OPEN SOURCE THREAT HUNTING AND LOG MANAGEMENT

A PROJECT REPORT

Submitted by

21BCY10123 Harshitaa Ashish
21BCY10128 Neha Lakshmanan
21BCY10193 Praise E Mathew
21BCY10196 Manish Kumar M

in partial fulfilment for the award of the degree
of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING
(Cyber Security and Digital Forensics)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

DECEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project report titled " **OPEN SOURCE THREAT HUNTING AND LOG MANAGEMENT** " is the bonafide work of Harshitaa Ashish (21BCY10123), Praise E Mathew (21BCY10193), Neha Lakshmanan (21BCY10128), Manish Kumar M (21BCY10196) who carried out the Capstone Project – DSN 4099 under my supervision. Certified further that to the best of my knowledge, the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. D. Saravanan, Assistant Professor Sr.,
School of Computing Science and Engineering,
VIT Bhopal University.

PROJECT SUPERVISOR,
Dr. Harihara Sitaraman S,Assistant
Professor Senior Grade 2
School of Computing Science and Engineering,
VIT Bhopal University.

The Capstone Project Examination is held on

ACKNOWLEDGEMENT

I first and foremost like to say a big thank you to the Lord Almighty for His presence and the huge blessings that I received during the work itself. Moreover, I would like to express my most heartfelt gratitude to Dr. D. Saravanan, Lead Programme Chair, Dr Adarsh Patel, Programme Chair Cyber Security and Digital Forensics for providing the tremendous encouragement throughout the course of this work. I would like to thank Dr. Harihara Sitaraman S, my internal guide, who procured valuable suggestions for work completion during the project. I would like to thank all the technical and teaching staff of the School Computing Science and Engineering for their direct or indirect support. What I lacked in many other ways, I made up in loyalty, determination and this is the last, but not least, my sane thanks to my parents that without their support I have worked day and night to bring the project in success.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success

LIST OF ABBREVIATIONS

Abbreviation	**Full Form**
ELK	Elasticsearch, Logstash, Kibana
OWASP ZAP	Open Web Application Security Project Zed Attack Proxy
IoCs	Indicators of Compromise
PCAP	Packet capture
SIEM	Security Information and Event Management
SOC	Security Operations Center
IDS	Intrusion detection system
IPS	Intrusion prevention system
IR	Incident response
CVE	Common vulnerabilities and exposures
CWE	Common weakness enumeration
API	Application programming Interface
OSINT	Open source intelligence
JSON	Javascript object notation
DNS	Domain name system
CIR	Cyber incident response
NVT	Network vulnerability test
CVSS	Common vulnerability scoring system
FOSS	Free and open source software

LIST OF FIGURES AND GRAPHS

LIST OF TABLES

ABSTRACT

Open Source Threat Hunting and Log Management, focuses on leveraging open source tools to enhance network security monitoring and threat detection. The purpose of the project is to develop a comprehensive framework for analyzing network activity, identifying potential vulnerabilities, and addressing security threats in a systematic manner.

The methodology incorporates a combination of tools such as Suricata, Zeek, and the ELK stack for traffic capture, analysis, and visualization.OWASP ZAP was utilized for vulnerability scanning of web applications, providing actionable insights into security flaws. Each tool's output was analyzed to uncover indicators of compromise , assess security gaps, and recommend mitigation strategies.

The findings reveal that open-source tools provide a robust and cost-effective solution for network threat hunting. The integrated use of these tools facilitates the identification of malicious activities, such as malware communication patterns, and enhances the overall security posture of networks and applications.

This work demonstrates the applicability of open-source frameworks in real-world threat management scenarios, offering a scalable and adaptable solution for organizations.

TABLE OF CONTENTS

Chapter 1: Project Description and Outline

1.1 Introduction

The project titled **"Open-source Threat Hunting and Log Management"** focuses on utilizing open-source tools for effective threat detection and network analysis. The project aims to integrate and leverage tools such as Suricata, Zeek, ELK Stack, OWASP ZAP, and OpenVAS to address cybersecurity challenges. These tools facilitate capturing, analyzing, and mitigating networkthreats through robust log management and detailed threat analysis. This approach emphasizes cost effectiveness and flexibility, ensuring that organizations can enhance their security posture without relying solely on proprietary solutions.

1.2 Motivation for the Work

The rise of sophisticated cyberattacks necessitates the development of advanced threat detection and management systems. Organizations often struggle with budget constraints, making open-source solutions an attractive alternative. This project is motivated by the need to explore the capabilities of open-source tools in threat hunting, providing a scalable, efficient, and affordable way to secure digital infrastructures. Moreover, this project aims to contribute to the growing community of cybersecurity practitioners by showcasing the power of collaboration and open innovation in solving modern cybersecurity challenges.

1.3 About the Project

This project integrates multiple open-source tools to create a comprehensive threat-hunting and log management system. The techniques include:

· **Suricata** for intrusion detection and real-time network traffic analysis.

· **Zeek** for extracting network logs and analyzing traffic patterns.

· **ELK Stack** for centralized log management and visualization of network data. ·
OWASP ZAP for web application vulnerability assessment.

· **OpenVAS** for conducting in-depth vulnerability scans of servers and applications.

By combining these tools, the project achieves a holistic view of network security, identifying Indicators of Compromise (IoCs), anomalies, and vulnerabilities to proactively address potential threats.

1.4 Problem Statement

Cybersecurity threats are evolving in complexity, yet many organizations lack the resources to invest in expensive proprietary solutions. There is a pressing need for a cost-effective, reliable, and scalable threat-hunting system that can detect, analyze, and mitigate network threats using open-source tools.

This project addresses this gap by creating an integrated platform capable of handling log management and providing actionable insights.

1.5 Objective of the Work

The objectives of this project are:

1. To leverage open-source tools for efficient threat detection and network analysis.
2. To demonstrate the capabilities of these tools in identifying vulnerabilities, anomalies, and potential attacks.
3. To develop a structured workflow for analyzing network traffic and security logs.
4. To provide insights into the effectiveness of open-source solutions in securing digital infrastructures.

1.6 Organization of the Project

Chapter 1: Project Description and Outline – Provides an overview of the project, including its motivation, problem statement, and objectives.

Chapter 2: Related Work Investigation – Reviews the related literature, methodologies, and tools that inform the project's direction.

Chapter 3: Requirement Artifacts – Discusses the functional and non-functional requirements essential for the project's success.

Chapter 4: Design Methodology and Its Novelty – Explains the design approach and highlights the innovative aspects of the project.

Chapter 5: Technical Implementations and Analysis – Details the technical implementation of the project, along with a thorough analysis of the results.

Chapter 6: Project Outcome and Applicability – Describes the outcomes of the project and its potential applications in real-world scenarios.

Chapter 7: Conclusions and Recommendations – Summarizes the findings of the project and provides recommendations for future work.

1.7 Summary

This chapter introduced the project, its motivation, techniques employed, problem statement, objectives, and organizational structure. The subsequent chapters delve deeper into the tools, methodologies, and outcomes of the project, showcasing the potential of open-source solutions in addressing cybersecurity challenges.

Chapter 2: Related Work Investigation

2.1 Introduction

This chapter explores the related work and research that form the foundation of the project, focusing on the tools and techniques used in open-source threat hunting and log management. It reviews core areas, existing methods, their pros and cons, and issues observed in current practices, providing the context necessary for the project's development and implementation.

2.2 Core Area of the Project

The core area of this project is **network threat detection, analysis, and log management using open-source tools**. It leverages tools like Suricata, Zeek, ELK Stack, OWASP ZAP and OpenVAS to address real-world cybersecurity challenges. These tools support tasks such as intrusion detection, network monitoring, vulnerability scanning, and centralized log management, collectively enhancing an organization's ability to respond to cyber threats effectively.

2.3 Existing Approaches/Methods

2.3.1 Approach/Method 1: Intrusion Detection with Suricata

Suricata is an open-source network intrusion detection and prevention system (IDS/IPS). It inspects network traffic in real-time, identifying malicious patterns based on predefined rules and behavior analysis. Suricata is widely used due to its scalability and ability to handle high-speed networks.

- **Example:** Using Suricata to detect malware and generate logs containing Indicators of Compromise (IoCs).

2.3.2 Approach/Method 2: Network Traffic Analysis with Zeek

Zeek is a powerful network security monitoring tool that transforms raw network traffic into detailed logs for further analysis. It provides insights into protocols, connections, and potential threats through its extensible scripting capabilities.

- **Example:** Using Zeek to analyze PCAP files, extract logs, and identify anomalies in network behavior.

2.3.3 Approach/Method 3: Vulnerability Scanning with OpenVAS

OpenVAS is an open-source vulnerability assessment system designed to identify weaknesses in servers, network devices, and applications. It generates detailed reports with severity ratings and recommendations for mitigation.

- **Example:** Scanning servers and applications to detect vulnerabilities like outdated software and misconfigurations.

2.4 Pros and Cons of the Stated Approaches/Methods

Approach	**Pros**	**Cons**
Suricata	- Real-time detection - High-speed network support - Open-source flexibility	- Rule management complexity - Limited scope for encrypted traffic
Zeek	- Detailed and structured network logs - Highly extensible scripting capabilities	- Requires expertise for configuration - Limited real-time capabilities
OpenVAS	- Comprehensive vulnerability detection	- High resource consumption during scans

	- Severity-based reports	- Potential false positives
ELK Stack	- Centralized logging and rich visualizations - Scalable and customizable	- Complex setup and resource-intensive - Requires continuous maintenance and monitoring
OWASP ZAP	-Comprehensive web application security testing - Open-source and extensible	- High false positive rate - Resource-intensive and requires configuration
RITA	- Automated network traffic analysis - Works well with Zeek logs	- Limited to network traffic data - Requires specialized knowledge for effective use
The Hive	- Incident response management with customizable workflows - Scalable and open-source	- Requires setup and integration with other tools - Resource-intensive and steep learning curve
Cortex	- Automated security data analysis - Integrates well with other security tools	- Can be resource-heavy - Requires integration with other tools for full functionality

2.5 Issues/Observations from Investigation

2.5.1. **Integration Complexity:** Combining multiple tools like Suricata, Zeek, and ELK requires seamless integration to avoid inefficiencies.

2.5.2. **Scalability Challenges:** Tools like OpenVAS may struggle to scale in large networks with numerous endpoints.

2.5.3. **Expertise Requirement:** Proper utilization of tools like Zeek and Suricata necessitates advanced skills and knowledge.

2.5.4. **Data Overload:** Generating and analyzing logs from multiple sources can result in overwhelming data, making it difficult to extract actionable insights.

2.5.5. **False Positives:** Tools like OpenVAS occasionally generate false positives, which can lead to unnecessary mitigation efforts.

2.6 Summary

This chapter reviewed the tools and methodologies relevant to the project. The pros and cons of each method, along with key observations, highlight the importance of integrating these tools effectively to address cybersecurity challenges. The insights gained from this investigation will guide the implementation and optimization of the project's threat hunting and log management framework.

Chapter 3: REQUIREMENT ARTIFACTS

3.1 Introduction

This project aims to build a network threat detection, analysis, and log management system using a suite of open-source cybersecurity tools. The tools integrated into this system are Suricata, Zeek,ELK Stack,The Hive,Cortex OWASP ZAP, Rita, and OpenVAS. These tools will collectively help monitor and analyze network traffic, identify security vulnerabilities, and provide insights into potential intrusions or threats within an organization's network. The project will enhance an organization's ability to detect, analyze, and respond to cyber threats by centralizing data collection, threat analysis, and reporting. This approach leverages the power of real-time data processing and visualization to offer a comprehensive cybersecurity solution.

3.2 Hardware and Software requirements

3.2.1 Hardware Requirements:

3.2.1.1 Processor (CPU):

· Minimum: Quad-core processor (Intel i5 or equivalent).

· Recommended: Octa-core processor (Intel i7 or equivalent) for high-speed processing and better scalability.

3.2.1.2. Memory (RAM):

· Minimum: 8 GB of RAM for handling typical log data.

· Recommended: 16 GB or more, especially when dealing with larger datasets or multiple tools running simultaneously.

3.2.1.3. Storage (Hard Disk):

· Minimum: 500 GB SSD for fast data access and storage.

· Recommended: 1 TB or more for storing large datasets generated by network monitoring, intrusion detection, and vulnerability scans.

3.2.1.4. Network Interface:

· Minimum: Gigabit Ethernet (1 Gbps) for efficient data transfer.

3.2.1.5 Backup and Redundancy:

· Use cloud-based or external storage solutions for redundancy and data backup.

3.2.2 Software Requirements

3.2.2.1. **Operating System**:

· **Linux-based OS** (Ubuntu, CentOS, or Debian recommended) for compatibility with most of the open-source tools used.

· **Windows** can be used in conjunction with Docker or virtual machines for specific tools that may require Linux environments.

3.2.2.2. **Required Tools**:

· Suricata (v6.x or higher) for intrusion detection and prevention.

· Zeek (v4.x or higher) for network monitoring and traffic analysis.

· OWASP ZAP (v2.x or higher) for web application security testing.

· Rita (v1.x or higher) for analyzing network traffic and detecting anomalies. · OpenVAS (v21.x or higher) for vulnerability scanning.

3.2.2.3. **Additional Software**:

· Docker for containerization of tools.

· Python 3.x for scripting and automating tasks.

· Java Runtime Environment (JRE) for tools like OWASP ZAP(ifused).

3.3 Specific Project requirements

3.3.1 Data requirement

3.3.1.1. **Data Sources**:

a. **Suricata and Zeek Logs**: Data from network traffic, intrusion detection events, and network monitoring logs.

b. **OpenVAS Scan Data**: Results of vulnerability scans on network devices and infrastructure.

c. **OWASP ZAP Data**: Results from web application security scans, including vulnerabilities and attack surface data.

d. **Rita Logs**: Network traffic data used for detecting anomalies and identifying suspicious activities.

3.3.1.2. **Data Format**:

· Structured formats like **JSON**, **CSV**, or **XML** are preferred for logs and reports generated by the tools.

3.3.1.3. **Data Volume**:

· Depending on network size and scan frequency, data volume could range from a few gigabytes to several terabytes.

3.3.1.4. **Data Storage**:

- Tools will generate log files that should be stored locally or in cloud-based storage for easy access and retrieval.

3.3.2 Functions requirement

The system should provide the following functionalities:

3.3.2.1. Network Traffic Monitoring:

a. Suricata and Zeek will monitor network traffic for signs of malicious activity, such as potential intrusions and threats.

3.3.2.2. Intrusion Detection and Prevention:

a. **Suricata** will detect and prevent network-based attacks, generating alerts for detected malicious traffic.

3.3.2.3. Vulnerability Scanning:

a. **OpenVAS** will scan network infrastructure for vulnerabilities, providing detailed reports with potential security gaps.

3.3.2.4. Web Application Security Testing:

a. OWASP ZAP will perform security scans on web applications, identifying common security flaws like SQL injection, cross-site scripting (XSS), etc.

3.3.2.5. Anomaly Detection:

a. **Rita** will help detect anomalies in network traffic, identifying suspicious or unexpected behavior that may indicate a cyber attack.

3.3.2.6. Log Management and Analysis:

a. The collected logs will be stored and managed centrally for easy retrieval and analysis. The system will provide methods for searching and analyzing logs manually or through automated scripts.

3.3.2.7. Reporting:

a. The system will generate reports based on data from **Suricata**, **Zeek**, **OWASP ZAP**, **Rita**, and **OpenVAS** to summarize detected threats, vulnerabilities, and anomalies.

3.3.3 Performance and security requirement

· **Performance**:

The system should be able to process and analyze large volumes of network data in real-time or near-real-time.

Suricata and Zeek should operate efficiently even in high-throughput environments with large amounts of network traffic.

· **Security**:

Secure communication protocols such as **TLS/SSL** should be used for transferring data between tools.

Logs and reports should be stored securely, with access control mechanisms in place to restrict unauthorized access.

Tools should be regularly updated to protect against new vulnerabilities and exploits.

3.3.4 Look and Feel Requirements

· **User Interface**:

The system should have an easy-to-navigate interface for managing the different tools and viewing logs and reports.

The interface should allow users to configure settings for each tool and trigger scans or analyses as needed.

· **Visualization**:

Dashboards should be designed to display information clearly, focusing on network traffic analysis, detected intrusions, and vulnerabilities.

The interface should provide real-time visualizations for threat monitoring and alerting.

·

Responsiveness:

The system interface should be responsive and able to adapt to different screen sizes for ease of use on desktops, tablets, and mobile devices.

3.4 Summary

This project integrates Suricata, Zeek, OWASP ZAP, Rita, and OpenVAS into a centralized system for network threat detection, anomaly analysis, vulnerability scanning, and log management. The system is designed to monitor network traffic, detect potential intrusions, perform security assessments on applications, and provide real-time visibility into network health. The project provides organizations with an open-source solution to enhance their cybersecurity efforts, offering tools for identifying, analyzing, and responding to security threats.

CHAPTER 4: DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal

The project adopts a hybrid approach combining network-based intrusion detection, threat analysis, and vulnerability assessment methodologies.

Goal: To create an integrated security solution that leverages open-source tools to identify, analyze, and mitigate cybersecurity threats effectively.

We have initially started with tools like Zeek, Suricata, ELK, TheHive, and Cortex were implemented to enable real-time network monitoring, threat detection, and centralized incident management and followed by tools like advanced tools like RITA (for traffic analysis and anomaly detection), ZAP Proxy (for web application vulnerability scanning), and OpenVAS (for network vulnerability scanning) were incorporated.

4.2 Functional modules design and analysis

The project is divided into functional modules to streamline implementation:

· <u>Network Monitoring and Traffic Analysis:</u>

Tools: Zeek and RITA

Functionality: To analyze network traffic for suspicious patterns, anomalies, and malicious activity.

· <u>Intrusion Detection System (IDS):</u>

Tools: Suricata

Functionality: Perform signature-based and anomaly-based detection to identify threats in real-time.

· <u>Threat Hunting and Incident Response:</u>

Tools: The Hive, Cortex, and ELK

Functionality: Facilitate centralized logging, automated threat enrichment, and case management for incident handling

· Vulnerability Assessment:

Tools: ZAP Proxy and OpenVAS

Functionality: Conduct automated scans for network and web application vulnerabilities

4.3. Software Architectural designs

- The system architecture is based on a modular design with the following key components:
    - Data Collection Layer: Zeek and Suricata collect network data and alerts.
- Data Processing Layer:
    - RITA analyses Zeek logs for detecting command-and-control traffic.
    - ELK performs centralized log aggregation and visualization.
- Application Layer:
    - The Hive and Cortex manages the incident response workflows.
    - ZAP Proxy and OpenVAS scan for vulnerabilities.
- Integration Layer: APIs and connectors enable seamless data flow between modules.

4.4 Subsystem services

Subsystems and their services include:

Monitoring Subsystem: Continuous traffic monitoring and alert generation (Zeek, Suricata). Threat Detection Subsystem: Anomaly detection and malicious traffic identification (RITA). Incident Management Subsystem: Case creation, enrichment, and collaboration (The Hive, Cortex). Vulnerability Assessment Subsystem: Identifying exploitable vulnerabilities in applications and networks (ZAP Proxy, OpenVAS).

4.5 User Interface designs

- The Hive Interface: Simplified case management with a dashboard view for tracking incidents.

· ELK Dashboard: Intuitive visualization of network logs and alerts.

· ZAP Proxy and OpenVAS Interfaces: Web-based UIs to configure and view scan results effectively.

· Integration Design: Unified interaction across subsystems through APIs and dashboards.

4.6 Summary

The design methodology emphasizes an integrated approach to cybersecurity by leveraging open source tools to create a comprehensive and robust threat detection, analysis, and mitigation system. The novelty lies in combining these tools cohesively to enhance automation, scalability, and usability.

