YENEPOYA INSTITUTE OF ARTS, SCIENCE, COMMERCE AND MANAGEMENT

YENEPOYA (DEEMED TO BE UNIVERSITY)

BALMATTA, MANGALORE

FINAL PROJECT REPORT ON

"ADVANCED THREAT SIMULATION IN A SOC ENVIRONMENT"

SUBMITTED BY

RIYA FATHIMA

22BDACC278

GUIDED BY: MS. YASHWINI

INDUSTRY MENTOR: SHASHANK

TABLE OF CONTENTS

LIST OF IMAGES

1. INTRODUCTION

1.1. OVERVIEW OF THE PROJECT

This project implements a real-time network monitoring and threat detection system focused on identifying Advanced Persistent Threats (APTs) using the ELK Stack — comprising Elasticsearch, Logstash, Kibana, and Filebeat, integrated with Suricata IDS. The system captures, processes, analyzes, and visualizes network traffic and alerts, enabling proactive security monitoring.

1.2. OBJECTIVE OF THE PROJECT

. The core objective is to:

• Detect malicious traffic in real time using Suricata.

• Visualize threat data using interactive dashboards in Kibana.

• Demonstrate how open-source tools can be combined to create a scalable, real-time security monitoring system.

1.3. PROJECT CATEGORY

This project falls under the category of Cybersecurity, Network Security Monitoring Intrusion Detection Systems (IDS), Security Information and Event Management (SIEM).

1.4. TOOLS AND PLATFORM TO BE USED

Kali Linux: Used to simulate real-world attacks.

Ubuntu: Host environment for Filebeat, Logstash, and Elasticsearch.

Suricata: IDS engine that generates security alerts from network traffic.

Filebeat: Collects and forwards Suricata logs to Logstash.

Logstash: Parses and formats the logs before indexing them into Elasticsearch.

Elasticsearch: stores the structured data in an index.

Kibana: Visualization layer to create dashboards from Elasticsearch data.

1.5. OVERVIEW OF TECHNOLOGIES USED

This project integrates various technologies across threat detection and visualization:

1.5.1 Hardware Requiremeents

Intel Core i7 processor (or equivalent)

Minimum 16GB RAM

256GB SSD storage

Full HD display (1920×1080 resolution)

1.5.2 Software Requirements

OS Ubuntu 20.4 LTS

Suricata v6.x

Filebeat v8.x

Docker

Docker-compose

Kali Linux

Browser for Kibana Access

1.6. STRUCTURE OF THE PROGRAM

The program follows a structured data pipeline:

Suricata generates logs for each detected anomaly or threat.

Filebeat reads logs from Suricata's output directory and forwards them to Logstash.

Logstash parses and transforms logs into structured format.

Elasticsearch stores this structured data in an index.

Kibana queries Elasticsearch to generate visualizations:

- Pie charts for attack types
- Bar graphs for source/destination IP counts

1.7. STATEMENT OF THE PROBLEM

With the growing sophistication of cyber threats, especially Advanced Persistent Threats (APTs), organizations need real-time visibility into their network activity to detect anomalies and malicious behavior. Traditional log analysis is time-consuming and inefficient.

This project addresses the need for a centralized, real-time monitoring platform using open-source tools to enhance network security, visibility, and threat response.

2. LITERATURE REVIEW

In today's digital era, the increasing sophistication of cyber threats demands equally sophisticated defense mechanisms. The conventional perimeter-based security approach is no longer sufficient, especially in environments where zero-day attacks, advanced persistent threats (APTs), and polymorphic malware are constantly evolving. This has led to a significant shift in cybersecurity paradigms—from reactive models to proactive and predictive detection systems that focus on real-time traffic analysis and threat intelligence. This literature review explores the foundation, trends, and tools relevant to the implementation of modern network intrusion detection systems (NIDS), particularly focusing on the open-source ecosystem comprising Suricata, Filebeat, and the ELK stack.

