

Advancements in Automated Malware Analysis Techniques: A Comprehensive Study

Anvi Chauhan[1], Praveen Ailawalia[2] and Cyrus Mehra[3]
[1-2]Department of Mathematics, University Institute of Sciences, Chandigarh University, Gharuan-Mohali, Punjab, INDIA
Email: anvi344@gmail.com, praveen_2117@rediffmail.com
[3]University Institute of Computing, Chandigarh University, Gharuan-Mohali, Punjab, INDIA
Email: cyrus000suv@gmail.com

***Abstract*— In our digitally interconnected world, malware poses a persistent threat to information security. This paper explores automated malware analysis as a vital component of cybersecurity, offering a comprehensive overview of its fundamental concepts, methodologies, and contributions to combating malicious software.**
The study focuses on core principles, examining static and dynamic analysis techniques, behavioral analysis, and signature-based detection methods. It evaluates their strengths and limitations, emphasizing their relevance in addressing diverse malware threats.
Results indicate the effectiveness of both static and dynamic analysis, with dynamic analysis showing slightly better performance. Combining these techniques enhances detection accuracy, particularly for polymorphic malware.
The conclusion underscores the promise of automated malware analysis in signature-based detection and suggests future research directions for improving effectiveness and efficiency.

***Index Terms*— Automated malware analysis, Malware detection, signature-based detection, Comparative study, Code emulation, Sandbox, Cybersecurity.**

I. INTRODUCTION

Automated software analysis has become integral to modern software development, employing diverse tools and techniques to scrutinize code for potential issues and vulnerabilities. This practice ensures the creation of high-quality software that aligns with user and stakeholder requirements. Despite significant advancements in automated analysis tools, there remains a need to assess their effectiveness and their potential to replace manual analysis.[1]
Malware behavior analysis, a crucial aspect, is commonly conducted in sandboxes and virtual environments. These tools allow analysts to observe malware behavior without compromising system security, facilitating analysis at various levels of abstraction—from low-level code examination to high-level behavioral scrutiny.[2]
This research paper examines the current state of automated software analysis, encompassing tools and techniques. It evaluates the advantages and limitations of automated analysis, considering accuracy and efficiency in comparison to manual approaches. Additionally, the paper delves into the future of automated software analysis and its potential impact on software development practices.

Grenze ID: 01.GIJET.10.2.514

II. METHODOLOGY

A. Dataset

The study uses a dataset of known malware samples obtained from public repositories. The dataset consists of 1,000 samples of various types of malware, including viruses, trojans, and worms.

B. Background and Context of Malware threats

In the digital age, where computers and interconnected networks are integral to our daily lives, the term "malware" has become a household term, but its origins and implications are often underestimated. [3] This discussion aims to provide a comprehensive overview of the background and context of malware threats, highlighting their historical development, impact, and the ever-evolving nature of this cybersecurity challenge. [1]

C. Historical Perspective

Malware, or malicious software, encompasses a range of harmful programs designed to disrupt or damage computer systems[4]. Initially simple, early malware focused on annoyance or destruction, such as viruses spreading via floppy disks. Advancements in computer technology led to more sophisticated threats, notably in the 1980s with viruses like "Brain" and the 1990s with Internet-enabled worms like "ILOVEYOU" causing global economic damage.

In the realm of "automated malware analysis," research objectives involve using techniques from machine learning, data mining, and computer security to streamline the analysis of malicious software. This includes understanding its behavior and potential impact on systems.

D. Dynamic Behavior Analysis

Research ways to automate the execution of malware samples in controlled environments to observe their dynamic behavior.

Explore techniques like sandboxing, virtualization, and emulation to analyze how malware interacts with the operating system and other software. [2]

E. Feature Extraction and Representation

Identify relevant features or attributes that can be extracted from malware samples to facilitate automated analysis.

Investigate methods for representing malware samples in a format that is amenable to machine learning algorithms.

