Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2s (2025)

# Yara-Based Emotet Malware Scanner – A Factual Analysis

**Dr Priya P Sajan[1], Midhun Mohan M G[2], Binny Jacob Kuriakose[3]**

[1]C-DAC, Thiruvananthapuram. priyasajan@cdac.in

[2]EY, Thiruvananthapuram. Midhun.Mohan@gds.ey.com

[3]EY, Thiruvananthapuram. Binny.Kuriakose@gds.ey.com

***Article History:***

***Received:*** *12-09-2024*

***Revised:*** *15-10-2024*

***Accepted:*** *28-10-2024*

**Abstract:**

Malware is malicious software which has the ability to compromise sensitive data and disrupt systems. At present malicious software is the most efficient tool used in compromising the security of computers or any other electronic devices connected to the internet. This has become a menace owing to the rapid progress in technologies such as encryption and data hiding techniques. A highly adaptable and persistent type of malware, Emotet is well-known for its capacity to propagate via phishing emails and send out further malicious payloads, by impairing network and data security. Emotet poses a special risk since it serves as a gateway for other viruses, such as ransomware and data thieves, which can cause serious data breaches, monetary losses, and interruptions to business operations. It poses a serious danger to both individual users and big enterprises due to its advanced evasion techniques and quick network spread. In this project, distinct file signatures and byte patterns that are distinctive of the malware are identified by YARA rules in order to detect Emotet. These patterns include specific sequences used in Emotet's payload and obfuscation techniques, allowing for precise detection within scanned files..

**Keywords:** Emotet, Malware Detection, YARA Rules, Cyber Security.

## I. INTRODUCTION

A cyber-attack is an attempt by hackers to take advantage of security holes in a computer system, network, or other device in order to harm or interfere with it. These assaults can be conducted for a number of reasons, including political objectives, financial gain, operational interruption, and the theft of confidential data. Malware, ransomware, phishing, and distributed denial-of-service (DDoS) attacks are examples of common techniques. Cyber-attacks can have serious consequences, ranging from harming one's finances and reputation to risking national security.

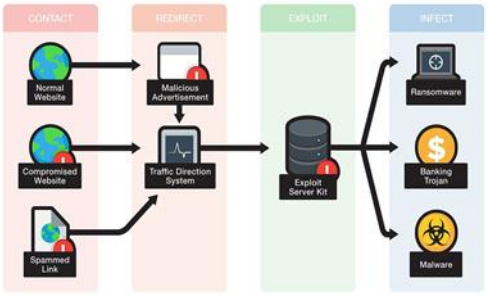


Fig. 1: Malware attack stages

A malware attack usually happens in the following stages: contact, redirect, exploit, and infect. Users may initially en- counter a compromised website, a legitimate website with a malicious advertisement displayed on it, or a link that has been spammed. These components reroute the user to an exploit server kit using a traffic direction scheme. Following that, this kit installs malware, including banking Trojans and ransomware, by taking advantage of system flaws. Once installed, this virus has the ability to corrupt files, steal private data, obstruct system operations, and seize control of the compromised computer. These cyber threats frequently come from malware downloads, phishing emails, or shoddy soft- ware. Strong cyber security measures are therefore necessary to shield people and businesses from these threats..

Malware attacks can happen via a number of different channels, including as email, phishing sites, and networks. Network-based attacks, which frequently target weakly se- cured systems to obtain widespread access, take use of flaws in network security to propagate malware among linked devices. An attacker often use email to pose as genuine senders and include malicious attachments or links that when accessed, might infect a recipient's device with malware. Phishing attacks utilize fake emails or websites to deceive people into disclosing critical information or clicking on dangerous links. Malware may be automatically downloaded and installed as a file on the user's hard drive or as a hidden process running in memory when the user clicks on these links, giving hackers unauthorized access to the system. This may lead to corruption of the system, data theft, and further spreading of malware to other systems within network.

Memory resident malware is malicious software that re- mains in a computer's random-access memory (RAM) after it has been initially run. Memory resident malware only works in memory, making it more difficult to identify and delete than standard malware, which writes itself to the disk and endures reboots. Emotet is such a type of malware that use memory-resident strategies to avoid detection and remain persistent. Emotet can function silently by inserting itself into the memory of valid processes, avoiding conventional file-based detection techniques. Emotet is extremely difficult to identify and eliminate using traditional security measures because of its capacity to live in memory and dynamically load additional payloads directly into RAM.