Network Intrusion Detection Systems (NIDS)

NIDS are crucial for monitoring network traffic and identifying unauthorized or malicious activity. Traditional intrusion detection systems (IDS) often relied on signature-based detection methods, which, while effective against known threats, fall short against novel or obfuscated attacks. Over time, hybrid models integrating signature-based and anomaly-based techniques emerged to enhance detection capabilities. According to research by Axelsson (2000), the importance of minimizing false positives in IDS cannot be overstated, as they contribute significantly to alert fatigue and undermine trust in automated systems.

Suricata: The Next-Gen Open-Source NIDS

Suricata is a high-performance, open-source network threat detection engine developed by the Open Information Security Foundation (OISF). Unlike earlier tools such as Snort, Suricata

supports multi-threading, protocol detection, file extraction, and TLS certificate logging. Its capability to process network traffic at high throughput while concurrently performing deep packet inspection (DPI) makes it highly scalable and efficient for modern network infrastructures.

Researchers and practitioners praise Suricata's built-in support for rulesets like Emerging Threats, which continuously update signatures based on real-world attack data. In recent academic evaluations (Kreibich et al., 2021), Suricata has been highlighted as one of the most flexible platforms for real-time network analysis and forensic inspection.

Log Aggregation with Filebeat

In any cybersecurity architecture, centralizing log collection is pivotal for both visibility and correlation. Filebeat, a lightweight shipper developed by Elastic, is designed to forward and centralize log data. It integrates seamlessly with both Suricata and the ELK stack. Filebeat's modular design allows users to enable preconfigured modules for common log types— including Suricata. This eliminates the complexity of building custom pipelines and ensures that logs are parsed and structured correctly before ingestion into Logstash or Elasticsearch.

The value of Filebeat lies in its efficiency and fault tolerance. With backpressure-sensitive queueing, low memory usage, and an ability to handle large volumes of data without data loss, Filebeat has become a cornerstone in modern security monitoring pipelines.

The ELK Stack: Elasticsearch, Logstash, Kibana

The ELK stack—Elasticsearch for storage and search, Logstash for parsing and transformation, and Kibana for visualization—has revolutionized how organizations monitor, analyze, and respond to network activity. Originally used for operational log analytics, ELK has grown into a popular platform for security information and event management (SIEM).

Elasticsearch's ability to index, search, and retrieve vast amounts of log data in near real-time enables instant querying of security events. Logstash, acting as the data processing pipeline, applies filters and enriches logs to add context—critical in incident response. Kibana then enables security teams to visualize this data through dashboards, offering insights into malicious IP addresses, event trends,

protocol usage, and alert signatures.

A comprehensive study by the SANS Institute (2022) emphasized the relevance of ELK in open-source SIEM solutions, particularly for small-to-medium enterprises (SMEs) seeking cost-effective but powerful analytics platforms.

Attack Simulation and Threat Modeling

To test the robustness of threat detection systems, simulated attacks are essential. Kali Linux, a Debian-derived penetration testing distribution, provides a suite of tools for ethical hacking. From network scanning to payload delivery, Kali allows controlled simulation of real-world attack vectors, enabling the calibration of Suricata's detection capabilities and the verification of alert accuracy in Filebeat and Kibana dashboard.

Numerous case studies (e.g., MITRE ATT&CK evaluations) validate the importance of red teaming and adversary emulation in refining defense strategies. Simulating attacks not only demonstrates system efficacy but also highlights areas for improvement in rule tuning, false- positive reduction, and incident triage.

Related Works and Gaps

Several open-source projects and academic prototypes have aimed to combine NIDS with ELK-based visualizations. However, many of these focus narrowly on either performance benchmarking or single-node setups without addressing the full operational lifecycle—from attack generation to alert visualization. Moreover, few projects emphasize usability from a beginner-friendly or educational perspective.

This project bridges that gap by offering an end-to-end implementation that combines Suricata, Filebeat, and ELK stack using Docker for simplicity. It prioritizes usability, real- time performance, and extensibility. Furthermore, it offers customizable dashboards in Kibana that can be adapted to diverse threat intelligence scenario.