F. Threat Intelligence and Attribution

Develop automated methods for attributing malware to specific threat actors or groups based on analysis of code, behavior patterns, and infrastructure.

Explore ways to extract indicators of compromise (IoCs) and generate threat intelligence reports automatically.

G. Evasion and Polymorphism Detection

Research ways to automatically detect evasive techniques employed by malware to evade traditional detection mechanisms.

Develop methods for identifying polymorphic malware that dynamically changes its code to evade signature-based detection.

H. Automated Reverse Engineering

Explore automated methods for reverse engineering malware to understand its inner workings, including code decompilation and function analysis.

Develop tools to generate high-level representations of malware functionality from its binary code.

I. Behavioral Profiling

Study techniques for creating behavioral profiles of malware to identify patterns and anomalies in their actions.

Develop automated systems that can recognize deviations from normal behavior and trigger alerts. [2]

J. Scalability and Real-time Analysis

Address the challenge of analyzing large-scale malware datasets in real-time or near-real-time.

Research methods for parallelizing and distributing automated analysis tasks to handle high volumes of malware samples.

K. Human-Machine Collaboration

Investigate ways to combine automated analysis with human expertise to improve the accuracy of malware analysis.
Design interfaces and workflows that allow analysts to interact with automated systems effectively.

III. Comparison Of Manual And Automated Malware Analysis

A. Manual Analysis

Manual analysis involves human experts dissecting malware samples to gain a deep understanding of their behavior and capabilities. Analysts delve into the code, reverse engineer binaries, and analyze network traffic to identify malicious patterns.

IV. Automated Analysis

Automated analysis employs tools, scripts, and algorithms to process and evaluate malware samples quickly. Sandboxing and virtual environments are used to execute malware and observe behavior in isolation

A. Advantages of Automated Analysis

Speed and Efficiency: Automated systems can process large volumes of samples quickly, identifying known threats rapidly.
Consistency: Automation ensures a consistent approach to analysis, reducing the risk of human errors and variability.
Scalability: Automated analysis is well-suited for processing a large number of samples and dealing with widespread threats.
Indicators and Signatures: Automated systems can generate IoCs, behavioral profiles, and signatures for identifying and mitigating threats.

V. Techniques Used In The Automated Malware Analysis

Automated malware analysis employs various techniques and tools to efficiently and rapidly dissect malicious software. These techniques aim to uncover the malware's behavior, characteristics, and potential impact on systems. Here are some key techniques used in automated malware analysis:

1. Static Analysis

Static analysis involves examining the malware's code and structure without actually executing it. This includes analyzing the binary code, disassembling executable files, and inspecting script contents. Static analysis can reveal information about functions, libraries, and potentially malicious patterns, aiding in identifying the malware's purpose and behavior.

2. Dynamic Analysis

Dynamic analysis focuses on observing the malware's behavior during execution. This technique involves running the malware in a controlled environment, such as a sandbox or virtual machine, to monitor its interactions with the operating system and other software. Dynamic analysis can reveal network communication, file modifications, system calls, and other actions taken by the malware.

3. Sandboxing

Sandboxing involves executing the malware in an isolated environment designed to mimic a real system. This technique helps researchers observe the malware's behavior without risking infection of the host system. [8] Sandboxes can also capture changes made by the malware to files, registry keys, and system memory.

4. Network Traffic Analysis

Analyzing the network traffic generated by the malware can provide insights into communication with command and control servers, data exfiltration, and other malicious activities. Automated systems monitor network traffic to detect unusual patterns and connections.

5. Signature-based Detection

Signature-based detection involves comparing the characteristics of a file or code snippet to known malware signatures in databases. If a match is found, the file is flagged as malicious. While effective for detecting known threats, this technique can miss novel or modified malware.

