IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024

Malware Detection and Analysis Using YARA Tool

Bhargavi Jadhav[1] and Malhar Jadhav[2]
Department of Computer Engineering
Cummins College of Engineering, Mumbai, India[1]
HDFC, Mumbai, India[2]

Abstract: *Malicious software, or malware, is one of the biggest threats to the vast amounts of data and files handled today. One widely used tool for malware detection is YARA(Yet Another Recursive Acronym), which uses YARArules to match suspicious content in files or network packets analyzed by antivirus engines. It works with most hosts running Windows, Linux, or Mac operating systems. However, while YARA is effective, its scanning process can be quite slow, especially with large datasets. This paper explores ways to optimize YARA's scanning process to make it faster and more efficient. We discuss various modes available within YARA to enhance performance, such as fast mode for quicker scanning and recursive mode for in-depth analysis. By fine-tuning these settings, we aim to reduce scan times without compromising detection accuracy.*

Keywords: YARA tool, Malware Detection, Yara Modes, Pattern Matching

I. INTRODUCTION

Malware refers to malicious software designed to infiltrate computer systems without the user's knowledge or permission. It can range from being intrusive and harmful to seemingly harmless, yet its presence can still cause significant disruption. To tackle this, industries commonly rely on techniques like dynamic analysis and pattern matching, as they provide effective ways to identify malware samples. There are various types of malware, including viruses, worms, Trojan horses, ransomware, and spyware, each differing in their purpose and method of spreading. Some forms enable unauthorized surveillance, trigger annoying pop-up ads, or send emails from a compromised account without the owner's awareness [1]. In more severe cases, malware can compromise the system entirely, leading to critical damage from within. Before diving into the malware detection tool and analysis, it is important to understand the different types of malware.

1.1 Ransomware

Ransomware is a form of malware that locks and encrypts the victim's essential files, demanding a ransom payment in exchange for restoring access. [2]
These attacks exploit weaknesses in systems, networks, software, or human behavior to infiltrate the target device [2]. The affected devices could range from computers and smartphones to printers, wearables, point-of-sale (POS) terminals, or other endpoints.

1.2 Worms

This is a type of standalone malware that replicates itself to spread to other computers. Worms can change, delete, or corrupt files on their own [3]. It is considered deathly as it can introduce other malware. There are different types of worms such as Email worms, Internet worms, and Net worms.

1.3 Botnet

A Botnet is a network of connected devices, such as computers, and smartphones, that are infected with malware and controlled remotely by a Botmaster.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024

These are malicious actors that operate without the owner's permission. Botnets are used for spam campaigns, data theft, and crypto-jacking.

II. MALWARE DETECTION USING YARA

YARA is a general-purpose tool that uses a rule-based approach to identify malware based on signature detection, such as text or binary patterns. Rules or descriptions are created from strings and logic, and they match patterns or features to classify the sample according to specific malware families or variants [1]. Figure (1) displays the general syntax of any YARA rule. While the metadata is not mandatory, the rule name, string, and condition are necessary to define any rule.

```
rule with_sqlite : sqlite
{
   meta:
        author="Julian J. Gonnzalez <info@seguridadparatodos.es>"
        reference="http://www.st2labs.com"
        description="Detection of Sqlite data in raw image"

   strings:
        $hex_strings={53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 33}

   condition:
        all of them
```

The programmability of YARA rules, whichanybody can develop or modify, and theincreased depth of malware information theygive are further benefits. Still, knowledge ofmalware analysis is necessary for designinggood YARA rules, as ineffective YARA rulesmight let malware avoid detection.[4].The pattern-based approach identifies harmful files by examining their structure, looking for specific indicators like strings, URLs, IP addresses, or processes. These patterns remain consistent, meaning any modification to the file's content would still exhibit the same core symptoms or characteristics.