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1. INTRODUCTION

3.1.1. Purpose

The purpose of this document is to define the software requirements involved in designing and deploying a virtual SOC environment where advanced network-based threats can be simulated, detected, and analyzed.

3.1.2. Scope of the Project

The scope of this project revolves around the creation and deployment of a virtual Security Operations Center (SOC) environment capable of detecting, analyzing, and visualizing advanced cyber threats through simulated attack scenarios. The key objective is to provide a hands-on, realistic, and controlled infrastructure where security incidents can be generated, captured, and monitored using open-source tools.

3.1.3. Intended Audience and Reading Suggestions

This document is intended for:

- Project supervisors and evaluators
- Cybersecurity researchers
- Stakeholders seeking insights into project goals and functionalities

Readers are advised to have a basic understanding of business Cybersecurity and cloud technologies.

3.1.4 Definitions, Acronyms, and Abbreviations

The following terms and abbreviations are used throughout this document:
SOC (Security Operations Center): A centralized unit that monitors, detects, responds to, and mitigates cybersecurity threats in an organization's infrastructure.
SIEM (Security Information and Event Management): A security solution that

aggregates and analyzes log data from across an organization to identify and respond to potential threats.

IDS (Intrusion Detection System): A system that monitors network or system activities for malicious actions or policy violations.

ELK Stack (Elasticsearch, Logstash, Kibana): A collection of open-source tools for searching, analyzing, and visualizing log data in real time.

Suricata: An open-source network threat detection engine that functions as an IDS, IPS, and network security monitoring tool.

Filebeat: A lightweight shipper that forwards and centralizes log data to Logstash or Elasticsearch.

Kibana: A data visualization and exploration tool used to view and analyze logs and metrics stored in Elasticsearch.

Logstash: A data processing pipeline that ingests data from multiple sources, transforms it, and sends it to a "stash" like Elasticsearch.

Dashboard: A graphical interface that displays essential business metrics and key performance indicators in a concise and interactive format.

APT (Advanced Persistent Threat): A prolonged and targeted cyberattack in which an intruder gains access to a network and remains undetected for an extended period.

3.1.5. References

Axelsson, S. (2000). *Intrusion Detection Systems: A Survey and Taxonomy*.
Kreibich, C., et al. (2021). *Evaluation of Modern Intrusion Detection Tools*.
SANS Institute (2022). *Using ELK Stack for Security Monitoring*.
MITRE ATT&CK Evaluations: https://attack.mitre.og/
Official Suricata Documentation: https://suricata.io
Elastic Filebeat and Logstash Docs: https://elastic.co/docs
Docker Containers 101 – Retrieved from https://www.youtube.com/@networkchuck

3.1.6. Overview

The document is structured to first provide a high-level description of the system, followed by specific requirements including functional, performance, and interface details. It concludes with design and deployment considerations.

3.2 OVERALL DESCRIPTION

3.2.1 Product Perspective

This document provides a high-level overview of the cybersecurity system, outlining its primary goals and key components. It covers functional requirements such as threat detection, performance expectations, and user interface needs. The document concludes with design and deployment considerations to ensure secure and efficient system implementation.

3.2.2 Product Features

This project functions as a modular, open-source alternative to commercial network monitoring and SIEM platforms. It is designed to operate as a plug-and-play component within a virtual Security Operations Center (SOC) environment. Each component of the ELK stack, alongside Suricata and Filebeat, contributes to building an integrated, end-to-end solution for real-time network threat detection, analysis, and visualization. Unlike traditional systems, this architecture is containerized using Docker, allowing for portability and scalability across different operating environments.

3.2.3 User Characteristics

Intended users include:

Security Researchers: Require detailed insights into network anomalies and threat behaviors.

Network Administrators: Utilize the platform to monitor traffic and receive real-time alerts.

Students and Educators: Use the platform for cybersecurity education and hands-on learning.

SOC Analysts: Need actionable visualizations and fast querying capabilities.