It is difficult to identify memory resident malware due to its stealthy nature and advanced evasion techniques. Unlike traditional malware that can be found by scanning files, memory resident malware runs only in the computer's RAM. This means it leaves very few traces on the hard drive, making it hard for regular antivirus programs to detect. Emotet is also known to be polymorphic, meaning it can change its code each time it runs. This is possible with polymorphic packers to create files with different hash and attributes and so many such other factors like encryption, code mutation etc. These challenges highlight the need for innovative detection approaches. This project focuses on leveraging YARA rules to enhance detection capabilities for Emotet malware, ensuring a more effective defense against these sophisticated threats.

## II. LITERATURE SURVEY

### *A. Emotet malware steals credit cards from Google Chrome users*

Since its initial release as a banking Trojan in 2014, the Emotet botnet has seen an enormous recovery and significant evolution. Emotet was originally a financial malware, but it eventually

evolved into a modular botnet under the control of the TA542 threat organization, also known as Mummy Spider. Emotet returned in November 2021 using TrickBot's infrastructure, despite a significant takedown by international law enforcement in early 2021 that momentarily interrupted its operations. The recovery has been characterized by a notable increase in activity, with accounts suggesting a spike of over 100 times since late 2021.

Recent events have brought attention to Emotet's flexibility and changing strategies. A new Emotet module intended to steal credit card information from Google Chrome user ac- counts was discovered by Proofpoint in June 2023.With particular attention to cardholder information, this module exfiltrate the stolen data to several C2 servers. Furthermore, Emotet has moved away from Microsoft Office macros, which were deactivated by default beginning in April 2022, and toward utilizing 64-bit modules and Windows shortcut files (.LNK) to carry out PowerShell instructions. These modifications show Emotet's ongoing efforts to strengthen its infection vectors and avoid detection.

Fig. 2: Proofprint

Cyber security is constantly at risk from the smart and persistent efforts of Emotes. The botnet's influence goes be- yond the original infections because it can propagate different secondary payloads like Trickbot and Qbot. This involves the introduction of new malware strains like Ryuk and Conti, as well as exploitation tools like Cobalt Strike. The return and continued development of Emotet highlight the necessity of regular security updates, persistent monitoring, and coordinated international efforts to mitigate its impact and protect against its sophisticated attacks.

*B. Emotet malware campaign impersonates the IRS for 2022 tax season.*

The return of Emotet malware in the midst of the 2022 tax season in the United States is a prime example of innovative techniques used by hackers to take advantage of current and relevant themes for phishing efforts. Originally infecting computers with infected Word or Excel documents through phishing emails, Emotet has developed to take advantage of tax season by posing as the Internal Revenue Service (IRS). These bogus emails pose as tax forms or returns, such W-9 forms, and persuade recipients to open the attached files by emphasizing how important they are. The victim's PC downloads the Emotet virus after opening the associated documents and enabling macros. After there, the malware sends more spam emails, collects email addresses for use in future attacks, and has the ability to install other malicious software, including the SystemBC remote access Trojan or Cobalt Strike, which may finally result in Conti ransomware attacks.


Fig. 3: Phishing email pretends to be from IRS

The exact phishing tactics employed in these campaigns are highlighted in Cofense's study. Secured by password zip files and misleading email subjects like "IRS Tax Forms W-9" and "INCOME TAX RETURN 2021" are some examples of these strategies. These methods are designed to circumvent security email gateways and trick users into enabling dangerous macros in the attached Excel files. The goal of these methods is to circumvent security email gateways and trick users into enabling dangerous macros in the attached Excel files. . While some people may find it difficult to extract the zip files using Windows' built-in utilities, third-party solutions such as 7- Zip can get around this problem, which somewhat reduces the campaign's effectiveness. Organizations must exercise extreme caution in light of the serious consequences of these infections, which include the potential for ransomware attacks and substantial data exfiltration. It is important to keep in mind that the IRS does not send unsolicited emails and that it only communicates via postal service. For this reason, any email claiming to be from the IRS should be viewed with suspicion and immediately deleted.

