

**Proactive Computer Security Final Project**

**By:**

Aalaa Mahmoud Zahran 2106136

Aya Mohamed Abdelrahman 20221380245

Ahmed Mohamed Abdelsalam 20201500633

John George 2106143

### **Introduction**

The **Zeus Banking Trojan**, also known as **Zbot**, is a sophisticated and highly effective piece of malware that has been one of the most impactful banking Trojans in cybersecurity history. Initially discovered in **2007**, Zeus was designed to steal sensitive financial information, including banking credentials, by leveraging advanced techniques such as **man-in-the-browser (MITB) attacks**, **keylogging**, and **command-and-control (C2) communications**. Although its original authors abandoned the project, the leaked source code has led to the emergence of numerous variants, making Zeus a significant ongoing threat to cybersecurity.

This project aims to simulate, detect, and analyze the Zeus Trojan using various tools and techniques, providing a comprehensive understanding of its behavior and impact. By leveraging advanced cybersecurity tools like **Suricata**, **Splunk**, **Volatility**, and **YARA**, the project demonstrates how modern defensive mechanisms can be used to detect and mitigate such threats. The analysis combines network monitoring, memory forensics, and signature-based detection, offering a multi-layered approach to malware analysis.

### **Project Goals**

The primary goals of this project are to:

1. Simulate the execution of the Zeus Trojan in a controlled environment to generate real-world indicators of compromise (IOCs).
2. Monitor and analyze the malware’s network activity using Suricata to detect suspicious patterns, including C2 communication.
3. Forward Suricata logs to Splunk to create correlation rules and visualize malicious activity through dashboards.
4. Perform memory analysis using Volatility to uncover active processes, injected code, and network artifacts related to Zeus.
5. Develop and apply custom YARA rules to detect Zeus artifacts in files and memory dumps, enhancing signature-based detection capabilities.

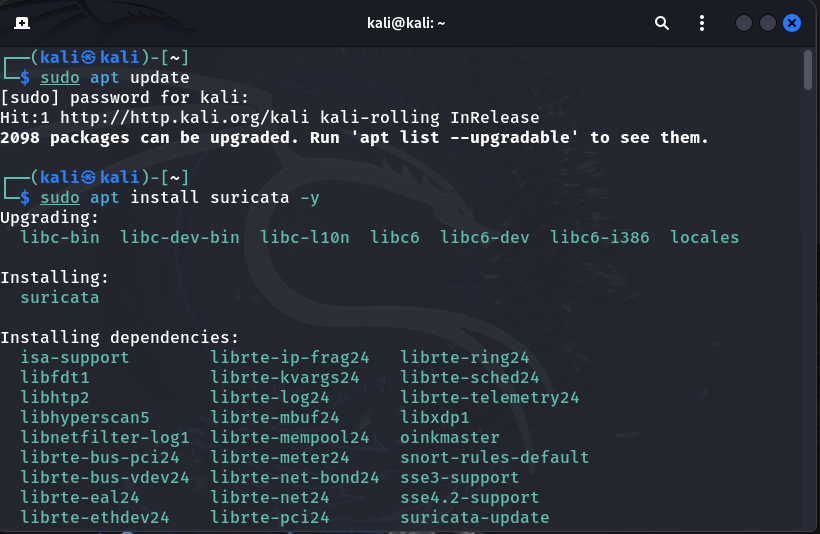
### **Scope of the Project**

This project covers the following key areas of malware analysis and detection:

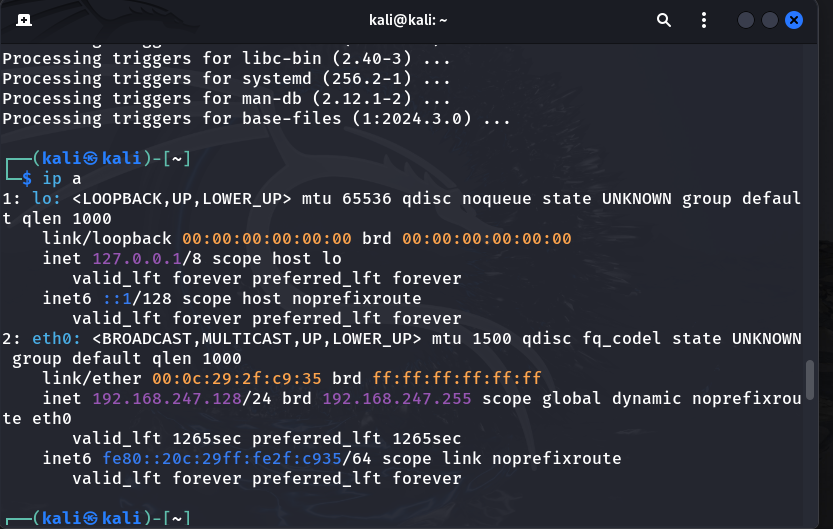
* **Malware Simulation:** Executing the Zeus Trojan in a sandboxed virtual machine (VM) to safely study its behavior and generate artifacts for further analysis.
* **Network Monitoring:** Using Suricata to capture and analyze network traffic generated by Zeus, identifying malicious activities such as C2 communication.
* **Log Correlation and Visualization:** Ingesting and analyzing logs with Splunk to correlate network anomalies with system events, providing a high-level overview of Zeus-related activities.
* **Memory Forensics:** Capturing and analyzing memory dumps using Volatility to detect active processes, injected modules, and hidden network connections.
* **Signature-Based Detection:** Writing and applying YARA rules to identify Zeus-specific patterns in binaries, configurations, and memory artifacts.

1. **Suricata**

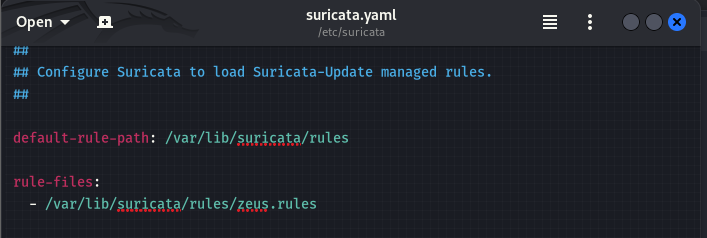
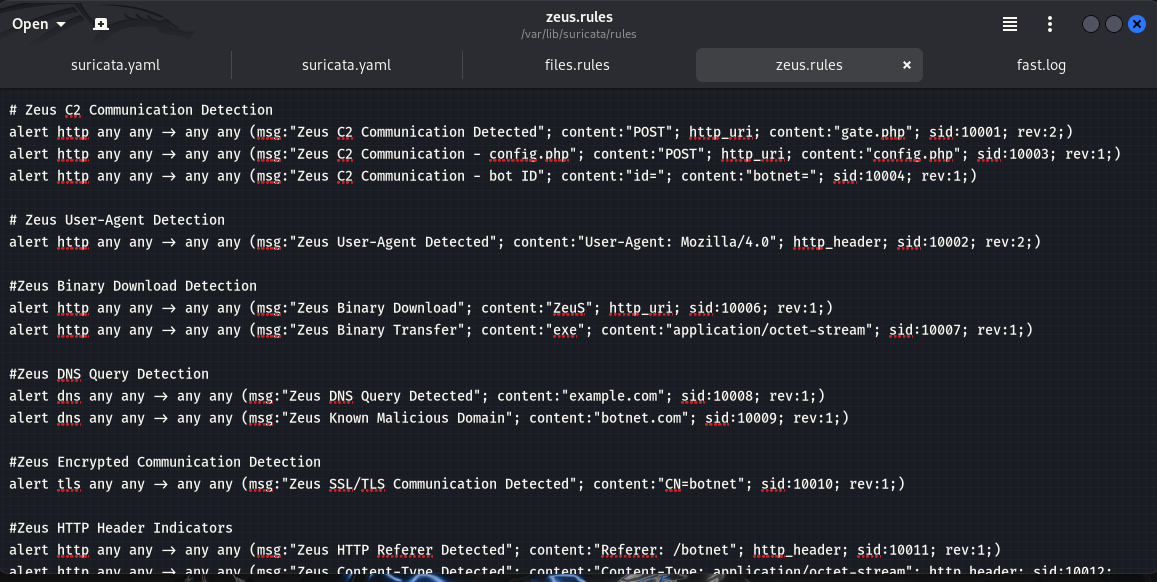
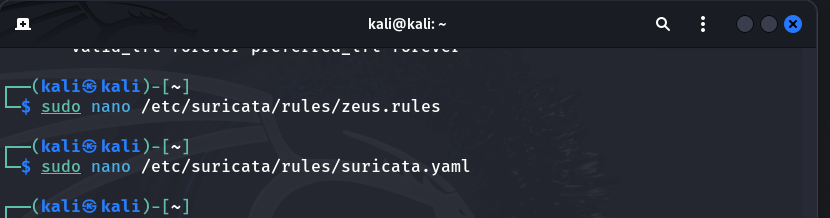
**Suricata** is an open-source tool designed for **network intrusion detection**, **prevention**, and **monitoring**. It provides real-time analysis of network traffic, helping detect malicious activities and threats. Suricata works by analyzing packets and matching them against pre-configured rules or custom rules, making it a valuable tool in identifying threats like the **Zeus Banking Trojan**.In this project, Suricata is used to monitor and analyze the network traffic generated by the Zeus Trojan. By leveraging its ability to detect suspicious patterns and generate alerts, Suricata helps identify malicious behaviors such as **command-and-control (C2) communications**, data exfiltration, or other network anomalies. Additionally, its integration with tools like Splunk enhances visibility and provides a comprehensive analysis of malicious activity.