6. Behavioral Analysis

Behavioral analysis focuses on identifying patterns of behavior that are indicative of malicious intent. This includes monitoring system calls, registry modifications, and file accesses during execution to identify actions that deviate from normal behavior.

7. Heuristic Analysis

Heuristic analysis involves using predefined rules or algorithms to identify potentially suspicious or malicious code patterns. While not as specific as signature-based detection, heuristics can identify previously unknown threats based on their behavior or structure.

8. Machine Learning

Machine learning techniques, such as supervised and unsupervised algorithms, can be employed to classify and detect malware based on features extracted from samples. Machine learning models can learn from historical data to identify new threats or anomalies.

VI. Combining Both Techniques

By integrating static and dynamic analysis, researchers can overcome limitations associated with each technique:

A. Behaviour Validation

Dynamic analysis validates the behavioral hypotheses generated from static analysis. If the static analysis indicates a specific behavior, dynamic analysis can confirm whether the behavior is actually exhibited during execution.

B. Contextual Understanding

Static analysis provides context for dynamic analysis by revealing code-level details and potential attack vectors. This information guides analysts in setting up a dynamic analysis environment that captures relevant behavior.

C. Code Unpacking and Decryption

Static analysis can identify code that's encrypted or packed to evade detection. Dynamic analysis allows the unpacked or decrypted code to be observed and analyzed in its fully functional state.

D. Evasion Techniques

Malware often employs techniques to evade automated analysis, such as checking for virtual environments. Combining both techniques allows analysts to identify and adapt to evasion attempts.

E. Custom Signatures

Static analysis can provide insights into unique code structures. Analysts can use this information to create custom signatures for detection during dynamic analysis.

F. Interaction with Network and Files

Dynamic analysis reveals how the malware interacts with the network and modifies files. This information can help assess the malware's impact on the system and its intended purpose.

VII. A Comparison Statement For Various Automated Tools Available In The Market

A. Experiment and testing on sample data

For the demonstration, we will be testing the sample of one of the most dangerous ransomware in the history of cyber-attacks.

WannaCry, also known as WannaCrypt, is a notorious ransomware attack that occurred in May 2017. It was one of the largest and most widespread cyberattacks in history.

WannaCry served as a wake-up call for organizations and individuals to take cybersecurity seriously and regularly update their systems to protect against ransomware and other cyber threats. It also raised awareness about the importance of cybersecurity practices and the potential consequences of not maintaining secure systems. [5]

B. VMRay Sandbox

VMRay's sandboxing solution typically involves running potentially harmful files or code in a virtualized and isolated environment to observe their behavior and identify any malicious activities. [6] This allows security teams to understand the nature of the threat, gather information about its capabilities, and develop strategies to defend against it.

C. Sample Details

MD5 : 3b9fa46d89fd099e914d6275cac9171f
SHA1 : 6236eff5dc07a222bbf60f4e62225f2052b1f55a
SHA256 : 042889c7d19b1c7aec30d171c1e147953907146ff5eb81e3bfc29ca83e962658
SSDeep :12288:yebLgPlu+QhMbaIMu7L5NVErCA4z2g6rTcbckPU82900Ve7zw+K+DF:zbLgddQhfdmMSirYbcMNgef0Y
Authentihash : 32aa0ad321e2bb8fe0b75854983ecb32f5a8fdff40839ce472988c30edcb9659
ImpHash : 2e5708ae5fed0403e8117c645fb23e5b
Size : 5,267,459 bytes
File Type : PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
Mime Type : application/x-dosexec
Extension : exe

D. Results and findings

We have analyzed the sample in the VMRay Environment and have observed that the VMRay analyzed the sample in the live test environment and classified the sample as Ransomware (as shown in Fig.1). The analysis took a timestamp of nearly 2.6 Minutes approx.

Fig. 1: sample in VMRay

Along with analyzing the sample in the real-time, the VMRay is also capable of analyzing(as shown in Fig.2) and extracting the IOCs for immediate containment actions.