### *C. Trojanized Tor browsers target Russians with crypto- stealing malware*

Captured by Trojans Installers for the Tor Browser have become a serious danger, especially for users in Eastern and Russian Europe. These malicious installers insert malware that monopolizes crypto currency transactions by taking advantage of the Tor Browser's need for privacy. These installers, which are distributed as improved or reportedly real versions of the official Tor browser, frequently include outdated browser versions in addition to hidden executables that install malware. This malware steals bitcoin transactions without the user's knowledge by keeping an eye on the clipboard for addresses, then replacing them with ones under the control of the attackers.

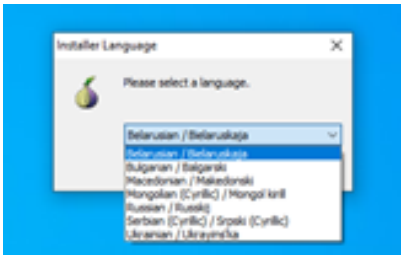


Fig. 4: Malicious Tor Browser language pack

The 2021 Russian website ban on the Tor Project increased the reliance on alternate, less secure download sources, which is why Russia and Eastern Europe were targeted. Although not a very innovative strategy, Kaspersky's detection of over 16,000 versions in 52 countries between August

2022 and February 2023 yielded over $ 400,000 in crypto currency that was stolen. In order to reduce these dangers and preserve the security of their system, users should make sure they download software from reputable sources and run quick tests to look for clipboard hijackers. This will protect their crypto currency transactions.

*D. Android file manager apps infect thousands with Sharkbot malware*

A serious cyber security threat has been highlighted by the recent discovery of malicious Android apps on the Google Play Store that pose as file managers and infect users with the Sharkbot banking Trojan. According to Bitdefender, apps such as ”X-File Manager” and ”FileVoyager” first seemed innocent, evading Google’s security checks by not installing the malicious payload. These apps later downloaded the Shark- bot malware from remote servers, which covers fake login forms over prompts from legitimate banking apps to steal user credentials. The campaign, which targets users in the UK and Italy, uses risky permissions common to file management apps to avoid suspicion.


Fig. 5: X-File Manager on Google Play

Users who previously downloaded these apps are still at risk even if they were removed from the Play Store. The malware’s capacity to carry out anti-emulation tests and to only activate on particular SIM cards emphasizes how evasive and targeted it is. Users should change their online banking passwords and remove any discovered rogue apps right once to guard against such risks. Furthermore, by identifying and eliminating harmful apps, trustworthy mobile security antivirus programs like Google Play Protect can offer vital protection. This instance highlights the necessity of proactive security measures and ongoing monitoring in order to counter growing malware threats such as Sharkbot.

*E. MRm-DLDet: a memory-resident malware detection frame- work based on memory forensics and deep neural network*

This paper addresses memory-resident malware detection strategies, emphasizing the difficulties presented by these elusive threats and the shortcomings of current detection techniques. The MRm-DLDet framework is compared by the authors with a number of memory forensics-based malware detection efforts, such as Bozkir et al., Malfind, Quincy, and Sihwail et al. To identify memory-resident malware, each of these techniques uses a unique set of techniques, includ- ing feature extraction, classification algorithms, and visual representation of memory dumps. The comparison analysis shows that MRm-DLDet performs better than these current methods in terms of recall, accuracy, precision, and F1-score, demonstrating how well the proposed model addresses the drawbacks of conventional detection techniques.

The paper also evaluates the MRm-DLDet framework against cutting-edge vision-based malware detection tech- niques and explores the applicability of vision-based ap- proaches in malware detection. Using visual representations of memory dumps, MRm-DLDet displays higher performance in identifying memory-resident malware by utilizing deep neural networks and attention processes. In order to improve the precision and effectiveness of detecting memory-resident threats that elude conventional detection systems, the research emphasizes the importance of integrating vision-based tech- niques into malware detection. The study also explores the difficulties and potential avenues for memory-resident malware detection, highlighting the danger of deep learning models over fitting as a result of small dataset sizes. The authors discuss the runtime over- head of deep learning-based detection methods and memory forensics, emphasizing the necessity of maximizing detection efficiency in practical applications. In order to provide prompt and effective detection of memory-resident malware, future research topics include using the MRm-DLDet framework in production environments, keeping an eye on harmful sample analysis reports for updated samples, and shortening training and testing timeframes for periodic model upgrades.

