# Self - Hosted Malware Analysis Lab

## Zeus Banking Trojan Analysis

***Aleena Plessey - 21BCE5767***
*Bachelor of Technology CSE*
*VIT Chennai*
*aleenaanna.plessey2021@vitstudent.ac.in*

***Greeshma Reddy - 21BCE5786***
*Bachelor of Technology CSE*
*VIT Chennai*
*basireddygreeshma.reddy2021@vitstudent.ac.in*

***Heet Chanchad - 21BCE5809***
*Bachelor of Technology CSE*
*VIT Chennai*
*heet.chanchad2021@vitstudent.ac.in*

***Abstract -*** This paper addresses the proliferation of malware in cybersecurity by introducing a self-hosted malware analysis lab. Combining static and dynamic analysis techniques, the lab effectively inspects, classifies, and mitigates malicious software threats. Through examples such as the Zeus banking trojan, it demonstrates the lab's efficacy in scrutinizing malware behavior and identifying security risks. Leveraging compartmentalization techniques and advanced tools, the lab provides a secure environment for dissecting malware samples without compromising network integrity. Practical considerations like architectural components, deployment strategies, and scalability factors are explored, guiding organizations in establishing their self-hosted malware analysis labs. Beyond malware analysis, the paper discusses the lab's utility in threat intelligence gathering, incident response training, and collaborative research initiatives. Continuously refining its capabilities and staying ahead of emerging threats, the lab serves as a crucial asset in enhancing cybersecurity resilience across industries and sectors.

***Keywords:*** *Self-hosted, Virtual environment, Malware, Zeus Trojan, Response Training, Static Analysis, Dynamic Analysis*

## I. INTRODUCTION

With the increasing threat of malware, having a solid defense strategy is crucial for safeguarding digital assets. Malware, like viruses and ransomware, can cause significant harm, from stealing sensitive information to disrupting operations. Traditional methods of detecting malware are often not enough to catch new and evolving threats, making dynamic analysis vital. This paper proposes creating a self-hosted malware analysis lab, providing a controlled environment to study and neutralize malware. Unlike relying solely on external services, hosting our analysis lab gives us more control and flexibility, which is essential for researchers and analysts. By building our lab, we can strengthen our cybersecurity defenses and better protect against emerging threats.

### *A. Motivation*

The purpose of the paper is to conduct a thorough malware analysis lab to overcome difficulties faced through traditional approaches. Traditional approaches to malware analysis often suffer from limitations in terms of flexibility, customization, and reproducibility. By creating a self-hosted lab, we gain the ability to tailor their environments to specific research needs, ensuring precise replication of experiments and promoting transparency in findings. Moreover, self-hosted labs serve as collaborative platforms, facilitating interdisciplinary cooperation between academia, industry, and cybersecurity practitioners. These labs empower researchers to experiment with novel analysis techniques, iterate rapidly in response to emerging threats, and contribute to the advancement of knowledge in the field. Additionally, self-hosted labs play a crucial role in education and training, providing students and professionals with hands-on experience in malware analysis and strengthening their capabilities in defending against evolving cyber

threats. Furthermore, self-hosted labs offer enhanced data privacy and security, ensuring that sensitive information remains protected within controlled environments.

### *B. Organization of paper*

Following the introduction, the paper's structure comprises sections such as background study, proposed methodology, results, conclusion, and future scope. The background study section reviews prior research in malware analysis, while the proposed methodology outlines the specific approach employed within the self-hosted malware analysis lab, with a particular focus on analyzing the Zeus banking trojan. The results section presents the findings of the Zeus banking trojan analysis, followed by a conclusion that summarizes the study's outcomes and discusses their implications. Finally, the future scope section delves into prospective avenues for development and enhancement within the realm of self-hosted malware labs.

## II. BACKGROUND STUDY

In this section, we provide a comprehensive overview of relevant studies done prior in the field of malware analysis, drawing insights from recent literature.