CHAPTER-5: TECHNICAL IMPLEMENTATION & ANALYSIS

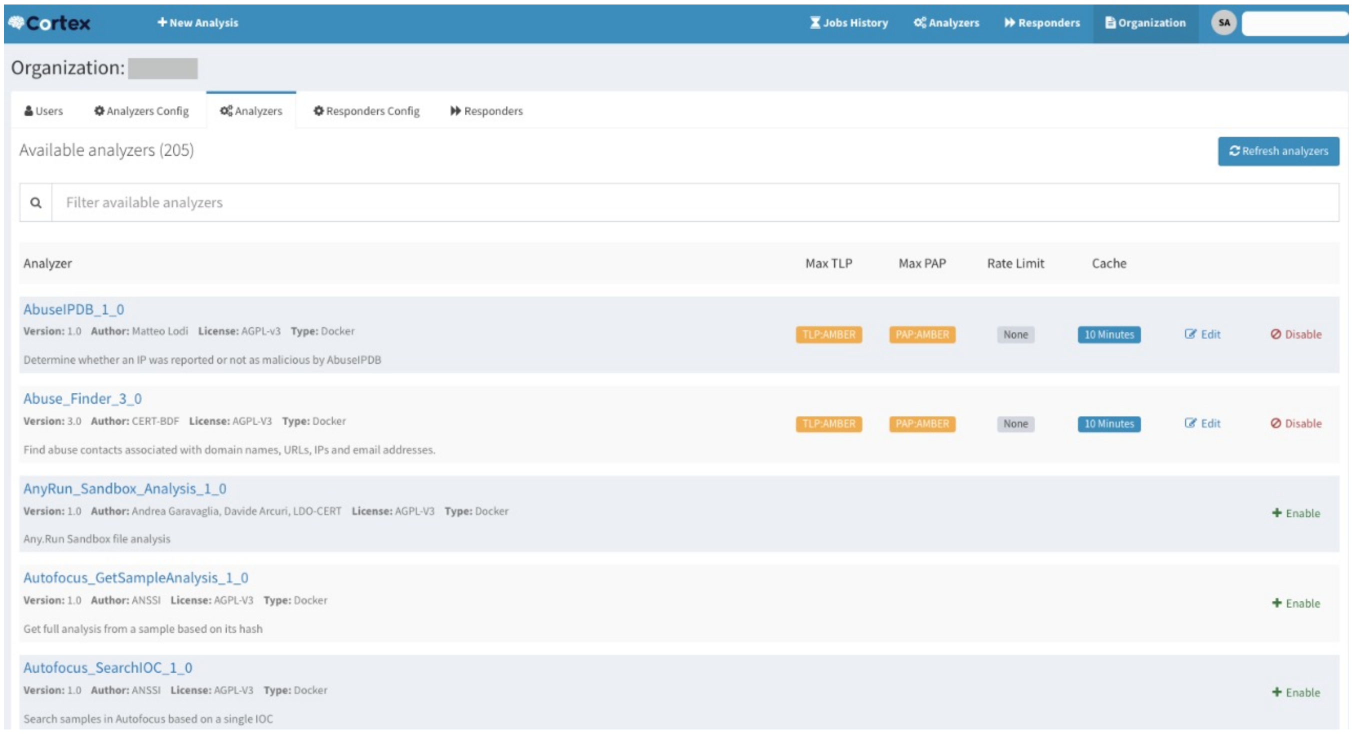
5.1 Outline

The Hive, Cortex, and OpenVAS were used for incident response and vulnerability management. The Hive and Cortex functioned flawlessly for case management and automated threat analysis, and each tool was essential in assessing threats and locating weaknesses.

JSON and CSV log files with threat and security issue data were imported for analysis in the Hive, which was used for effective case management. Based on the threats found, severity levels were assigned to newly created and monitored cases. By keeping an eye on tasks and observations, the platform made it possible for centralised incident management, which expedited the handling of security incidents.

Cortex

By automating the threat analysis process, Cortex was a vital addition to The Hive. It made it possible to choose particular analysers designed for various observable data kinds, like domains, IP addresses, and URLs. For effective threat assessment, this flexibility made sure that the appropriate analyser was used with the right kind of data. Based on its analysis, Cortex produced comprehensive reports that classified test outcomes as In Progress, Failure, or Success in order to offer useful insights. Its smooth interface with The Hive expedited the automation process, greatly cutting down on incident response times and improving workflow in general.



Organization: [redacted]

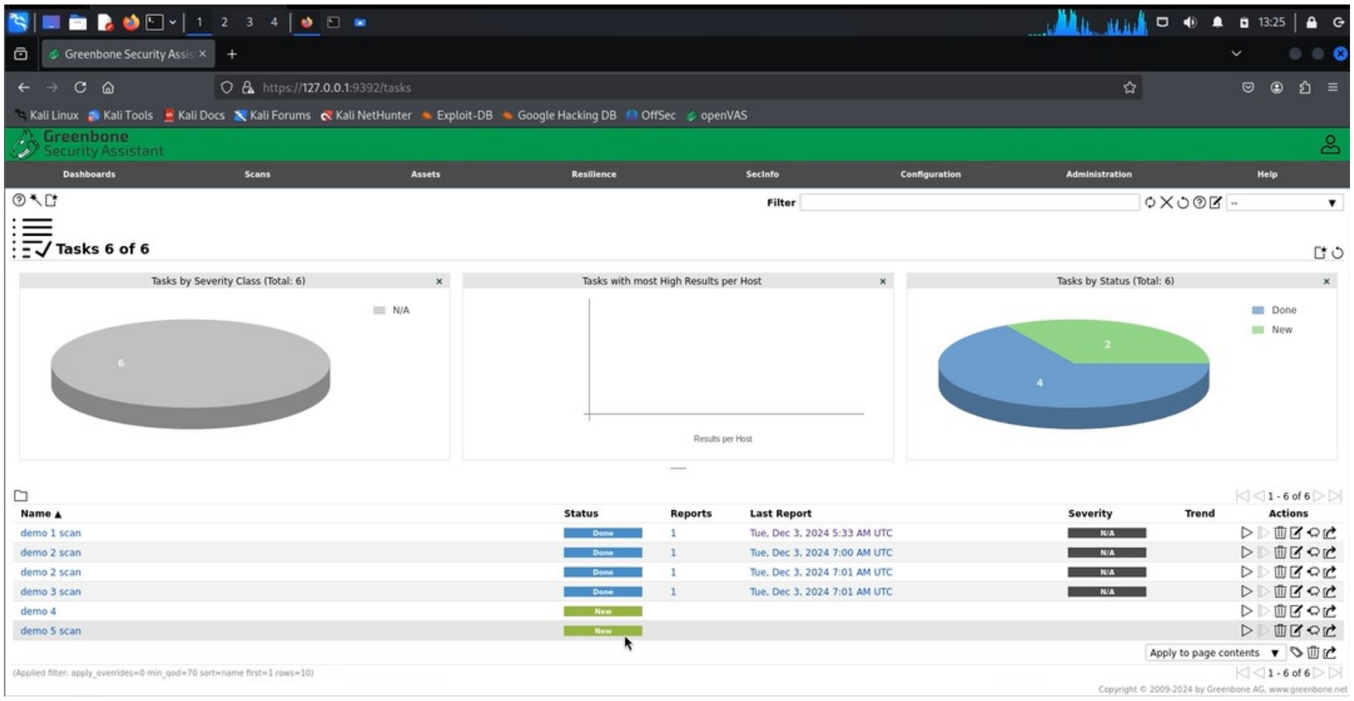
Available analyzers (205)

Analyzer	Max TLP	Max PAP	Rate Limit	Cache		
AbuseIPDB_1_0 **Version:** 1.0 **Author:** Matteo Lodi **License:** AGPL-v3 **Type:** Docker Determine whether an IP was reported or not as malicious by AbuseIPDB	TLP:AMBER	PAP:AMBER	None	10 Minutes	Edit	Disable
Abuse_Finder_3_0 **Version:** 3.0 **Author:** CERT-BDF **License:** AGPL-V3 **Type:** Docker Find abuse contacts associated with domain names, URLs, IPs and email addresses.	TLP:AMBER	PAP:AMBER	None	10 Minutes	Edit	Disable
AnyRun_Sandbox_Analysis_1_0 **Version:** 1.0 **Author:** Andrea Garavaglia, Davide Arcuri, LDO-CERT **License:** AGPL-V3 **Type:** Docker Any.Run Sandbox file analysis						Enable
Autofocus_GetSampleAnalysis_1_0 **Version:** 1.0 **Author:** ANSSI **License:** AGPL-V3 **Type:** Docker Get full analysis from a sample based on its hash						Enable
Autofocus_SearchIOC_1_0 **Version:** 1.0 **Author:** ANSSI **License:** AGPL-V3 **Type:** Docker Search samples in Autofocus based on a single IOC						Enable

Fig 1:Cortex

OpenVAS

Conducting thorough vulnerability assessments against IP addresses, network devices, and applications—including the Damn Vulnerable Web Application (DVWA)—was made possible in large part by OpenVAS. By effectively identifying possible security flaws and categorising them into High, Medium, and Low severity levels, it made it possible to prioritise mitigation efforts. The technology produced thorough reports that gave teams a comprehensive picture of the vulnerabilities, enabling them to take well-informed action to reduce risks and improve network security in general.


Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
Tasks 6 of 6
Tasks by Severity Class (Total: 6)
N/A
6
Tasks with most High Results per Host
Results per Host
Tasks by Status (Total: 6)
Done
New
2
4

Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
demo 1 scan	Done	1	Tue, Dec 3, 2024 5:33 AM UTC	N/A		
demo 2 scan	Done	1	Tue, Dec 3, 2024 7:00 AM UTC	N/A		
demo 2 scan	Done	1	Tue, Dec 3, 2024 7:01 AM UTC	N/A		
demo 3 scan	Done	1	Tue, Dec 3, 2024 7:01 AM UTC	N/A		
demo 4	New					
demo 5 scan	New					

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

1 - 6 of 6

Copyright © 2009-2024 by Greenbone AG, www.greenbone.net

Fig 2:OpenVas

Suricata

Suricata was used to examine the Dridex PCAP file, offering important information about malicious activity. It produced thorough logs, like fast.log, which were crucial in locating malicious payload signatures and revealing Dridex malware communication patterns. Suricata played a crucial role in determining the behaviour and extent of the malware by confirming the existence of the Dridex infection through the detection of key Indicators of Compromise (IoCs).

Zeek

The Specula PCAP file was processed using Zeek, which resulted in thorough logs that provided insightful information about network activity. Conn.log, which recorded network connection information, and dns.log, which recorded DNS queries and answers, were important logs. Furthermore, important events and warnings were highlighted by notice.log and reporter.log, which made it possible to comprehend possible network threats and anomalies more clearly.

RITA

The logs produced by Zeek were further examined by RITA (Real Intelligence Threat Analytics), which offered a more thorough understanding of network activity. In order to spot irregularities, identify possible dangers, and find patterns in the data's behaviour, it built a structured database.

RITA improved the capacity to react to network threats by providing actionable insights into suspicious activity through the visualisation of traffic patterns.

ELK Stack

The logs produced by Suricata were processed using the ELK Stack, which consists of Elasticsearch, Logstash, and Kibana. While Logstash parsed the data into an organised and manageable format, Elasticsearch indexed and stored the logs, guaranteeing effective retrieval. The data was then displayed using Kibana as graphs and charts, which made it simpler to spot trends and irregularities in the network activity.

OWASP ZAP

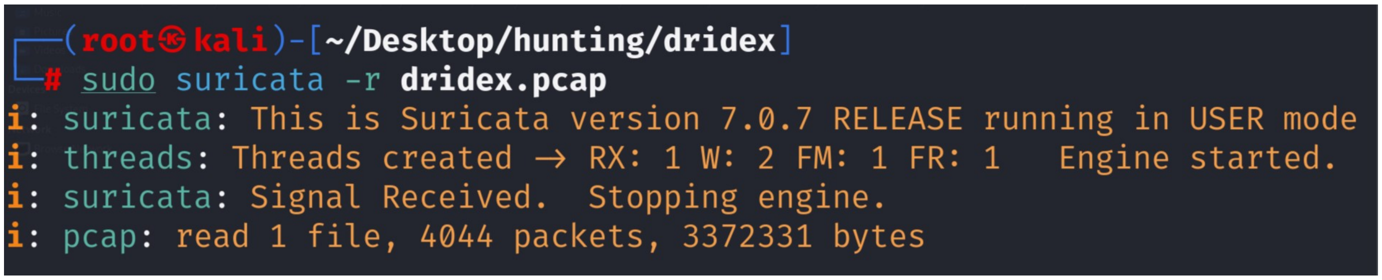
The Mutillidae application was scanned for vulnerabilities using OWASP ZAP. Firefox's FoxyProxy was set up to intercept and examine web traffic. The application was crawled and spidered by the tool to find hidden files and URLs. Clear instructions for remediation efforts were provided by the resulting reports, which described vulnerabilities like Cross-Site Scripting (XSS) and were ranked by severity.

Zeek Online Tool

Through the analysis of pre-captured PCAP files, Zeek's web tool proved its adaptability. The network data was parsed into comprehensive logs, such as conn.log, which revealed information about source and destination activity. This feature demonstrated Zeek's usefulness for both online and offline analysis, making it an important instrument for in-depth network traffic research.

5.2 Implementation:

Suricata :

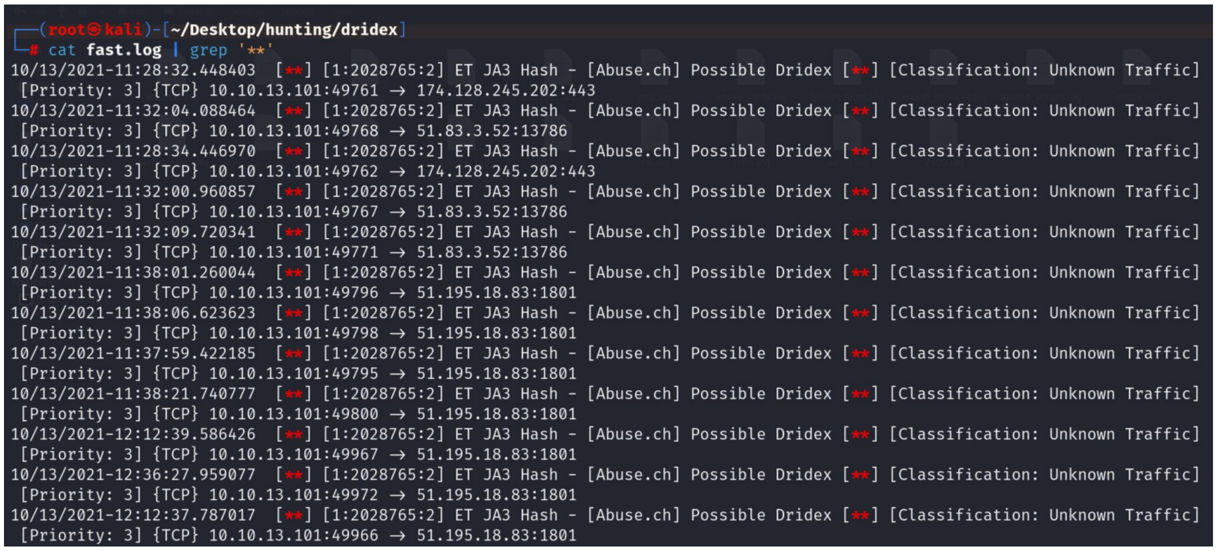


```
┌──(root㉿kali)-[~/Desktop/hunting/dridex]
└─# sudo suricata -r dridex.pcap
i: suricata: This is Suricata version 7.0.7 RELEASE running in USER mode
i: threads: Threads created → RX: 1 W: 2 FM: 1 FR: 1   Engine started.
i: suricata: Signal Received.  Stopping engine.
i: pcap: read 1 file, 4044 packets, 3372331 bytes
```

Fig 3:Suricata Implementation

Explanation:

The picture shows the sudo suricata -r dridex.pcap command being executed, in which Suricata analyzes a PCAP file that contains suspected Dridex malware communication. The report demonstrates that Suricata version 7.0.7 was operating in USER mode, with the engine initialized and threads successfully established. It shows that a single PCAP file with 4,044 packets and 3,372,331 bytes was read and processed, setting the stage for additional traffic analysis.

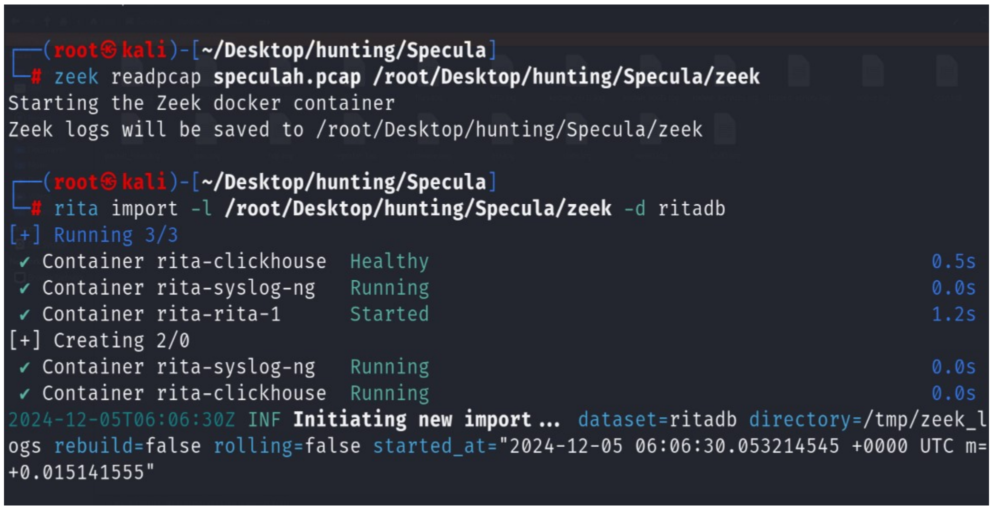


```
┌──(root㉿kali)-[~/Desktop/hunting/dridex]
└─# cat fast.log | grep '**'
10/13/2021-11:28:32.448403  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49761 → 174.128.245.202:443
10/13/2021-11:32:04.088464  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49768 → 51.83.3.52:13786
10/13/2021-11:28:34.446970  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49762 → 174.128.245.202:443
10/13/2021-11:32:00.960857  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49767 → 51.83.3.52:13786
10/13/2021-11:32:09.720341  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49771 → 51.83.3.52:13786
10/13/2021-11:38:01.260044  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49796 → 51.195.18.83:1801
10/13/2021-11:38:06.623623  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49798 → 51.195.18.83:1801
10/13/2021-11:37:59.422185  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49795 → 51.195.18.83:1801
10/13/2021-11:38:21.740777  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49800 → 51.195.18.83:1801
10/13/2021-12:12:39.586426  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49967 → 51.195.18.83:1801
10/13/2021-12:36:27.959077  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49972 → 51.195.18.83:1801
10/13/2021-12:12:37.787017  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49966 → 51.195.18.83:1801
```

Fig 4: **Fast.log**

Explanation

The results of filtering Suricata's fast.log file for noteworthy entries using the cat fast.log | grep "" command are shown in the second image. Several connections are marked in the logs as "[Abuse.ch] Possible Dridex," according to matches with known JA3 hashes and Dridex-related signatures. Source and destination IP addresses, ports, and classifications (such as "Unknown Traffic") are among the comprehensive metadata included in each entry. These logs serve as a foundation for a more thorough examination of the network traffic by highlighting particular instances of possibly harmful activity.



```
┌──(root㉿kali)-[~/Desktop/hunting/Specula]
└─# zeek readpcap speculah.pcap /root/Desktop/hunting/Specula/zeek
Starting the Zeek docker container
Zeek logs will be saved to /root/Desktop/hunting/Specula/zeek

┌──(root㉿kali)-[~/Desktop/hunting/Specula]
└─# rita import -l /root/Desktop/hunting/Specula/zeek -d ritadb
[+] Running 3/3
 ✔ Container rita-clickhouse  Healthy                                          0.5s
 ✔ Container rita-syslog-ng   Running                                          0.0s
 ✔ Container rita-rita-1      Started                                          1.2s
[+] Creating 2/0
 ✔ Container rita-syslog-ng   Running                                          0.0s
 ✔ Container rita-clickhouse  Running                                          0.0s
2024-12-05T06:06:30Z INF Initiating new import ... dataset=ritadb directory=/tmp/zeek_l
ogs rebuild=false rolling=false started_at="2024-12-05 06:06:30.053214545 +0000 UTC m=
+0.015141555"
```