### *F. Evaluation of Threat Information Quality Provided by Twitter*

The paper ”Evaluation of Threat Information Quality Pro- vided by Twitter” examines the dependability, detail, and timelines of threat information from Twitter and a cyber-security news website. The study highlights how crucial it is for both security specialists and non-experts to have up-to-date and trustworthy threat information. Although a few studies have assessed the quality of threat intelligence available on Twitter, the medium is acknowledged as a significant source of Open-Source Intelligence (OSINT). The research gathered tweets pertaining to Emotet assaults and juxtaposed them with pieces from a news website covering cyber security. The findings indicated that while Twitter was not as reliable as the news website, it was superior in terms of timeliness and information. The usage of Chat Generative Pre-trained Transformer (ChatGPT) for automated discrepancy detection in web page content was also introduced by the study. In general, the study emphasizes how important it is to evaluate the accuracy of threat intelligence on social media sites like Twitter in order to implement efficient cyber security procedures.

### *G. Malware Reverse Engineering to Find the Malicious Ac- tivity of Emotet*

A methodology for employing reverse engineering methods to examine Emotet malware is provided in the paper ”Mal- ware Reverse Engineering to Find the Malicious Activity of Emotet”. By quickly discovering the payload and identifying crucial function calls that result in malicious activity, the researchers hope to shorten the time needed to analyze Emotet. With the use of CFF Explorer, Ghidra, and other tools, as well as testing on malware samples, the framework is able to detect Emotet malware with an accuracy of up to 90.9%/. The growth of Emotet from a banking credential stealer to a sophisticated threat employing a variety of payloads for malicious actions is also covered in the study, emphasizing how crucial it is to comprehend its behavior in order to develop efficient detection and mitigation techniques.

The literature review section of the paper examines previous studies on malware analysis, machine learning methods for malware detection, and malware activity monitoring tools such as PEStudio and Procmon. The evaluation highlights how important it is to comprehend the control flow graph and

make use of tools for PE file structure analysis in order to spot anomalies and malicious activity. The paper also makes recommendations for future research paths to improve code to identify damaged or corrupted PE files and expand the framework for detecting other kinds of malware. Overall, the study emphasizes the significance of ongoing research and development in the field of malware detection and mitigation and offers insightful information about how to resist Emotet malware through reverse engineering analysis.

### *H. Emotet is back*

First discovered in 2014 as a banking Trojan, Emotet has since matured into much more - notably acting out an instrumental role of downloading secondary-stage payloads such as Trickbot and IcedID from January 2020 onwards. That it has returned since a joint Interpol and Eurojust takedown in January 2021 shows how long-lived the threat is. The module- based architecture of Emotet enables it to lie dormant until attackers want to make good use out from the malware, adding a degree of flexibility in carrying multi-phased attacks. This evolution has allowed Emotet to drop other payloads such as CobaltStrike (which can deploy Ransomware like Ryuk), making it a multipurpose cybercriminal tool. The main Emotet infection method is phishing emails with malicious Office document attachments. The Emotet DLL is downloaded, and its execution occurs when the victim enables macros in the document. Over which being a connection with C2 servers, Emotet communicates with a port of 80 or 8080 for HTTP, and 443 for HTTPS. Emotet is much easier to detect than before, enhanced by the Global Threat Alerts and Secure Endpoint's machine learning, which examines network traffic patterns, identifies which are malicious that make subsequent IPs and domains, and provides informative threat targets. Although there is no solid evidence of Emotet's use in the Exploit or association with the Log4J vulnerability, certain IP addresses utilized by Emotet C2 servers were involved in Log4J Exploits. Depending on whether this seems like the infrastructure shared attack or the opportunist exploiting it.

