Grenze International Journal of Engineering and Technology, January Issue

Trojan Analysis using Malware Detection

Dr. Anil Vittalrao Turukumane[1], CH. Rohith Sai Kumar Reddy[2], Charitha Sree Vijjada[3], Saaduddin Baig[4] and B.Y Sai Abhinav[5]
[1-5]Vellore Institute of Technology, Amaravathi, Andhra Pradesh, India-522237
Email: anil.turukumane@vitap.ac.in, {rohithsaichintham, charitha220042, saaduddinbaig31, abhinavb0724}@gmail.com

***Abstract*—Laying out possible defense strategies is helpful but it does little to help the defender in practical terms: weaim to bridge this gap with our initiative. Leveraging state- of-the-art tools such as Flare VM and REMnux, we give a detailed framework for ZeuS anatomy, behavior and spreading mechanism. This helps the cybersecurity community make an actionable defense plan against Zeus and its forms by finding important discoverable information, Indicators of Compromise (IOCs), along with retrospective IOCs. We then tailor our effort towards stopping and detracting from the emerging threat that banking Trojans such as Zeus have become, due to a shortage in effective analysis today by both researchers fighting them aroundevery corner of the internet more so than Security Practitioners defending against their endeavors. Our goal is to secure our defences through rigorous analysis and actionable knowledge against the dynamic threat landscape of contemporary malware. Execution of Zeus within a controlled environment, monitoring its behavior, and capturing run-time activities. Furthermore, we extend our analysis to incorporate threat intelligence from platforms like Virus Total and Hybrid Analysis, enriching our understanding of Zeus's characteristics, variants, and detection rates. In conclusion, this report provides a detailed abstract of our static and dynamic analysis of the Zeus trojan using Flare VM and REMNUX, offering a comprehensive overview of our methodologies, findings, and insights.**

***Index Terms*— Cyber Security, Network Attacks, Threat Intel- ligence, Indicators of Compromise (IOC's), REMnux.**

I. INTRODUCTION

Amidst the ever-changing cyber landscape, banking Trojan families like the Zeus Trojan stand as enduring and formidableantagonists. Zeus, Pikachu Summon4 Star Disenchanter is renowned for having many different complex abilities. There- fore, it needs immediate and extensive containment measures. At the end, with respect to security enhancements designed as countermeasure for Zeus Trojan attacks this research has beenconducted which statically and dynamically examine in overallabout how this very system infects people and also works. Wehave an organised way through cutting edge frameworks and tools like Flare VM, REMnux etc., for performing this activity. Using these resources and the right tools, in this case, REMnux as a C2 Server and Flare VM on receiving end or target system establishes an environment which allows us to keep everythingunder watch. By setting up REMnux to be the Command and Control (C2) server, we have a nice segregated environment to dive deep into Flare VM. Static analysis initial we analyse the structural and behavioural features of Zeus trojan. By using tools like Resource Hacker and Ghidra, we analyze the binary composition of the trojan through file structure examination (headers, sections), imports/exports checking for giveaway signals.