Dai et al. [1] underscore the shortcomings of conventional signature and heuristic-based technologies in malware detection, advocating for the adoption of a holographic platform for behavioral profiling. However, this approach lacks adaptability to emerging malware threats and relies heavily on user-triggered events for detection. Fukushima et al. [2] propose a malware detection method achieving a 60% detection rate while minimizing false positives compared to traditional behavior-based antivirus software. Yet, the efficacy of their approach remains limited, partly due to the constraints of the dataset used for evaluation. Ucci et al. [3] emphasize the growing significance of machine learning in malware analysis, categorizing research based on objectives and features for analyzing Portable Executable (PE) files. Meanwhile, J. Bermejo Higuera et al. [4] advocate for a SAMA (Static, Analysis, and Memory Analysis) approach to malware analysis, emphasizing the importance of behavioral analysis in uncovering malicious intent, despite its susceptibility to obsolescence in the face of rapidly evolving malware techniques. De Vincente Mohino et al. [5] stress the importance of incorporating static, dynamic, and memory analysis to gain a comprehensive understanding of malware behavior, particularly in the context of vulnerabilities in Linux operating systems. Finally, R. S. Kunwar et al. [6] address the grave threat posed by malicious code to computer systems, discussing the division of malware analysis into static and dynamic approaches, alongside the challenges posed by evasion techniques employed by malware developers to conceal their source code.

## III. PROPOSED METHODOLOGY

In this section, we present our proposed methodology for the self-hosted malware analysis lab, specifically tailored for analyzing the Zeus banking trojan.

Our methodology incorporates both static and dynamic analysis techniques, synergizing their respective strengths to achieve robust malware detection and comprehension. By combining insights from various studies, we are devising an effective framework capable of enhancing the detection and understanding of malware threats, particularly in the context of the Zeus banking trojan.

1. *Oracle Virtual Box*: Oracle VirtualBox, is a free and open-source virtualization platform that allows users to create and run virtual machines on their host operating system. VirtualBox provides features such as snapshotting, which enables users to save the current state of a virtual machine and revert to it later if needed. It is widely used by researchers, developers, and IT professionals for testing, development, and experimentation in virtualized environments.
2. *Flare VM:* With its adaptable framework and advanced features, Flare VM equips security professionals with the tools necessary for malware analysis, detection, and classification within their own controlled environments. Leveraging its robust text classification techniques and flexible architecture, Flare VM provides a comprehensive solution for identifying and understanding the intricacies of malicious code.
3. *REMnux:* REMnux, a streamlined and comprehensive toolkit, simplifies the setup of a self-hosted malware lab. With its curated selection of security tools and pre-configured environment, REMnux offers an efficient solution for analysts to establish a controlled ecosystem for malware analysis.

To establish a self-hosted malware lab, integrate Flair VM and REMnux. First, set up REMnux, a Linux distribution specialized for malware analysis, and then install Flair VM on the same system. Utilize REMnux' s suite of tools for malware analysis, complemented by Flair VM's natural language processing capabilities for analyzing text-based artifacts associated with malware. This integrated approach allows for comprehensive analysis of malware samples within a controlled environment, empowering analysts to understand and combat malicious threats effectively.

## A. *Topology*

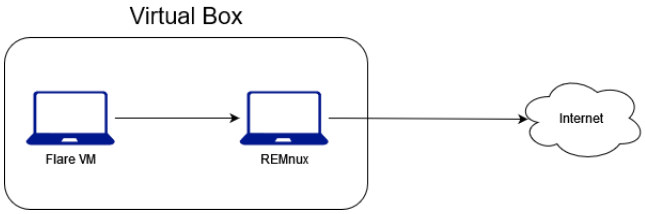


*Figure 1: Topology*

We put our self-hosted malware analysis system to the test by scrutinizing the Zeus banking trojan. This notorious malware, renowned for its sophisticated techniques and pervasive threat, serves as an ideal benchmark to evaluate the effectiveness of our security measures.

## B. *Static Analysis*

a) VirusTotal, an online service to analyze files gave us a community score of 65/75 which shows that this file is a highly malicious file. This is a good start on fingerprinting the executable file.

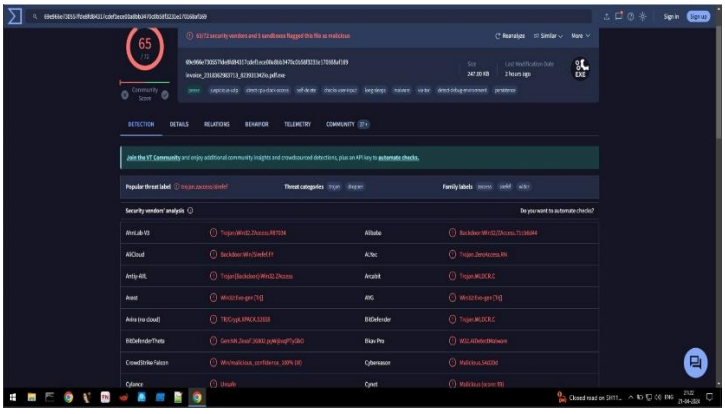

*Figure 2: Analysis of Trojan file using **VirusTotal***

b) PEStudio is a specialized tool used for analyzing executable Windows files. Using this software, which is already pre-installed on FlareVM, we get a detailed report including the hash values, filetype and size. We see that this file is actually an exe file and not a pdf. We also find out that it links to a domain correct.com which yielded no useful results. We also found out all the API calls that the executable calls. Some API calls were used to find clipboard data, keystrokes and environment variables which indicates this a powerful program when executed.

