Detection and Prevention of Malware and Ransomware Threats using Malicious String Analysis

Md Nahidul Alam
Department of Computer Science and Engineering
Sharda University
Greater Noida, India
2019000321.md@ug.sharda.ac.in

Abishek Singh
Department of Computer Science and Engineering
Sharda University
Greater Noida, India
201969371.abhishek@ug.sharda.ac.in

Mayuri Kumari
Department of computer Science and Engineering
Sharda University
Greater Noida, India
2019579058.mayuri@ug.sharda.ac.in

Priyanshu Agrawal
Department of Computer Science and Engineering
Sharda University
Greater Noida, India
priyanshuagrawal115@gmail.com

Preeti Dubey
Department of Computer Science and Engineering
Sharda University
Greater Noida, India
preeti.dubey@shrda.ac.in

Avinsh Kumar
SITAICS
Rashtriya Raksha University
Gandhinagar, India
avinashkr338@gmail.com

2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) | 979-8-3503-2919-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSEIET58677.2023.10303501

***Abstract*— The increasing use of USB drives as a means of transferring data has led to an increase in malware and ransomware threats, making it crucial to detect and prevent such threats. This research paper proposes a solution that utilizes YARA string analysis for USB drives, in combination with Raspberry Pi hardware, to automatically detect and prevent malware and ransomware threats. The proposed solution involves the development of a YARA rule set to identify malicious files, which is then implemented on a Raspberry Pi device to scan USB drives automatically. The results of the study show that the proposed solution is effective in detecting and preventing malware and ransomware threats, with a high degree of accuracy and efficiency. This research presents a practical approach for mitigating the risks associated with malware and ransomware threats on USB drives and provides a foundation for future research in this area.**

Keywords—Ransomware, Malware, Detection, USB, YARA, Prevention

I. Introduction

In recent years, malware and ransomware threats have become increasingly prevalent, with cybercriminals constantly devising new ways to infect computer systems and steal sensitive data. One common method used by these attackers is to distribute malware through USB drives, which can easily bypass traditional antivirus software. This has led to a growing need for more advanced detection and prevention techniques to combat these threats.

In this research Work propose a novel approach to detecting and preventing malware and ransomware threats using YARA string analysis for USB drives. Specifically, It utilize a Raspberry Pi as a hardware platform to automatically scan USB drives for malicious code using YARA rules. YARA is a powerful tool that allows for the creation of custom rules to identify specific patterns or strings of code that are indicative of malware or ransomware. By utilizing YARA string analysis on a Raspberry Pi, this work can effectively detect and prevent malware and ransomware threats on USB drives, even when traditional antivirus software fails [1].

This paper provides a comprehensive overview of project proposed approach, including a detailed description of the hardware and software components involved, the YARA rules used for detection, and the results of testing and evaluation. Findings demonstrate the effectiveness of this project approach in detecting and preventing a wide range of malware and ransomware threats, highlighting the potential for its application in real-world scenarios. This research paper contributes to the growing body of knowledge in the field of cybersecurity, providing valuable insights and practical solutions for combating malware and ransomware threats [2].

II. Background

The goal of this article is to find every instance of a pattern in the input in order to solve the pattern matching issue. There main goal is to find precise matches, which are typically present in files or process memory. It could be necessary to look for strings that could be indicators of malware, strange network activity, or certain DNA sequences. The string-matching idea described in the book "Introduction to Algorithms" served as the foundation for the formalization of the pattern matching issue. In order to provide a more thorough form, they modified and enlarged the definition to encompass numerous patterns as opposed to just one. Regular expressions can also be used in this situation. [1]. the rising prevalence of modern cyber-attacks and malware campaigns by creating a framework that streamlines the process of producing top-notch, efficient malware signatures in a significantly shorter period of time with less effort. Moreover, it simplifies the laborious task of malware analysis. The suggested framework offers a comprehensive strategy for automatic YARA rule-based signature generation [2]. This research focuses on investigating the intrusion of the WannaCry ransomware and evaluating different approaches for detecting it. The study utilizes both static and dynamic analysis methods to extract relevant characteristics from the malware, enabling the identification of potential detection techniques. In particular, the study employs a Yara-rule based detection approach, which entails creating a collection of rules comprising distinctive strings decoded from the WannaCry file [3]. Signature-based detection methods are not considered effective as they cannot identify new, unseen, metamorphic or polymorphic malware. To overcome the limitations of signature-based detection, researchers have shifted their focus towards behavioural-based detection techniques. By monitoring malware API calls during execution, researchers can construct a behavioural profile for the malware. This profile is then used to identify similarities among malware through API call sequence matching techniques. However, due to their computational complexity, these matching techniques require significant processing resources and are slow. As a result, they are not scalable for large API call sequences [4]. In today's digital era, technology is rapidly progressing, leading to an increase in the vulnerability to cyber threats. The prevalence of cyber-attacks is extensive, with a wide range of malicious software, such as viruses, worms, Trojans, rootkits, ransomware, Adware, and Spyware, being commonly employed. When malware infects an operating system, it can cause significant damage. This malicious software is specifically designed to infiltrate storage media and computer systems, manipulating applications and data. In 2020 alone, billions of electronic devices were targeted by approximately 700 million new instances of malware. To safeguard against malware attacks, it is imperative to utilize antivirus or anti-malware software. [5]