3.2.4 Operating Environment

Hardware: Intel Core i5, 8–16GB RAM, 256GB SSD, Full HD Display

Software: MacOS, Kali Linux, Ubuntu

3.2.5 Design and Implementation Constraints

Requires elevated privileges for network packet capturing (Suricata) Log processing pipeline must handle high throughput without data loss.

3.2.6 General Constraints

Continuous network activity needed for meaningful detection and requires configuration tuning to reduce false positives.

3.2.7 Assumptions and Dependencies

Assumptions:

Users are assumed to have administrative privileges for setting up containers and services.

All tools used (Suricata, ELK Stack, Kali Linux) are assumed to be open-source and free for educational use

Network interfaces are assumed to be correctly bridged to capture traffic across Docker containers.

Dependencies:

Threat detection rulesets (e.g., Emerging Threats) are assumed to be regularly updated.

Availability of internet access for downloading containers and updates

Docker and Docker Compose must be pre-installed on host machines ELK Stack services must be correctly configured to interoperate.

3.3 SPECIFIC REQUIREMENTS

3.3.1 External Interface Requirements

3.3.1.1 User Interface

Web-based dashboard in Kibana for log visualization.

Filters for timestamp, source IP, destination IP, alert signature.

Pie charts, bar graphs, and line charts for activity monitoring.

3.3.1.2 Hardware Interface

The system operates on standard PC/laptop hardware with internet access. Persistent storage is used through Docker volumes to retain Elasticsearch indices across sessions.

3.3.1.3 Software Interface

Suricata: Analyzes traffic and generates alerts in real time.

Filebeat: Forwards Suricata logs to Logstash.

Logstash: Parses and structures log data.

Elasticsearch: Indexes and stores data for quick search.

Kibana: Visualizes data through interactive dashboards.

Docker & Docker-Compose: Manages containerized components.

Kali Linux: Simulates threats for detection testing.

3.3.2 Functional Requirements

Detect network threats in real time using Suricata.

Forward Suricata-generated logs using Filebeat.

Parse and structure logs via Logstash.

Index and store processed data in Elasticsearch.

Visualize alerts and trends through Kibana dashboards.

Simulate cyber-attacks using Kali Linux for validation.

Ensure Docker containers communicate correctly for data flow.

Allow users to filter and search logs through the Kibana interface.

Maintain rule updates for accurate threat detection.

Generate alerts based on predefined threat signatures.

3.3.3 Performance Requirements

System must detect threats within 2 seconds of occurrence.

Dashboards must reflect new entries in near real-time (within 5 seconds).

Elasticsearch should support query response time < 1 second under normal load.

3.3.4 Design Constraints

The entire system must use open-source tools.

Suricata must have access to bridged network interfaces for accurate monitoring.

Kibana dashboards rely on defined Elasticsearch index patterns.

3.3.5 Other Requirements

Unauthorized access attempts must be logged.

All services must auto-start using Docker Compose.

Detection rules (e.g., Suricata) should be customizable.

4.SYSTEM ANALYSIS AND DESIGN

4.1 INTRODUCTION

This section describes the system architecture, design methodology, and integration strategy used to build the network monitoring solution. The project integrates open-source tools in a modular, containerized environment to detect and visualize threats in real-time.

4.2 METHODOLOGY

The system adopts a modular and containerized architecture using Docker and Docker Compose. Major tools— Logstash, Elasticsearch, and Kibana— are deployed in a container. Logs generated by Suricata are shipped via Filebeat to Logstash, structured and indexed by Elasticsearch, and visualized through Kibana dashboards. This approach ensures portability, scalability, and simplified maintenance.

