# ****Network Monitoring System Report****

### ****Project Title:**** Network Monitoring System

## Project Team

* **Abhishikth Parry** – Project Lead, System Configuration & Security Analysis
* **Aakash Sivakumar** – Intrusion Detection & Log Management
* **Rajiv Bhardwaj** – Data Visualization
* **Sandeep Kandula** – Performance Monitoring

## ****1. Introduction****

### ****Objective:****

The objective of this project is to establish a **Network Monitoring System** that enables real-time traffic monitoring, intrusion detection, anomaly detection, automated reporting, and data visualization. This system enhances **network security, performance optimization, and proactive threat detection** using industry-standard tools.

## ****2. System Architecture****

## ****3. Tools and Technologies Used****

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Tools Used** | **Functionality** | **Integration Details** |
| **Real-Time Traffic Monitoring** | Suricata, Tcpdump, Wireshark | Captures and inspects live network traffic | Suricata logs traffic, Tcpdump/Wireshark used for deeper analysis |
| **Intrusion Detection System (IDS)** | Suricata | Detects malicious network activity | Suricata logs forwarded to Elasticsearch via Filebeat |
| **Performance Metrics Analysis** | **Prometheus, Grafana, Node Exporter** | Collects and visualizes system/network metrics | Grafana visualizes Prometheus metrics |
| **Anomaly Detection** | **Suricata, Filebeat, Elasticsearch, Kibana** | Detects unusual behavior based on traffic logs | Suricata logs sent to Elasticsearch via Filebeat, visualized in Kibana |
| **Automated Reporting** | **Grafana, Kibana (via Elasticsearch)** | Generates automated reports and alerts | Grafana alerts via webhooks, Kibana alerting being finalized |
| **Data Visualization** | Grafana, Kibana | Creates dashboards for logs and network data | Kibana visualizes Elasticsearch data, Grafana displays Prometheus data |

## ****4. Implementation and Configuration****

### ****4.1 Real-Time Traffic Monitoring (Suricata, Tcpdump, Wireshark)****

#### ****Step 1: Install Suricata****

sudo apt update

sudo apt install -y suricata

#### ****Step 2: Configure Suricata****

sudo nano /etc/suricata/suricata.yaml

Modify the af-packet section to match the correct network interface.