**In the screenshot, you performed the following steps:**

1. **Updated the Package List**:
   1. Used sudo apt update to refresh the list of available software packages from the Kali Linux repositories.
2. **Installed Suricata**:
   1. Used sudo apt install suricata -y to install Suricata along with its required dependencies.
   2. The -y flag automated the installation process by confirming prompts.

**In this screen, the following actions occurred:**

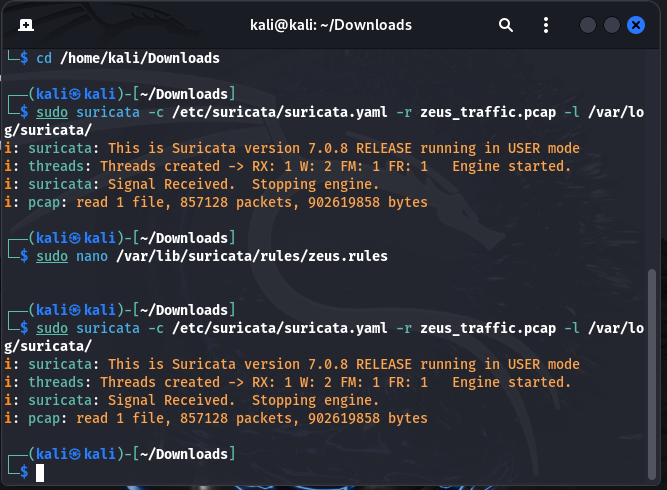
1. **Processed Installation Triggers:**
   1. After installing Suricata, the system completed setup tasks for dependencies (e.g., libc-bin, systemd, man-db). These are required for Suricata to function correctly.
2. **Displayed Network Interface Information:**
   1. The command ip a was executed to display information about the system's network interfaces:
      1. **Interface lo (Loopback):**
         1. Used for internal communication within the system.
         2. IP address: 127.0.0.1.
      2. **Interface eth0:**
         1. The primary network interface connected to the network.
         2. Assigned the IP address 192.168.247.128/24.
         3. Shows the MAC address (00:0c:29:cf:9c:35) and additional details such as broadcast and IPv6 addresses.