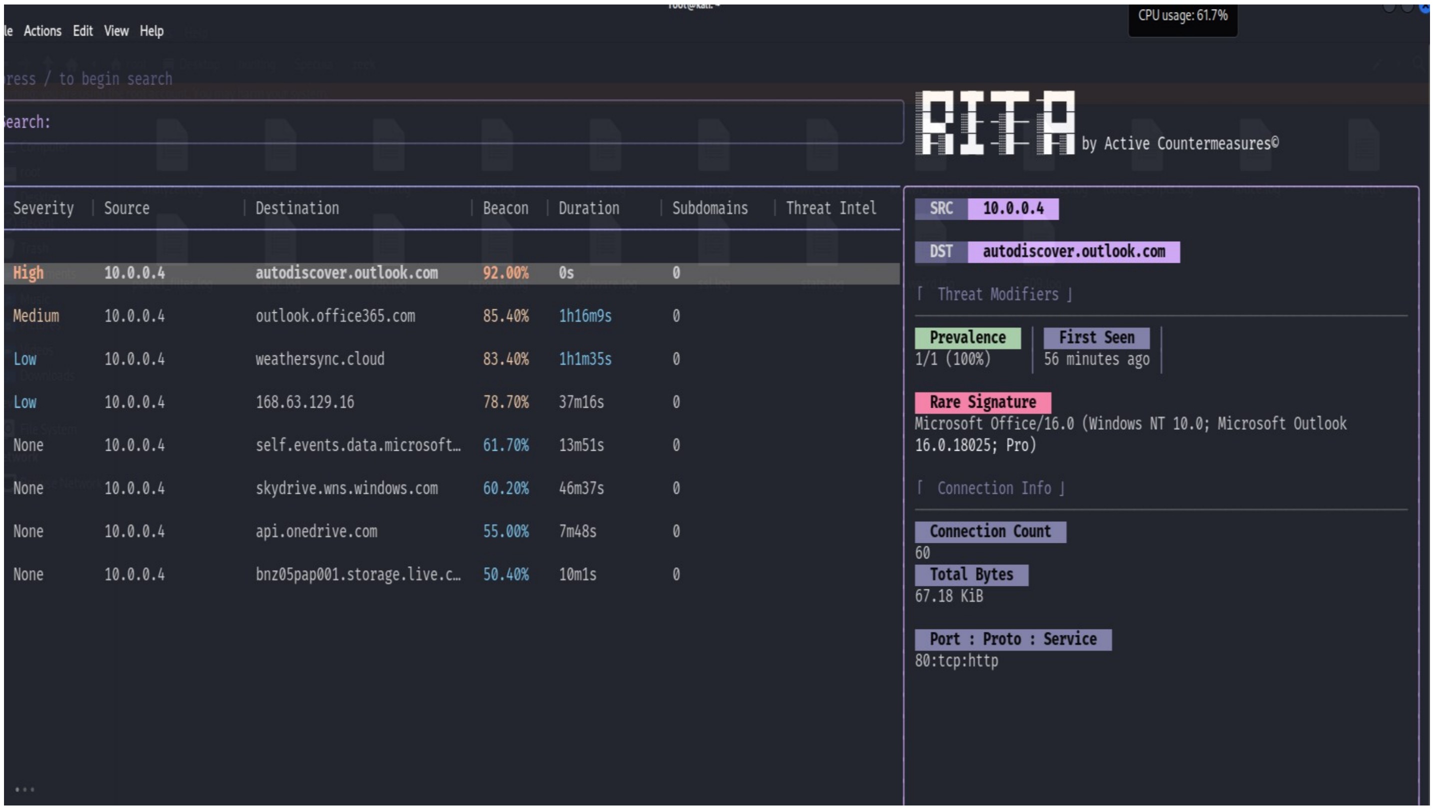
Fig.5: PCAP Processing with Zeek and RITA Import

Explanation:

The process for importing and examining the Specula PCAP file is demonstrated by this terminal output. The sequence includes:

Zeek Command: To process the Specula PCAP file and produce logs in a designated directory (/root/Desktop/hunting/Specula/zeek), run zeek readpcap.

RITA Command: To import the Zeek-generated logs into the RITA database (ritadb) for additional analysis, use rita import. The logs are successfully ingested and prepared for threat hunting operations, as seen by the status.



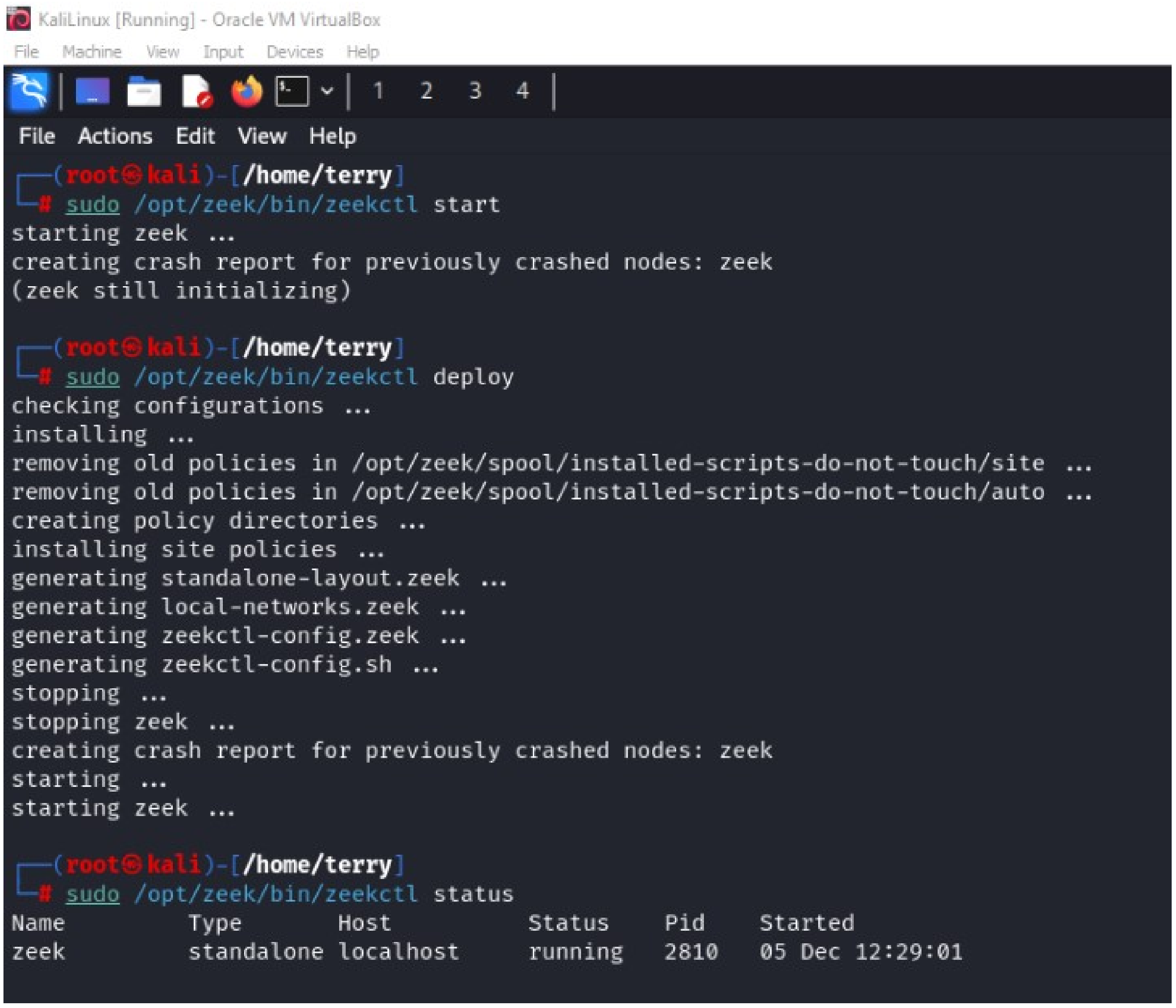
```
ress / to begin search
earch:
                                                                          RITA by Active Countermeasures©

Severity | Source    | Destination               | Beacon | Duration | Subdomains | Threat Intel     SRC 10.0.0.4
                                                                                                    DST autodiscover.outlook.com
High       10.0.0.4    autodiscover.outlook.com    92.00%   0s         0
                                                                                                    [ Threat Modifiers ]
Medium     10.0.0.4    outlook.office365.com       85.40%   1h16m9s    0
                                                                                                    Prevalence  | First Seen
Low        10.0.0.4    weathersync.cloud           83.40%   1h1m35s    0                            1/1 (100%)  | 56 minutes ago

Low        10.0.0.4    168.63.129.16               78.70%   37m16s     0                            Rare Signature
                                                                                                    Microsoft Office/16.0 (Windows NT 10.0; Microsoft Outlook
None       10.0.0.4    self.events.data.microsoft…  61.70%  13m51s     0                            16.0.18025; Pro)

None       10.0.0.4    skydrive.wns.windows.com    60.20%   46m37s     0                            [ Connection Info ]

None       10.0.0.4    api.onedrive.com            55.00%   7m48s      0                            Connection Count
                                                                                                    60
None       10.0.0.4    bnz05pap001.storage.live.c…  50.40%  10m1s      0                            Total Bytes
                                                                                                    67.18 KiB

                                                                                                    Port : Proto : Service
                                                                                                    80:tcp:http
...
```

Fig.6 : Ingesting Logs into RITA and Creating a Database

Explanation:

A connection identified as posing a high-severity danger is indicated by the high alert in the study. Communication was made between the source IP 10.0.0.4 and autodiscover.outlook.com, which was classified as a "Rare Signature" and had a 92% beaconing score. With metadata pointing to several attempts and a noteworthy signature connected to Microsoft Office and Outlook traffic, this connection stands out for its high prevalence and possible malicious purpose. This alert calls for an urgent investigation since it raises the possibility of suspicious activities, such as beaconing or possible exploitation of the autodiscover.outlook.com endpoint.

Zeek Deployment: Start, Deploy, and Status


KaliLinux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
File Actions Edit View Help

```
┌──(root㉿kali)-[/home/terry]
└─# sudo /opt/zeek/bin/zeekctl start
starting zeek ...
creating crash report for previously crashed nodes: zeek
(zeek still initializing)

┌──(root㉿kali)-[/home/terry]
└─# sudo /opt/zeek/bin/zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
creating crash report for previously crashed nodes: zeek
starting ...
starting zeek ...

┌──(root㉿kali)-[/home/terry]
└─# sudo /opt/zeek/bin/zeekctl status
Name         Type       Host          Status    Pid    Started
zeek         standalone localhost     running   2810   05 Dec 12:29:01
```

Fig. 7 : Zeek Deployment

Explanation:

The `zeekctl` utility is a command-line tool used to manage Zeek operations. The `deploy` command applies configuration changes and restarts Zeek services. During deployment, it removes outdated policies, installs updated site policies, and configures settings using scripts like `zeekctl config.sh`. It also resolves any issues caused by previous crashes and ensures that Zeek is properly configured. Following deployment, the `zeekctl start` command starts the Zeek service based on the applied configurations, enabling Zeek to actively monitor and analyze network traffic in real time.

Command to Check Zeek Status:

zeekctl status:

The `zeekctl status` command displays the current operational status of the Zeek service. The output includes several key details: the instance name (e.g., "zeek"), the deployment type (such as "standalone" for single-host configurations), and the host running the service. Additionally, it indicates the current status of Zeek (whether it is running or stopped) and provides the process ID (PID) of the active Zeek service. This information is essential for monitoring and troubleshooting Zeek deployments.

5. Suricata: Start, Enable, Status, and Scan

Command to Start the Suricata Service:

Starting Suricata

The `start suricata` command, executed via `systemctl`, starts the Suricata service. Once initiated, Suricata begins monitoring network traffic or processing data based on its configuration, ensuring immediate functionality.

Enabling Suricata on Boot

The `systemctl enable suricata` command configures Suricata to start automatically during system boot. It creates a persistent systemd symlink for efficient service management, allowing for seamless operation without manual intervention after a reboot.

Checking Suricata's Status

The `status suricata` command verifies if the Suricata service is active and running. It outputs relevant details, including the service name (e.g., `suricata.service`), its current status (running or stopped), the main PID of the active process, and metrics such as CPU and memory usage, offering insights into its resource consumption.

Running Suricata with Configuration and Interface

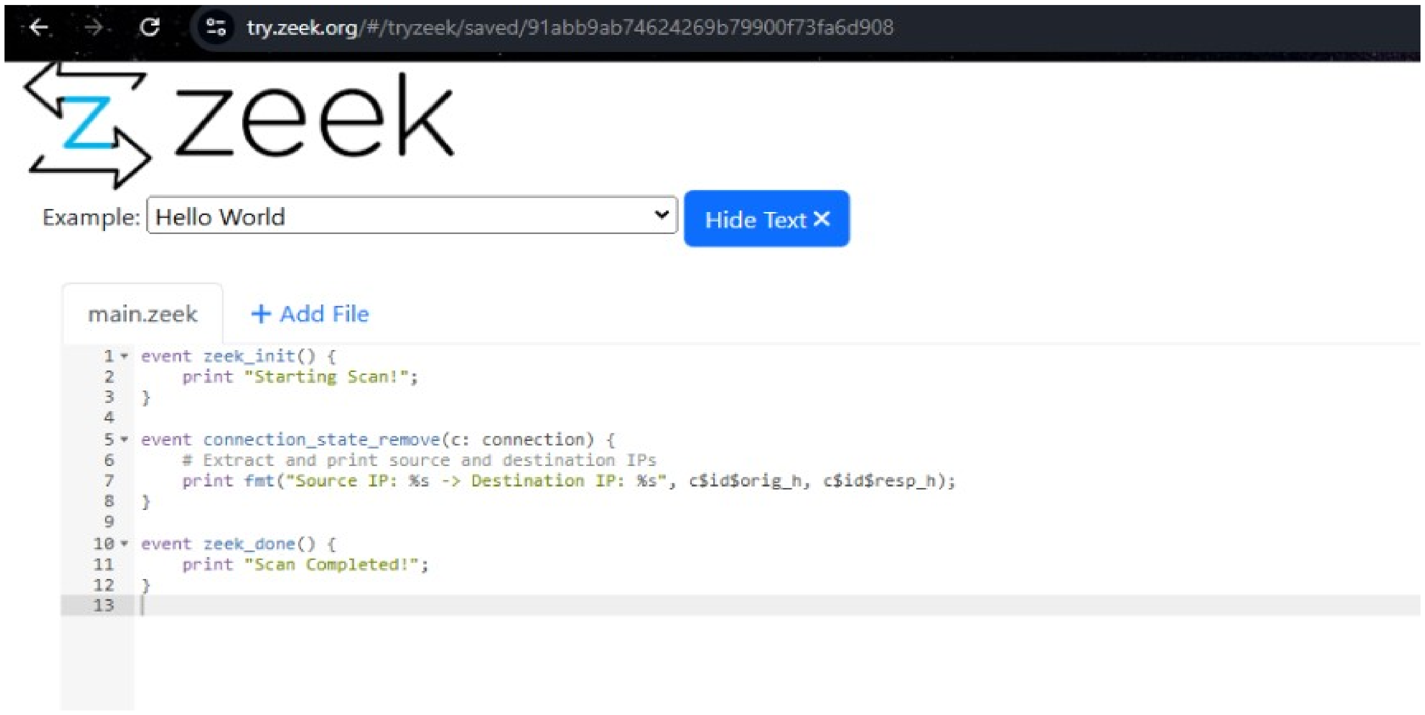
The `sudo suricata -c /etc/suricata/suricata.yaml -i eth0` command starts Suricata with a specified configuration file and network interface. The `-c` flag specifies the configuration file location (e.g., `/etc/suricata/suricata.yaml`), while the `-i` flag defines the network interface (e.g., `eth0`) to capture and analyze traffic.

ls /var/log/suricata/:

Suricata generates logs that are stored in the `/var/log/suricata/` directory, which can be reviewed for insights into network activity and threat detection. Key logs include `eve.json`, a JSON-formatted log that provides detailed event tracking; `fast.log`, which offers a concise summary of detected threats or anomalies; and `stats.log`, which contains performance metrics to evaluate Suricata's operational efficiency. These logs are instrumental for monitoring and analyzing network security events.

Zeek Online Code

Code for Zeek Online Analysis:



try.zeek.org/#/tryzeek/saved/91abb9ab74624269b79900f73fa6d908

Example: Hello World

Hide Text ✕

main.zeek + Add File

```
event zeek_init() {
    print "Starting Scan!";
}

event connection_state_remove(c: connection) {
    # Extract and print source and destination IPs
    print fmt("Source IP: %s -> Destination IP: %s", c$id$orig_h, c$id$resp_h);
}

event zeek_done() {
    print "Scan Completed!";
}
```

Fig 8 : Code for Zeek Online Analysis

Explanation:

The `zeek_init` function is triggered at the start of the analysis process and serves as an indicator that the scan has begun. It prints the message "Starting Scan!" to signify the initiation of network monitoring or traffic analysis.

The `connection_state_remove(c: connection)` function is responsible for extracting and printing the source and destination IP addresses for every network connection analyzed. The source IP is accessed via `c$id$orig_h`, while the destination IP is obtained through `c$id$resp_h`. These IPs are then formatted into a structured message using the `fmt()` function and displayed, providing clear insights into network interactions during the analysis

· **zeek_done**:

The `zeek_done` function is triggered at the conclusion of the analysis process. It prints the message "Scan Completed!" to signal the successful completion of the scan, providing a clear endpoint for the monitoring or traffic analysis session.

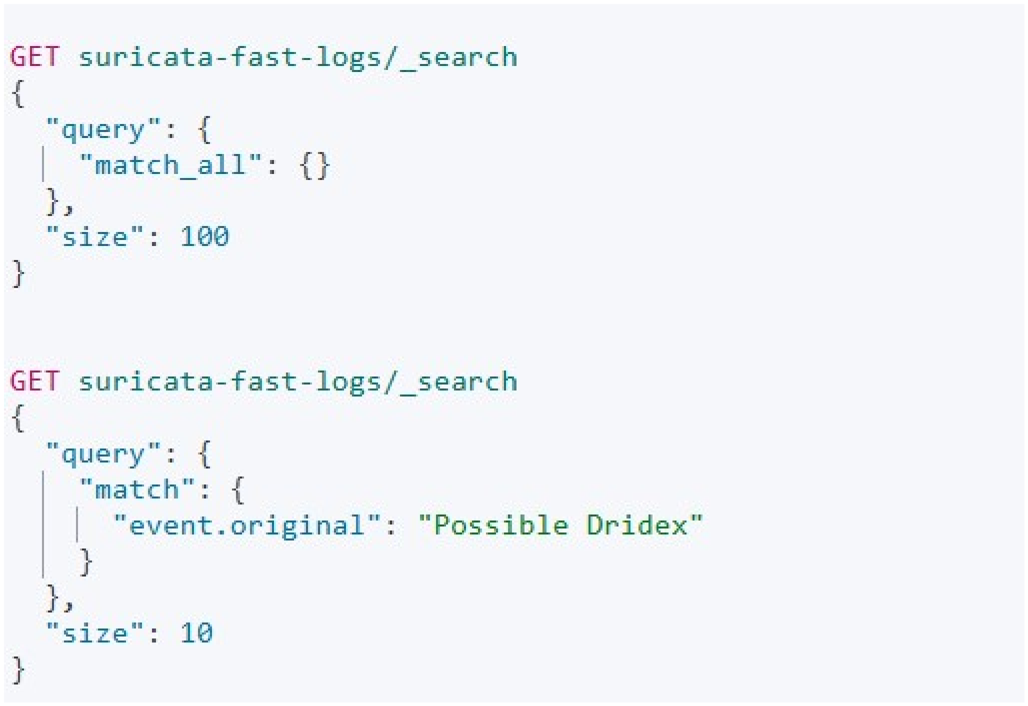
Implementation Steps:

To analyze network data, input the PCAP file or live network feed into the Zeek online tool. Once the data is uploaded, execute the Zeek script to parse the connections and extract meaningful insights. During this process, the tool outputs information in real time, providing progress markers and displaying connections as they are processed.