Fig. 2: analyzing IOC

The IOCs can be filtered and categorized on the basic of various categories like File Type, File Name, IP and Processes.

E. Sandbox Preview

Fig. 3: The sandbox preview

The sandbox environment analyzed the sample(Fig.3) and created the process overview:

1. Deletes file after execution
2. Delays execution

3. Modifies operating system directory
4. Creates process with hidden window
5. Installs system service
6. Connects to remote host
7. Drops PE file
8. Executes dropped PE file (as shown in Fig. 4)

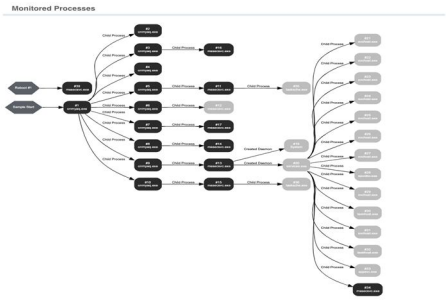

Fig. 4: monitored processes

F. Hybrid Analysis Sandbox

Hybrid Analysis is a well-known online sandbox tool and malware analysis platform that is used by cybersecurity professionals, researchers, and organizations to analyze and understand the behavior of potentially malicious files and executables. It combines both static and dynamic analysis techniques to provide comprehensive insights into the behavior of files and their potential threat level.

We have analyzed the same sample in the Hybrid Analysis sandbox and observed the similar functionality but with in an efficient way.

Hybrid Analysis had performed the risk assessment (as shown in Fig. 5) for the respective sample and clearly specify that Network Behaviour, System behavior, and spreading sensitivity.

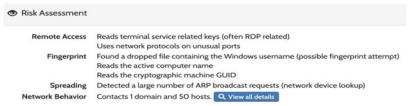

Fig. 5: Risk assessment of Hybrid sandbox

Hybrid analysis is mainly associated with the combination and the usage of the various other integrated technologies like Crowdstrike Falcon, VirusTotal, and Defender but with the limited functionality via API.

G. VirusTotal Sandbox

VirusTotal is not primarily a sandbox tool, but rather a web-based service and platform that provides a wide range of security-related features, including file and URL analysis. While it doesn't offer the dynamic analysis capabilities of a traditional sandbox, it's a valuable resource for assessing the safety and reputation of files and URLs. Here's how VirusTotal works and some of its key features:

H. File and URL Scanning

Users can upload files or submit URLs to VirusTotal for analysis. The platform scans these items using a multitude of antivirus engines and security tools to check for known malware signatures and indicators of compromise.

I. Multi-Engine Scanning

VirusTotal integrates with over 70 different antivirus engines, allowing it to provide a comprehensive assessment of the submitted files and URLs.

J. Results and Reports

After scanning, VirusTotal generates detailed reports that show which antivirus engines detected the submitted item as malicious or suspicious. It also provides additional context and information about the file or URL.

K. Community Contributions

Like Hybrid Analysis, VirusTotal benefits from a user community that contributes analysis results, comments, and additional information to help assess the reputation and safety of files and URLs.

L. Integration

VirusTotal can be integrated into existing security workflows and tools through its API. This allows for automated scanning of files and URLs as part of an organization's security processes.

M. Threat Intelligence

VirusTotal offers various threat intelligence features, such as historical scans, metadata analysis, and connections to related domains and IP addresses, which can help security analysts investigate and understand potential threats.

N. YARA Rules

Users can apply custom YARA rules to their submissions to enhance the detection of specific patterns or behaviors. We have performed the similar analysis with the VirusTotal and have observed (as shown in Fig. 6) one of the best feature supported by the VirusTotal is that the web based tol is capable of analysing and finding the relations like Contacted URLs, Contacted Domains, Contacted IP addresses, Execution Parents, and Dropped Files for further containment actions.