While the last few bytes of the file remain static, the rest of the signature can vary. To address this, we identified a recurring pattern that ensures the file consistently begins with the same sequence of bytes. Additionally, these indicators help uncover trends and relationships common to malicious files, enhancing detection accuracy. Magic bytes, also known as magic numbers, are unique sequences of bytes found at the beginning of a file that serve as identifiers for the file's format or type [6]. YARA uses these bytes in their rules to quickly identify malicious files that may attempt to masquerade as legitimate ones. Here is an example: -

Application	Type	Details
JPEG Images	.jpg or .jpeg	Start with the magic bytes FF D8 FF
PNG Images	.png	Begin with 89 50 4E 47 0D 0A 1A 0A
PDF Documents	.pdf	Have magic bytes %PDF at the beginning
ZIP Files	.zip	Often start with 50 4B 03 04

YARA has a performance tool that allows various parameters to pass in the command line such as: -

- *Recursive Scanning:*Recursive scanning ensures that all files within the specified directory and its subdirectories are examined. This is crucial for thorough analysis, especially in environments with complex directory structures where files might be spread across multiple levels.
- *Matching Strings:* Outputs all the matched strings within a rule, providing transparency and insight into the detection process.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024

- *Fast Scanning:* Optimizes efficiency by halting further scanning of a file once a rule is matched, saving processing time without compromising accuracy.
- *Metadata:* Displays key metadata associated with the matched files or strings, aiding in file classification and forensic analysis.

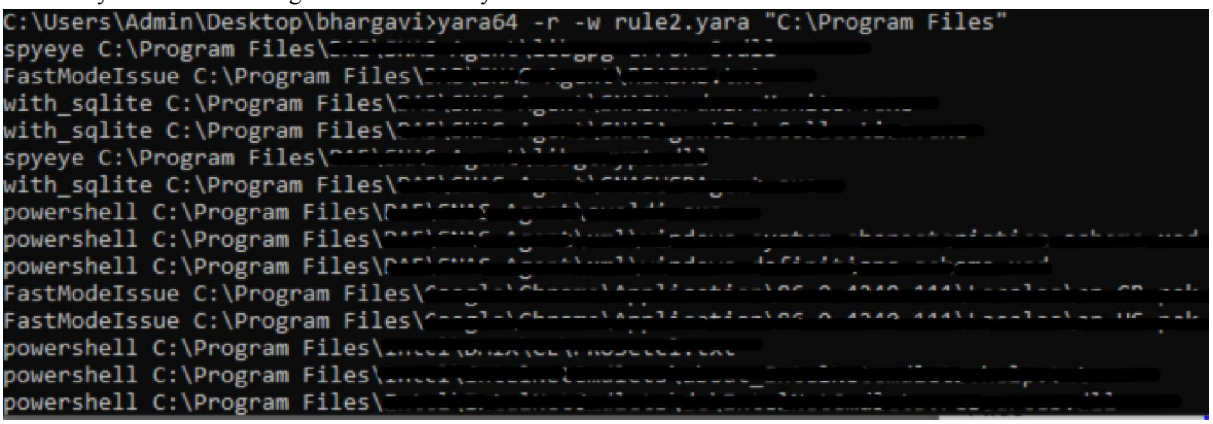
These functionalities make YARA an indispensable tool for cybersecurity experts, enabling structured, efficient, and scalable malware detection and analysis.

III. TESTING AND EVALUATION

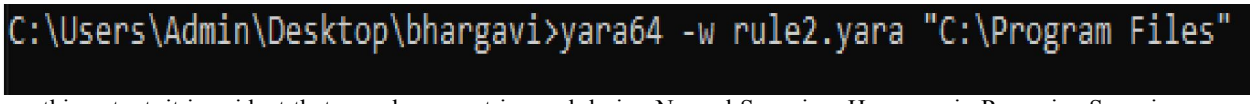
To demonstrate the efficiency of YARA's parameters in optimizing scan times, we conducted a focused experiment comparing different scanning modes. For this test, we used a rule file, **rule2.yara,** which contains 500 rules, to scan a file sized approximately **1.59 GB**. The goal was to highlight the performance differences between **Normal Scanning** and **Recursive Scanning** modes, both in terms of speed and depth of analysis.