Output Example:

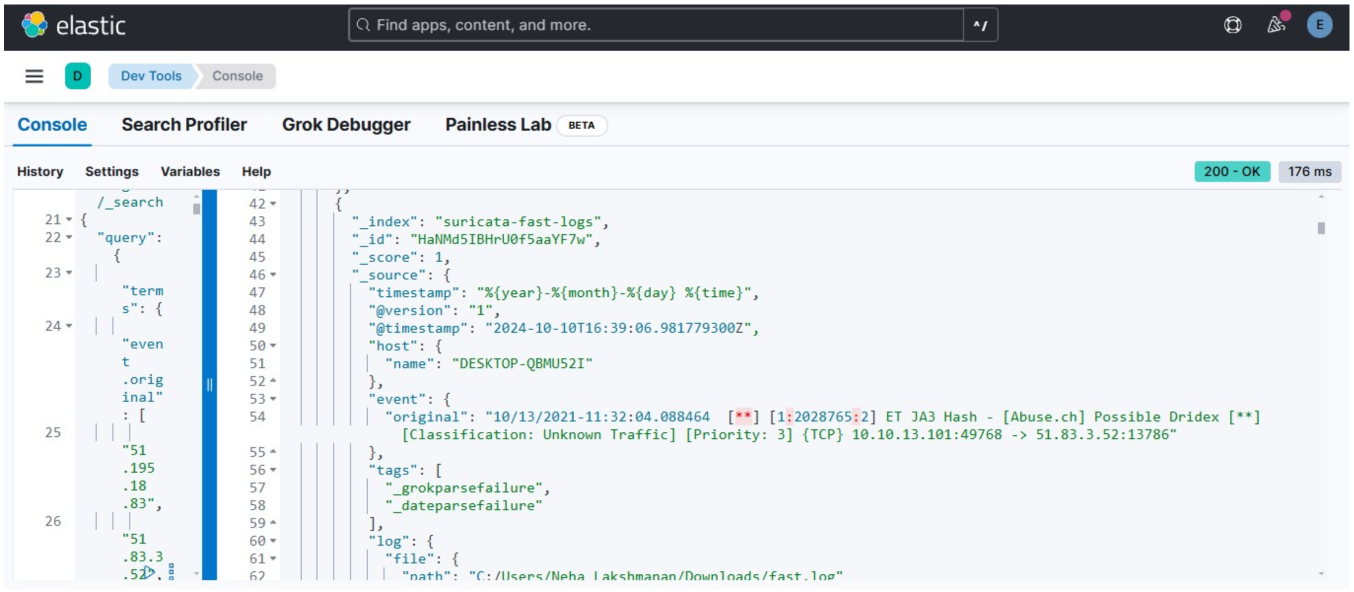
For each connection, the source and destination IP addresses are printed in a structured format. For example, a source IP of `172.16.238.1` connecting to a destination IP of `172.16.238.135` will be displayed. The connections are processed sequentially, and the results continue to be printed until the scan is fully completed, offering a comprehensive view of network interactions.

Querying Suricata Logs in Kibana (Fast Logs Count):



```
GET suricata-fast-logs/_search
{
  "query": {
    "match_all": {}
  },
  "size": 100
}


GET suricata-fast-logs/_search
{
  "query": {
    "match": {
      "event.original": "Possible Dridex"
    }
  },
  "size": 10
}
```

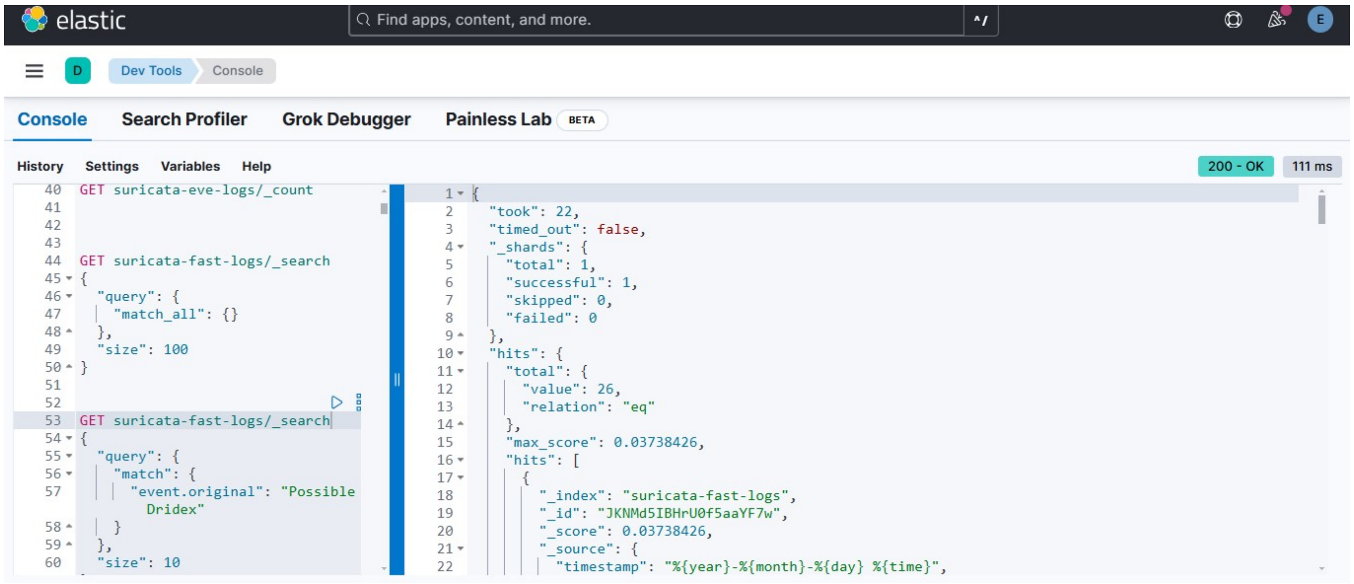
Fig 9- Querying Fast logs



```
/_search
{
  "query":
    {
      "terms": {
        "event.original": [
          "51.195.18.83",
          "51.83.3.52",
```

```
{
  "_index": "suricata-fast-logs",
  "_id": "HaNMd5IBHrU0f5aaYF7w",
  "_score": 1,
  "_source": {
    "timestamp": "%{year}-%{month}-%{day} %{time}",
    "@version": "1",
    "@timestamp": "2024-10-10T16:39:06.981779300Z",
    "host": {
      "name": "DESKTOP-QBMU52I"
    },
    "event": {
      "original": "10/13/2021-11:32:04.088464  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**]
        [Classification: Unknown Traffic] [Priority: 3] {TCP} 10.10.13.101:49768 -> 51.83.3.52:13786"
    },
    "tags": [
      "_grokparsefailure",
      "_dateparsefailure"
    ],
    "log": {
      "file": {
        "path": "C:/Users/Neha Lakshmanan/Downloads/fast.log"
```

Fig 10.1-Output of fast log query



```
GET suricata-eve-logs/_count



GET suricata-fast-logs/_search
{
  "query": {
    "match_all": {}
  },
  "size": 100
}


GET suricata-fast-logs/_search
{
  "query": {
    "match": {
      "event.original": "Possible
        Dridex"
    }
  },
  "size": 10
```

```
{
  "took": 22,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 26,
      "relation": "eq"
    },
    "max_score": 0.03738426,
    "hits": [
      {
        "_index": "suricata-fast-logs",
        "_id": "JKNMd5IBHrU0f5aaYF7w",
        "_score": 0.03738426,
        "_source": {
          "timestamp": "%{year}-%{month}-%{day} %{time}",
```

Fig10.2- Output of fast log query

Implementation Steps:

1. **Command**:

   json

   Copy code

   GET suricata-fast-logs/_search

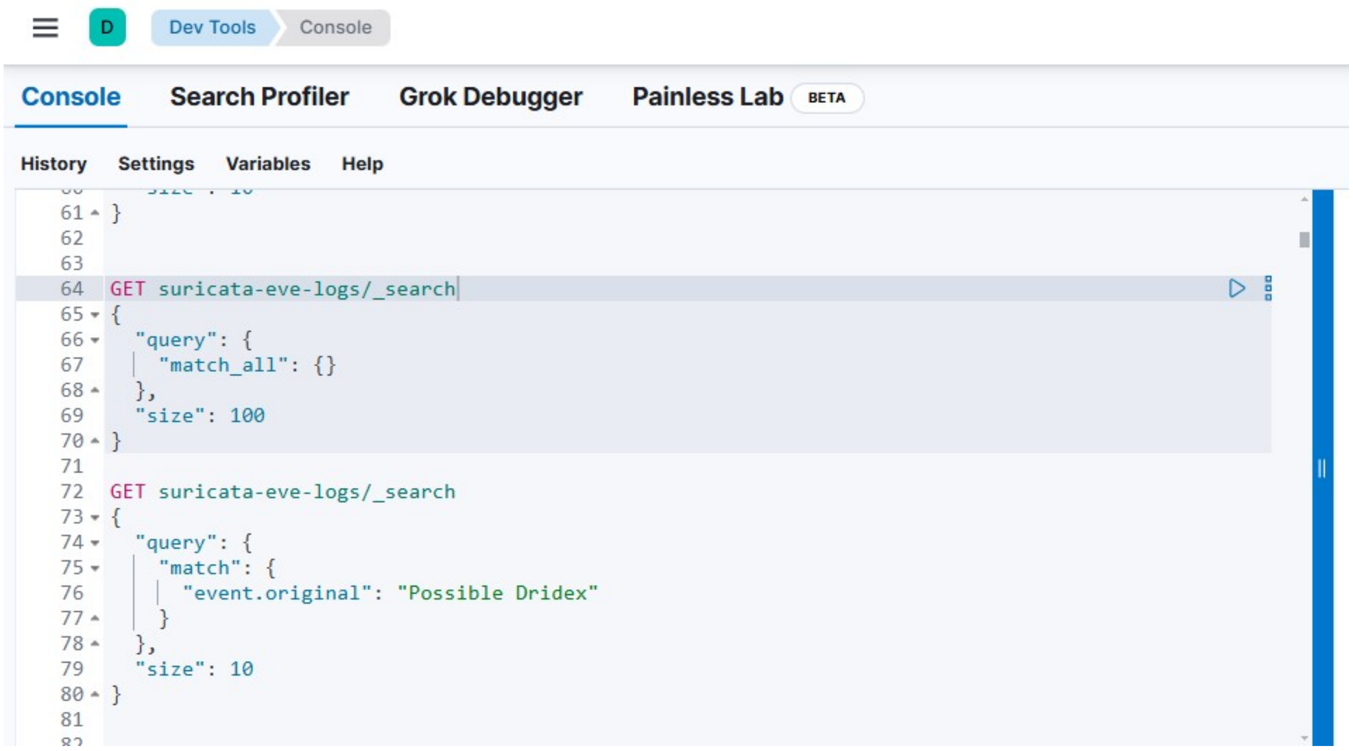
   {

     "query": {

```
    "match_all": {}
  },
  "size": 100
}
```

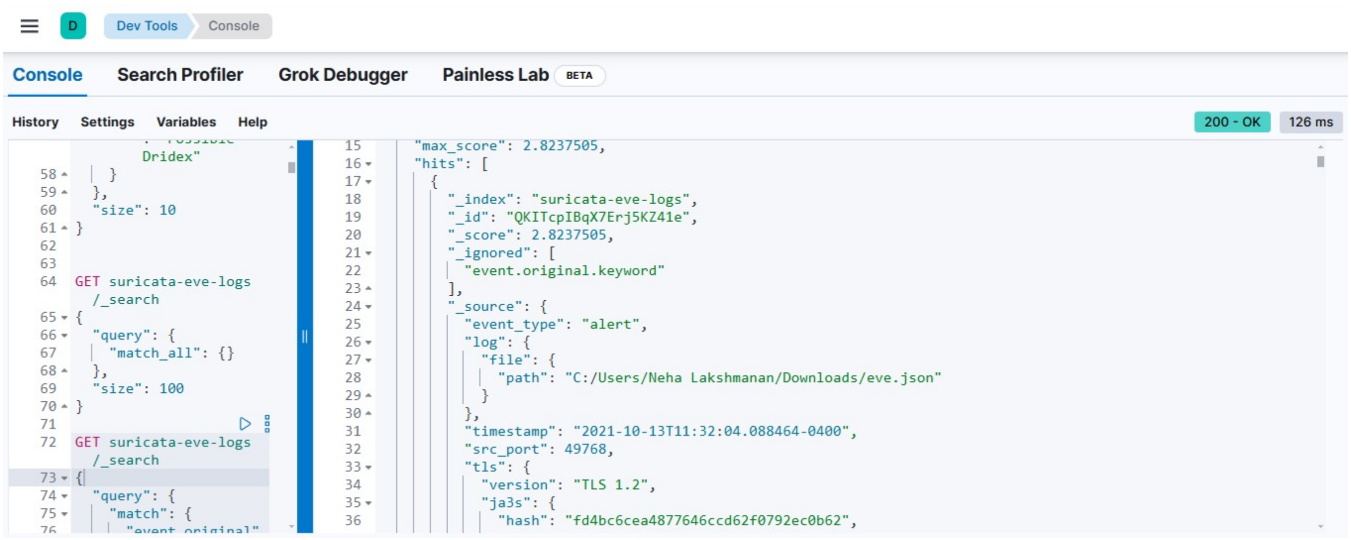
This query retrieves the first 100 entries from the suricata-fast-logs index in Elasticsearch. The match_all parameter ensures that all available logs in the index are fetched, while the size parameter limits the results to 100 logs. The output is a JSON response that includes metadata such
as hits.total and _index, along with individual log entries. This process helps in extracting raw logs for further analysis or review.

Querying Suricata Logs in Kibana (Eve Logs Count)



```
}


GET suricata-eve-logs/_search
{
  "query": {
    "match_all": {}
  },
  "size": 100
}

GET suricata-eve-logs/_search
{
  "query": {
    "match": {
      "event.original": "Possible Dridex"
    }
  },
  "size": 10
}
```

Fig 11-Querying Eve Log Data



```
      Dridex"
    }
  },
  "size": 10
}


GET suricata-eve-logs
  /_search
{
  "query": {
    "match_all": {}
  },
  "size": 100
}

GET suricata-eve-logs
  /_search
{
  "query": {
    "match": {
      "event.original"
```

```
  "max_score": 2.8237505,
  "hits": [
    {
      "_index": "suricata-eve-logs",
      "_id": "QKITcpIBqX7Erj5KZ41e",
      "_score": 2.8237505,
      "_ignored": [
        "event.original.keyword"
      ],
      "_source": {
        "event_type": "alert",
        "log": {
          "file": {
            "path": "C:/Users/Neha Lakshmanan/Downloads/eve.json"
          }
        },
        "timestamp": "2021-10-13T11:32:04.088464-0400",
        "src_port": 49768,
        "tls": {
          "version": "TLS 1.2",
          "ja3s": {
            "hash": "fd4bc6cea4877646ccd62f0792ec0b62",
```

Fig 11.1- Output of Eve Log Query

Implementation Steps:

1. **Command**:

json

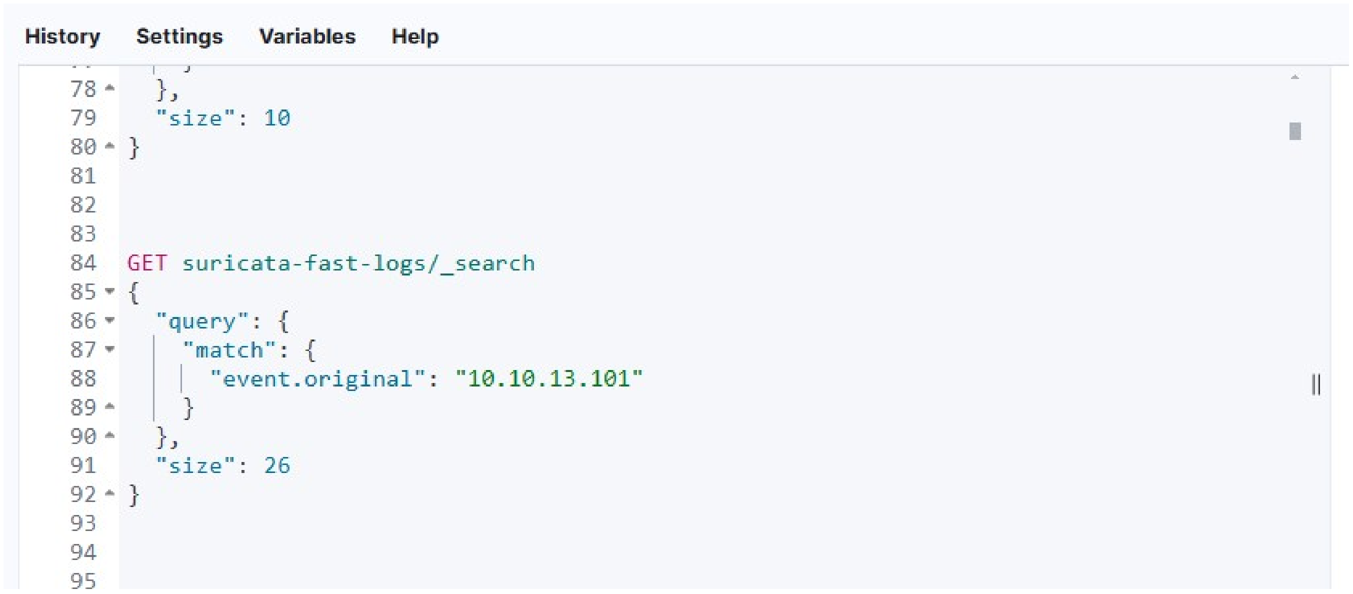
Copy code

```
GET suricata-eve-logs/_search
{
  "query": {
    "match": {
      "event.original": "Possible Dridex"
    }
  },
  "size": 100
}
```

This query searches the suricata-eve-logs index for entries where the

event.original field contains the text "Possible Dridex." The match parameter filters
logs to include only those relevant to Dridex malware, and the size parameter restricts the output to the first 100 matching entries. The result is a JSON response containing metadata such as _index and _id, along with the matched event data, which provides targeted insights into Dridex-related activities.

Refining the Query with Specific IP Filtering



History Settings Variables Help

```
  },
  "size": 10
}



GET suricata-fast-logs/_search
{
  "query": {
    "match": {
      "event.original": "10.10.13.101"
    }
  },
  "size": 26
}
```

Fig 12 -Drilling down to a specific IP.

Implementation Steps:

1. **Command**:

   json

   Copy code

   ```
   GET suricata-fast-logs/_search
   {
     "query": {
       "match": {
         "event.original": "10.10.13.101"
       }
     },
     "size": 26
   ```

```
}
```

To pinpoint traffic linked to a specific IP suspected of malicious activity, the event.original field is queried for entries containing "Possible Dridex" within the suricata-fast-logs index. This targeted approach ensures the retrieval of logs related to potentially harmful behavior. The query is refined to match specific conditions, such as "Possible Dridex," and the size parameter is reduced to 10, focusing on the most recent or critical logs. The output provides detailed logs that highlight suspicious activity, including the event_type (e.g., alerts) and timestamps, offering precise insights into the traffic patterns associated with the IP.

5.3 Working Layout of Forms

Form Title: Incident Response and Vulnerability Management

Network Traffic and Malware Analysis

Section 1: Analyzing Malware with Suricata

Suricata is primarily used to identify malicious behavior in network communications. To do this, load the Dridex PCAP file into Suricata, then examine the logs that are produced—especially the fast.log file. Malicious payload signatures and malware-related communication patterns are important indicators of compromise (IoCs) to examine. This procedure aids in determining possible effects and implementing the required steps to lessen the threat by verifying the infection.

Section 2: Network Analysis with Zeek

Zeek is used to better understand the behavior of network traffic. It creates important logs like notice.log and reporter.log to emphasize important events and warnings, conn.log to track network connections, and dns.log to record DNS queries and resolutions. For instance, the logs might show a DNS query for a dubious domain or warnings about odd traffic patterns, providing information to identify and fix network irregularities.

Section 3: Behavioral Trends with RITA

Network traffic anomalies and behavioral patterns are found using RITA. It is used to provide comprehensive behavioral analysis reports from Zeek logs. These reports offer practical insights to enhance security and reduce risks by identifying abnormalities like beaconing traffic or lateral movement within the network.

Centralized Log Management and Vulnerability Scanning

Section 1: Log Management with ELK

Network activity is centralized and visualized using the ELK Stack to facilitate efficient log management. Elasticsearch is initially used to process logs in order to index and retrieve them quickly. After that, Logstash parses them into a structured manner, allowing for easy management. Lastly, data visualization using charts and graphs, including pie charts or bar graphs, is done with Kibana dashboards. For example, it is possible to visualize repeated connections to a known malicious IP, which would immediately trigger a firewall-level blocking of the IP.

Section 2: Vulnerability Scanning with OWASP ZAP

OWASP ZAP analyzes traffic and application activity to help find vulnerabilities in online applications. The program (such as Mutillidae) is crawled to find hidden files and links, and Firefox is set up with FoxyProxy to intercept web traffic. According to severity categories (High, Medium, Low), ZAP reports are examined for vulnerabilities such as SQL Injection (SQLi) and Cross-Site Scripting (XSS). To safeguard the application, for instance, mitigating measures like sanitizing user input would be necessary for a high-severity finding like Reflected XSS.