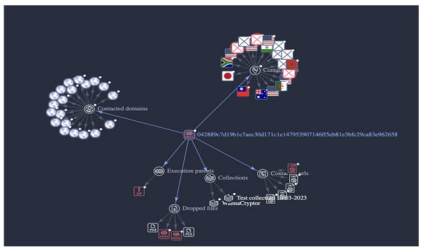

Fig. 6: Analysis with Virus Total

VT is also cabapable for generating and classifying the bahabiour on the basic of MITRE ATT&CK Tactics and Techniques.
These sandbox platforms are very useful in understand the behaviour of the malware or ransomware. The Network behaviour can be understand on the basic of HTTP Requests, DNS Resolutions, IP Traffic, JA3 Digests and TLC Connections. [6]. The File or System behaviour is associated with File system actions, Files Opened, Registry Keys Opened, Registry Keys Set, Registry Keys Deleted, Processes Created, Shell Commands and Services Created. [3]

VIII. Significance Of Machine Learning In Automated Malware Analysis:

Addressing these challenges requires a combination of approaches, including continuous tool development, machine learning techniques, hybrid analysis (combining automated and manual approaches), and collaboration among security experts. As the cyber threat landscape evolves, so too must the techniques and technologies used in automated malware analysis. [7]
Machine learning (ML) plays a significant role in modern malware analysis, offering advanced capabilities for detecting, classifying, and understanding malicious software. [8]. ML techniques leverage patterns and features extracted from malware samples to build predictive models that can aid in automated analysis and threat detection. Here's how machine learning is utilized in malware analysis:

1. Malware Detection

ML models can be trained to recognize patterns in malware samples, enabling them to classify files as either benign or malicious. These models learn from labeled datasets containing examples of both malicious and benign files, and they can subsequently classify new samples based on the patterns they've learned.

2. Feature Extraction

Machine learning algorithms can automatically extract relevant features from malware samples. These features can include API calls, system calls, byte sequences, file headers, and more. These extracted features serve as input to ML models for classification and analysis.

3. Behavioral Analysis

ML can help in understanding the behavior of malware by learning normal patterns of system behavior and identifying deviations. Anomalies in system calls, network traffic, or other behaviors can signal the presence of malware.

4. Heuristic Rule Generation

ML can be used to generate heuristic rules that identify suspicious or malicious patterns in code. These rules can then be used by security tools for automated analysis.

5. Zero-Day Threat Detection

ML models can learn to detect previously unknown threats by identifying anomalies that deviate from known patterns. This is especially valuable for detecting zero-day exploits.

6. Dynamic Analysis

ML can enhance dynamic analysis by analyzing the behavior of malware during execution. By learning from past executions, ML models can predict whether a particular behavior is malicious or benign.

7. Feature Engineering

Machine learning techniques help identify which features are most relevant for distinguishing between malicious and benign samples. This process, known as feature engineering, enhances the accuracy of classification models.

8. Ensemble Methods

Ensemble methods combine multiple ML models to improve accuracy and robustness. They can help reduce false positives and negatives in malware detection.

IX. Case Studies In Automated Malware Analysis

Case studies offer real-world insights into how automated malware analysis tools and techniques have been applied to identify, analyze, and mitigate various types of malicious software. [7]. These studies provide valuable lessons about the effectiveness, challenges, and outcomes of using automated approaches in cybersecurity. Here are a few examples:

A. Stuxnet Worm

The Stuxnet worm is a well-known case where automated malware analysis played a crucial role. Stuxnet targeted industrial control systems and was designed to disrupt Iran's nuclear program. Researchers used automated analysis to reverse engineer the worm, uncovering its complex behavior, propagation methods, and specific targeting of Siemens industrial software. This case highlighted the importance of automated analysis in dissecting sophisticated threats and understanding their intricate functionalities.