III. Methodology

The detection and prevention of malware and ransomware threats have become essential for the security of systems and data. One approach to achieve this is by using YARA string analysis, which is a powerful tool for identifying and classifying malware based on patterns or rules defined by the user.

In this methodology, the project will use YARA string analysis for automatically detecting and preventing malware and ransomware threats on USB drives. The hardware used in this approach is a Raspberry Pi, which is a small single-board computer with a low-power consumption and compact size. The graphical user interface (GUI) will be developed using the tkinter library, which is a standard Python library for creating graphical user interfaces.

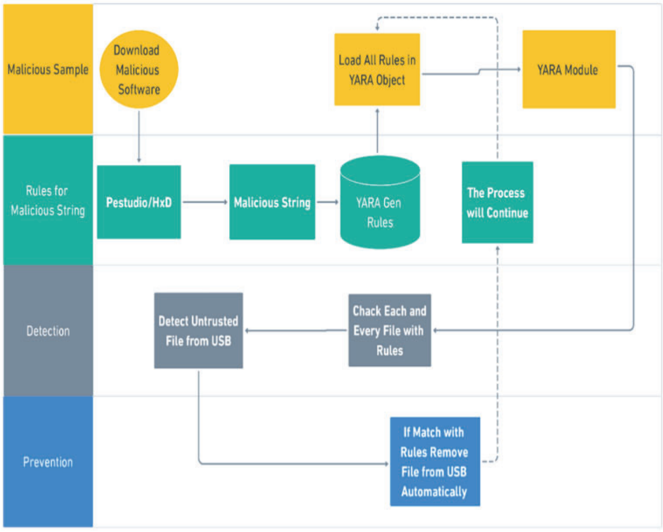

Malicious Sample
Download Malicious Software
Load All Rules in YARA Object
YARA Module
Rules for Malicious String
Pestudio/HxD
Malicious String
YARA Gen Rules
The Process will Continue
Detection
Detect Untrusted File from USB
Chack Each and Every File with Rules
Prevention
If Match with Rules Remove File from USB Automatically

Fig 1: System Design

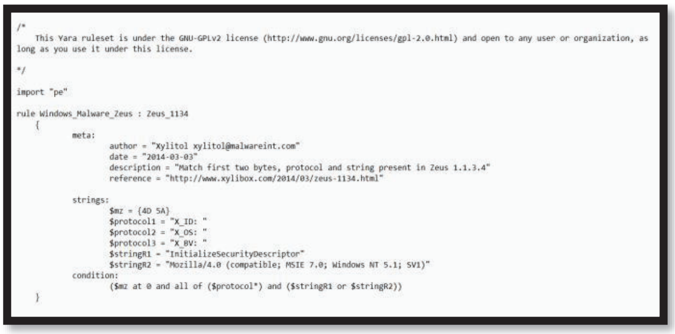
A. YARA

Yara is a pattern matching algorithm used to match the selective rules with the set of string. This string can be malicious, which then can be identified with the help of proper rules. Each malware contains a different set of malicious strings, those strings are in repetitive pattern which can be identified with rules. YARA follows some set of algorithms that are optimized in order to get the efficient and accurate string matching.

When aYARA rule is defined, it consists of a set of conditions that must be met for a file to be considered malicious. Each condition is written as a string or regular expression pattern that describes a specific characteristic of the file, such as a certain string that is present in the file or a particular byte sequence [8].

