STM JOURNALS

Journal of
Network Security

ISSN: 2395-6739 (Online)
ISSN: 2321-8517 (Print)
Volume 12, Issue 1, 2024
January–April
DOI (Journal): 10.37591/JoNS

http://computerjournals.stmjournals.in/index.php/JoNS/index

Review JoNS

IMAT: Intuitive Malware Analyzer Tool

J. Dhiviya Rose[1,*], Kaushal Tiwari[2], Priyanshee Sethi[2],
Tanya Goyal[2], Sakshi Sati[2]

Abstract

Malware refers to malicious software intentionally created to damage or exploit computer systems, networks, and devices. Malware can steal information, damage computers, and cause other problems disrupting normal computer operations, or gaining unauthorized access to systems. Our proposed system, "IMAT (Intuitive Malware Analyzer Tool)" uses special Python tools like VirusTotal and YARA to look for and understand malware. Imagine having a guard for your computer that checks all the files to make sure they are safe. That is what our Malware Analyzer does, a helpful tool created using Python. The proposed system is designed to check files where it looks at files to see if they might be harmful. It can also ask VirusTotal, a big online database, if the file is known to be bad. Finding Bad Patterns which are common in malware using YARA helps it catch even new kinds of malware. It is designed to create easy-to-read reports so that people can understand what it found and how to stay safe. Our Malware Analyzer proves beneficial for individuals seeking to safeguard their computers against malicious software. It makes finding and stopping malware easier, which helps everyone stay more secure online. In this project, we will explain how to use our analyzer to protect your digital world.

Keywords: Malware, YARA, VirusTotal, analyzer, security

INTRODUCTION

In our modern world, where we heavily depend on computers and the internet, cyber security is a top priority [1]. Malware, which includes all kinds of harmful software, poses a constant and ever-changing danger, putting both individuals and organizations at risk. The proposed system, centered around malware analysis using Python, is designed to address this pressing issue. With a particular focus on automation and resource integration, it aims to provide an efficient and comprehensive solution for the identification and analysis of various types of malwares. The scope of the proposed system involves malware analysis using Python and various APIs, is aimed at addressing the critical problem of identifying and analyzing malicious software, which poses a significant threat to computer systems and data security.

***Author for Correspondence**
J. Dhiviya Rose
E-mail: dhiviyarj@ddn.upes.ac.in

[1]Assistant Professor-Selection Grade, School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi, Dehradun, Uttarakhand, India
[2]Student, School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi, Dehradun, Uttarakhand, India

Received Date: February 03, 2024
Accepted Date: February 08, 2024
Published Date: April 04, 2024

Citation: J. Dhiviya Rose, Kaushal Tiwari, Priyanshee Sethi, Tanya Goyal, Sakshi Sati. IMAT: Intuitive Malware Analyzer Tool. Journal of Network Security. 2024; 12(1): 13–18p.

The motivation to execute this project is clear for several reasons:

1. *Cybersecurity:* Malware poses a severe threat to data security, privacy, and the stability of computer systems. Your project directly addresses the need for better malware detection and analysis techniques, contributing to enhanced cybersecurity.
2. *Efficiency:* Automation is key in addressing the vast number of malware samples generated daily. By automating the analysis process, your project can significantly reduce the time and effort required to identify and respond to malware threats.

3. *Resource integration:* The use of VirusTotal API keys and the YARA module is motivated by the need for a comprehensive malware analysis toolkit. Integrating these resources provides a more thorough and accurate analysis, enhancing the effectiveness of your project.
4. *Practical application:* The project's practical application is evident. It provides a tool that can be used by cybersecurity professionals, researchers, and organizations to bolster their defenses against malware threats.