Malware Simulation and Analysis

Section 1: Setting Up Simulation

Dridex and other malware can be simulated to investigate their behaviour in a controlled setting. This entails setting up an environment to mimic the malware's exploit mechanism, starting the simulation, and watching the results in a secure environment..

Section 2: Traffic Capture and Analysis

Traffic capture is a critical step in understanding the communication patterns of Dridex malware. Key actions include monitoring network behaviour for payload delivery and command-and-control communication. Analyzing the captured traffic helps extract IoCs, such as IP addresses, domains, and protocols, which are critical for identifying and mitigating similar threats in real-world scenarios.

Section 3: Insights and Outcomes

The final stage involves analyzing findings to enhance defense strategies against malware. Observed communication protocols and exploit delivery mechanisms provide crucial insights into the malware's operational workflow. These findings inform the development of improved detection and prevention techniques, strengthening overall network security.

Comprehensive Network Analysis

Section 1: Offline Network Analysis with Zeek

Zeek's offline network analysis is looking at previously recorded traffic data to find trends and abnormalities. Zeek creates important logs including notice.log to draw attention to irregularities, dns.log to record DNS activity, and conn.log to monitor connection trends. Reporter.log also offers notifications and warnings regarding possible dangers. For instance, by identifying beaconing data aimed at dubious domains, this analysis can help identify criminal activity and enhance network security in general.

Section 2: Live Network Capture with Suricata

Suricata is used to keep an eye on network traffic in real time and identify hazards as they arise. Logs like fast.log are produced when Suricata is allowed to capture network packets, offering information about threats and irregularities in real time. This makes it possible to quickly identify suspicious activity, like traffic from IP addresses that have been identified, and take appropriate action, like blocking malicious traffic, to safeguard the network.

Section 3: Online Analysis with Zeek Tool

Zeek is a robust network monitoring solution utilized for the real-time evaluation of network traffic, offering comprehensive insights into network operations and possible security vulnerabilities. It produces logs such as "conn.log," which hold details about network connections and can identify anomalies like unusual traffic patterns or questionable communication behaviors. Zeek's capacity to examine packets and generate detailed logs aids security teams in swiftly detecting and addressing threats. Through the examination of these logs, administrators can identify harmful activities, track breaches, and respond quickly to reduce risks, thus maintaining a safe network environment.

5.4 Prototype submission

The Hive, Cortex, OpenVAS

Suricata, Zeek, RITA

Together, the Hive, Cortex, and OpenVAS improve incident response and vulnerability management by offering a comprehensive method for identifying, evaluating, and reducing threats. As a unified platform for security incident management and tracking, The Hive streamlines incident tracking for increased productivity by enabling the import of JSON or CSV files for in-depth processing and visualization of task statuses, observables, and severity levels. This is enhanced by Cortex, which integrates easily with The Hive to provide automated workflows that decrease human labor and speed up threat identification by automating threat analysis using numerous analyzers such as PassiveDNS and Abuse Finder.

OpenVAS enhances this framework by performing thorough vulnerability assessments of IP addresses, applications, and network devices, producing detailed reports that classify vulnerabilities by severity, assisting teams in prioritizing and promptly addressing critical weaknesses. Collectively, these tools enhance organization, increase visibility, and bolster the security stance of systems and networks, leading to a more secure and robust environment.

Key Features:

Suricata, Zeek, and RITA collectively offer a robust process for identifying and examining security threats. Suricata analyzes PCAP files, including those with Dridex malware, to produce logs like fast.log, which assist in recognizing Indicators of Compromise (IoCs) by identifying harmful payloads and communication behaviors, validating infections, and facilitating thorough threat assessment.

Zeek enhances this by examining network traffic from PCAP files such as Specula, producing essential logs like conn.log for connection details, dns.log for DNS query and resolution records, as well as notice.log and reporter.log for anomalies and important alerts, providing a thorough insight into network activity.

RITA subsequently processes logs generated by Zeek to detect anomalies, threats, and behavioral trends, offering organized insights into actions such as beaconing traffic and lateral movement. Collectively, these instruments optimize malware detection, network behavior assessment, and threat recognition, enhancing effectiveness and precision in managing security threats.

ELK Stack and OWASP ZAP

This prototype integrates the ELK Stack for centralized log management along with OWASP ZAP for web vulnerability scanning, forming an all-encompassing workflow that tackles both network and application security. The ELK Stack manages logs produced by tools such as Suricata, where Elasticsearch indexes and stores logs for quick access, while Kibana offers visualizations to highlight patterns, anomalies, and threats in the network.

In the meantime, OWASP ZAP examines web applications like Mutillidae for vulnerabilities by setting up FoxyProxy to intercept requests and crawl or spider applications to reveal hidden URLs and files. It creates comprehensive reports that classify vulnerabilities according to severity levels and provide practical mitigation suggestions. Collectively, this unified workflow streamlines log examination, provides comprehensive insights into application weaknesses, and guarantees that organizations can prioritize remediation actions, improving overall security.

Zeek and Suricata

This prototype combines Zeek and Suricata to assess network traffic through both offline and online methods for detecting anomalies and threats. Zeek analyzes network traffic data from PCAP files offline, creating comprehensive logs including conn.log for connection details, dns.log for DNS requests, and notice.log and reporter.log for emphasizing important network occurrences. Suricata enhances this by providing live network capture, observing real-time traffic, and creating fast.log for prompt threat detection and anomaly recognition.

Furthermore, the Zeek online tool enables rapid analysis by uploading PCAP files, offering insights via logs and visual representations that emphasize anomalies and traffic trends. Collectively, these tools provide offline and real-time analysis, efficient threat detection, and in-depth insights into network activity.

5.5 Test and validation

The Hive, Cortex, OpenVAS

The testing and validation strategy guarantees the smooth integration and effectiveness of The Hive, Cortex, and OpenVAS in incident response processes. Test scenarios for The Hive comprise importing JSON and CSV log files to verify parsing, generating and modifying cases with severity levels, delegating tasks, and assessing notification and observation features.

Cortex is evaluated for the correct operation of analyzers like VirusTotal and Abuse Finder against observables including domains and IPs, guaranteeing precise integration with The Hive and the production of automated results, along with logged failures and useful error messages.

OpenVAS undergoes testing through scans on different targets, such as web apps, IP ranges, and network devices, confirming the correctness of vulnerability reports and severity levels, as well as validating tracking of mitigations. The validation standards emphasize achieving successful integration between The Hive and Cortex, producing automated outcomes, delivering detailed and precise vulnerability reports from OpenVAS, and ensuring efficient task and case monitoring within The Hive.

Suricata, Zeek, RITA

The testing and validation strategy guarantees the effectiveness and functionality of Suricata, Zeek, and RITA in identifying malware and assessing network traffic. For Suricata, the assessment entails examining a PCAP file, like one with Dridex malware, and confirming the production of precise logs (fast.log) that pinpoint Indicators of Compromise (IoCs), comprising harmful payloads and communication trends. Zeek is evaluated by analyzing a PCAP file to confirm the generation of logs (conn.log, dns.log, notice.log, and reporter.log), making sure they correctly reflect DNS requests, connection information, and important notifications.

RITA is verified through the processing of Zeek logs, ensuring that a database is established for anomaly detection, and identifying particular activities such as beaconing or lateral movement. The validation standards emphasize confirming that logs produced by Suricata and Zeek align with anticipated results from recognized malware traffic, RITA efficiently detects and illustrates anomalies and trends, and the combination of these tools yields actionable intelligence for improved threat detection.

ELK Stack and OWASP ZAP

The test and validation strategy assesses the functionalities of the ELK Stack for centralized log management and OWASP ZAP for scanning web vulnerabilities. In the ELK Stack, the assessment includes importing Suricata logs into Elasticsearch to confirm correct indexing, utilizing Kibana to generate visual representations like pie charts and bar graphs to detect patterns or irregularities, and evaluating the capacity to search and filter logs according to IP addresses, ports, or events.

For OWASP ZAP, the assessment involves setting up FoxyProxy in Firefox to capture requests, exploring and scanning the Mutillidae application to reveal concealed URLs and files, and producing vulnerability reports to confirm the precision of identifying problems such as Cross-Site Scripting (XSS) or SQL Injection.

The validation standards guarantee that the ELK Stack accurately parses and visualizes logs without data loss or mistakes, while ZAP successfully detects vulnerabilities with precise severity assessments and practical remediation advice.

Zeek and Suricata

The evaluation plan for testing and validation assesses the performance of Zeek and Suricata for analyzing network traffic in real-time and offline. Zeek undergoes testing by examining PCAP files offline to confirm the creation of logs including conn.log, dns.log, notice.log, and reporter.log, ensuring precise identification of irregularities like strange connection behaviors or DNS activities.

Suricata is evaluated for its ability to capture real-time traffic, ensuring that logs like fast.log are produced promptly and efficiently identify suspicious traffic or anomalies in live datasets.

Furthermore, the Zeek online tool undergoes testing by uploading previously captured PCAP files to ensure that the parsed logs are correctly visualized for analysis. The validation standards guarantee that Zeek delivers precise and detailed traffic data from both offline and online sources, while Suricata identifies real-time threats through actionable log outputs, and using these tools together underscores their combined strengths for thorough network traffic analysis.

5.6 Behavioural Analysis

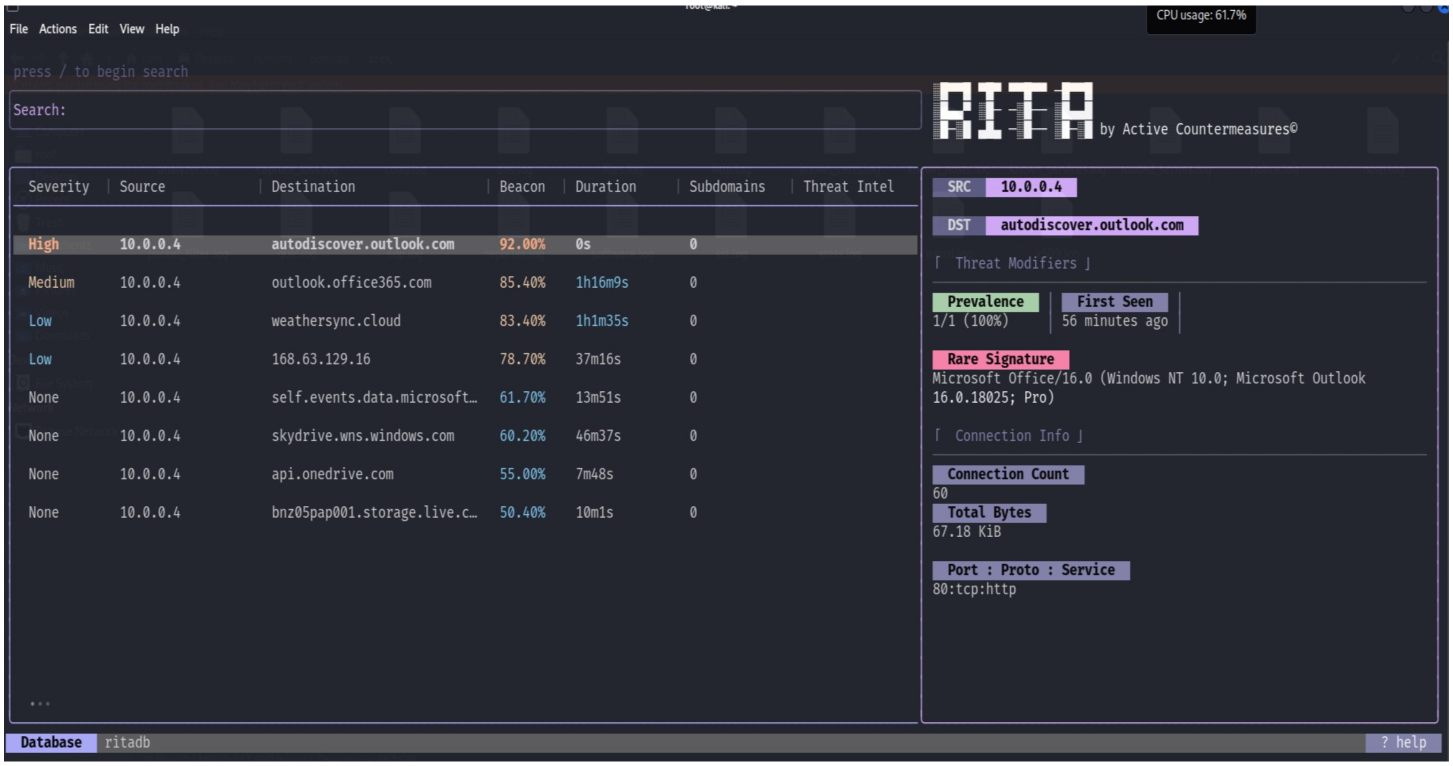
Suricata Fast Logs (Dridex Detection Results):

Fig 13 : Suricata Fast Logs

The fast.log file created by Suricata underscores possible Dridex malware involvement via frequent detections of JA3 Hash classifications identified as Possible Dridex, a pattern closely linked to the Dridex banking trojan. The Priority Level (3) indicates threats that are moderate to low, which may

suggest initial reconnaissance or attempts at communication. Monitored traffic reveals steady malicious patterns between the compromised host (10.10.13.101) and various external IP addresses (174.128.245.202, 51.195.18.83).

Numerous TCP connections aiming at port 443 (HTTPS) were detected, potentially suggesting data exfiltration or interaction with command-and-control (C2) servers. The consistency in connection priorities and classifications indicates automated actions typical of malware. In summary, the results indicate that Dridex utilizes encrypted channels to avoid being detected, necessitating further investigation, like packet inspection, to comprehend payload details and C2 server activity.



```
press / to begin search
Search:

Severity | Source     | Destination                  | Beacon | Duration | Subdomains | Threat Intel

High       10.0.0.4     autodiscover.outlook.com       92.00%   0s         0
Medium     10.0.0.4     outlook.office365.com          85.40%   1h16m9s    0
Low        10.0.0.4     weathersync.cloud              83.40%   1h1m35s    0
Low        10.0.0.4     168.63.129.16                  78.70%   37m16s     0
None       10.0.0.4     self.events.data.microsoft…    61.70%   13m51s     0
None       10.0.0.4     skydrive.wns.windows.com       60.20%   46m37s     0
None       10.0.0.4     api.onedrive.com               55.00%   7m48s      0
None       10.0.0.4     bnz05pap001.storage.live.c…    50.40%   10m1s      0
...

SRC  10.0.0.4
DST  autodiscover.outlook.com
⌈ Threat Modifiers ⌋
Prevalence     | First Seen
1/1 (100%)     | 56 minutes ago
Rare Signature
Microsoft Office/16.0 (Windows NT 10.0; Microsoft Outlook
16.0.18025; Pro)
⌈ Connection Info ⌋
Connection Count
60
Total Bytes
67.18 KiB
Port : Proto : Service
80:tcp:http
```

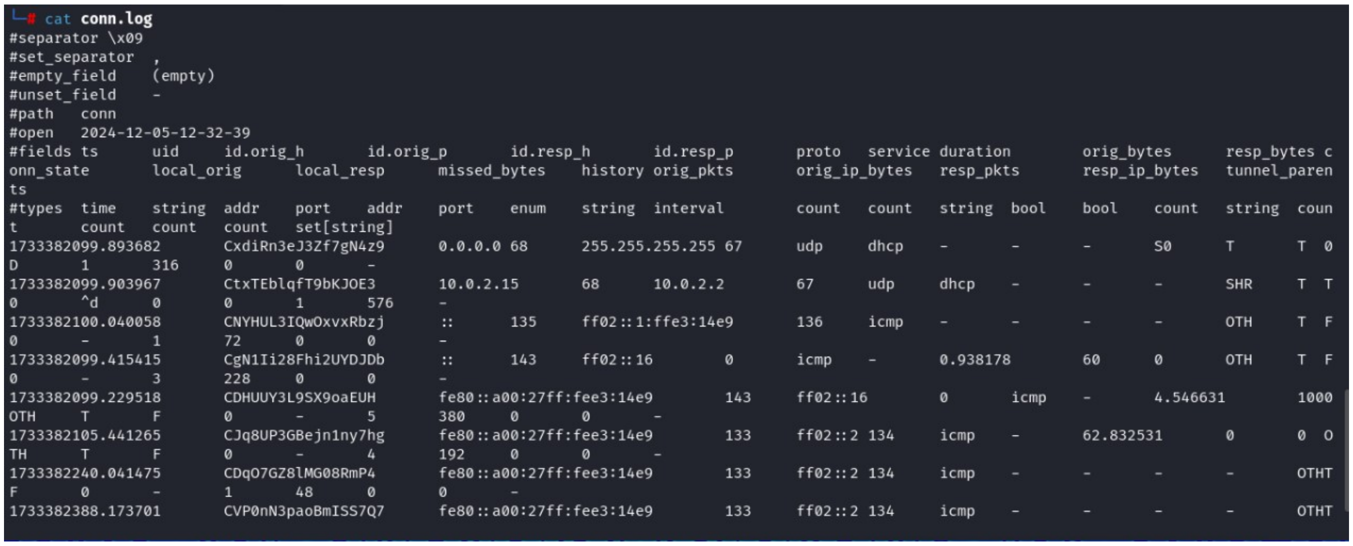
Fig 14: RITA Database (Beaconing and Suspicious Connections)

The examination performed with the RITA (Real Intelligence Threat Analytics) tool uncovers important insights into network traffic, emphasizing beaconing behavior, subdomain enumeration, and threat indicators. A significant beaconing rate (92%) to autodiscover.outlook.com indicates frequent communication, a typical trait of malware beaconing. The source IP (10.0.0.4) exhibits a

100% frequency in interaction with the destination, suggesting steady, possibly automated, or harmful intent. The identification of an unusual signature associated with Microsoft Outlook's communication behaviors additionally suggests potential exploitation of legitimate services to conceal harmful actions.

Connection information points to the utilization of port 80 (HTTP), implying unencrypted communication possibly intended for data exfiltration or command relays, whereas subdomain examination uncovers brief sessions (less than 2 minutes), signifying efforts to evade detection. Anomalies involve the misuse of the genuine domain autodiscover.outlook.com, frequently associated with phishing or improperly set up email services, necessitating an examination of DNS and email settings. Suggested measures involve performing reverse DNS lookups, examining payloads for harmful indicators, and reviewing email settings to safeguard against the illicit use of Outlook services.