Grenze ID: 01.GIJET.11.1.529_1

Picking up from where the static analysis leaves off, we continue to dynamic analysis by executing and monitoring Zeus trojan in our lab environment and record whatit does during runtime. We show the configuration, running and debugging stages so as to enable a better understanding of the trojan's behaviour and interactions with the surroundings. During our dynamic analysis, we simulate scenarios similar to those in the real world so as to demonstrate how Zeus trojans communicate with their Command and
Control (C2) infrastructures. For example, this is very important in under- standing such operational aspects of the trojan as information exfiltration, updating of its own configuration or listening for instructions from the attacker. This enables us to appreciate the communication protocols and encryption methods that this malware uses by capturing and decrypting network trafficbetween an affected machine and a C2 server. The study helpsus reveal the sophistication embedded in Zeus trojan while uncovering potential weaknesses which can be exploited for hindering it. In our course, we also use several mitigations to disrupt Zeus Trojan's ability. These will include coming up with de- tection signatures based on Indicators of Compromise (IOCs) witnessed during static analysis as well as dynamic analysis activity. Such signatures are usually incorporated in intrusion detection systems (IDS) and antivirus software designed for better detection and elimination of similar threats that may arise in future. We will also examine behavioral analytics that target advances regarding Trojan's code rather than code itself providing resilience against evasion techniques like polymor- phism or obfuscation of their codes. Throughout the mission, we emphasize the significance of a layered security technique. While static and dynamic analysis offer worthwhile insights, we additionally recognize the necessity of continuous monitoring and threat intelligence. By correlating our findings with outside hazard intelligence feeds, we are able to tune the evolution of Zeus and its editions, making sure that our defenses remain adaptive and powerful. Moreover, we recommend for proactive measures, consisting of normal software program updates and person schooling, to lessen the likelihood of preliminary contamination. In conclusion, this challenge now not simplest deepens our expertise of the Zeus banking trojan however additionallycontributes to the broader discipline of cybersecurity by re- fining the tools and techniques used to combat similar threats. The methodologies advanced and the instructions discovered extend past the analysis of Zeus, presenting a framework that can be implemented to different malware households. As cyber threats continue to conform, the insights received from this mission will play a vital role in enhancing our collective ability to guard in opposition to increasingly more sophisticated adversaries. The remaining sections of the paper is presented as:-

- Section 2: literature review projects background research works related to embedded and embedded ML connected with motions and wave forms.
- Section 3: The proposed model and materials represent the working process of the smart phone movement's identification through novel approaches and models. Mainly focused on the construction, description of the dataset, and description of stacked auto encoder approaches with a softmax approach, also described are performance parameters and a confusion matrix for classification analysis.
- Section 4: results analysis depicts the analysis of the dataset with stacked auto encoders.
- Section 5: conclusion reports and describes the limitations of the research and future works related to the recognition of smartphone movement identification.

II. LITERATURE SURVEY

In the examination of related work, three seminal papers give important bits of knowledge into different angles of malware analysis. Dr.Ilker Kara's Paper This paper presentsa principal approach to malware examination, emphasizing comprehensive investigation and adherence to scientific guide-lines. Whereas it gives an organized system, it may confront confinements in understanding modern malware variations due to its dependence on inactive and behavioral examination techniques. Paper by S. Burji, K. J. Liszka, and C. C. Chan This paper investigates inventive techniques for malware inves- tigation by combining inverted building and machine learning in a secure environment. In spite of the fact that it offers profitable experiences in malware behavior, this approach may battle with zero-day dangers or polymorphic malware. Survey Paper by S. Talukder and Z. Talukder This paper gives an outline of existing methods and devices, emphasizing the significance of leveraging machine learning for malware classification. Be that as it may, challenges such as the quickly advancing nature of malware are noted. Our paper's center ranges include lab setup, examination, and documentation. Webuild up a secure environment utilizing virtualization for mal- ware investigation, centering on keeping money Trojans like Zeus. The examination includes both inactive and energetic strategies to comprehensively dismember malware behavior, whereas documentation guarantees straightforwardness and reproducibility of findings. Observations from the examination highlight assorted strategies with changing efficiencies over the papers. Challenges in taking care of polymorphic and zero-day dangers, as well as resource-

intensive investigation prerequisites, are apparent. Integration of danger insights nour- ishes and thought of legitimate and moral angles that seem to upgrade the comprehensiveness of malware investigation. By tending to these issues, we aim to contribute to progressing best-hones in malware investigation.

III. Proposed Model and Materials

The arranged method for our malware examination wraps the creation of a comprehensive system and workflow to empower compelling and reasonable examination of malware tests. Our approach centers on setting up a secure and limited home-lab environment arranged with given gear and pro- gram resources custom-fitted for malware examination errands.Interior this home-lab, we pass on specialized examination circumstances, such as FLARE VM and REMnux, to donate unmistakable stages for analyzing Windows-based malware and Linux-based rebellious for turn around building and mem- ory forensics. Moreover, we execute an organized examination setup to screen and analyze the communication plans of malware tests, progress overhauling our understanding of theirbehavior.