File Size	Normal Scanning	Recursive Scanning
1.59 GB	11 seconds	1 minute 32 seconds

As the results show, **Normal Scanning** completed the scan significantly faster, taking only **11 seconds,** while **Recursive Scanning** required **1 minute and 32 seconds** to process the same file. This stark difference highlights the inherent trade-off between speed and thoroughness. While Normal Scanning efficiently analyzes files at the top level of a directory, it lacks the depth to uncover threats hidden within nested subdirectories. On the other hand, Recursive Scanning ensures a comprehensive analysis, delving deep into nested folders and subdirectories to detect hidden threats, albeit at the cost of increased scan time. To further evaluate this, we applied recursive scanning with the rule2.yara file on the Program Files directory:



Below is the output from the command line when executing Normal Scanning with the same rule file on the same subdirectory:



From this output, it is evident that no rules were triggered during Normal Scanning. However, in Recursive Scanning, multiple rules, such as "powershell" and "spyeye," were successfully triggered. This suggests that none of the defined patterns in rule2.yara matched the files at the top level of the directory. While Normal Scanning is fast and efficient, its inability to detect threats in nested directories highlights its limitations for comprehensive malware analysis. This result underscores the need for recursive scanning or other enhancements to ensure a deeper and more thorough examination of files across all levels of a directory structure.

To further optimize scan times and enhance the efficiency of YARA's scanning process, we implemented advanced techniques such as **recursion with multi-threading** and **recursion with fast scanning**. These approaches aim to

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024

significantly reduce the time required to analyze large files or directories while maintaining accuracy in malware detection.

Multi-Threading: A Boost to Performance

Multi-threading is a computational technique that divides a task into multiple smaller processes (threads) that run concurrently, leveraging the power of multi-core processors [5]. By parallelizing the scanning process, multi-threading allows multiple sections of a file or multiple files to be scanned simultaneously, greatly accelerating the overall operation. For this experiment, we configured the system to use **10 threads**, balancing efficiency and resource utilization.

Fast Scanning: Speed with Smart Optimization

Fast scanning is another enhancement that complements recursion. In this mode, YARA halts scanning within a file as soon as a rule matches, skipping further analysis for that file. This approach significantly reduces redundant computations, focusing only on critical areas of the files while still delivering accurate results.

We tested these methods on two datasets: a **1.59 GB file** and an **85.6 GB file**, comparing the performance of standard recursive scanning, recursion with multi-threading, and recursion with fast scanning. The results are summarized below:

File Size	Recursion Scanning	Recursion with Multi-threading (10 threads)	Recursion with Fast Scanning
1.59 GB	1 minute 32 seconds	21 seconds	19 seconds
85.6 GB	2 hours 24 minutes 18 seconds	1 hour 15 minutes 32 seconds	52 minutes 38 seconds

For the 1.59 GB file:

- Standard Recursive Scanning took 1 minute 32 seconds, serving as the baseline for comparison.
- With Multi-Threading, the scan time dropped dramatically to 21 seconds, showcasing a substantial improvement in speed.
- When combining recursion with Fast Scanning, the time was further reduced to just 19 seconds, demonstrating how selective processing can shave off critical seconds.

For the 85.6 GB file:

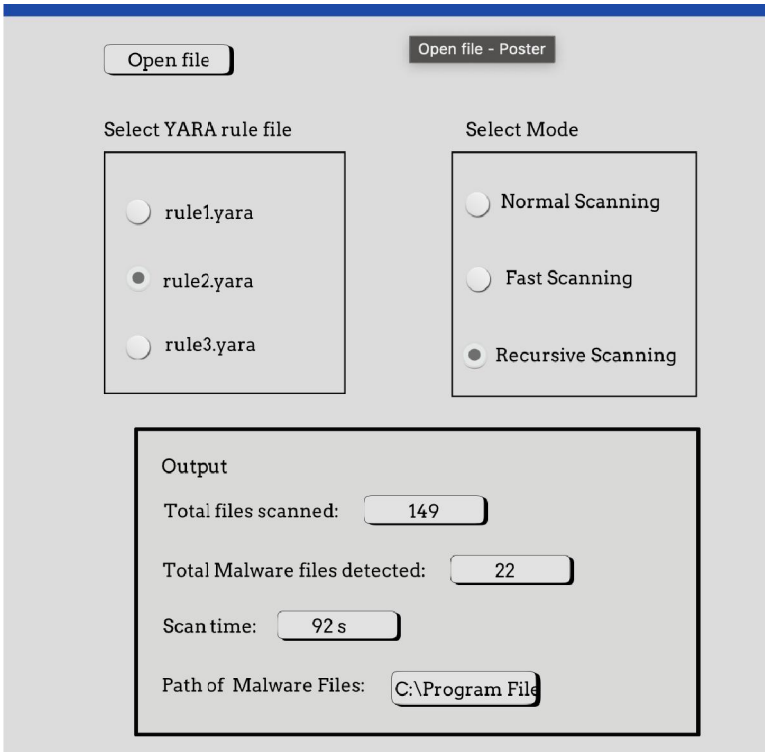
- Recursive Scanning without any enhancements required 2 hours 24 minutes 18 seconds, reflecting the challenges of scanning large files thoroughly.
- By introducing Multi-Threading, the time was cut nearly in half, completing in 1 hour 15 minutes 32 seconds.
- Fast Scanning further optimized the process, completing the task in 52 minutes 38 seconds, effectively slashing the scan time by more than half compared to standard recursion.
- The performance boost from **multi-threading** is particularly evident in both datasets, as it leverages parallel processing to speed up operations.

IV. SOFTWARE DEVELOPMENT

Manually configuring command-line parameters can be cumbersome and error-prone, especially for users managing large directories or complex rule files. To address this challenge, the GUI brings a user-friendly and visually organized approach to performing YARA scans.

The process begins with a straightforward "Open file" button, allowing users to seamlessly browse their directories and select the target file or folder for scanning. This eliminates the need for users to input lengthy file paths manually, making the setup process both faster and less prone to error.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024



In the first subsection, users can effortlessly select one of the predefined YARA rule files by simply clicking on the desired option. This feature saves time and reduces the complexity of typing out rule file paths.The second subsection provides a versatile mode selection system, allowing users to choose between **Normal Scanning, Fast Scanning**, and **Recursive Scanning**. For added flexibility, users can enable multiple modes simultaneously based on their scanning needs. For example, combining Fast and Recursive Scanning offers both speed and depth in malware detection. The final section dynamically updates with real-time scan results. Key metrics such as the **total number of files scanned, files flagged as containing malware**, and **total scan time (in seconds)** are displayed prominently. Additionally, the GUI provides the **file path** of detected malicious files, enabling users to quickly locate and address potential threats without sifting through directories manually.

V. CONCLUSION

The test and validation results demonstrate how YARA's malware detection capabilities can be significantly improved through optimizations like multi-threading and fast scanning, reducing scan times while maintaining accuracy. These enhancements, along with a user-friendly GUI, make YARA more efficient and accessible for analyzing large files and complex directories. As malware evolves, such advancements ensure YARA remains a critical tool in combating cybersecurity threats.

VI. ACKNOWLEDGMENT

We extend our sincere gratitude to our mentor, Soham Buddhiwant, for their invaluable guidance and encouragement throughout this project. We also appreciate the contributions of our peers, whose insights helped refine our work.

REFERENCES

[1] “A YARA-based approach for detecting cyber security attack types” by Kubra YILDIRIM, Mustafa Emre DEMIR, Tugce KELES, Arif Metahan YILDIZ, Sengul DOGAN, Turker ,2023.
[2] “Security Awareness”, July 2017, University of California, [Online] Available: https://security.ucop.edu/resources/security-awareness/ransomware.html

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, December 2024

[3] M. Sajedul Talukder, Zahidur Talukder, "A survey on Malware detection and analysis tools", IJNSA Vol. 12, No. 2, March 2020.
[4] MilanB. Radulović, Veljko M. Milutinović, "Multithreading", 2018. [Online] Available: https://www.sciencedirect.com/topics/computer-science/multithreading
[5] A. Lockett, "Assessing the Effectiveness of YARA Rules for Signature-Based Malware Detection and Classification," 2021, [Online]. Available: http://arxiv.org/abs/2111.13910.
[6] Reyadh Hazim Mahdi, Hafedh Trabelsi, "Detection of Malware by Using YARA Rules", IMCS-SSD, 2024