The prime beneficiaries of the malware analysis project includes everyday computer users, whether at home or in the workplace, can benefit from the improved security that project can offer [2]. It can help protect their personal data and digital devices from malware infections. Companies and institutions that rely on digital systems and data for their operations will benefit from enhanced cybersecurity. The project can help them protect sensitive information, customer data, and intellectual property from malware threats [3]. If the project is open-source, it can benefit a wider community of developers and security enthusiasts, allowing them to contribute, enhance, and utilize your tool for their security needs.

The key components of this project are as follows:

1. *URL scanning using VirusTotal API key [4]:* Malicious URLs are often a gateway to malware distribution. By integrating the VirusTotal API key, we can automatically scan URLs for potential threats, improving our ability to preemptively block access to malicious sites.
2. *Image scanning using VirusTotal API key:* Malware can be hidden within image files, making them a potential vector for attack. With the VirusTotal API, we can automate the analysis of image files, ensuring that we can detect any embedded threats.
3. *Executable (EXE) scanning using VirusTotal API key:* Executable files are a common carrier for malware. By utilizing the VirusTotal API, we can automatically scan and analyze EXE files for any signs of malicious activity.
4. *PDF scanning using YARA module:* PDF files are frequently exploited to deliver malware. The integration of the YARA module allows us to analyze the content and structure of PDF files, enhancing our ability to detect hidden threats.

BACKGROUND STUDIES

The literature studies was conducted with two tools. Examples of websites or software that use the VirusTotal API and YARA, along with their respective pros and cons:

1. *Joe Sandbox:* Joe Sandbox is a comprehensive and advanced malware analysis platform that integrates the VirusTotal API and YARA for in-depth analysis and threat detection. Joe Sandbox provides highly advanced malware analysis capabilities, leveraging the VirusTotal API for broader coverage of antivirus engines and YARA for custom rule-based detection [5]. This combination enhances detection accuracy. Users can create and apply custom YARA rules, tailoring the analysis to specific requirements. This customization is valuable for identifying targeted attacks or specific malware families.
   Detailed Reports: The platform generates detailed and well-structured analysis reports, making it easier for security professionals to understand the behavior and impact of malware. However, it is a commercial product and can be costly, making it less accessible for small organizations and individuals.
   Complexity: Due to its advanced features, Joe Sandbox can have a steeper learning curve. It may require significant expertise to use effectively, which could be a limitation for less experienced users.
2. *Cuckoo Sandbox:* Cuckoo Sandbox is an open-source malware analysis platform that integrates with the VirusTotal API and YARA to provide malware analysis and detection capabilities. Cuckoo Sandbox is open-source and free to use. This makes it accessible to a wide range of users and organizations. Users can extend the functionality of Cuckoo Sandbox by creating and adding custom YARA rules and plugins. This extensibility allows for tailored analysis and the incorporation of additional analysis tools. There is an active user and developer community around Cuckoo Sandbox, providing support and resources for users [6]. However, setting up

Cuckoo Sandbox can be complex and requires technical expertise, particularly for custom rule and plugin development. Similar to other advanced analysis tools, Cuckoo Sandbox can be resource-intensive and may require dedicated hardware for optimal performance.
Limited to Public Signatures: Like other tools relying on the VirusTotal API, Cuckoo Sandbox's detection capabilities are limited to publicly available antivirus signatures.

SYSTEM DESIGN

The proposed system was designed with the objective to develop a user-friendly and comprehensive Malware Analyzer using Python, integrating various techniques, including machine learning, static analysis, dynamic analysis, and signature-based detection. The project aims to achieve the following specific goals to develop robust detection mechanisms to improve accuracy in identifying a wide range of malware types, including new and polymorphic variants. Create an intuitive and accessible interface that allows both cybersecurity experts and non-experts to easily analyze and understand malware samples. Seamlessly integrate machine learning, signature-based detection, static analysis, and dynamic analysis into a unified tool for comprehensive malware analysis. Ensure that the Malware Analyzer is resource efficient, making it accessible for users with varying computing resources. Figure 1 shows the process flow for the proposed system with the below given components.