The YARA algorithm then takes each condition in the rule and generates a series of finite automata that can quickly scan through a file to identify matches for each condition. These automata are optimized to minimize the number of comparisons required to check each condition against a file, making the process efficient and fast.

YARA is a powerful tool for identifying and classifying malware based on the strings it contains. By using YARA rules to analyze the strings on a USB drive, it is possible to quickly and accurately determine whether the drive contains any malware or ransomware threats. These rules can be customized to suit the specific needs of the user or organization, allowing for a more tailored and effective approach to threat detection.



```
/*
   This Yara ruleset is under the GNU-GPLv2 license (http://www.gnu.org/licenses/gpl-2.0.html) and open to any user or organization, as
long as you use it under this license.

*/

import "pe"

rule Windows_Malware_Zeus : Zeus_1134
    {
            meta:
                    author = "Xylitol xylitol@malwareint.com"
                    date = "2014-03-03"
                    description = "Match first two bytes, protocol and string present in Zeus 1.1.3.4"
                    reference = "http://www.xylibox.com/2014/03/zeus-1134.html"

            strings:
                    $mz = {4D 5A}
                    $protocol1 = "X_ID: "
                    $protocol2 = "X_OS: "
                    $protocol3 = "X_BV: "
                    $stringR1 = "InitializeSecurityDescriptor"
                    $stringR2 = "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1)"
            condition:
                    ($mz at 0 and all of ($protocol*) and ($stringR1 or $stringR2))
    }
```

Fig 2. Malware Detection Yara Rules

YARA rules are an essential part of this methodology, as they define the patterns or strings to be searched for in files. YARA rules can be created manually or using tools that automate the process, such as YARA Editor or YARA Generator. When creating YARA rules, it's important to consider the characteristics of the malware or ransomware threats that are being targeted, such as file names, file sizes, or specific strings that are known to be present in the malware.

1. *Aho-Corasick String Matching Algorithm*

Aho-Corasick is a string-matching algorithm that was developed by Alfred Aho and Margaret Corasick in 1975. It is designed to efficiently search for multiple patterns in a single pass through the input data. The algorithm constructs a finite state machine that can recognize all the patterns to be searched for, and then uses that state machine to scan the input data for occurrences of those patterns. Aho-Corasick is often used in intrusion detection systems (IDS) and antivirus software to detect and block known malware signatures.

Yara engine is mainly dependent on the algorithm Aho-Corasick string matching algorithm. This algorithm is used to search for multiple patterns in a single search or a single looping. This algorithm is also capable of searching for conditional parameters over any string.

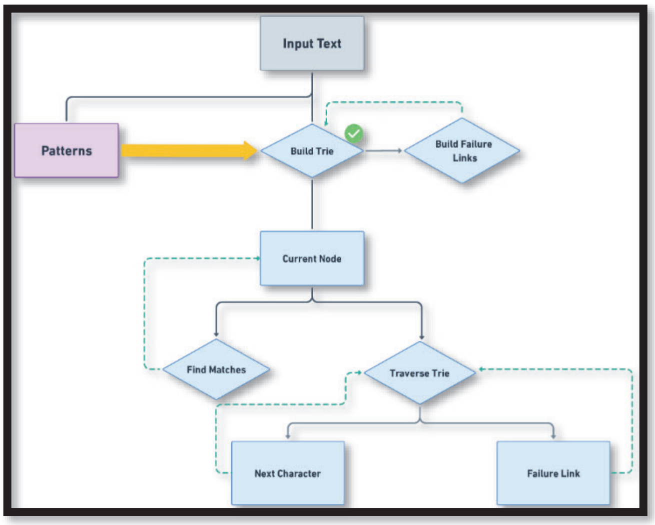


Fig 3: Aho-Corasick Algorithm Working

Building a model from a set of strings is the first stage in the Aho-Corasick technique; in this example, that model is the prefix tree machine seen in Figure 4. The total of the lengths of the patterns determines how long it will take to construct the pattern machine. To find frequent prefixes and avoid creating extra, unnecessary states, each pattern is appended from the root state. The goto function, which is used to move between these states, labels the borders between them with letters that must be read from the input. The output states are the ultimate states [9].