4.3 DATA FLOW DIAGRAM

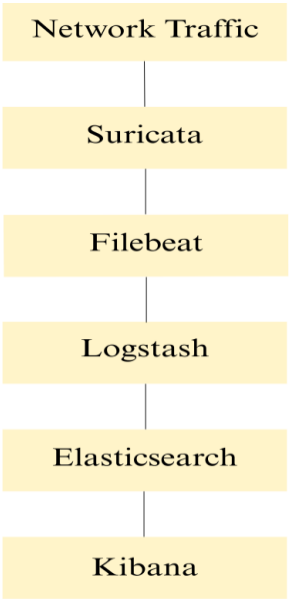


fig 4.1 Data flow Diagram

4.4 SYSTEM ARCHITECTURE

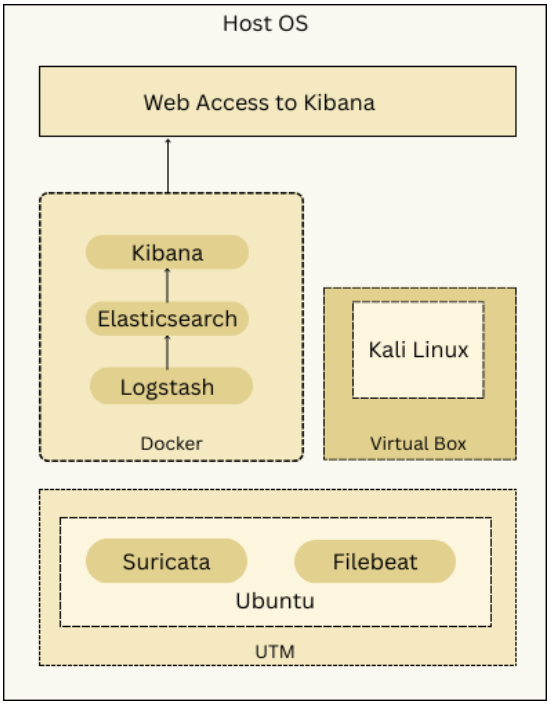


fig.4.2 System Architecture

4.5 SYSTEM DESIGN IMPLEMENTATION

The system is implemented using a modular architecture built entirely on containerized components. This design ensures scalability, flexibility, and ease of replication across different environments. Each module in the architecture plays a distinct role in achieving seamless end-to-end threat detection, log processing, and visualization. Docker and Docker- Compose are used to orchestrate these components efficiently within a virtualized environment on the host system (macOS).

4.5.1 Use Case

Real-time Detection and Visualization of Network-Based Attacks.

Primary Actors:

- SOC Analyst: Monitors real-time alerts and responds to detected threats.
- Attacker (Kali Linux): Simulates attack scenarios such as port scanning, brute force, and DoS.

- System Components: Suricata, Filebeat, Logstash, Elasticsearch, Kibana.

Preconditions:

- Docker container and all the images (Logstash, Elasticsearch, Kibana) are up and running.
- Network bridge configuration allows Suricata to capture traffic from Kali Linux targeting the Ubuntu container.
- Threat rulesets such as Emerging Threats are properly configured in Suricata.

Basic Flow:

1. The attacker launches an attack using Kali Linux targeting the Ubuntu machine.
2. Suricata inspects the network packets in real-time and detects signatures matching known threats.
3. Suricata logs the events in the form of JSON-formatted alert logs.
4. Filebeat, acting as a log shipper, monitors the Suricata log directory and forwards new log entries to Logstash.
5. Logstash applies filtering and transformation pipelines to parse the logs into structured format.
6. Structured data is sent to Elasticsearch, which indexes and stores the data for search and analytics.
7. Kibana queries Elasticsearch to retrieve and visualize alerts on interactive dashboards.
8. SOC Analyst uses Kibana to observe attack patterns, analyze source and destination IPs, and validate the efficacy of detection rules.

Postconditions**:**

- Attack events are successfully logged, parsed, indexed, and visualized.
- SOC Analyst receives actionable insights with timestamped details and attack metadata.