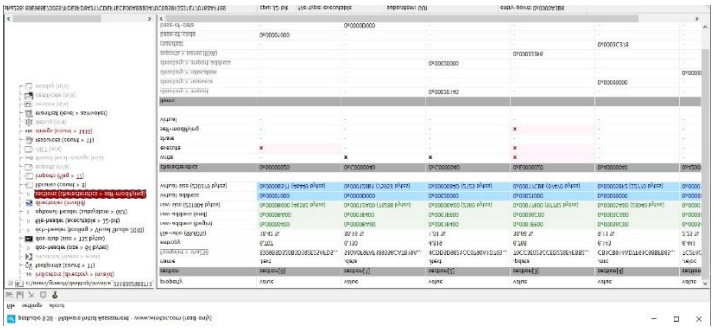

*Figure 3: Analysis of Trojan file using **pestudio***

c) FLOSS aids analysts in understanding the behavior and capabilities of malware by revealing hidden commands, URLs, and other crucial information. In the analysis of the FLOSS output, we can see there are many functions names of randomly generated characters which is a common technique used by malware authors to obfuscate the overall functionality of the code.

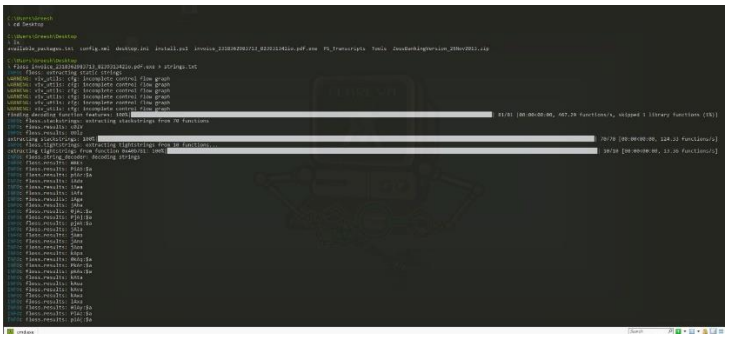

*Figure 4: Analysis of Trojan file using **Floss***

## C. *Dynamic Analysis*

We use FlareVM as the environment to detonate the binary and REMnux machine is going to impersonate the DNS service and act as a C2 server.