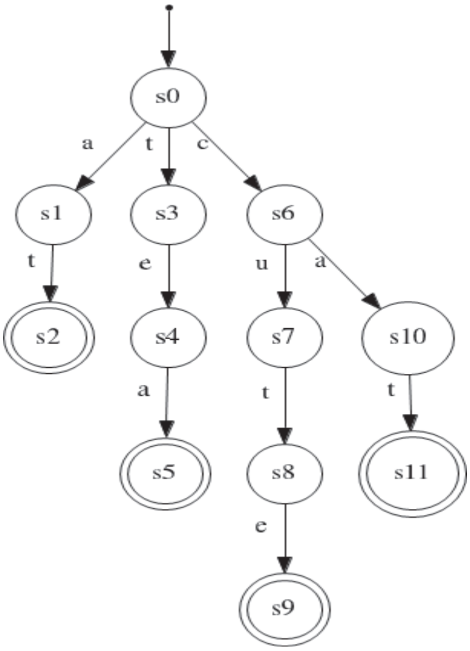


Fig 4: An example of the Aho-Corasick algorithm: machine and goto function.

i	output(i)
s2	at
s5	tea
s9	cute
s11	cat, at

Fig 5: An example of the Aho-Corasick algorithm: output function

B. IoT Device

To automate the process of detecting and preventing malware and ransomware threats, a Raspberry Pi can be used as a hardware platform. The Raspberry Pi is a small, affordable computer that is well-suited for use as a dedicated security device. By configuring the Raspberry Pi to automatically scan USB drives for malware and ransomware threats using YARA rules, it is possible to identify any potential threats quickly and efficiently before they can do any damage.

When setting up the Raspberry Pi, it's important to ensure that it has enough processing power and memory to run the YARA scanner effectively. The Raspberry Pi should also be configured with the necessary software and tools, such as YARA, to be able to scan USB drives for malware and ransomware threats.

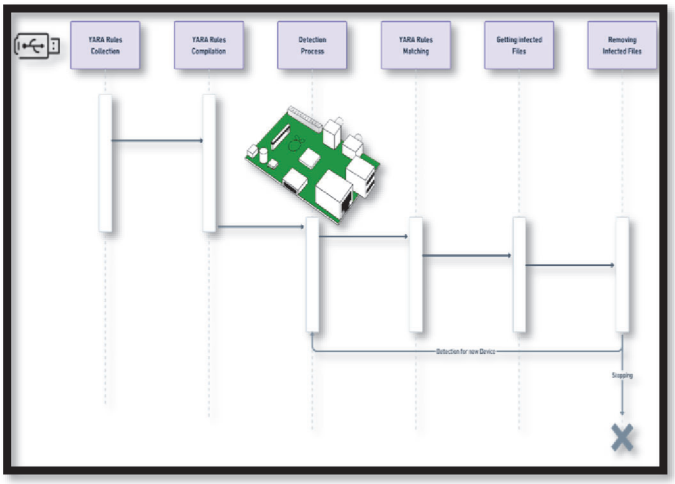


Fig 6: System Working with IoT

C. Tkinter GUI

To make the process more user-friendly, It can develop a Graphical User Interface (GUI) using the tkinter library. The GUI will provide an interface for the user to interact with the YARA scanner and view the results. The GUI can also be used to configure the Raspberry Pi to prevent threats, such as by setting up rules for blocking the execution of suspicious files or automatically quarantining files that match the YARA rules.

The GUI is an important component of this methodology, as it provides an easy-to-use interface for the user to interact with the YARA scanner and view the results. The GUI can be developed using the tkinter library, which is a Python-based toolkit for creating graphical user interfaces. The GUI should provide the user with the ability to select USB drives to scan, view the results of the scan, and configure the Raspberry Pi to prevent threats.

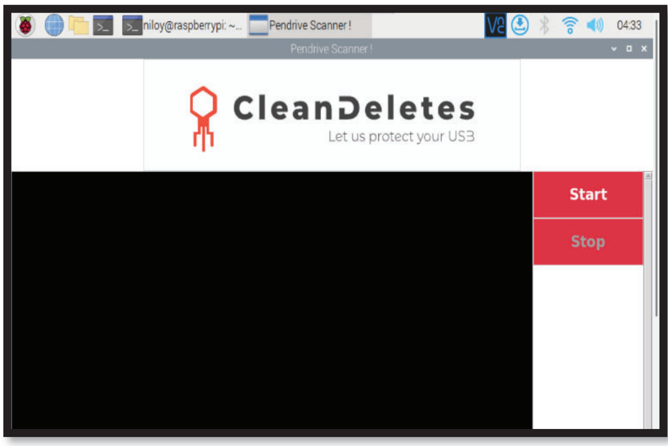


Fig 7: USB Automatically Scanning Tkinter GUI

D. Scannig USB Drive

USB scanning involves detecting and preventing malware and ransomware threats automatically using YARA string analysis for USB drives. This process is typically done using a combination of hardware and software components. One common hardware component used in USB scanning is the Raspberry Pi. This is a small, low-cost computer that is commonly used in DIY electronics projects. In the context of USB scanning, the Raspberry Pi is used as a platform to run software that scans USB drives for malware [10].