Alternate Flows:

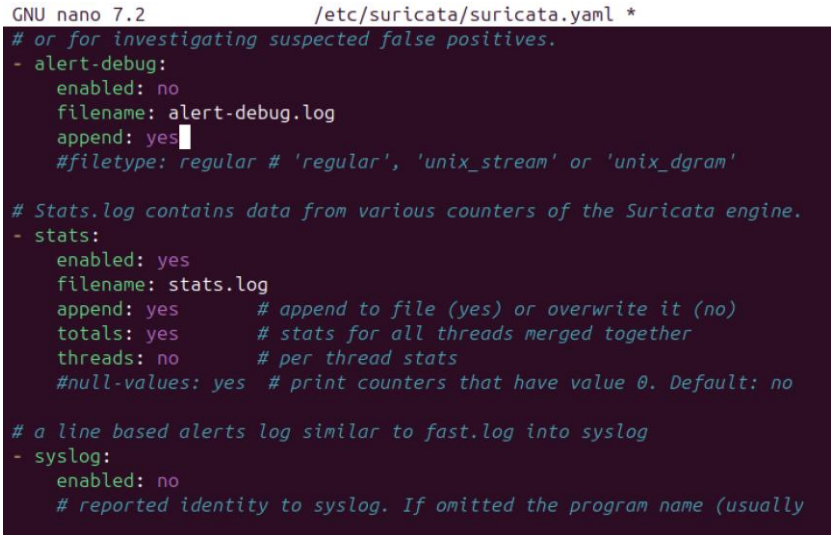
- If the container network is misconfigured, Suricata may fail to detect traffic, and no alerts will be generated.

- If Filebeat fails to forward logs, alerts will not appear in Kibana, indicating pipeline disruption.

4.6 Suricata and Logstash Configuration

Suricata, a high-performance network IDS/IPS engine, is configured to monitor packet traffic and generate JSON-formatted alerts. These alerts include critical metadata related to potential threats and anomalies.

Logstash acts as the intermediary that ingests these Suricata logs, applies parsing filters, and structures the data appropriately for indexing by Elasticsearch. The configuration ensures minimal latency between detection and visualization, forming a reliable data pipeline. Both Suricata and Logstash configurations are designed to be scalable and adaptable, enabling seamless integration into larger security infrastructures if required in the future.



Configuring Suricata in YAML

4.7USER INTERFACE DESIGN

The user interface of the system is designed to ensure clarity, accessibility, and real time insights into network security threats. The UI is primarily delivered through Kibana, a powerful open-source data visualization dashboard that works seamlessly with Elasticsearch. By integrating Kibana with the output from Suricata and Logstash,

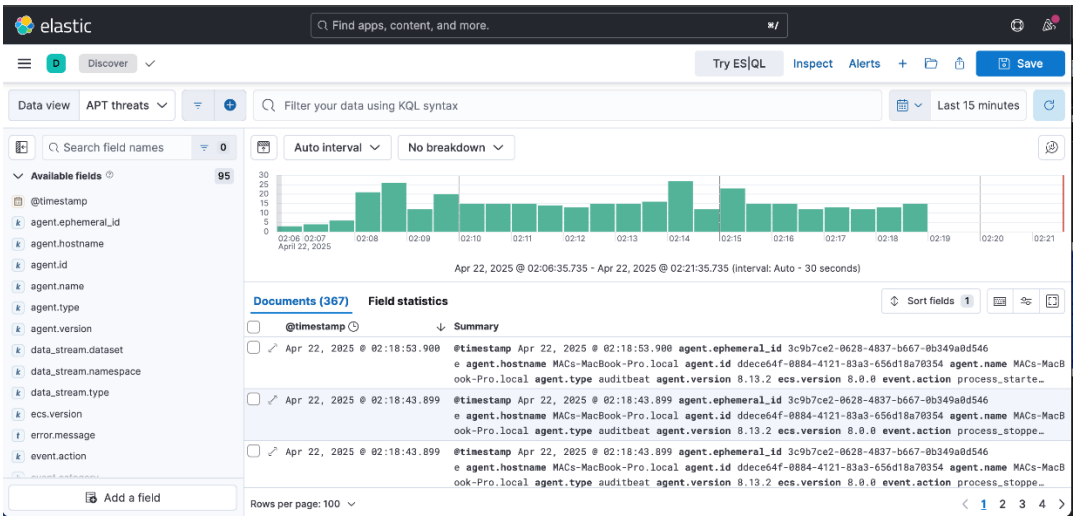
the system enables users to visually analyze alerts, monitor network traffic, and detect anomalies with ease.