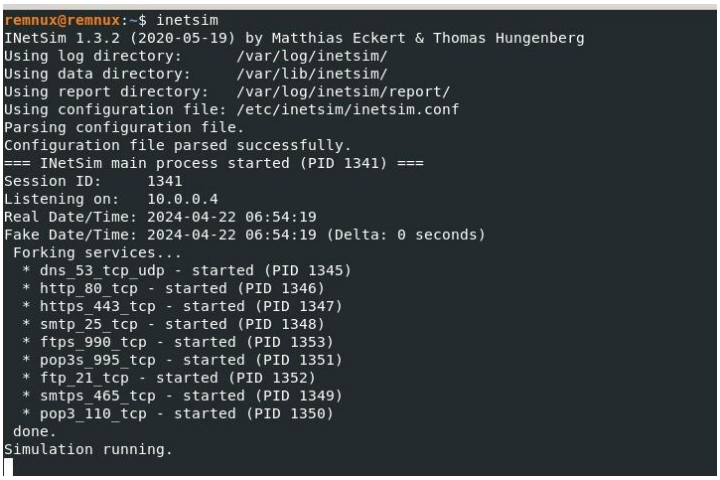


*Figure 5: inetsim starting*

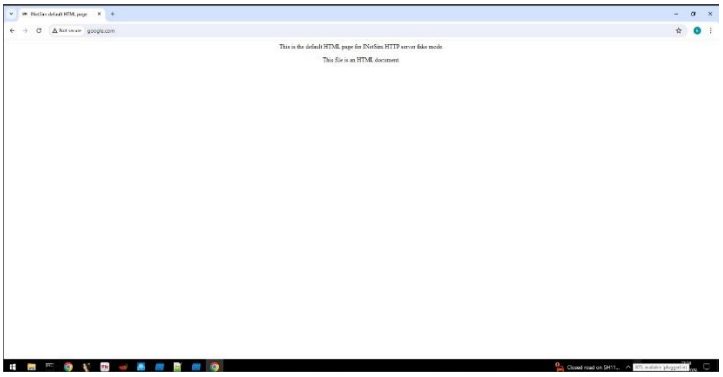

*Figure 6: Default HTML web page*

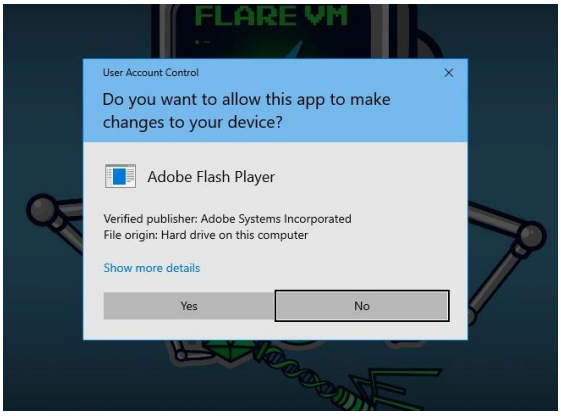


*Figure 7: Installation Prompt while detonating Malware*

a) Process Monitor is a monitoring tool developed by Microsoft to provide real-time visibility into the activities happening on our FlareVM machine. On detonating the malware, we see that it deletes its self from the Desktop after executing can view all the parent and child processes happening as well as the hidden sessions where commands are being executed.

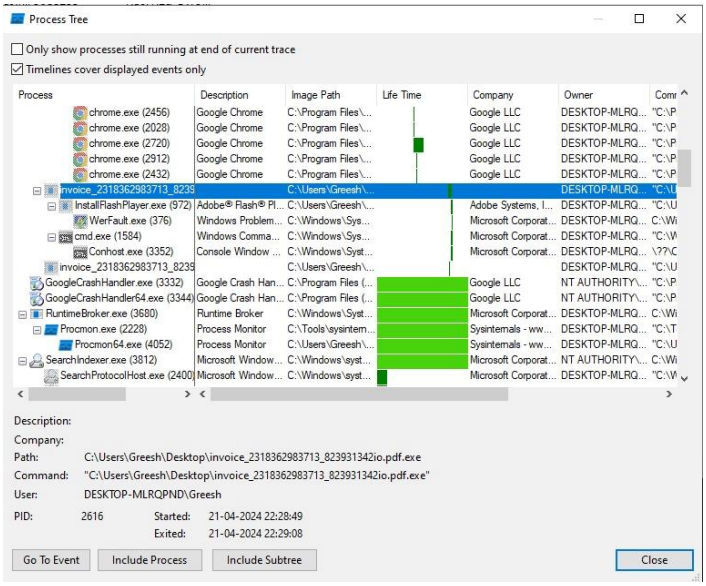


*Figure 8: Process Monitor Analysis*

b) Wireshark is then used to check if there are network-based indicators for this binary. This checks if our binary reaches out to any C2 server. We see an HTTP request sent to get Flash Player and a hardcoded website is found which is not found as a malicious website. No other hard coded IP requests were found.

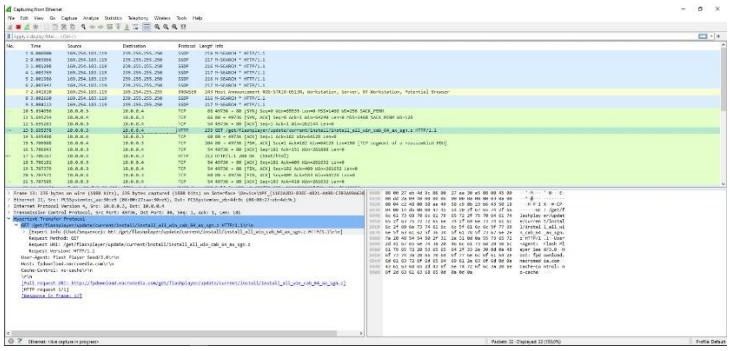

*Figure 9: Wireshark Packet Filtering*

*Figure 10: Analysis of suspicious HTTP Request Packet*

## V. RESULT

This section presents the outcomes of our analysis focused on detecting and understanding the Zeus banking trojan using our proposed methodology. We highlight the effectiveness of our approach, which combines behavioral profiling and comprehensive analysis techniques. Importantly, our results showcase the successful detection and characterization of the Zeus trojan, demonstrating the practical applicability of our methodology in real-world malware analysis scenarios.