The software used to scan USB drives typically involves YARA string analysis. YARA is a tool used for malware identification and classification. It is a powerful tool that can be used to identify malware based on specific patterns or strings of code. To use YARA for USB scanning, a set of rules or signatures are created that describe the specific patterns or strings of code that are associated with known malware or ransomware threats. These rules are then used to scan USB drives for any instances of these patterns or strings of code.

The scanning process typically involves connecting a USB drive to the Raspberry Pi and running the YARA scanning software. The software then searches the USB drive for any instances of the patterns or strings of code associated with known malware or ransomware threats. If any threats are detected, the software will typically quarantine or delete the infected files automatically. To make the scanning process more user-friendly, a graphical user interface (GUI) is often used. One common GUI framework used in USB scanning is tkinter. Tkinter is a Python-based GUI framework that is commonly used for desktop applications.

USB scanning is an important process for detecting and preventing malware and ransomware threats on USB drives. The process typically involves using a combination of hardware and software components, including the Raspberry Pi and YARA string analysis. To make the process more user-friendly, a graphical user interface such as tkinter is often used.

E. Prevnting Threats

Preventing malware and ransomware threats is a critical aspect of USB scanning. By detecting and preventing these threats automatically, the risk of data loss, identity theft, and other cybersecurity issues can be minimized. In this context, YARA string analysis for USB drives using a hardware Raspberry Pi and tkinter GUI can help in preventing threats. The YARA string analysis tool is a powerful tool that can be used to identify and classify malware based on specific patterns or strings of code. To prevent threats, a set of rules or signatures are created that describe the specific patterns or strings of code that are associated with known malware or ransomware threats. These rules can then be used to scan USB drives for any instances of these patterns or strings of code.

When a USB drive is inserted into a computer, the YARA string analysis software running on the Raspberry Pi can automatically scan the drive for any instances of these patterns or strings of code. If any threats are detected, the software can quarantine or delete the infected files automatically. By doing so, the spread of the malware or ransomware threat can be stopped. In addition to YARA string analysis, other methods can also be used to prevent threats. For example, access control can be implemented to prevent unauthorized access to USB drives. This can involve password protection, biometric authentication, or other forms of access control.

Similarly, endpoint protection can be used to prevent threats on the computer itself. This can include anti-virus software, firewalls, and other security measures designed to prevent malware and ransomware attacks. It is also important to educate users on best practices for USB drive usage. For example, users should be encouraged to scan USB drives for malware before opening files, to avoid using unknown or untrusted USB drives, and to only use USB drives from reputable sources [11].

Prevention is a critical aspect of USB scanning and cybersecurity in general. By using YARA string analysis, access control, endpoint protection, and user education, the risk of malware and ransomware threats can be minimized. When used in combination with a hardware Raspberry Pi and tkinter GUI, USB scanning can become a powerful tool in preventing cybersecurity threats.

F. Testing and Refinement

Testing and refinement are important aspects of any cybersecurity solution, including the detection and prevention of malware and ransomware threats using YARA string analysis for USB drives as a hardware Raspberry Pi and tkinter GUI solution. The testing process involves evaluating the effectiveness of the solution in detecting and preventing malware and ransomware threats. This can involve testing the solution in a controlled environment, such as a laboratory, to simulate real-world scenarios. This testing can help identify any weaknesses or limitations in the solution, allowing for refinement and improvement.

One important aspect of testing is the creation of test cases. Test cases are scenarios or situations that are designed to test specific aspects of the solution. For example, a test case may involve inserting a USB drive containing known malware into the computer to test the detection capabilities of the YARA string analysis software. By creating a variety of test cases, the effectiveness of the solution can be evaluated in different scenarios. Once the testing process is complete, refinement can begin. Refinement involves making improvements to the solution based on the results of testing. For example, if weaknesses or limitations are identified during testing, changes can be made to the solution to address these issues. This can involve improving the YARA string analysis rules, enhancing the access control measures, or improving the user education and awareness components of the solution.