The interface supports real-time monitoring, customizable dashboards, and interactive filtering, making it intuitive even for users with minimal background in cybersecurity. This approach minimizes complexity while maximizing the visibility of potential threats, thus playing a vital role in improving response time and system transparency.

4.7.1 Kibana Dashboard

The Kibana Dashboard serves as the visual front-end for the ELK stack. It transforms raw data and alerts generated by Suricata into dynamic, human-readable insights. Users can monitor a wide array of metrics, such as source IPs, destination ports, alert signatures, and timestamps in real-time.

This dashboard is accessed via web browser on the host machine allowing interaction with the containerized data environment without compromising isolation. It is fully customizable, supporting multiple types of visualizations including bar charts, line graphs, data tables and geographic maps. The ability to drill down into specific events empowers users to perform root cause analysis and identify attack patterns effectively.



5.TESTING

5.1 INTRODUCTION

Testing plays a pivotal role in validating the robustness, accuracy, and reliability of the threat detection and visualization system developed using the ELK stack. This phase ensures that all components — from data ingestion to alert generation and dashboard visualization — perform as expected under simulated real-world attack conditions. By methodically testing each module, the integrity of the data flow and the system's responsiveness to network threats can be evaluated and improved.

5.2 Testing Objective

The primary objective of testing is to verify the effectiveness of the ELK-based monitoring solution in identifying and visualizing network threats generated during simulated attacks. It aims to confirm that:
Suricata accurately detects malicious network traffic and generates alerts in real time.
Filebeat successfully forwards these logs to Logstash without data loss or delay.
Logstash correctly parses and structures incoming data for Elasticsearch.
Elasticsearch stores the processed logs efficiently, allowing fast retrieval.
Kibana accurately visualizes the data, enabling users to identify threats through interactive dashboards.
The testing process also ensures that the Dockerized architecture operates seamlessly and that communication between the containers and the host environment remains stable throughout the data pipeline.

5.3 TEST CASES

5.3.1 Network Attack Simulation

Objective:
To simulate real-world cyberattacks from an external attacker machine (Kali Linux) to observe how the system handles and detects malicious traffic.

Test Procedure:

Kali Linux was used to perform various attacks such as port scanning using nmap, brute-force login attempts using hydra, and exploitation using metasploit. The target was an Ubuntu system configured with open services to mimic vulnerabilities. Traffic between the two was monitored by Suricata.

Result:

Suricata successfully detected multiple types of attacks, generating alert logs with accurate metadata like IP addresses, ports, and attack types. These logs were captured in EVE JSON format and forwarded seamlessly to the rest of the ELK pipeline.

5.3.2 Real-time Threat Detection

Objective:

To validate the real-time nature of detection and forwarding in the system using Suricata and Filebeat.

Test Procedure:

- Triggered sequential attack types and monitored the generation of alerts.
- Observed how quickly these logs reached Logstash and were processed.

Result:

The delay between attack execution and alert generation was minimal (less than 2–3 seconds on average). Filebeat efficiently picked up and sent logs to Logstash, where they were parsed and forwarded to Elasticsearch with no observable packet or data loss.

5.3.3 Data visualization in Kibana

Objective:

- To assess the accuracy and clarity of data representation within Kibana dashboards.
- Test Procedure:

- Verified whether logs were indexed by Elasticsearch.
- Opened Kibana from the host MacOS browser and checked for visualizations such as bar graphs, tables, and timelines.
- Searched for attack events using filters like source IP, event type, and timestamps.

Result:

Kibana displayed all relevant events in a structured and interactive dashboard. Alerts were easy to filter and analyze. Visualizations clearly showed spikes during attack periods and gave an intuitive overview of the network behavior, proving its effectiveness for threat monitoring.