B. WannaCry Ransomware

The WannaCry ransomware attack spread globally in 2017, affecting hundreds of thousands of systems. Automated malware analysis tools were instrumental in quickly identifying the ransomware's behavior, propagation techniques, and the kill switch domain. [5] These insights aided security professionals in developing countermeasures and containing the attack's impact.

C. Emotet Botnet

The Emotet botnet is an example of how automated analysis can be used to track and mitigate the activities of a large-scale cybercriminal operation. By monitoring network traffic, system behavior, and patterns of malware distribution, researchers and security teams were able to map out Emotet's infrastructure, enabling law enforcement to dismantle the botnet.

X. Outcomes And Lessons Learned

*Efficient Threat Detection***:** Automated malware analysis accelerates threat detection, allowing security teams to quickly identify new and emerging threats, assess their behaviour, and develop mitigation strategies.
Behavioural Insights: Automated analysis tools provide insights into the behaviour of malware, uncovering patterns and actions that can be used to develop effective countermeasures.
Scalability: These tools enable the processing of large volumes of malware samples, making it possible to analyze and categorize threats on a massive scale.

Rapid Response: Automated analysis aids in swift incident response by providing timely information about the nature of the threat and potential impacts
Adaptability to Evasion Tactics: While malware authors deploy evasion techniques, automated systems can adapt by learning from new data and evolving tactics.
Early Warning: Automated analysis tools can serve as early warning systems, detecting and flagging potential threats even before they are widely recognized
Human Expertise Integration: Despite automation, human analysts play a critical role in contextual understanding, creative problem-solving, and decision-making based on the insights provided by automated systems.
False Positives and Negatives: Case studies reveal that while automated tools are powerful, they can produce false positives and negatives. Combining automated analysis with human expertise mitigates these shortcomings.
Collaboration and Threat Intelligence: Automated analysis platforms facilitate collaboration among researchers and analysts, sharing insights and threat intelligence that contribute to a collective defense against cyber threats.
Continuous Learning: Case studies underscore the importance of continuous learning and adaptation in the realm of automated malware analysis to keep pace with evolving attack techniques.

REFERENCES

[1] Mohaisen, O.Alrawi and M.Mohaisen, "High-fidelity, behavior-based automated malware analysis and classification" Computers and Security, Volume 52, Pages 251-266,July 2015 ,10.1016/j.cose.2015.04.001
[2] Mohaisen and O Alrawi, " High-fidelity, behavior-based automated malware analysis and classification," lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Volume 52,Pages 251-266,July 2015,10.1007/978-3-319-15087-1_9
[3] J.M Ceron., C.B Margi and L.Z Granville," From traffic containment to network reconfiguration in malware-analysis systems," Computer Networks, Volume 129, Part 1, Pages 261-272,24 December 201710.1016/j.comnet.2017.10.003
[4] Lever,P. Kotzias, D.Balzarotti, J. Caballero and M. Antonakakis," Lustrum of Malware Network Communication: Evolution and InsightsProceedings" - IEEE Symposium on Security and Privacy,10.1109/SP.2017.59
[5] Y.L. Aung,M. Ochoa, J. Zhou," A Practical Attack Detection and Live Malware Analysis System for IoT Threat Intelligence,"Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 13640,Dec 2022,10.1007/978-3-031-22390-7_19
[6] A.Lakhotia, V.Notani and C.LeDoux,"Malware economics and its implication to anti-malware situational awareness,"2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, June 2018 CyberSA 201810.1109/CyberSA.2018.8551388
[7] J.Van Randwyk ,K. Chiang, L. Lloyd and K. Vanderveen," An automated malware analysis environment,"Proceedings - International Carnahan Conference on Security Technology,Jan 2019,10.1109/CCST.2008.4751322
[8] A.R. Chukka ,and V.S. Devi,".Detection of Malicious Binaries by Applying Machine Learning Models on Static and Dynamic Artefacts," International Conference on Internet of Things, Big Data and Security, IoTBDS – Proceedings