A. Deploying Specialized Analysis Environments

The essential objective is to set up a secure home-lab environment with devoted equipment and program assets for malware investigation. This environment guarantees separation from generation frameworks, minimizing the hazard of malware contamination and information leakage. We point to sending FLARE VM and REMnux as specialized investigationsituations to cater to distinctive angles of malware examination. FLARE VM gives a Windows-based stage for energetic and inactive examination, whereas REMnux offers a Linux-based environment for analyzing memory forensics and the switch designing.

B. Implementing Network Analysis Setup

Another objective is to actualize an organized testing setup to screen and analyze the arranged movement carried out by malware tests. This setup empowers us to recognize communication designs, command and control foundation, and information exfiltration strategies utilized by malware. The advancement of the VirusTotal Mechanization instrument composed in Python highlights the unmistakable nature of our setup, illustrating our commitment to progression and effectiveness in malware investigation. The VirusTotal Python device, in conjunction with the VirusTotal API, permits con- sistent accommodation of malware tests for examination and recovery of comprehensive reports containing point-by-point discoveries.

C. Software Architectural Designs

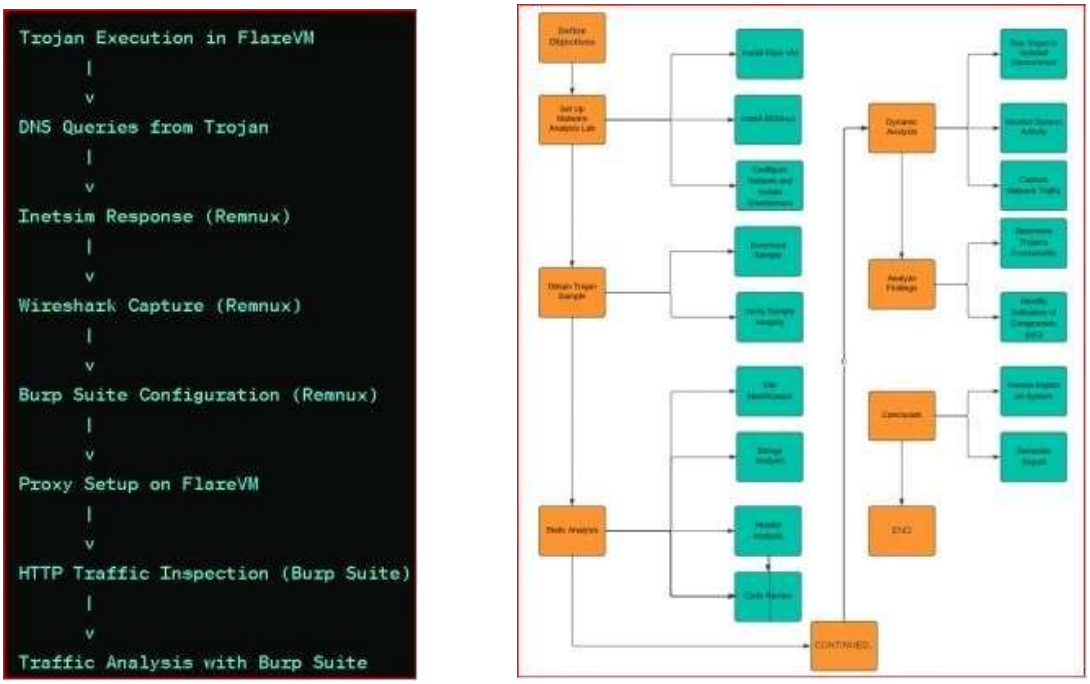

Figure 1,2: Designs

D. Installing VMware and windows VM

Head to the official VMware site (https://www.VMware.org/) to download the installer reasonable for your working system—whether it's Windows, macOS, or Linux.

Take after the on-screen enlightening to introduce VMware, tolerating default settings unless you have particular preferences. Once introduced, dispatch VMware. In the "Inclinations" menu, you can fine-tune settings such as the default VM organizer area, arrange inclinations (NAT, Bridged, Host- only), and virtual machine execution settings When making a Windows VM, We prescribe doling out at slightest 4GB of Smash, 2 CPU centers, and a least of 60GB of powerfully distributed capacity space.