1. Zeek conn.log



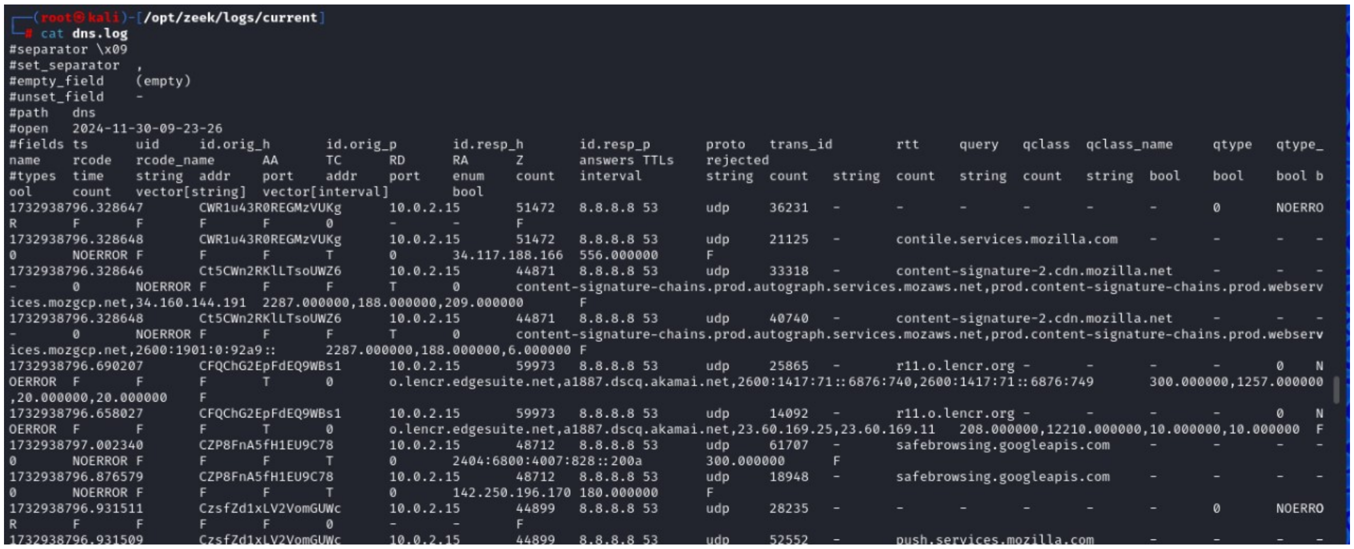
```
└─# cat conn.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2024-12-05-12-32-39
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   service duration        orig_bytes      resp_bytes c
onn_state       local_orig      local_resp      missed_bytes    history orig_pkts       orig_ip_bytes   resp_pkts       resp_ip_bytes   tunnel_paren
ts
#types  time    string  addr    port    addr    port    enum    string  interval        count   count   string  bool    bool    count   string  coun
t       count   count   count   set[string]
1733382099.893682       CxdiRn3eJ3Zf7gN4z9      0.0.0.0 68      255.255.255.255 67      udp     dhcp    -       -       -       S0      T       T  0
D       1       316     0       0       -
1733382099.903967       CtxTEblqfT9bKJOE3       10.0.2.15       68      10.0.2.2        67      udp     dhcp    -       -       -       SHR     T  T
0       ^d      0       0       1       576     -
1733382100.040058       CNYHUL3IQwOxvxRbzj      ::      135     ff02::1:ffe3:14e9       136     icmp    -       -       -       -       OTH     T  F
0       -       1       72      0       0       -
1733382099.415415       CgN1Ii28Fhi2UYDJDb      ::      143     ff02::16        0       icmp    -       0.938178        60      0       OTH     T  F
0       -       3       228     0       0       -
1733382099.229518       CDHUUY3L9SX9oaEUH       fe80::a00:27ff:fee3:14e9        143     ff02::16        0       icmp    -       4.546631        1000
OTH     T       F       0       -       5       380     0       0       -
1733382105.441265       CJq8UP3GBejn1ny7hg      fe80::a00:27ff:fee3:14e9        133     ff02::2 134     icmp    -       62.832531       0       0  O
TH      T       F       0       -       4       192     0       0       -
1733382240.041475       CDqO7GZ8lMG08RmP4       fe80::a00:27ff:fee3:14e9        133     ff02::2 134     icmp    -       -       -       -       OTHT
F       0       -       1       48      0       0       -
1733382388.173701       CVP0nN3paoBmISS7Q7      fe80::a00:27ff:fee3:14e9        133     ff02::2 134     icmp    -       -       -       -       OTHT
```

Fig 15:Zeek conn.log

The conn.log file generated by Zeek contains comprehensive metadata regarding network connections, delivering important insights into network activity. It features characteristics like id.orig_h and id.resp_h for source and destination IPs, id.orig_p and id.resp_p for source and destination ports, proto for the type of protocol (e.g., TCP, ICMP), and service to show if the connection relates to a particular service such as HTTP or DNS. The log additionally records the connection status, including established (SF) or rejected (RST), allowing for traffic pattern analysis.

This data is crucial for recognizing unusual protocol usage and uncovering possible reconnaissance actions such as network scans.

Zeek dns.log

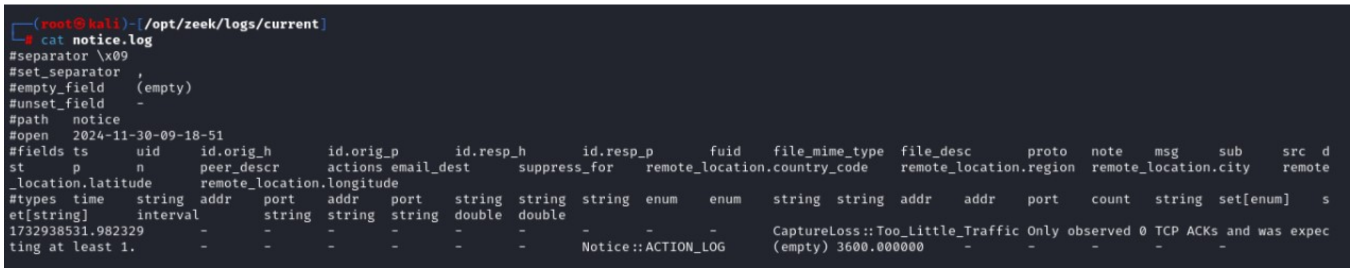


```
┌──(root㉿kali)-[/opt/zeek/logs/current]
└─# cat dns.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   dns
#open   2024-11-30-09-23-26
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id        rtt     query   qclass  qclass_name     qtype   qtype_
name    rcode   rcode_name      AA      TC      RD      RA      Z       answers TTLs    rejected
#types  time    string  addr    port    addr    port    enum    count   interval        string  count   string  count   string  bool    bool    bool b
ool     count   vector[string]  vector[interval]        bool
1732938796.328647       CWR1u43R0REGMzVUKg      10.0.2.15       51472   8.8.8.8 53      udp     36231   -       -       -       -       -       -       0       NOERRO
R       F       F       F       F       0       -       -       F
1732938796.328648       CWR1u43R0REGMzVUKg      10.0.2.15       51472   8.8.8.8 53      udp     21125   -       contile.services.mozilla.com    -       -       -       -
0       NOERROR F       F       F       T       0       34.117.188.166  556.000000      F
1732938796.328646       Ct5CWn2RKlLTsoUWZ6      10.0.2.15       44871   8.8.8.8 53      udp     33318   -       content-signature-2.cdn.mozilla.net     -       -       -
-       0       NOERROR F       F       F       T       0       content-signature-chains.prod.autograph.services.mozaws.net,prod.content-signature-chains.prod.webserv
ices.mozgcp.net,34.160.144.191  2287.000000,188.000000,209.000000       F
1732938796.328648       Ct5CWn2RKlLTsoUWZ6      10.0.2.15       44871   8.8.8.8 53      udp     40740   -       content-signature-2.cdn.mozilla.net     -       -       -
-       0       NOERROR F       F       F       T       0       content-signature-chains.prod.autograph.services.mozaws.net,prod.content-signature-chains.prod.webserv
ices.mozgcp.net,2600:1901:0:92a9::      2287.000000,188.000000,6.000000 F
1732938796.690207       CFQChG2EpFdEQ9WBs1      10.0.2.15       59973   8.8.8.8 53      udp     25865   -       r11.o.lencr.org -       -       -       -       0       N
OERROR  F       F       F       T       0       o.lencr.edgesuite.net,a1887.dscq.akamai.net,2600:1417:71::6876:740,2600:1417:71::6876:749       300.000000,1257.000000
,20.000000,20.000000    F
1732938796.658027       CFQChG2EpFdEQ9WBs1      10.0.2.15       59973   8.8.8.8 53      udp     14092   -       r11.o.lencr.org -       -       -       -       0       N
OERROR  F       F       F       T       0       o.lencr.edgesuite.net,a1887.dscq.akamai.net,23.60.169.25,23.60.169.11   208.000000,12210.000000,10.000000,10.000000     F
1732938797.002340       CZP8FnA5fH1EU9C78       10.0.2.15       48712   8.8.8.8 53      udp     61707   -       safebrowsing.googleapis.com     -       -       -       -
0       NOERROR F       F       F       T       0       2404:6800:4007:828::200a        300.000000      F
1732938796.876579       CZP8FnA5fH1EU9C78       10.0.2.15       48712   8.8.8.8 53      udp     18948   -       safebrowsing.googleapis.com     -       -       -       -
0       NOERROR F       F       F       T       0       142.250.196.170 180.000000      F
1732938796.931511       CzsfZd1xLV2VomGUWc      10.0.2.15       44899   8.8.8.8 53      udp     28235   -       -       -       -       -       -       0       NOERRO
R       F       F       F       F       0       -       -       F
1732938796.931509       CzsfZd1xLV2VomGUWc      10.0.2.15       44899   8.8.8.8 53      udp     52552   -       push.services.mozilla.com       -       -       -       -
```

Fig 16:Zeek dns.log

The dns.log file records DNS resolution queries, providing essential information about possible malicious behavior. It displays queried domains in the query field, where rare or atypical domains could suggest interaction with Command and Control (C2) servers. Response attributes, including id.resp_h (IP of the destination DNS server) and qtype (query types such as A, AAAA, or CNAME), offer further context for analysis. Regular requests to unfamiliar domains or non-traditional TLDs (e.g., .xyz, .info) might indicate harmful intent, and trends suggestive of DNS amplification or tunneling attacks can likewise be detected via this log.

Zeek Notice.log



```
┌──(root㉿kali)-[/opt/zeek/logs/current]
└─# cat notice.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   notice
#open   2024-11-30-09-18-51
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       fuid    file_mime_type  file_desc       proto   note    msg     sub     src     d
st      p       n       peer_descr      actions email_dest      suppress_for    remote_location.country_code    remote_location.region  remote_location.city    remote
_location.latitude      remote_location.longitude
#types  time    string  addr    port    addr    port    string  string  string  enum    enum    string  string  addr    addr    port    count   string  set[enum]       s
et[string]      interval        string  string  string  double  double
1732938531.982329       -       -       -       -       -       -       -       -       -       CaptureLoss::Too_Little_Traffic Only observed 0 TCP ACKs and was expec
ting at least 1.        -       -       -       -       -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
```

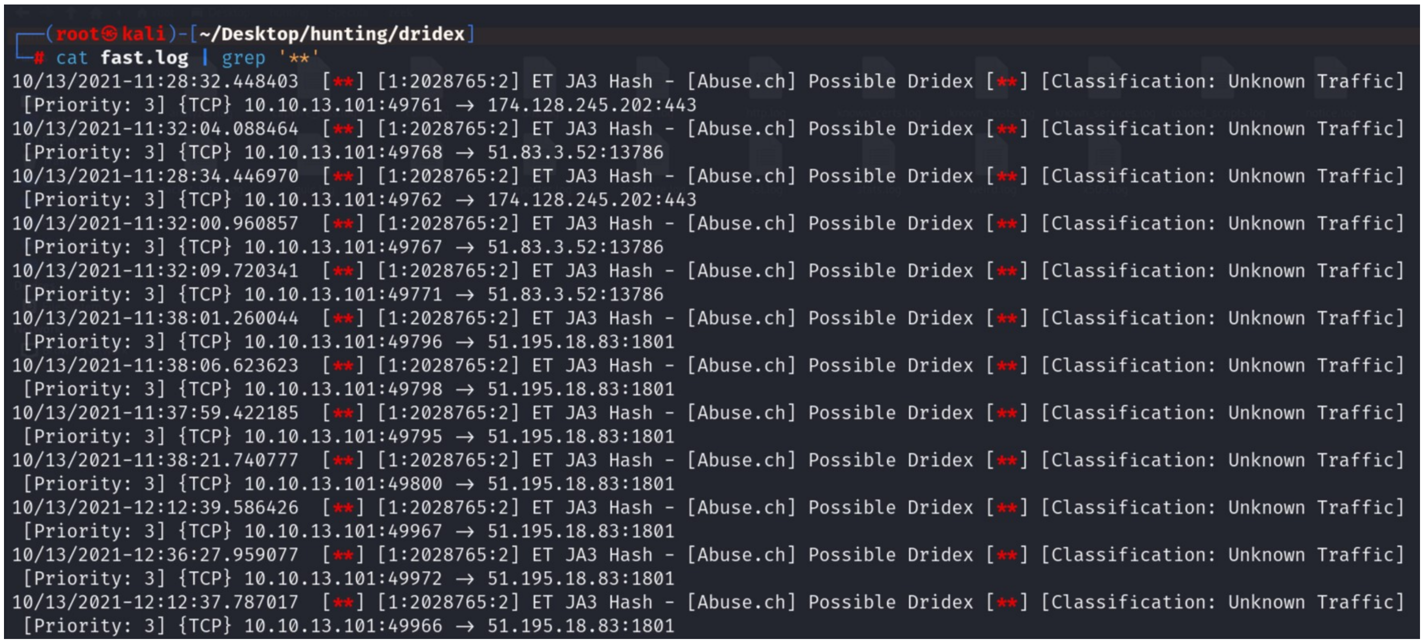
Fig 17:Zeek Notice.log

The notice.log file records network irregularities or warnings, including possible traffic problems like CaptureLoss or unfinished transmissions. For instance, elevated CaptureLoss values may signify packet loss, which might result in a lack of comprehensive visibility into network traffic. The reporter.log emphasizes errors or warnings, such as "Invalid TCP/UDP checksums" or checksum offloading issues, which may lead to incorrect log creation and necessitate additional scrutiny to guarantee dependable network analysis

Suricata eve.json

The eve.json file generated by Suricata presents comprehensive logs, encompassing event types and payload assessment, providing a glimpse into network activity. Important fields like event_type classify identified activities such as alerts or flows, whereas src_ip and dest_ip indicate the source and destination IP addresses for every event. The file contains indicators like alerts for questionable behaviors, including Command and Control (C2) connections or attempts to exploit vulnerabilities. By linking event data, the file facilitates the separation of particular threats and the recognition of their behavioral trends, assisting in thorough threat assessment.

Suricata Fast.log



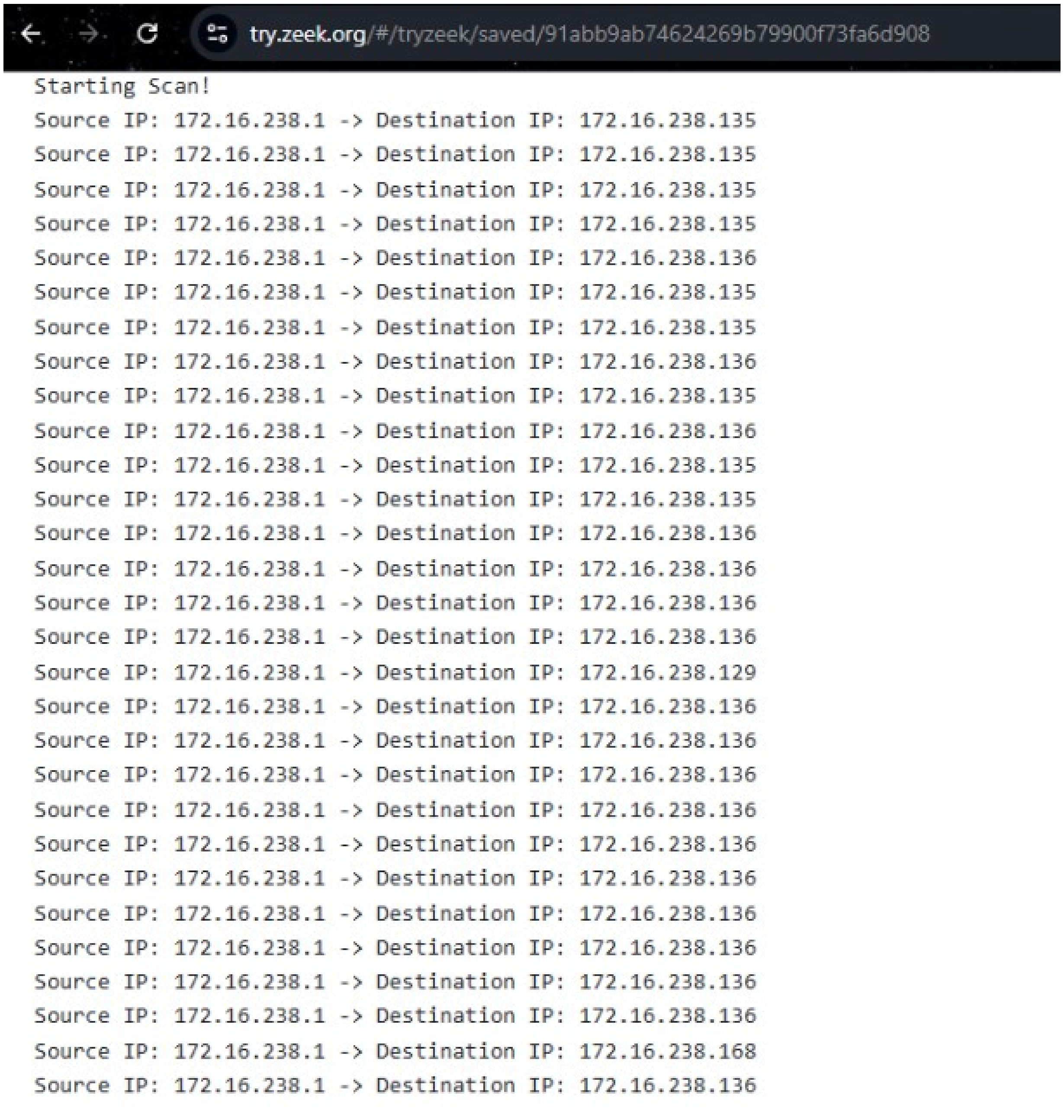
```
┌──(root㉿kali)-[~/Desktop/hunting/dridex]
└─# cat fast.log | grep '**'
10/13/2021-11:28:32.448403  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49761 → 174.128.245.202:443
10/13/2021-11:32:04.088464  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49768 → 51.83.3.52:13786
10/13/2021-11:28:34.446970  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49762 → 174.128.245.202:443
10/13/2021-11:32:00.960857  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49767 → 51.83.3.52:13786
10/13/2021-11:32:09.720341  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49771 → 51.83.3.52:13786
10/13/2021-11:38:01.260044  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49796 → 51.195.18.83:1801
10/13/2021-11:38:06.623623  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49798 → 51.195.18.83:1801
10/13/2021-11:37:59.422185  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49795 → 51.195.18.83:1801
10/13/2021-11:38:21.740777  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49800 → 51.195.18.83:1801
10/13/2021-12:12:39.586426  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49967 → 51.195.18.83:1801
10/13/2021-12:36:27.959077  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49972 → 51.195.18.83:1801
10/13/2021-12:12:37.787017  [**] [1:2028765:2] ET JA3 Hash - [Abuse.ch] Possible Dridex [**] [Classification: Unknown Traffic]
 [Priority: 3] {TCP} 10.10.13.101:49966 → 51.195.18.83:1801
```

Fig 18 :Suricata Fast.log

The fast.log file delivers summarized alerts, allowing swift assessment of threats. It features patterns like "QUIC failed decrypt," which could suggest possible encrypted communications. Highlighted

IP addresses must be checked against established malicious activity databases, while the alert priority levels aid in focusing on critical threats that need urgent attention, thus enhancing the incident response workflow.

Zeek Online Tool Output



try.zeek.org/#/tryzeek/saved/91abb9ab74624269b79900f73fa6d908

```
Starting Scan!
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.135
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.129
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.168
Source IP: 172.16.238.1 -> Destination IP: 172.16.238.136
```

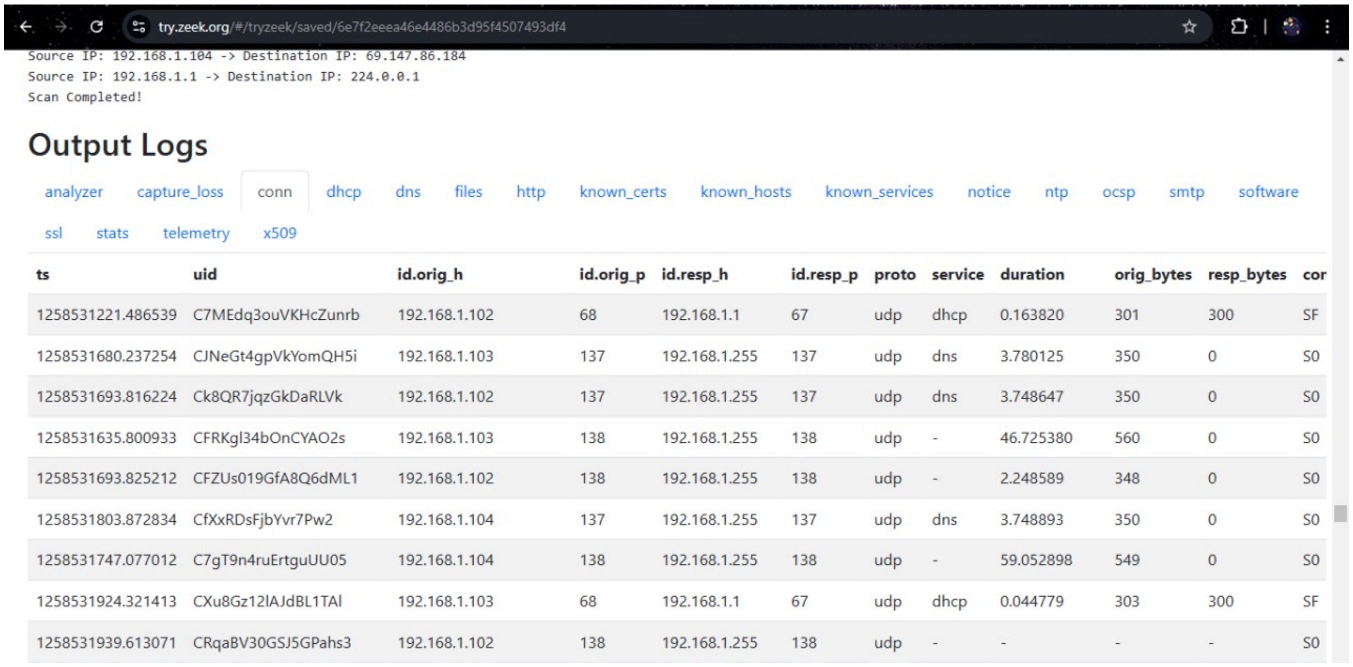
Fig 19 :Zeek Online Tool Output

The Zeek tool available online analyzes PCAP data to offer insights into network behavior. Its examination emphasizes dynamic links between source and destination IPs, providing an overview of network traffic flows that assists in comprehending and tracking communication trends.