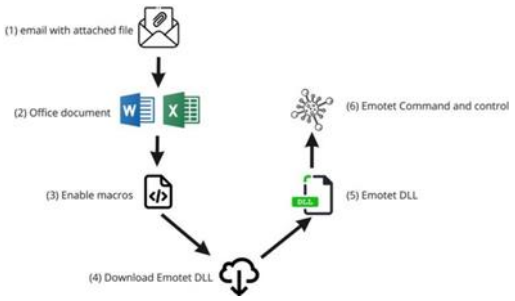


Fig. 6: Emotet attack flow

The implementation of two-factor authentication, isolating compromised machines, limiting the use of PowerShell, and rejecting dubious email attachments are just a few of the security procedures that have been improved as a result of recent spikes in Emotet detections. By taking these steps, we hope to lessen the effects of Emotet's comeback and shield networks from its changing array of threats.

### *I. Emotet growing slowly but steadily since November resurgence*

Earlier this year, Europol and Eurojust coordinated a take- down of the Emotet botnet - long hailed as one of the world's most dangerous malware strains due to its far-reaching influence and

sophisticated infection methods. Emotet now in- fects 130,000 systems across 179 countries despite a dramatic decrease from the height of its in containing control over1.6 million devices. The resurfacing of this strain in November 2021, with the help of Trickbot malware and carried out by Conti ransomware gang demonstrates its continuous evolution and difficult to deal structure. Emotet specifically has changed its payload strategy to now bypass Trickbot and instead load the Cobalt Strike pentesting tool, which opens up remote access of compromised networks at rapid pace making it a more efficient adversary for other cybercriminal campaigns.

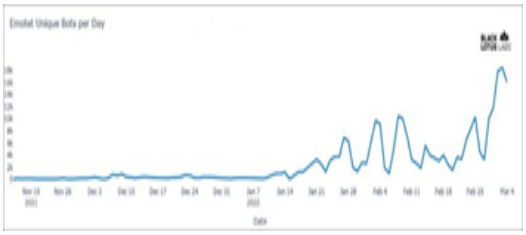

Fig. 7: Emotet bot distribution shifting up a gear

The botnet's "Epoch 3" versions have seen critical evolution of its encryption mechanisms and information gathering capabilities, which threat analysts such as those at Black Lotus Labs diligently monitor. Support for protecting network traffic using Elliptic Curve Cryptography (ECC) more efficiently than the older RSA encryption. And the use of a process list module after connecting to command and control (C2) servers implies better system profiling techniques. Emotet benefits from its continually updated infrastructure, featuring 200 distinct C2 servers mostly hosted in the United States and Germany. The spike in engagement is particularly evident among those with many ageing Windows systems, including Japan, India and Indonesia (where vulnerabilities like exploit CVE-2021-43890 the AppX Installer spoofing bug are being used). Although Mircosoft is trying to fight this loop hole by freezing OEM updates for machines with pirated Windows- the most likely software used in Caribbean offices based on sustainability projections -the persistence of systems using pirated versions (i.e., no automatic update patching) preventing virtual blackguards like Emotet from spreading, underscores crypto-massacre's continuing saga within global cyber security.

### *J. Trojan Sources:Invisible Vulnerabilities*

The concept of Trojan Source attacks—where source code is maliciously encoded to appear different to compilers and human reviewers—is introduced by Boucher and Anderson in their paper "Trojan Source: Invisible Vulnerabilities". They show how nuanced text encoding methods, like Unicode, can be used to introduce vulnerabilities that are difficult for conventional code review procedures to find. In order to address this new class of vulnerabilities, the authors emphasize the necessity of coordinated disclosure and compiler-level safe- guards. A wide range of programming languages, including C, C++, C, JavaScript, Java, Rust, Go, Python, SQL, Bash, Assembly, and Solidity are also covered in their discussion of the ramifications of these assaults.

The authors also stress the need of the coordinated industry- wide disclosure process for these vulnerabilities because they impact a variety of editors, compilers, and repositories. This coordinated disclosure effort highlights the value of coop- eration in tackling security risks that cut across numerous platforms and languages and offers insights into how various stakeholders react to

vulnerability releases.

Overall, by highlighting the need of both safe usability and compiler verification, the study advances the topic of security usability from the developer's point of view. The authors push for strong safeguards against Trojan Source assaults and pose significant doubts regarding the reliability of compilers. The report also lists the major contributions, which include defining Trojan Source assaults, offering instances in different programming languages, suggesting compiler-level protections, recording the coordinated disclosure procedure, and raising doubt on compiler reliability.