**The actions across the three screens were as follows:**

1. **Creating and Editing Zeus Rules**:
   1. Used nano to create and edit the file /etc/suricata/rules/zeus.rules.
   2. Added custom Suricata rules to detect Zeus-related activities, such as:
      1. **C2 Communication** (e.g., specific POST requests).
      2. **User-Agent Strings** linked to Zeus.
      3. **Binary Downloads**, **DNS Queries**, and **Encrypted Communications**.
2. **Configuring Suricata to Use Zeus Rules**:
   1. Edited the suricata.yaml configuration file using nano to include the newly created zeus.rules.
   2. Updated the rule-files section in the YAML file to load the zeus.rules file during Suricata's execution.

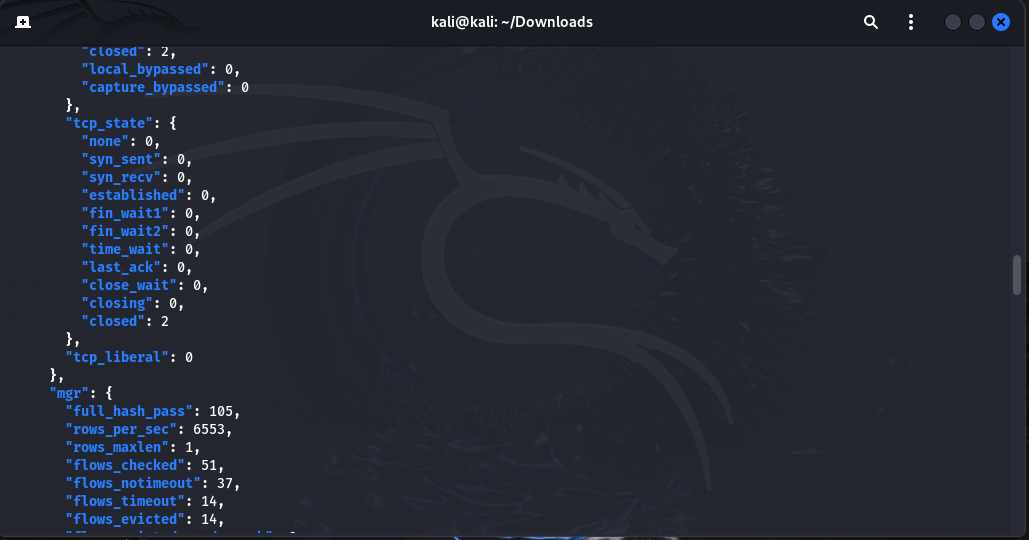
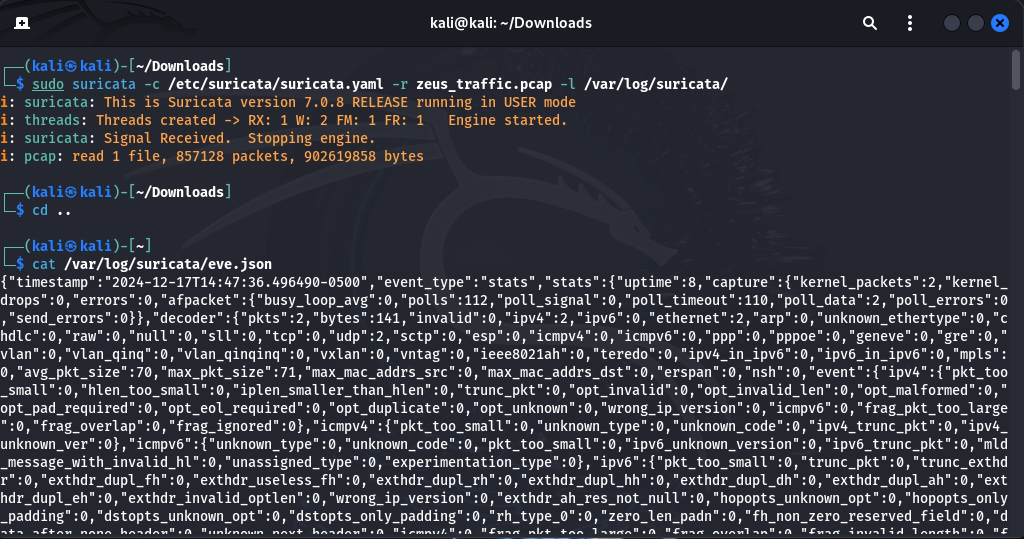
### **Summary of What Happened:**