Zeek Online Logs

The logs produced by the Zeek online platform feature statistical metrics like capture packet loss, protocol distribution, and event triggers. These areas are essential for evaluating overall network well-being and grasping behavioral trends, offering a complete picture of network performance and irregularities.

Zeek Online conn.log


try.zeek.org/#/tryzeek/saved/6e7f2eeea46e4486b3d95f4507493df4

```
Source IP: 192.168.1.104 -> Destination IP: 69.147.86.184
Source IP: 192.168.1.1 -> Destination IP: 224.0.0.1
Scan Completed!
```

Output Logs

ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	cor
1258531221.486539	C7MEdq3ouVKHcZunrb	192.168.1.102	68	192.168.1.1	67	udp	dhcp	0.163820	301	300	SF
1258531680.237254	CJNeGt4gpVkYomQH5i	192.168.1.103	137	192.168.1.255	137	udp	dns	3.780125	350	0	S0
1258531693.816224	Ck8QR7jqzGkDaRLVk	192.168.1.102	137	192.168.1.255	137	udp	dns	3.748647	350	0	S0
1258531635.800933	CFRKgl34bOnCYAO2s	192.168.1.103	138	192.168.1.255	138	udp	-	46.725380	560	0	S0
1258531693.825212	CFZUs019GfA8Q6dML1	192.168.1.102	138	192.168.1.255	138	udp	-	2.248589	348	0	S0
1258531803.872834	CfXxRDsFjbYvr7Pw2	192.168.1.104	137	192.168.1.255	137	udp	dns	3.748893	350	0	S0
1258531747.077012	C7gT9n4ruErtguUU05	192.168.1.104	138	192.168.1.255	138	udp	-	59.052898	549	0	S0
1258531924.321413	CXu8Gz12lAJdBL1TAl	192.168.1.103	68	192.168.1.1	67	udp	dhcp	0.044779	303	300	SF
1258531939.613071	CRqaBV30GSJ5GPahs3	192.168.1.102	138	192.168.1.255	138	udp	-	-	-	-	S0

Fig 20 :Zeek Online conn.log

Similar to the local `conn.log` file, the online-generated log contains detailed information about connection states, IP addresses, and protocol metadata. It highlights specific network flows, enabling targeted investigations into suspicious traffic patterns for more effective threat analysis.

Dashboard Visualization (Pie Chart)

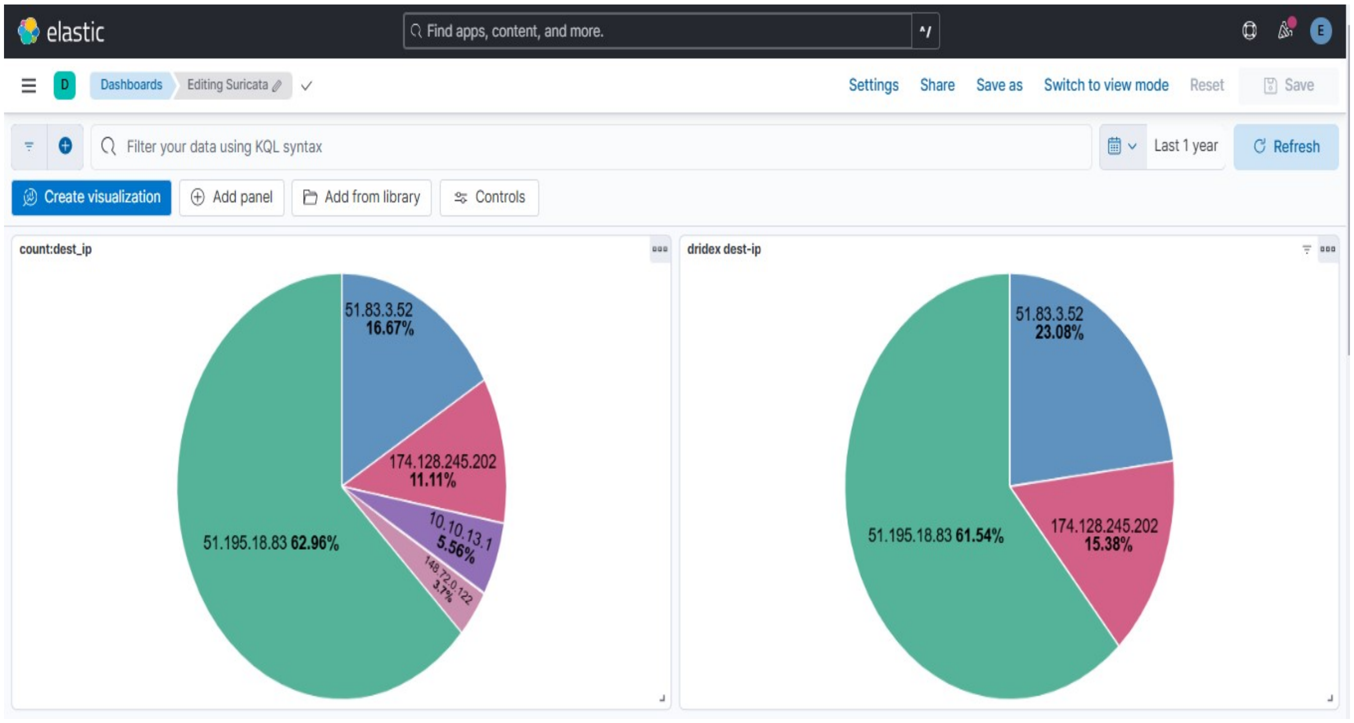
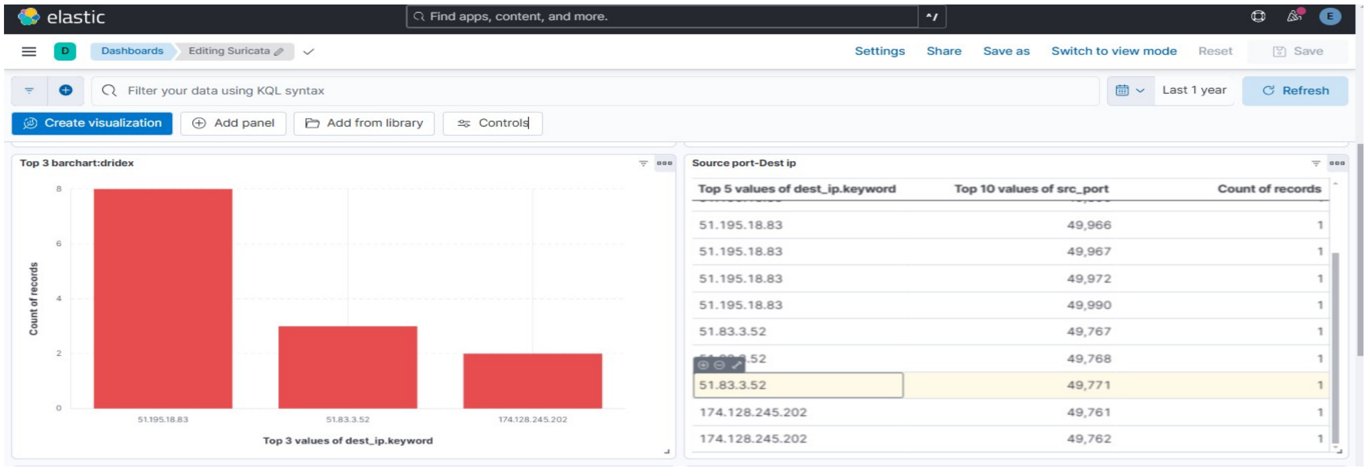


Fig 21 : Dashboard Visualization (Pie Chart)

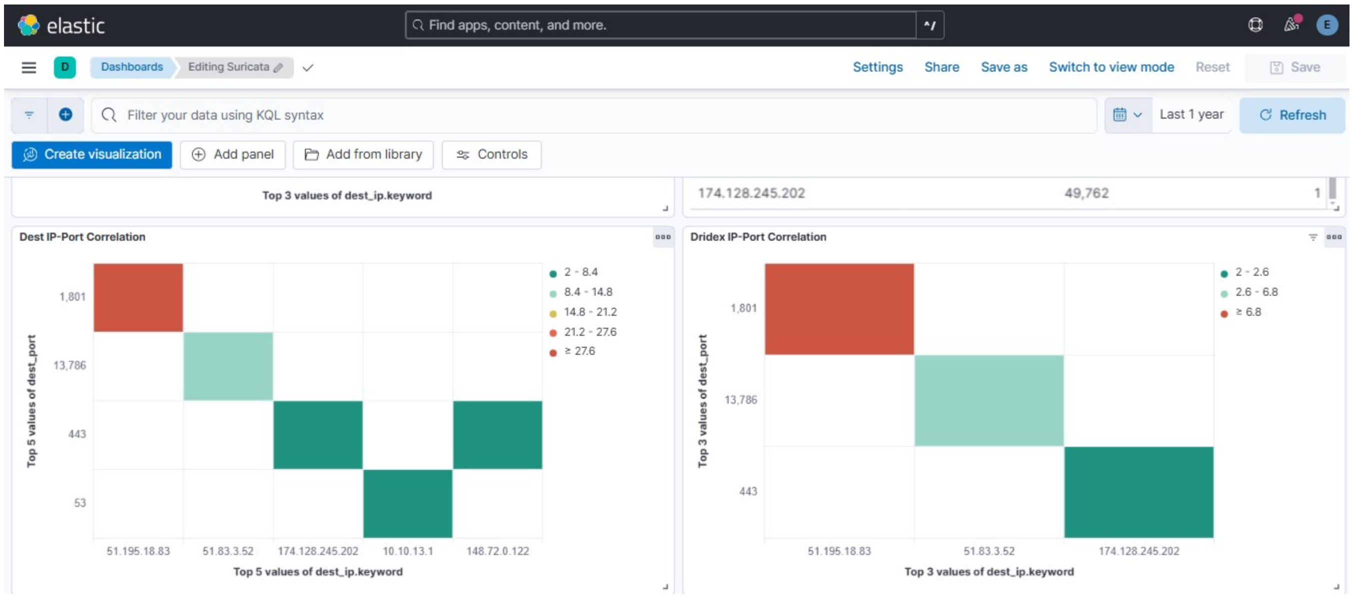
This visualization uses pie charts to provide a high-level summary of network traffic distribution, highlighting key insights such as the largest traffic segment (62.96%) involving IP `51.195.18.83`, indicating a dominant presence or activity. Other significant IPs include `51.83.3.52` (16.67%) and `174.128.245.202` (11.11%). Visual summaries like these help analysts quickly identify dominant sources or destinations of traffic, enabling them to prioritize further investigation into highly active or potentially anomalous IPs.

Dashboard Visualization (Bar Charts and Correlations)



Top 5 values of dest_ip.keyword	Top 10 values of src_port	Count of records
51.195.18.83	49,966	1
51.195.18.83	49,967	1
51.195.18.83	49,972	1
51.195.18.83	49,990	1
51.83.3.52	49,767	1
51.83.3.52	49,768	1
51.83.3.52	49,771	1
174.128.245.202	49,761	1
174.128.245.202	49,762	1

Fig22: **Dashboard Visualization (Bar Charts)**



174.128.245.202	49,762	1

Fig 23: **Dashboard Visualization (Correlations)**

This dashboard improves visual analysis through bar charts and correlation matrices, providing greater insights into network traffic. Bar charts showcase the occurrence of particular destination IPs and ports, for instance, 51.195.18.83, which shows the maximum activity, offering immediate insight into possibly suspicious trends. Correlation matrices illustrate the connections between destination IPs and ports, highlighting traffic flow trends that could suggest lateral movement or interaction among compromised hosts. Collectively, these tools aid in prioritizing IPs or ports that need additional scrutiny, based on traffic levels and noted correlations.

5.7 Summary

The Hive, Cortex, OpenVAS

Identification, analysis, and mitigation of security risks became easier with the integration of The Hive, Cortex, and OpenVAS, which created a thorough incident response and vulnerability management framework. Effective prioritizing of high-risk incidents was made possible by the Hive, which acted as the backbone and enabled centralized incident management through case creation, log imports (JSON/CSV), and task assignments with severity tracking.

Using analyzers like PassiveDNS and VirusTotal, Cortex smoothly connected with The Hive to automate the examination of observables like domains, IPs, and file hashes, providing instantaneous, actionable insights right within the UI.
By conducting vulnerability scans on IPs, web apps like DVWA, and network devices, OpenVAS enhanced this configuration and produced comprehensive reports that categorized vulnerabilities according to their level of severity. Effective mitigation measures were guided by these reports, which highlighted critical vulnerabilities including unpatched software and poor authentication. By integrating automated threat analysis, centralized case recording, and thorough vulnerability reporting, this workflow expedited incident response, greatly cutting down on reaction times and improving network security in general.

Suricata, Zeek, RITA

Summary:

Malware and network traffic were thoroughly analyzed using Suricata, Zeek, and RITA. Together, these technologies were able to examine PCAP files, find malware, and spot unusual network activities.

The Dridex PCAP file was analyzed using Suricata, with an emphasis on identifying Indicators of

Compromise (IoCs) such strange communication patterns and malicious payload signatures. Because of Suricata's capacity to produce comprehensive logs (fast.log), it was possible to identify crucial actions linked to the Dridex virus and validate its existence in network traffic.

Zeek produced thorough logs from the Specula PCAP file, which allowed for a deeper comprehension of network events. Important logs like notice.log (alerts), dns.log (DNS queries), and conn.log (connection details) revealed certain trends and questionable activity. These logs provided detailed information on the traffic, such as odd connection statuses and unexpected DNS results.

Zeek logs were subsequently ingested using **RITA** (Real Intelligence Threat Analytics), which produced a structured database for additional examination. RITA gave useful insights into network anomalies by spotting behavioral patterns like beaconing traffic, lateral movement, or other questionable activity. A thorough process for malware identification and network analysis was guaranteed by the combination of Suricata, Zeek, and RITA.

Outcome:

High-precision malware detection and network anomaly identification were made possible by this procedure. It illustrated how layered technologies could cooperate to efficiently analyze traffic and offer useful information for network defense.

ELK Stack and OWASP ZAP

Summary:

A strong framework for centralized log management and web application vulnerability assessment was developed by combining the ELK Stack with OWASP ZAP. This method covered web application security as well as network activity analysis.

The ELK Stack was essential to the analysis and visualization of the logs produced by Suricata. Kibana dashboards were used to see the logs, Logstash was used to parse them, and Elasticsearch was used to index them. Real-time insights into network traffic patterns were provided by these dashboards, which also highlighted irregularities such recurring connections to dubious IP addresses. Visualizations such as pie charts and bar graphs made it simpler to identify trends and conduct effective incident investigations.

Mutillidae, a purposefully weak web application, was scanned using **OWASP ZAP**. In order to intercept and analyze HTTP requests, FoxyProxy was configured in Firefox. The application's hidden files and URLs were discovered by the ZAP crawling and spidering features. In thorough reports, vulnerabilities including SQL Injection and Cross-Site Scripting (XSS) were found and ranked according to their seriousness. These studies offered practical suggestions for reducing the vulnerabilities.

Result:
A thorough approach to network and application security was made possible by the combination of OWASP ZAP and ELK Stack. ZAP guaranteed a comprehensive evaluation of web application vulnerabilities, whereas ELK expedited the examination of Suricata data. When combined, they improved security posture and visibility.

Zeek and Suricata

Summary:

Zeek and Suricata were used to do thorough network traffic analysis, integrating offline and real time capabilities to efficiently identify threats and anomalies.
Pre-captured PCAP files were processed using Zeek (Offline), which produced comprehensive logs including notice.log (alerts), dns.log (DNS activity), and conn.log (connection details). These logs highlighted odd activity that might point to malicious activity and offered insightful information about connection patterns and DNS requests.
Suricata (Real-Time): Suricata recorded network traffic in real time and stored it for quick analysis. Proactive reactions were made possible by the created logs (fast.log), which revealed persistent threats and anomalies. Its real-time features showed how dangers may be identified as they materialized.

Zeek Online Tool: Pre-captured PCAP data were analyzed using Zeek's online version. It provided instant insights into source and destination connections, services utilized, and communication patterns by parsing network traffic into logs and visualizing the information.
Result:

Zeek and Suricata together demonstrated a flexible network analysis architecture that could handle both live and previously recorded data. A strong and comprehensive strategy for keeping an eye on and safeguarding network activity was guaranteed by this procedure.

CHAPTER-6:PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

The Hive, Cortex, OpenVAS

A simplified procedure for handling security incidents and vulnerabilities was demonstrated by the integration of The Hive, Cortex, and OpenVAS. Cortex automated the threat analysis process, OpenVAS assisted with network security by identifying and classifying vulnerabilities, and case management was carried out centrally using the Hive. Together, these tools significantly minimize response times and manual labor, allowing security teams to concentrate on and address the most critical concerns.

Suricata, Zeek, RITA

We were able to create an integrated pipeline that not only analyzed network traffic but also identified malware and behavioral abnormalities by combining Suricata, Zeek, and RITA. Zeek had in-depth opinions on network protocols and suspicious behavior, whereas Suricata's enormous log generation indicated Indicators of Compromise (IoC). RITA further enhanced the procedure by identifying patterns such as beaconend lateral movement, which aided in distinguishing and reacting to irregularities.

ELK Stack and OWASP ZAP

Project Outcome:

But this project also successfully demonstrated the use of OWASP ZAP for web vulnerability assessment and the ELK stack for centralized log management. By enabling dashboards and charts that allowed anomalies and patterns in network activity to be found, the ELK stack greatly simplified the process of visualizing network activity. In addition, OWASP ZAP provided practical information on how to make online applications susceptible, what kinds of attacks may be created, and potential mitigation techniques.

Zeek and Suricata

Project Outcome:

Zeek and Suricata were integrated to give a network analysis framework that integrated offline and real-time data. Because Zeek's logs were so thorough, they shed light on network behavior, and

Suricata's real-time features made it possible to identify threats right away. The versatility of the Zeek tool in highlighting actionable insights was demonstrated through analysis of pre-captured data.

Relevance:

- Real-time threat identification and reaction: ongoing network surveillance.
- PCAP files are used in post-event analysis to investigate how previous network operations have transpired.
- Delivering instruction and training in cybersecurity as well as practical expertise in threat identification and traffic analysis.

For SOCs, network administrators, and cybersecurity training programs looking to strengthen their network defense approach, this is actually a fantastic workflow.

6.2 Key Implementations Outlines of the System

The Hive, Cortex, and OpenVAS

1. **The Hive & Cortex Integration**:

    a. **Incident Response Workflow**: Cortex was used for automated threat analysis, and The Hive was used for case management. It ensures a seamless transition between security cases and prompt resolution.

    b. **Data Import and Case Creation**: I generated cases in The Hive and assigned Cortex threat analysis tasks using imported JSON/CSV log files of threats and security concerns.

    c. **Automated Analysis**: Cortex analyzed the imported observables to facilitate analysis and provided actionable intelligence on potential exploits and vulnerabilities.