## III. PROPOSED SYSTEM

The project is a web application built using Flask that lets users upload files to be scanned for malware using YARA rules. The program configures an upload folder, loads YARA rules from a directory, and establishes the web interface's routes.

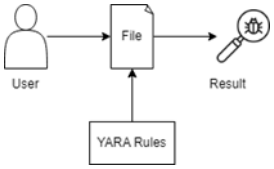


Fig. 8: System Architecture


Fig. 9: File Uploading

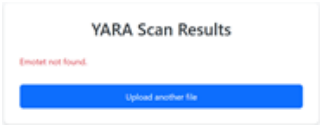


Fig. 10: YARA Scan Result

A file upload form can be found via the main route (/), and the file upload procedure is handled via the /upload route. The application uses the loaded YARA rules to scan a file after it is uploaded, saving it to the chosen folder. The user sees an error message if no file is selected. A results page is then presented with the scan's findings, including any matches discovered by the YARA guidelines. The user interface, which was created using Bootstrap, intends to offer a visually appealing and intuitive experience. Detailed error messages will be incorporated for improved usability. In order to identify Emotet malware, this project focuses on identifying a variety of file types, including ELF, PE, PDF, GIF, RAR, PNG, MP3, and ISO files.

Upon initializing a dictionary to store the rule files, the 'load yara rules from directory(directory

path)‘ method iterates over the files in the specified directory to gather those that have ‘.yar‘ or ‘.yara‘ extensions, builds their entire file paths, and employs ‘yara.compile‘ to compile these YARA regulations. If the compilation is successful, the method returns the compiled rules; if not, it finds any problems throughout the compilation process, emits an error message, and returns ‘None‘.

```
# Function to load YARA rules from a directory
def load_yara_rules_from_directory(directory_path):
    rule_files = {}
    for file_name in os.listdir(directory_path):
        if file_name.endswith('.yar') or file_name.endswith('.yara'):
            file_path = os.path.join(directory_path, file_name)
            rule_files[file_name] = file_path

    try:
        rules = yara.compile(filepaths=rule_files)
        print(f"Successfully loaded YARA rules from {directory_path}")
        return rules
    except yara.Error as e:
        print(f"Error compiling YARA rules: {e}")
        return None

# Function to scan a file using the loaded YARA rules
def scan_file(file_path, rules):
    try:
        matches = rules.match(file_path)
        return matches
    except yara.Error as e:
        print(f"Error scanning file: {e}")
        return []

# Load the YARA rules
rules_directory_path = 'utility'
yara_rules = load_yara_rules_from_directory(rules_directory_path)
```

Fig. 11: Source code snapshot

The second function, ‘scan file(file path, rules)‘, looks for matches between the file and the rules using a file path and the pre-installed YARA rules as parameters. It detects problems during the matching process, produces an error notice, and replaces the found matches with an empty list. The general process entails defining the path to the directory containing the YARA rule files, loading and assembling the YARA rules from this directory, and then employing the rules to perform file scanning. Both functions include error handling to guarantee that the application can manage problems during rule compilation and file scanning without crashing.

Flask is a simple and adaptable web framework for Python, it is perfect for creating web apps that require little setup. It offers necessary tools for processing HTTP requests, routing, and templating. Flask is used in this project to construct a web interface that allows users to upload files for malware detection. The upload form is displayed by the main route (/), and the uploaded files are processed, scanned using YARA rules, and the results are displayed on a separate page by the ‘/upload route. The backend operations and user interface of the application may be easily developed and managed because to Flask’s ease of use and capabilities.

*A. YARA Rules*

YARA is a useful tool for identifying and classifying mal- ware samples; malware researchers often refer to it as the ”pat- tern matching Swiss knife.” Using textual or binary patterns, users can describe malware families that can subsequently be used to search files or processes for these patterns. A Boolean expression that specifies the circumstances in which a file or process will be matched by a YARA rule is combined with a series of strings to produce a rule. These criteria are useful for identifying and categorizing malware since they are flexible and clear, enabling security professionals to assess threats by examining the unique characteristics of malware that has been identified. To increase detection capabilities and strengthen cyber security defenses overall, threat intelligence, incident response, and research all heavily rely on YARA. YARA rules are highly adaptive to different kinds of malware because they support intricate logic and conditions. Additionally, YARA

is open-source, which makes it available to a broad spectrum of security experts and organizations. YARA rules are utilized in this project to check uploaded files for malware signatures.