6. SYSTEM SECURITY

6.1 INTRODUCTION

System security plays a critical role in the design and deployment of modern network-based applications, especially those involved in monitoring and detecting threats. The primary objective of implementing security measures in this project is to ensure the confidentiality, integrity, and availability of the data processed by the ELK Stack and the associated components. Since this project simulates real-world cyberattacks and collects sensitive threat intelligence, it is vital that the architecture itself remains resilient to tampering, unauthorized access, or data leakage. Each layer of the system—from the Dockerized services to the interfaces and data pipelines—has been reviewed and configured with security in mind.

6.2 SOFTWARE SECURITY

The ELK Stack (Elasticsearch, Logstash, and Kibana) is at the core of this project's threat detection and visualization pipeline. To protect this core, several key security practices have been implemented:

All ELK services run in isolated Docker networks, limiting external exposure and minimizing the attack surface. This ensures that containers cannot be directly accessed from outside without proper routing.

Suricata logs are generated in structured JSON format and only accessible to internal services like Filebeat. No direct access to log files is permitted, ensuring the integrity of raw threat data.

Minimal Host Exposure: Only Kibana is exposed to the host machine (macOS) via the browser for visualization. Even this access is controlled through port binding and user authentication to prevent unauthorized access.

7. CONCLUSION

The project successfully demonstrates an end-to-end simulation of cyberattacks and the corresponding detection pipeline using open-source tools like Suricata and the ELK Stack. Through a controlled setup, threat scenarios were executed from a Kali Linux attacker machine targeting an Ubuntu system, while real-time detection was enabled by Suricata and logs were efficiently processed, indexed, and visualized using Filebeat, Logstash, Elasticsearch, and Kibana respectively. This implementation proves the effectiveness of intrusion detection and log analytics platforms in identifying and analyzing suspicious activities.

The system not only ensures a modular and scalable architecture by leveraging Docker containers, but also provides a clear and interactive visual interface via Kibana, making it user-friendly for security analysts and researchers. With the proper application of secure practices, the architecture stands resilient and adaptable to future improvements in cybersecurity monitoring.

8.FUTURE ENHANCEMENTS

Integration with MITRE ATT&CK Framework: Mapping detected threats to the MITRE ATT&CK matrix would provide a deeper contextual understanding of attacker behavior, tactics, and techniques, thereby helping security teams prioritize threats and implement proactive defense strategies.

Automated Threat Response: Incorporating Security Orchestration, Automation, and Response (SOAR) tools can help in triggering predefined actions in response to specific alerts, enabling faster containment and remediation.

Machine Learning for Anomaly Detection: By integrating ML-based models into the pipeline, the system could detect subtle anomalies or zero-day patterns that traditional signature-based systems might overlook.

These future enhancements would make the system a more robust and intelligent threat detection and response platform, ready to be adopted in enterprise-grade environments.

9. BIBLIOGRAPHY

Bejtlich, R. (2013). *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. No Starch Press.

Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)*. NIST SP 800-94.

Elasticsearch: The Definitive Guide. Clinton Gormley & Zachary Tong. O'Reilly Media.

"Security Information and Event Management (SIEM) – Concepts and Implementation." SANS Institute Whitepaper.

Analysis of APT attack cases targeting domestic companies using Dora RAT (Andariel Group). AhnLab SEcurity intelligence Center Report

Vacca, J. R. (2014). *Computer and Information Security Handbook*. Academic Press.

Ali, A., & Khan, M. (2021). *Using Open-Source IDS Tools to Detect Cyber Threats*. Journal of Cybersecurity and Information Systems.

ELK Stack as a SIEM Tool: Comparative Study and Real-World Applications.International Journal of Computer Applications (IJCA), 2021.

The Elastic Stack Documentation (Elasticsearch, Logstash, Kibana, Beats). Available at: https://www.elastic.co/docs

Docker Documentation. Docker Inc. Available at: https://docs.docker.com

Filebeat Documentation - Lightweight Log Shipper. Elastic. Available at: https:// www.elastic.co/beats/filebeat

Logstash Configuration Reference. Elastic. Available at: https://www.elastic.co/guide/en/ logstash

OWASP Foundation. Security Best Practices. Available at: https://owasp.org

10. APPENDIX

I. Bar chart of the hits