2. **OpenVAS Vulnerability Scanning**:

    a. **Scanning Targets**: IP addresses, network equipment, and programs on platforms such as DVWA and TryHackMe hosts were all checked for vulnerabilities.

    b. **Report Analysis**: Prioritizing and highlighting security problems allowed for the systematic usage of the generated reports to assist minimize vulnerabilities.

    c. **Security Hardening**: Using scan results, network configurations were strengthened, and vulnerabilities were addressed.

Suricata, Zeek, and RITA

1. **Suricata for Malware Detection**:

    a. **Dridex PCAP Analysis**: Dridex PCAP Analysis: Identified malicious activity like Dridex's Indicators of Compromise (IoCs) — including payload signatures and communication patterns — by analyzing network traffic with Suricata and generating the fast.log.

2. **Zeek for Network Logging**:

    a. **Specula PCAP Analysis**: Processed Specula traffic captures with Zeek to produce fine grained network logs (conn.log and dns.log), including protocol usage, connection details, and anomalies.

3. **RITA for Anomaly Detection**:

    a. **Ingesting Logs**: Zeek fed Zeek generated logs into RITA to create a database to analyse anomalies and trends.

    b. **Threat Insights**: RITA also identified potential threats such as unusual beaconing behavior and suspicious connection and alerted through actionable insights. **ELK Stack and OWASP ZAP**

1. **ELK Stack for Centralized Log Analysis**:

    a. **Suricata Integration**: Centralized, managed, and visualized network activities in processed log (fast.log, eve.json) with ELK stack (Elasticsearch, Logstash, Kibana). b. **Threat Detection**: Made pie charts, bar graphs and correlation matrices to highlight patterns, anomalies, and high-risk activities.

   c. **Efficient Filtering**: Utilised advanced filtering specifically targeting certain IP's or event types to identify any malicious traffic or targeted threats.

2. **OWASP ZAP for Web Vulnerability Scanning**:

   a. **Mutillidae Scan**: Crawled and spidered for hidden files and insecure links in a comprehensive security assessment of Mutillidae.

   b. **Active and Passive Scanning**: They identified vulnerable configurations insecure, injection flaws, weak authentication mechanisms.

   c. **Actionable Reports**: It generated detailed vulnerability reports which prioritizes fixes by severity and likelihood of exploitation.

Zeek and Suricata for Network Traffic Analysis

1. **Zeek for Comprehensive Insights**:

   a. **Local Deployment**: Zeek was deployed to capture and analyze live network traffic. In particular, the logs conn.log and dns.log offered a wealth of protocol and connexion data.

   b. **Zeek Online**: With the online version of Zeek I used to analyze pre captured PCAP files, for understanding what kind of traffic was going through it and point out some suspicious activity.

2. **Suricata for Live Network Capture**:

   a. **Real-Time Monitoring**: Live capture of specific protocols and connexions with Suricata. It is shown that encryption issues and anomalies are reflected in the logs (fast.log).

   b. **Detailed Analysis**: Examined the generated logs found for threats like encrypted traffic abnormality, misuse of protocol..

3. **Collaborative Insights**:

   a. **Combined Use of Tools**: Complementary roles of Suricata for real time monitoring and Zeek as an in depth protocol observer for unified network behaviour.

   b. **Visualization & Reporting**: Structured logs that foster enhanced understanding of potential exploits and anomalous patterns for targeted remediation.

6.3 Significant project outcomes

The Hive, Cortex, and OpenVAS

6.3.1. **Enhanced Incident Response Workflow**:

a. Integrated The Hive and Cortex for automated threat analysis, cutting a massive amount of time out of detecting and responding to security incidents.

b. This drove inefficiencies out of the case management process by establishing structured log analysis and root cause resolution security threat analysis processes.

c. It served as an enabler to correlate security events and observables with the actionable intelligence to remediate faster.

6.3.2. **Vulnerability Detection and Mitigation**:

a. Critical vulnerabilities in network devices, applications and IP hosts – including DVWA and TryHackMe lab – were found during the OpenVAS scans.

b. Served as a prioritized list of vulnerabilities that put the most rules at the highest levels of severity so their mitigation will have the most focused effect in securing these high-risk areas.

c. It helped secure the overall network security posture by patching items, hardening the configurations and closing down vulnerabilities mentioned in reports.

Suricata, Zeek, and RITA

1. **Malware Activity Detection**:

a. Through the Suricata logs we successfully detected Dridex malware activity using Dridex IoCs such as malicious payload signatures and communication patterns. b. It was shown to be able to sense and respond in real time to traffic activity suspicious of malware infections.

2. **In-Depth Network Traffic Analysis**:

a. The logs from Zeek describe a great level of detail about protocol usage, DNS queries, and connection detail which gave us a better visibility of what was happening on the network.

b. One such use case was to identify any suspicious trends within Specula PCAP traffic, such as beaconing and unauthorized connections, leveraging RITA.

3. **Actionable Threat Intelligence**:

a. Using RITA I generated a comprehensive database of anomalies and behavioural patterns that offered a strong foundation for proactive threat hunting.

b. Understanding of network traffic anomalies gives better understanding of network traffic anomalies to respond to them preemptively.

ELK Stack and OWASP ZAP

1. **Centralized Log Management and Visualization**:

a. I processed large volumes of Suricata logs in ELK stack and transformed raw data to visual insights in form of dashboards and correlation charts..

b. Through advanced filtering and analysis with Kibana, were able to detect high risk IPs, abnormal traffic patterns and event anomalies.

c. It allowed faster decision making by looking at critical network events in an easily interpreted form.

2. **Web Application Security Enhancement**:

a. Mutillidae app was found to suffer key vulnerabilities by OWASP ZAP, I.E. Hidden files, insecure configurations, and weak authentication mechanisms.

b. Led to the reporting and prioritising recommendations for a more secure and hardened web application environment provided detailed reports.

c. Gave the active and passive scanning for identifying and mitigating web vulnerabilities.

Zeek and Suricata for Network Traffic Analysis

1. **Comprehensive Network Monitoring**:

   a. A detailed breakdown of the network traffic was provided by Zeek logs that consisted of protocols, connection states and potentially anomalous activity.

   b. Through using con.log, dns.log, and other logs, I identified and categorized network behaviors to understand what behaviors normal and what behaviors were weren't.

2. **Real-Time Traffic Analysis with Suricata**:

   a. With Suricata, real time traffic was captured and analyzed for encryption errors and unusual traffic signatures.

   b. Proactively identified potential threats by live monitoring of and logging.

3. **Synergized Tool Usage**:

   a. Combined real time detection with in depth traffic analysis and demonstrated the complementary use of Suricata and Zeek.

   b. Used the Zeek Online Tool to leverage user responsibility of the pre-captured PCAP files, to extract actionable insights of potential threats and anomalies.

4. **Improved Security Posture**:

   a. It provided actionable insights into what network behavior looks like and as a result, these insights helped develop and implement better monitoring and anomaly detection strategies.

   b. It improved the organization's capacity to catch, analyze, and react in a proactive way to network based threats.

6.4 Project applicability on Real-world applications

The Hive, Cortex, and OpenVAS

1. **Incident Response and Threat Management**:

   a. The Hive and Cortex combine to offer a real-world security incident management solution. Enterprises can adopt this workflow to:

i. Automate threat analysis to streamline incident response.

ii. Get quick insights to log events and take faster action during critical security breaches.

b. Applicable for SOC's to handle daily security incidents seamlessly.

2. **Vulnerability Assessment and Risk Mitigation**:

a. OpenVAS is a critical tool for identifying and prioritizing vulnerabilities in enterprise networks, enabling:

i. Vulnerability scans to secure endpoints, servers, and applications on a regular basis.

ii. Aiding reporting with detailed reporting that aids in things like GDP, HIPAA, and PCI DSS.

b. Applicable by organizations that need to implement ongoing vulnerability management to keep a tight security posture.

Suricata, Zeek, and RITA

1. **Advanced Threat Detection**:

a. In enterprise environments one of the tools used to detect sophisticated malware such as Dridex is Surcata, Zeek or Corelight. Use cases include::

i. Monitoring real-time network traffic to identify indicators of compromise (IoCs).

ii. Investigating anomalies and suspicious behaviors in internal networks.

b. Applicable for industries dealing with targeted attacks, such as financial institutions, government organizations, and healthcare.

2. **Proactive Threat Hunting**:

a. The integration of **RITA** enables organizations to:

i. Identify patterns of beaconing, lateral movement, or exfiltration.

ii. Create a network anomaly robust database for future threat prediction.

b. Application to organisations that desire to pursue proactive security strategies to prevent cyber incidents from occurring..

ELK Stack and OWASP ZAP

1. **Centralized Log Management and Threat Analysis**:

   a. The **ELK stack** is widely used for:

i. Log Aggregation and visualisation in Enterprise Environment.

   ii. Giving to those businesses of all sizes that need efficient log management and visualisation to optimise their IT security.

   b. Applicable for businesses of all sizes that need efficient log management and visualization to optimize their IT security.

2. **Web Application Security**:

   a. **OWASP ZAP** is an essential tool for securing web applications by:

   i. Detecting vulnerabilities such as SQL injection, cross site scripting (XSS), and with insecure configuration settings.

ii. Solving problems with security in DevOps teams' CI/CD pipelines

   b. For organisations with customer facing web applications, that includes e-commerce platforms, SaaS providers, and banking websites.

1. **Network Traffic Analysis for Threat Insights**:

   a. Captured traffic provides valuable insights into:

i. Attack methods of malware delivery, and communication channels..

   ii. Needs to be reinforced areas upon the network defences..

   b. It can be widely applied to cybersecurity training programmes, research labs, and organisations trying to increase their malware detection systems..

Zeek and Suricata for Network Traffic Analysis

1. **Comprehensive Network Security Monitoring**:

   a. **Zeek** and **Suricata** offer powerful capabilities for:

i. Monitoring live traffic for suspicious patterns and anomalies.

   ii. Forensic data that proves past incidents.

b. For businesses, ISPs and critical infrastructure organisations, to ensure network security.

2. **Security Tool Synergy**:

a. The complementary use of **Suricata** and **Zeek** ensures:

i. In detection of both live threats and latent anomalies in pre-captured data. ii. Analyzing heterogeneous network environments, including cloud and hybrid infrastructures, with versatility.

b. Useful for a systems architecture where there are large numbers of network connected devices, all of which may require multiple layers of threat detection.

3. **Training and Education**:

a. The tools are ideal for training cybersecurity professionals to:

i. Develop expertise of traffic analysis and threat detection on networks.

ii. Learn protocols, logs, and security alerts in real world scenarios.

b. Applicable on a wide scale in academic programmes, cybersecurity bootcamps and in professional training courses.

6.5 Inference

The Hive, Cortex, and OpenVAS

6.5.1. **Improved Incident Response Workflow**:

a. The Hive and Cortex integration has shown us how incorporation of automation and case management can make a huge difference when it comes to incident response in actual world security operations. It speeds up two things – first, reducing the manual efforts so it is easier to make decisions quickly in critical events, and second, the real user interface that helps to delivery the data in a real user interface.

b. Vulnerability assessment with OpenVAS highlights its importance to keep your infrastructure secure. Applying to identify and manage risks is highly significant for organisations to protect sensitive information and remain compatible.

2. **Inference**:

a. These tools are critical for enterprise level security, looking at the kind of actionable insights and automated analysis. They are vital as they address growing threats in very interconnected business environments.

b. Finance, healthcare and e-commerce organisations can use these solutions to stay ahead of cyber risks.

6.5.2 Suricata, Zeek, and RITA

6.5.2.1. **Enhanced Threat Hunting**:

a. The live and historical traffic analysis capabilities of Suricata and Zeek allows organisations to see the network's security state on a masterful level. Finally, the integration of the tools with RITA further supplements threat hunting with malicious beaconing activity and other malware behaviours.

b. Fielded case studies demonstrate that proactive monitoring and anomaly detection can prevent advanced persistent threats (APTs).

6.5.2.2. **Inference**:

a. The are perfect for SOCs and red teams who seek to find and mitigate hidden threats. It also means that their application guarantees real time detection and it also offers forensic evidence for deeper investigations.

b. Capabilities that shield against sophisticated cyberattacks can be extremely useful for the critical infrastructure industries, such as government agencies and utilities. **ELK Stack and OWASP ZAP .**

1. **Effective Log Management and Web Security**:

a. By centralising and visualising the logs, the ELK stack helps organisations find their way through the minefield of logs, identify the anomalies and stop the threats quickly. This helps it to ensure compliance with robust logging infrastructure.

b. Vulnerability scanning and detailed reports provided by OWASP ZAP reinforce that secure application development is important so that organisations can avoid breaches that exploit software weaknesses.

2. **Inference**:

a. The ELK stack and OWASP ZAP are a merging of operational efficiency and application security. The real world challenge of handling large amounts of data and secure web interfaces is addressed.

b. Specifically, these tools have helped startups and organisations gradually moving towards DevSecOps practise to achieve end to end security from development to operations.

Zeek and Suricata for Network Traffic Analysis

1. **Multi-Layered Network Defense**:

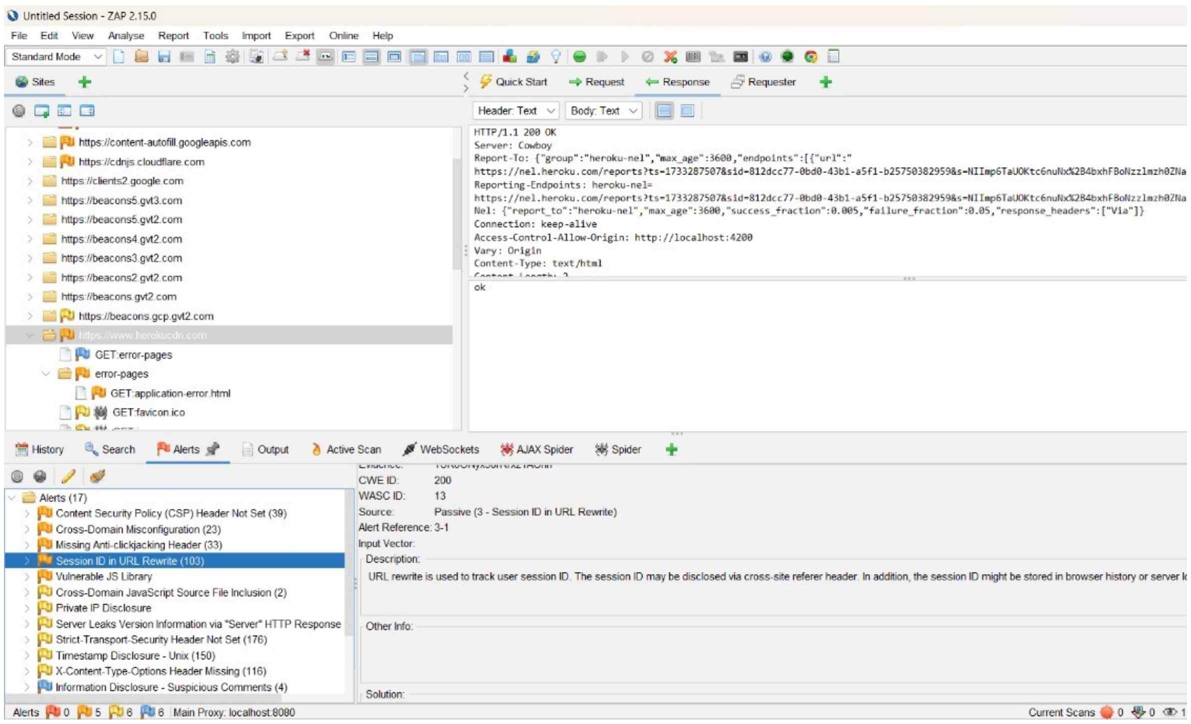
   a. With zeek's comprehensive logs into dumps files and Suricata's instrumentation at line level, we provide a solid base detecting threats at multiple stages. All their capabilities can be used whether in a reactive or proactive security strategy.

   b. Our use of Zeek Online illustrates the diversity of these tools in the adaptability to different environments and encoding languages, and more importantly, their ability to quickly analyse pre captured data.

2. **Inference**:

   a. The flexibility of these tools ensures their applicability in organizations ranging from small to large size. Using these solutions can help businesses implement specific plans for network monitoring and responding to incidents.

   b. Academic institutions and training centers may include these tools in curricula to train future cyber professionals on the latest available techniques.



```
https://content-autofill.googleapis.com
https://cdnjs.cloudflare.com
https://clients2.google.com
https://beacons5.gvt3.com
https://beacons5.gvt2.com
https://beacons4.gvt2.com
https://beacons3.gvt2.com
https://beacons2.gvt2.com
https://beacons.gvt2.com
https://beacons.gcp.gvt2.com
https://www.herokucdn.com
    GET:error-pages
    error-pages
        GET:application-error.html
    GET:favicon.ico
```

```
HTTP/1.1 200 OK
Server: Cowboy
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"
https://nel.heroku.com/reports?ts=1733287507&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=NIImp6TaUOKtc6nuNx%2B4bxhFBoNzzlmzh0ZNa
Reporting-Endpoints: heroku-nel=
https://nel.heroku.com/reports?ts=1733287507&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=NIImp6TaUOKtc6nuNx%2B4bxhFBoNzzlmzh0ZNa
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: http://localhost:4200
Vary: Origin
Content-Type: text/html

ok
```

```
Alerts (17)
  Content Security Policy (CSP) Header Not Set (39)
  Cross-Domain Misconfiguration (23)
  Missing Anti-clickjacking Header (33)
  Session ID in URL Rewrite (103)
  Vulnerable JS Library
  Cross-Domain JavaScript Source File Inclusion (2)
  Private IP Disclosure
  Server Leaks Version Information via "Server" HTTP Response
  Strict-Transport-Security Header Not Set (176)
  Timestamp Disclosure - Unix (150)
  X-Content-Type-Options Header Missing (116)
  Information Disclosure - Suspicious Comments (4)
```

```
CWE ID:          200
WASC ID:         13
Source:          Passive (3 - Session ID in URL Rewrite)
Alert Reference: 3-1
Input Vector:
Description:
 URL rewrite is used to track user session ID. The session ID may be disclosed via cross-site referer header. In addition, the session ID might be stored in browser history or server l
Other Info:
Solution:
```

```
Alerts 0 5 6 6  Main Proxy: localhost:8080          Current Scans 0 0 1
```

Fig24:OWASP ZAP Result

CHAPTER 7: CONCLUSIONS AND RECOMMENDATION

7.1 Outline

The results of this project show that a modular cybersecurity system can accurately and timely detect, analyze, and mitigate threats. The solution integrates tools for network monitoring, traffic analysis, incident response and vulnerability scanning, enabling solution of critical security requirements.

7.2 Limitation/Constraints of the System

· Scalability: In the large scale network environments, the system may not address the performance bottleneck.

· Tool Dependencies: Finding that there is a reliance on open source tools inherently puts you at risk of having compatibility and maintenance problems.

· Advanced Threats: Further, the system may require additional features to detect advanced persistent threat (APTs).

7.3 Future Enhancements

· Machine Learning Integration: Implement advanced threat prediction and anomaly detection models.

· Improved Scalability: Optimize the system for deployment in enterprise-scale networks. · Additional Tools: Include tools for endpoint detection and response (EDR) to cover broader attack surfaces.

· Automated Playbooks: Enhance incident response capabilities with pre-defined, automated workflows.

7.4 Inference

The project demonstrates a significant step toward building an efficient and cost-effective cybersecurity solution. It highlights the potential of open-source tools in addressing real-world security challenges. With targeted improvements, the system can serve as a scalable and adaptable security framework.

REFERENCES

1. Zeek_Detailed: https://docs.zeek.org/en/master/
2. Zeek: https://github.com/zeek/zeek
3. OpenVAS :https://greenbone.github.io/docs/latest/
4. ELK: https://www.elastic.co/docs
5. Suricata: https://suricata.io/documentation/
6. The Hive: https://github.com/TheHive-Project
7. The Hive_Detailed: https://docs.thehive-project.org/cortex/
8. OWASP ZAP https://www.zaproxy.org/docs/
9. Certification:https://www.coursera.org/account/accomplishments/certificate/9WRDGMM65B8U
10. Certification_Udemy:https://www.udemy.com/share/1065AI3@0kDIzIo_JSDtrTScVm1dxj0tPeG61TyF__3tWvNeSTD7zKFEfaksD5NGbxWbZk4VZQ==/