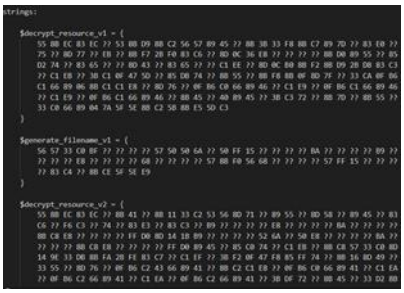

Fig. 12: Yara Rules

A file is saved in a specific upload folder when it is uploaded using the Flask web application. After that, the program loads the YARA rules from a designated directory and applies them to the file that was uploaded. These criteria look for particular patterns and traits linked to several kinds of malware, including Emotet. Informing the user about potential risks in the uploaded material, the results of any matches are shown on a separate results page. Through this integration, the project can better identify and guard against malware by utilizing YARA's robust detection capabilities.

## IV. CONCLUSION AND FUTURE SCOPE

The research paper developed a reliable method using YARA rules in a easy to use Flask web application to identify Emotet malware. The project's primary results include using a large number of YARA rules to identify various Emotet related file formats, creating a responsive web interface using Bootstrap, and enhancing user experience by using error management and user-friendly design.

While Virus Total facilitates file scanning using Yara rules, its adoption by organizations is limited due to the risk of data breaches associated with publicly visible file uploads. Implementing a web application within an access-controlled environment mitigates this risk by obviating the need to upload suspicious files to a publicly accessible platform, thereby providing a more secure solution for organizations.

The project's goals of user accessibility and efficient virus detection were both met. Additional malware varieties could be covered by the detection capabilities in future, and machine learning could be included for adaptive threat detection. This research paper seems to establishes a framework for efficient malware analysis and user-centered design in cyber security software.

## REFERENCES

[1] Emotet malware now steals credit cards from Google Chrome users, Bleeping Computer. [Online]. Available: https://www.bleepingcomputer.com/news/security/emotet-malware-now-steals-credit-cards-from-google-chrome-users/. [Accessed: 17-Jul-2024].

[2] Emotet malware campaign impersonates the IRS for 2022 tax season, Bleeping Computer. [Online]. Available: https://www.bleepingcomputer.com/news/security/emotet-malware-campaign-impersonates-the-irs-for-2022-tax-season/amp/. [Accessed: 17-Jul-2024].

[3] Trojanized Tor browsers target Russians with crypto-stealing malware, Bleeping Computer. [Online]. Available: https://www.bleepingcomputer.com/news/security/trojanized-tor- browsers-target-russians-with-crypto-stealing-malware/

[4] Android file manager apps infect thousands with SharkBot malware, Bleeping Computer. [Online]. Available: https://www.bleepingcomputer.com/news/security/android-file-manager- apps-infect-thousands-with-sharkbot-malware/.
[5] J. Liu, Y. Feng, and X. Liu, ”MRm-DLDet: a memory-resident malware detection framework based on memory forensics and deep neural network,” Cybersecurity, vol. 6, p. 21, 2023. Available: https://doi.org/10.1186/s42400-023-00157-w.
[6] G. Manohar, S. Chandran, and A. U, ”Malware Reverse Engineering to Find the Malicious Activity of Emotet,” in ATDE, 2023, doi: 10.3233/ATDE221253.
[7] P. Boucher, ”Learning and Detecting Malware with Neural Networks,” presented at the 2023 USENIX Security Symposium. Available: https://www.usenix.org/system/files/sec23fall-prepub-151-boucher.pdf.
[8] Emotet is back, Cisco Blogs. [Online]. Available: https://blogs.cisco.com/security/emotet-is-back.
[9] Emotet growing slowly but steadily since November resurgence, Bleeping Computer. [Online]. Available: https://www.bleepingcomputer.com/news/security/emotet-growing- slowly-but-steadily-since-november-resurgence/.
[10] https://personales.upv.es/thinkmind/dl/journals/sec /sec$_v$16$_n$34$_2$023/sec$_v$16$_n$34$_2$023$_1$.pdf