### *1. Malware Analysis:*

The Zeus Banking Trojan on execution installs a few different programs. It specifically installed Flash Player which on closer inspection seems to be a legitimate binary. However, it also installs a malicious DLL file in the background. It establishes persistence through a Google update and infiltrates the Google Update binary. So, any time Google Chrome is updated, the binary file will be executed. We extracted a domain called "fpdownload.micromedia.com" using Wireshark while doing network packet analysis which also seems to be a legitimate domain name verified using the wayback machine.
Using VirusTotal to view the review on the malicious DLL file downloads during the first execution of the invoice, we see that 51 security vendors have flagged this file as having some malicious capability.

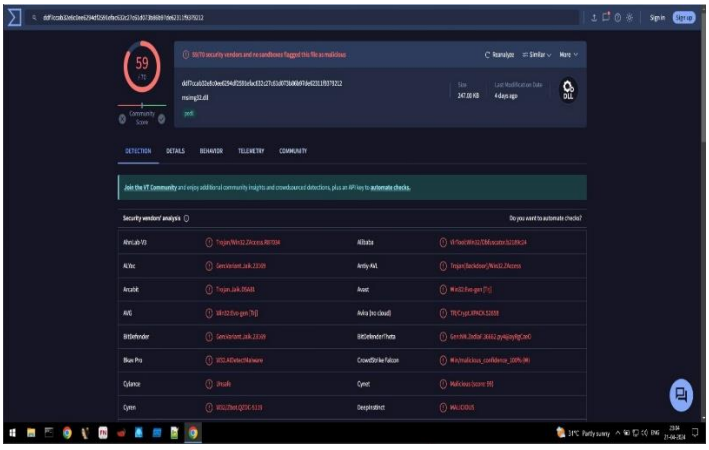

*Figure 11: Analysis of hidden downloaded DLL file*

## VI. CONCLUSION

In conclusion, the development and implementation of a self-hosted malware analysis lab represent a crucial step forward in the ongoing battle against cyber threats. Through our research we have demonstrated the effectiveness of such a lab by carefully scrutinizing the well-known Zeus Banking Trojan. By leveraging tools like Process Monitor, Wireshark and FLOSS we were able to conduct static as well as dynamic analysis on the binary.

## VII. FUTURE SCOPE

In this section, we delve into the prospective avenues of development and enhancement within the realm of self-hosted malware labs.
As the cybersecurity landscape evolves, the future of these labs holds immense potential for further innovation and refinement. Our exploration encompasses key areas such as automation, advanced analytical techniques, scalability, resource optimization, and enhanced collaboration. We can automate this process of malware analysis which can streamline workflows and improve efficiency, which can lead to rapid responses to emerging threats. Future research can be done to enhance advanced automation techniques to improve scalability.
Employing machine learning and artificial intelligence can also enhance detection accuracy and identify previously unknown malware variants.

## REFERENCES


[1] S. Y. Dai, Y. Fyodor, M. W. Wu, Y. Huang, and S. Y. Kuo, "Holography: a behavior-based profiler for malware analysis," Software: Practice and Experience, vol. 42, no. 9, pp. 1107-1136, 2012.

[2] Fukushima, Y., Sakai, A., Hori, Y., & Sakurai, K. (2010, October). A behavior-based malware detection scheme for avoiding false positive. In *2010 6th IEEE workshop on secure network protocols* (pp. 79-84). IEEE.

[3] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," Computers & Security, vol. 81, pp. 123-147, 2019.

[4] J. Bermejo Higuera, C. Abad Aramburu, J. R. Bermejo Higuera, M. A. Sicilia Urban, and J. A.

Sicilia Montalvo, "Systematic approach to malware analysis (SAMA)," Applied Sciences, vol. 10, no. 4, pp. 1360, 2020.

[5] J. J. de Vicente Mohino, J. Bermejo-Higuera, J. R. Bermejo Higuera, J. A. Sicilia, M. Sánchez Rubio, and J. J. Martínez Herraiz, "MMALE a methodology for malware analysis in linux environments," Computers, Materials & Continua, vol. 67, no. 2, pp. 1447-1469, 2021.

[6] R. S. Kunwar and P. Sharma, "Malware analysis: Tools and techniques," in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, March 2016, pp. 1-4.