* **Command Explanation:**
  + -T: Tests the Suricata configuration file without starting the service.
  + -c /etc/suricata/suricata.yaml: Specifies the configuration file (suricata.yaml) to validate.
* **Outcome:**
  + Suricata successfully verified the configuration file and loaded it without errors.
  + The message confirms that Suricata is configured correctly and ready to run using the specified configuration.

**This screen:**

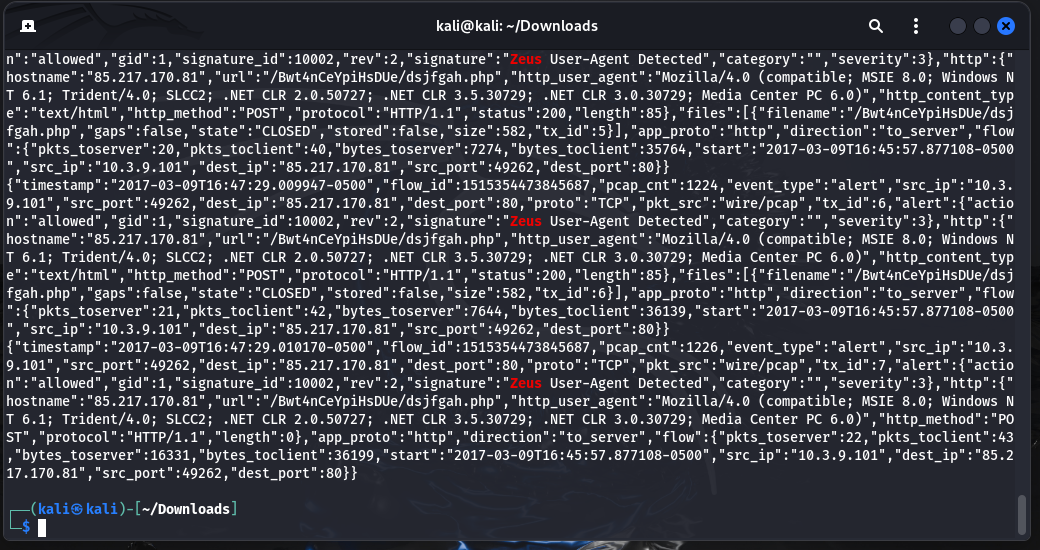
1. **Analyzed PCAP File:**
   1. Ran Suricata to analyze zeus\_traffic.pcap using the specified configuration file and logged results in /var/log/suricata/.
2. **Updated Rules:**
   1. Edited the zeus.rules file to fine-tune Zeus-specific detection.
3. **Reprocessed PCAP:**
   1. Re-ran Suricata to apply updated rules and reanalyze the PCAP file.

This process ensured effective detection of Zeus-related activity based on the latest rule updates.

**This screen:**

**Viewed Logs**:

* 1. Used the cat command to display the contents of eve.json, which contains detailed information about the analyzed traffic, including statistics and any detected events.

**This screen:**

* **Zeus Detected:** Suricata identified Zeus-related activity in the analyzed zeus\_traffic.pcap file.
* **Key Findings:**
  + Detected events include **Zeus User-Agent** patterns, such as HTTP requests containing the User-Agent string "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)".
  + Alerts point to communication between the source IP (10.3.9.101) and destination IP (85.217.170.81) on port 80.
* **Outcome:**
  + Logs confirm the detection of Zeus behavior based on custom Suricata rules.
  + The generated alerts provide evidence of suspicious traffic for further investigation.