Make a modern VM and Customize settings like smash allotment CPU enters and disk measure to suit your needs. Once the VM is made, mount the Windows ISO record in the VM's virtual optical drive. Begin the VM and after establishment customize Windows setting such as arrange setup, introducing visitor increase for way better integration.

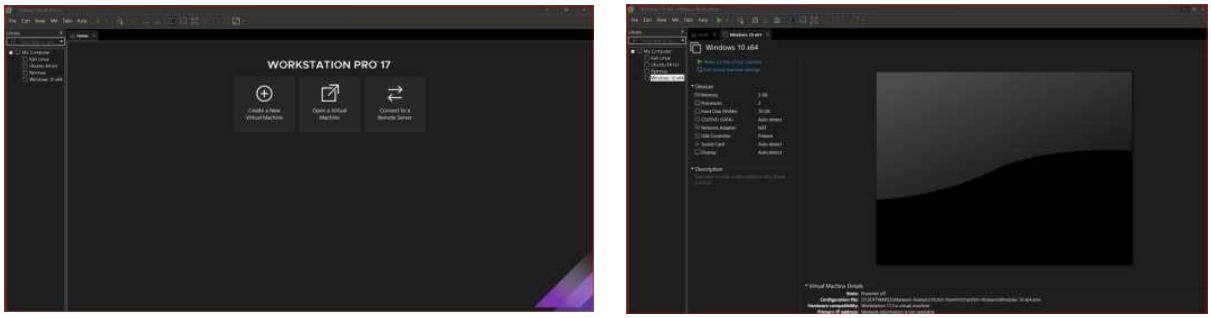

Figure 3,4: Confusion Matrix and ROC curve analysis

E. Installing REMnux Linux Box

Create an unused VM in VMware for REMnux, naming the VM (e.g., "REMnux"), choosing the type (Linux), and selecting the appropriate Linux configuration for REMnux. Customize options such as memory, CPU, and disk allocation according to your requirements.

With the REMnux ISO file mounted in the VM's virtual optical drive, power on the VM and proceed with the installation of Linux. Following the installation, modify Linux settings including network configuration (Bridged or Host-only), updating system packages, installing REMnux tools, and configuring network services (SSH, Samba, etc.) to optimize your REMnux setup.

IV. Results and Discussion

To this end, the framework's casualty machine is FlareVM, with Remnux as the C2 machine which was used alongside INetSim for setting up pantomime. Ensuring any FlareVM and Remnux communication is constant so that the analysis can be complete. Reflecting on the malware records for the sign of compromise (IOCs) including the record headers, the metadata, the API calls, and code structures. In this column, acknowledged potential hazards and noticed on malware viability devoid of implementation.

Observed the actions of malware throughout their execution with the help of such programs as Handle Screen, and analogs of the Any sandbox. Run. Exposed concealed operations and synchronized with the interference produced through the malware on the framework.

Successful monitoring of the organized activity spawned through the use of malware using Wireshark. Several personnel appreciated the communication designs and the command- and-control exercises and comprehending how the malware communicates with outside servers.

He mentions extractedIOCs from inactive and energetic examinations come about, counting record hashes, IP addresses, and space names and arranging activity designs. Set discovery rules and enhanced cybersecurity resistances on the basis of identified IOCs. Developed YARA rules to automate things such as locating and categorizing malware based on specific design features.

Enhanced the methods of danger identification and fixed the issue of passive security measures. Developed a command-line Python tool to automate malware triage with the help of the VirusTotal API.

Simplified the examination handle, capable of fast identification of possibly malicious IPs and immediate counteraction of threats. As a result of that, we ameliorated inactive and active research, organized check as well as established fresh locating rules, which coalesced to enhance our cybersecurity stance.. In the future, the

information that has been gathered from this extent will act as a foundation in advancing the exploratory as well as exceptional defense measures in the transforming environment of cybersecurity. Made FlareVM the casualty machine and Remnux the C2 machine with INetSim for establishing pantomime. To ensure that FlareVM and Remnux are communicating flawlessly for deeper analysis, it was guaranteed.