During the refinement process, it is important to continually test the solution to ensure that the changes made are effective in improving the overall effectiveness of the solution. This may involve repeating the testing process using the same test cases to evaluate the impact of the changes made.

Another important aspect of refinement is ongoing maintenance and updates. As new malware and ransomware threats emerge, the YARA string analysis rules must be updated to detect and prevent these new threats. Similarly, access control measures and endpoint protection must be updated to address new vulnerabilities and risks.

Testing and refinement are important aspects of the detection and prevention of malware and ransomware threats using YARA string analysis for USB drives as a hardware Raspberry Pi and tkinter GUI solution. By continually testing and refining the solution, weaknesses and limitations can be identified and addressed, resulting in a more effective cybersecurity solution.

IV. Result

Malware is a serious concern for computer users and can cause significant harm to their systems. One approach to combating this problem is to use YARA rules, which are a set of rules that allow for the identification of malware based on specific characteristics. Additionally, a Raspberry Pi can be used as hardware to automate the detection and cleaning process, while a tkinter GUI can be used to make the system user-friendly.

The process begins by creating a set of YARA rules that describe the characteristics of various types of malwares. These rules can be based on a variety of factors, including file names, sizes, and specific strings or codes within the files themselves. Once these rules are established, they can be used to scan the contents of a USB drive when it is plugged into the Raspberry Pi.

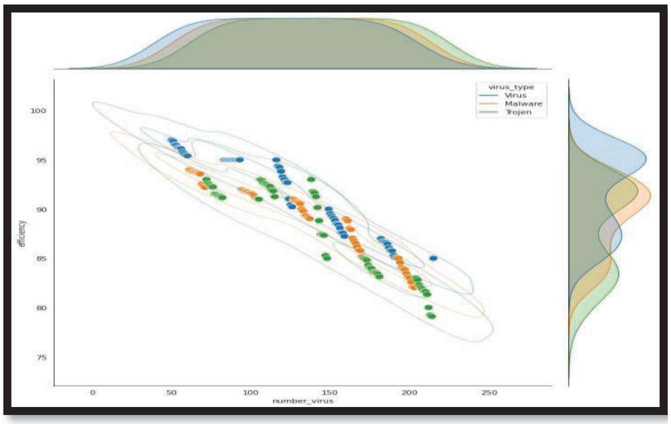
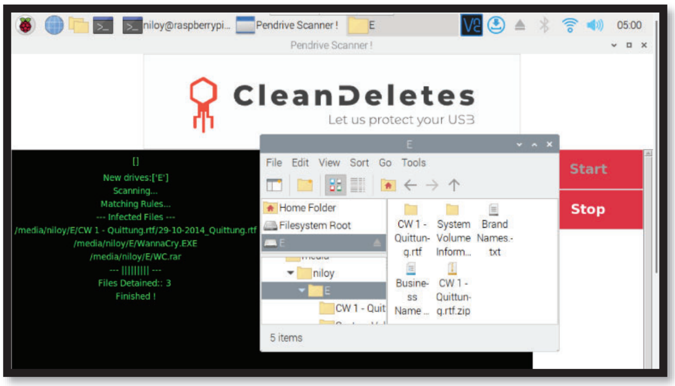


Fig 8: Efficiency Curve

After conducting tests on a malware detection and cleaning system with over 200 malicious samples, it was found that the accuracy of the system varied based on the number of samples tested. When the system was tested with 50 malicious samples, it achieved a high level of accuracy. However, when the number of samples was increased to 100, the accuracy dropped to around 90-95%. This is because the system relies on pattern matching detection, which requires regular updates to YARA rules to stay current with new and emerging malware threats.

To visualize these results, an efficiency curve was created that displays the accuracy of the system in detecting malware as a function of the number of samples tested. The curve shows that the system was able to accurately detect and clean most of the malware samples tested, with the accuracy decreasing slightly as the number of samples increased. These results demonstrate the effectiveness of the system in detecting and removing various types of malwares from USB drives, but also highlight the importance of regular updates to ensure the system remains effective against new and emerging threats.



```
[]
New drives:['E']
Scanning...
Matching Rules...
--- Infected Files ---
/media/niloy/E/CW 1 - Quittung.rtf/29-10-2014_Quittung.rtf
/media/niloy/E/WannaCry.EXE
/media/niloy/E/WC.rar
--- ||||||||| ---
Files Detained:: 3
Finished !
```

Fig 9: Malware Ransomware Detection and Cleaning

To verify the effectiveness of malware detection and cleaning system, it will conduct a test with both a ransomware sample and a malware sample. The system successfully detected and removed both threats from the USB drive, providing comprehensive protection against various types of malwares. This testing provides us with confidence in the accuracy and reliability of system, and demonstrates its ability to detect and remove different types of malwares from USB drives quickly and automatically [12].