Intuitive malware analyzer tool is designed to automate the identification and analysis of malware, which is a critical component of cybersecurity. In a world where digital threats continue to evolve and pose risks to individuals, organizations, and society, the project aims to provide an effective and practical solution for addressing these cybersecurity challenges. The primary goal of your project is to create a software tool that streamlines the process of analyzing potentially malicious digital content, including URLs, image files, executable (EXE) files, and PDF documents. To achieve this, you have integrated external APIs, such as the VirusTotal API for scanning URLs, images, and EXE files, and the YARA module for PDF analysis.

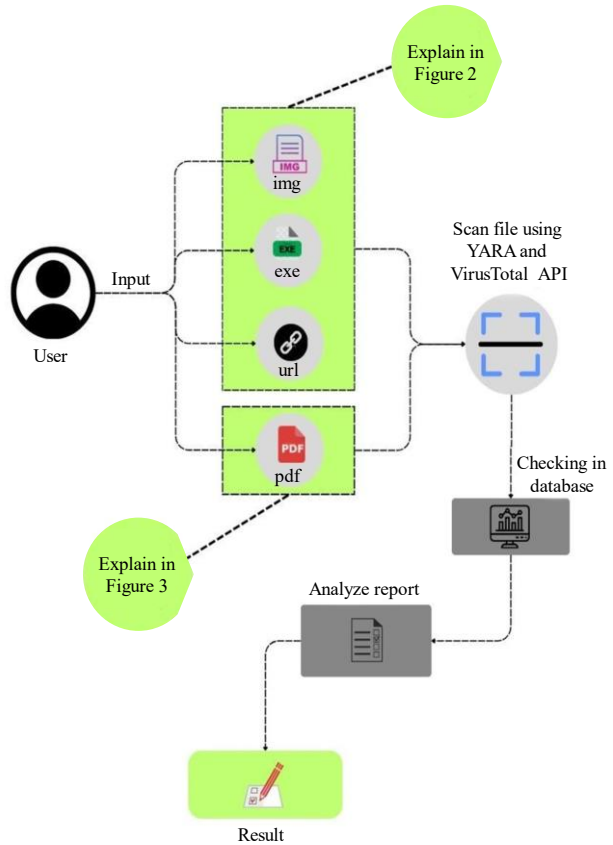

Figure 1. Proposed system design.

The scanning that we do in our project is:

1. *URL scanning:* This feature allows users to submit URLs for scanning and receive analysis results. It leverages the VirusTotal API to check URLs against multiple antivirus engines and blacklists, enabling early threat detection for potentially malicious websites.
2. *Image scanning:* By automating the analysis of image files using the VirusTotal API, this feature helps identify hidden threats within image content. It is valuable for detecting malware concealed within images.
3. *EXE scanning:* The executable (EXE) scanning feature uses the VirusTotal API to automatically scan and analyze EXE files. It offers a useful tool for detecting and studying potentially malicious software that comes in the form of executable files.
4. *PDF scanning:* This feature utilizes the YARA module to analyze the structure and content of PDF files. By doing so, it enhances the ability to detect concealed threats within PDF documents, which are commonly exploited for malware distribution.

VirusTotal API and YARA Module

VirusTotal is a widely used online service and repository for analyzing and scanning files and URLs for malware and other security threats. The VirusTotal API is an interface that allows developers and security professionals to programmatically access the VirusTotal services and integrate them into their own applications and workflows [7]. Here is what the VirusTotal API enables to submit files or URLs to the VirusTotal API, which will then scan them using multiple antivirus engines and other security checks. The API provides detailed reports on the results, including information about the presence of malicious content. The API can be integrated into security tools, scripts, and applications to automate the process of scanning files and URLs for malicious content as shown in Figure 2. Users can make use of the API to create custom solutions for their specific security needs.