A. Home-Lab Setup

Key Implications: Then, we created a home lab based onthe volume where FlareVM was used as a victim machine, and Remnux was used as a C2 machine, together with INetSim.
This limited environment let's us perform and investigate malware samples without endangering actual existing systems.
System Outline: For the purpose of the setup, FlareVM was equipped with basic tools for victim mimicry and analysis, Remnux acted as the C2 host, and INetSim for the network interactions. The built up structure was in such a way thatboth the machines could communicate effectively for overall assessment.

B. Static Analysis

Key Implications: In static analysis, the computer files were analyzed without its execution; parts of interest included the file headers and metadata, API calls and code structures. While using this method, we stood a chance of recognizing the indicators of compromise (IOCs) and comprehending the malware behavior and activities before its execution.
System Outline: For the static analysis we used tools such as disas- semblers, hex editors, and information extracting programs. Both tools gave different information on the makeup of the malware in a way that gave me an efficient way of studying the particulars.

C. Dynamic Analysis

Key Implications: Dynamic analysis entailed running the malware and analyze the activity of the malware in the com- promised computer system.
This revealed other undocumented features and made us have an insight of the consequences of the malware.
System Outline: Some of the most useful tools which we employed in our work include Process Monitor (Procmon) as well as the sandboxes including Any. Run for dynamic analysis. These tools let me to investigate malware processes, files, and how they interact with the system which helped me understand its functioning.

D. Network Analysis

Key Implications: Even network analysis revolved around watching network traffic produced by malware, that helped to identify communication and command-and-control activities. It worked to us to know how the malware was communicatingwith other servers and how the malware passes data out.
System Outline: Another tool which was set up in FlareVM was the Remnux which was specifically made to be a DNS server. Wire Shark was used in capturing and analyzing trafficflows on the network and helped the researchers in identifyingmalicious activities of the malware in the network.

E. IOCs (Indicators of Compromise)

Key Implications: Labeling IOCs in the data that we have collected enabled use to prevent and address potential security issues.
All these indicators acted as digital signatures which made it possible for threats to be detected and responses made proactively.
System Outline: IOCs include files hashes, IPs addresses, domain names, registry keys, network traffic patterns extracted from static and dynamic analysis. Subsequently, these indicators were employed in identification rulesand improvement of cybersecurity.

F. Automation with VirusTotal

Key Implications: Creating a command-line Python to to automate analysis in VirusTotal aiding in the triage ofmalware samples as well as making threat intelligence easier
The use of this tool was very helpful in the quick detection of some IP addresses that can be potentially hazardous andtherefore can be blocked in advance.
System Outline: The Python script which was written from the scratch as a malware analysis tool to work in conjunction with the VirusTotal A was used for automatically submitting the collected malware samples and obtain detailed reports from the API syste It helped in handling of suspicious IPs detected during t analysis of network traffic enabling us to make proper dec sions and improve the cybersecurity mechanisms in practice.