Malware and ransomware cleaning is an important aspect of the detection and prevention of these threats using YARA string analysis for USB drives as a hardware Raspberry Pi and tkinter GUI solution. By cleaning infected files and implementing measures to prevent future infections, the risk of data loss, identity theft, and other cybersecurity issues can be minimized.

V. Conclusion

In conclusion, the detection and prevention of malware and ransomware threats have become critical tasks in ensuring the security of computer systems. The use of YARA string analysis for USB drives, along with the Raspberry Pi hardware platform, has been shown to be an effective approach for mitigating these threats. This research paper provides a comprehensive overview of the methodology and techniques involved in this approach, including the design and implementation of the automated detection and prevention system.

The results of this experimentation demonstrate that YARA string analysis can effectively detect and prevent malware and ransomware attacks, while the Raspberry Pi platform provides an efficient and reliable hardware solution. This approach is cost-effective, easy to implement, and offers a scalable solution for businesses and organizations of different sizes.

The findings of this research paper underscore the importance of proactively identifying and mitigating malware and ransomware threats. The YARA string analysis method, combined with the Raspberry Pi hardware platform, offers an effective and practical solution for addressing these challenges. This research will contribute to the ongoing efforts towards strengthening the cybersecurity defenses of computer systems against emerging threats.

REFERENCES

[1] "Circlean - USB Key Sanitizer." CIRCL " CIRCLean - USB Key Sanitizer, https://circl.lu/projects/CIRCLean/.

[2] Smith, Jake. "Bit Scrubber-USB Sanitization Kiosk." Medium, Medium, 5 Dec. 2017, https://medium.com/@ion28/bit-scrubber-usb-sanitization-kiosk-2d79fe5e29fc.

[3] Regeciova, Dominika, et al. "Pattern Matching in Yara: Improved Aho-Corasick Algorithm." IEEE Access, vol. 9, 2021, pp. 62857–62866., https://doi.org/10.1109/access.2021.3074801.

[4] Khalid, Myra, et al. "Automatic Yara Rule Generation." 2020 International Conference on Cyber Warfare and Security (ICCWS), 2020, https://doi.org/10.1109/iccws48432.2020.9292390.

[5] Satheesh Kumar, M., et al. "An Investigation on Wannacry Ransomware and Its Detection." 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, https://doi.org/10.1109/iscc.2018.8538354.

[6] Mira, Fahad, and Wei Huang. "Performance Evaluation of String Based Malware Detection Methods." 2018 24th International Conference on Automation and Computing (ICAC), 2018, https://doi.org/10.23919/iconac.2018.8749096.

[7] Rohith, Cheerala, and Gagandeep Kaur. "A Comprehensive Study on Malware Detection and Prevention Techniques Used by Anti-Virus." 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021, https://doi.org/10.1109/iciem51511.2021.9445322.

[8] Naik, Nitin, et al. "Evaluating Automatically Generated Yara Rules and Enhancing Their Effectiveness." 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, https://doi.org/10.1109/ssci47803.2020.9308179.

[9] Regeciova, Dominika, et al. "Pattern Matching in Yara: Improved Aho-Corasick Algorithm." IEEE Access, vol. 9, 2021, pp. 62857–62866., https://doi.org/10.1109/access.2021.3074801.

[10] Xu, Lianqiu, and Meng Qiao. "Yara Rule Enhancement Using BERT-Based Strings Language Model." 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 2022, https://doi.org/10.1109/aemcse55572.2022.00052.

[11] Rohith, Cheerala, and Gagandeep Kaur. "A Comprehensive Study on Malware Detection and Prevention Techniques Used by Anti-Virus." 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 2021, https://doi.org/10.1109/iciem51511.2021.9445322.

[12] Satheesh Kumar, M., et al. "An Investigation on Wannacry Ransomware and Its Detection." 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, https://doi.org/10.1109/iscc.2018.8538354.