YARA is an open-source tool and rule-based language designed for identifying and classifying malware and other security threats. The working model of YARA is shown in Figure 3. YARA allows users to define their own rules using a simple and flexible syntax. These rules can be used to match on file content, strings, regular expressions, and more. This makes it possible to create tailored rules to detect specific malware families, behaviors, or characteristics [8]. YARA rules are applied to files, processes, or network traffic to identify patterns or attributes that match the defined rules. When

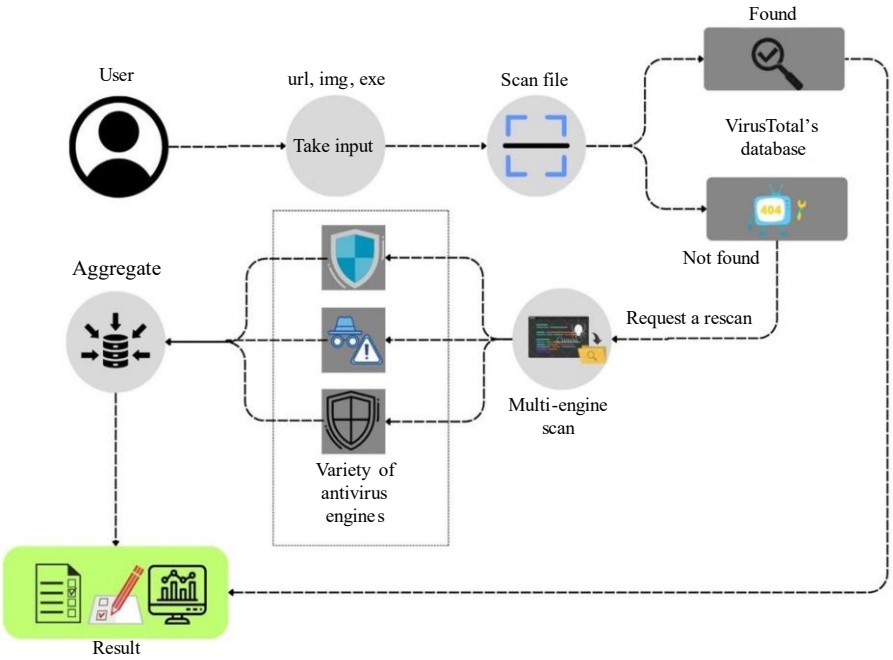

Figure 2. Virus total API design.

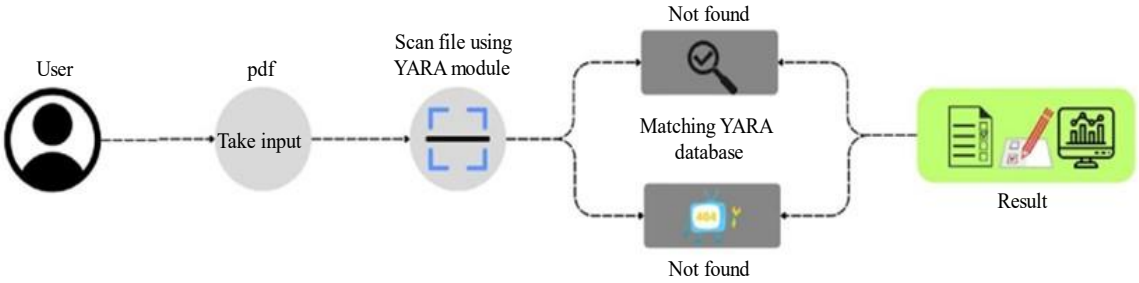

Figure 3. YARA design.

a match is found, it indicates the potential presence of malware or a security threat. YARA can be integrated into various security tools and systems, including antivirus software, intrusion detection systems (IDS), and malware analysis platforms.

SYSTEM IMPLEMENTATION

Design the analyzer with an open-source architecture to encourage community contributions and enable scalability and adaptability to evolving malware threats. Implement mechanisms for identifying and analyzing unknown and zero-day threats effectively. Documentation and Educational Resources: Provide comprehensive documentation and educational materials to help users make the most of the Malware Analyzer [9]. By achieving these objectives, the project aims to contribute to the cyber security community by providing a valuable tool that empowers users to enhance their malware detection and analysis capabilities, ultimately leading to a safer digital environment.