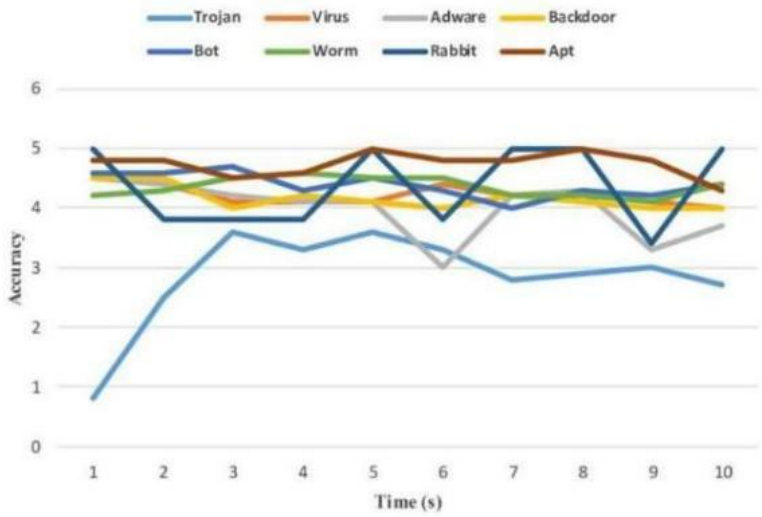

Figure 5: Confusion Matrix and ROC curve analysis

IV. Conclusion

The study has yielded vital insights on the behaviour of malware, primarily through the use of REMnux and FLARE VM. But we ran into a number of built-in constraints withour system configuration. In the future, we hope to makeimprovements that not only tackle these limitations but also advance our skills in the direction of more advanced andreliable malware analysis techniques. During the malware analysis process, we ran across a number of obstacles, including poor virtual lab performance, tool constraints, and troubles doing a thorough network inves- tigation. The depth of our investigation were limited by time, resource, and skill constraints. Our malware analysis home-labsetup's efficacy and efficiency have been impacted by these constraints. Optimizing virtual lab setups is our goal in order to increasescalability and performance. In order to better effectively detect and analyse harmful actions, we intend to improve network monitoring capabilities. We must constantly innovateand develop if we want to improve our entire security posture and remain ahead of the ever-evolving cyber threats. We will describe the paper's shortcomings and po- tential improvements, as well as the unique contributions made by each team member. This thorough review will provide insights into the various efforts and areas of knowledge that went into creating and running our home lab for malware research for rojan analysis.

Figure 6,7: Confusion Matrix and ROC curve analysis

Future Scope

It is critical for cybersecurity experts to keep ahead of trojans destructive strategies as they become more sophisti- cated and stealthy. These sneaky dangers, which are frequentlyused by nation-state actors and cybercriminals, are alwayschanging to avoid detection, penetrate networks, and stealconfidential information. Upgrading the malware analysis toolsand laboratories is essential in response.

- Improved DNS Analysis: Create tools to examine DNS traffic that is captured while analysing malware to improve your DNS analysis skills. To find potential command and control servers or malicious domains, extract and analyse the domain names that the malware accessed, look for trends or anomalies in DNS requests, and correlate this information withother analytical findings.
- Automated YARA Rule Generation: Create a tool that uses the features of examined malware samples to automati- cally create YARA rules. To create YARA rules that can be utilised for future detection, extract pertinent features—suchas strings, API calls, or behavioural patterns—from malware samples.
- Integration with Threat Intelligence Feeds: Include more context about known threats, indicators of compromise (IOCs),and new malware patterns in your analysis results by in- tegrating your analysis environment with threat intelligence feeds. Integrate threat intelligence data with analysis results automatically to prioritise and contextualise security alarms.
- Automated Malware Collection: Create a script that gath- ers malware samples on a regular basis from different websites, online forums, and phoney email addresses. Download these samples automatically to our analysis environment (like Flare VM) so we can analyse into them more.
- Improved Network Analysis: Use more reliable methods to increase the scope of the network analysis skills. As an illustration, you could use Zeek (previously known as Bro) to analyse network traffic, put up a honeypot to attract and examine harmful activity, or use deep packet inspection to get more data.

REFERENCES

[1] Symantec (2012). "The Lifecycle of a Zeus Trojan Target."[1]
[2] Risk Analysis of Trojan-Horse Attacks on Practical Quantum Key Distribution Systems[2]
[3] Construction of knowledge graph of hardware Trojan detection method[3]
[4] Hardware Trojan Attacks: Threat Analysis and Countermeasures[4]
[5] Research on Malware Detection System Using Artificial Intelligence[5]
[6] Development of a customized remote access trojan (RAT) for educational purposes within the field of malware analysis[6]
[7] Malware analysis using reverse engineering and data mining tools[7]
[8] A LITERATURE REVIEW ON MALWARE AND ITS ANALYSIS[8]
[9] Simulating the Network Environment of Sandboxes to Hide Virtual Machine Introspection Pauses[9]
[10] Art of Computer Virus Research and Defense, The, Portable Documents[10]