The proposed system is implemented using a multi-core processor (e.g., Intel Core i5 or equivalent) for efficient performance. At least 4 GB of RAM, but more is preferable for handling large files and multiple scans concurrently. Python 3.x should be installed on the system. Ensure that necessary Python libraries, such as requests and any others used in the code, are installed. Valid API keys for the external services are using, such as the VirusTotal API. Ensure these keys are obtained and configured correctly.

Interpreting a malware analysis report requires understanding various terms, acronyms, and abbreviations commonly used in the field of cybersecurity and malware analysis. YARA is a tool used for identifying and classifying malware based on defined rules and patterns. In your project, you utilize the YARA module to analyze PDF files for potentially malicious content. A scan report is a document that summarizes the results of a malware analysis. It includes information about detected threats, their characteristics, and the scan process [10]. An API key is a unique code or credential that allows access to a specific API service. In your project, you require API keys to interact with services like VirusTotal. Signature-based detection is a method of identifying malware by comparing files or code to known signatures of known threats. Antivirus software often uses this approach.

CONCLUSIONS

The problem addressed in this proposed system includes Malware, including viruses, Trojans, worms, and other forms of malicious software, constantly evolves and threatens computer systems. Detecting and identifying malware is a complex and ongoing challenge for cybersecurity professionals. Manually analyzing every potential malware sample is impractical due to the sheer volume of new malware variants. Automation is essential for efficient analysis and identification. Access to resources like VirusTotal API keys and the YARA module helps improve the accuracy and effectiveness of malware analysis. However, effectively integrating these resources into a single project can be challenging.

REFERENCES

1. Talukder S. Tools and techniques for malware detection and analysis. arXiv preprint arXiv:2002.06819. 2020 Feb 17.

2. Zolkipli MF, Jantan A. Malware behavior analysis: Learning and understanding current malware threats. In 2010 IEEE 2nd International Conference on Network Applications, Protocols and Services. 2010 Sep 22; 218–221.
3. Aslan Ö, Samet R. Investigation of possibilities to detect malware using existing tools. In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). 2017 Oct 30; 1277–1284.
4. Peng P, Yang L, Song L, Wang G. Opening the blackbox of virustotal: Analyzing online phishing scan engines. In Proceedings of the Internet Measurement Conference. 2019 Oct 21; 478–485.
5. Maier D, Müller T, Protsenko M. Divide-and-conquer: Why android malware cannot be stopped. In 2014 IEEE 9th International Conference on Availability, Reliability and Security. 2014 Sep 8; 30–39.
6. Jamalpur S, Navya YS, Raja P, Tagore G, Rao GR. Dynamic malware analysis using cuckoo sandbox. In 2018 IEEE 2nd international conference on inventive communication and computational technologies (ICICCT). 2018 Apr 20; 1056–1060.
7. Leka C, Ntantogian C, Karagiannis S, Magkos E, Verykios VS. A Comparative Analysis of VirusTotal and Desktop Antivirus Detection Capabilities. In 2022 IEEE 13th International Conference on Information, Intelligence, Systems & Applications (IISA). 2022 Jul 18; 1–6.
8. Naik N, Jenkins P, Cooke R, Gillett J, Jin Y. Evaluating automatically generated YARA rules and enhancing their effectiveness. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI). 2020 Dec 1; 1146–1153.
9. Chakkaravarthy SS, Sangeetha D, Vaidehi V. A survey on malware analysis and mitigation techniques. Comput Sci Rev. 2019 May 1; 32: 1–23.
10. Ucci D, Aniello L, Baldoni R. Survey of machine learning techniques for malware analysis. Comput Secur. 2019 Mar 1; 81: 123–47.