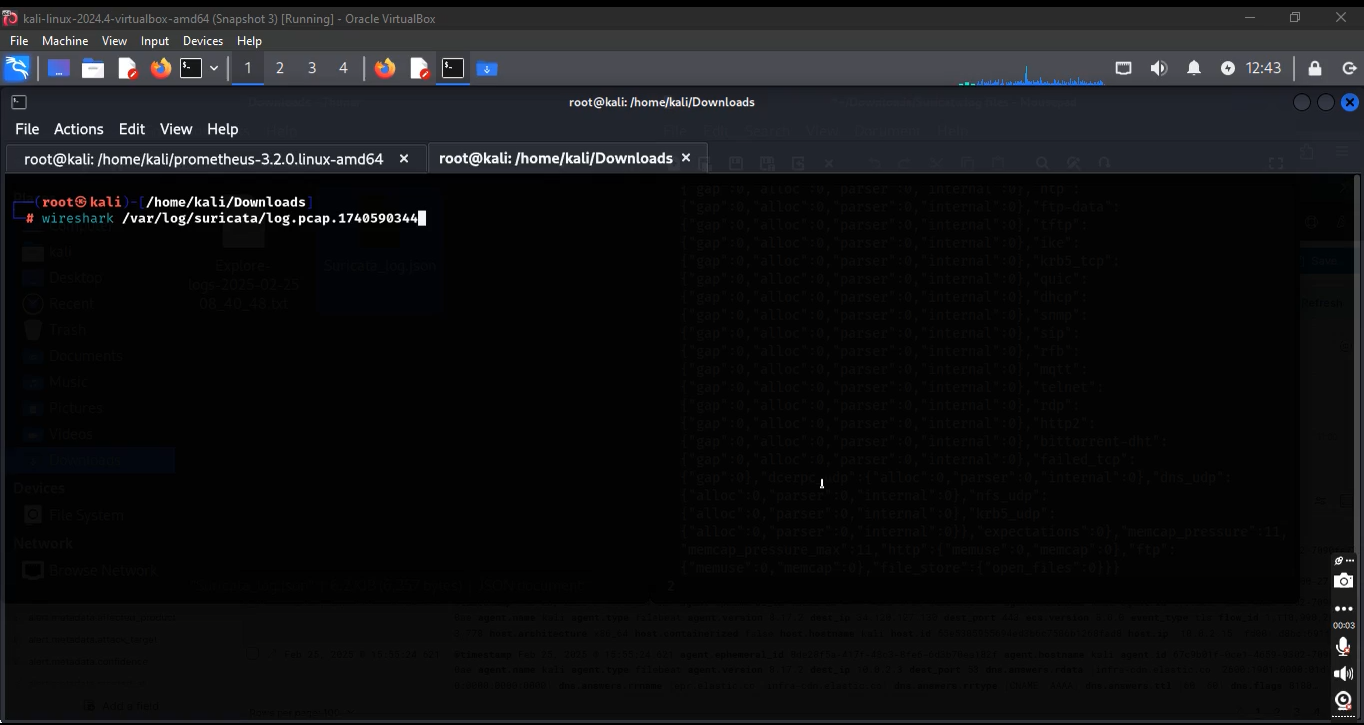
#### ****Step 3: Start Suricata****

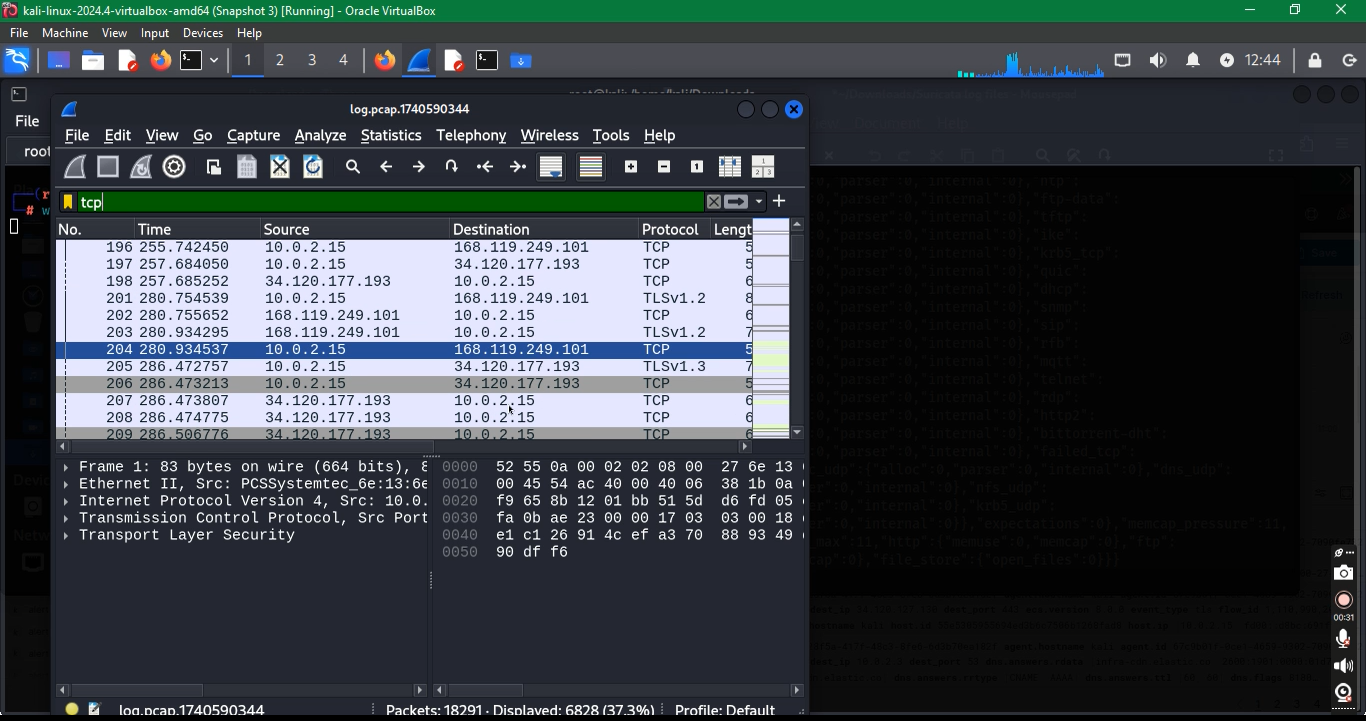
sudo systemctl start suricata

sudo systemctl enable suricata

#### ****Step 4: Verify Logs****

sudo tail -f /var/log/suricata/eve.json





### ****4.2 Log Forwarding with Filebeat****

#### ****Step 1: Install Filebeat****

sudo apt install -y filebeat

#### ****Step 2: Enable Suricata Module****

sudo filebeat modules enable suricata

#### ****Step 3: Configure Filebeat for Elasticsearch Output****

Edit the configuration file:

output.elasticsearch:

hosts: ["http://localhost:9200"]

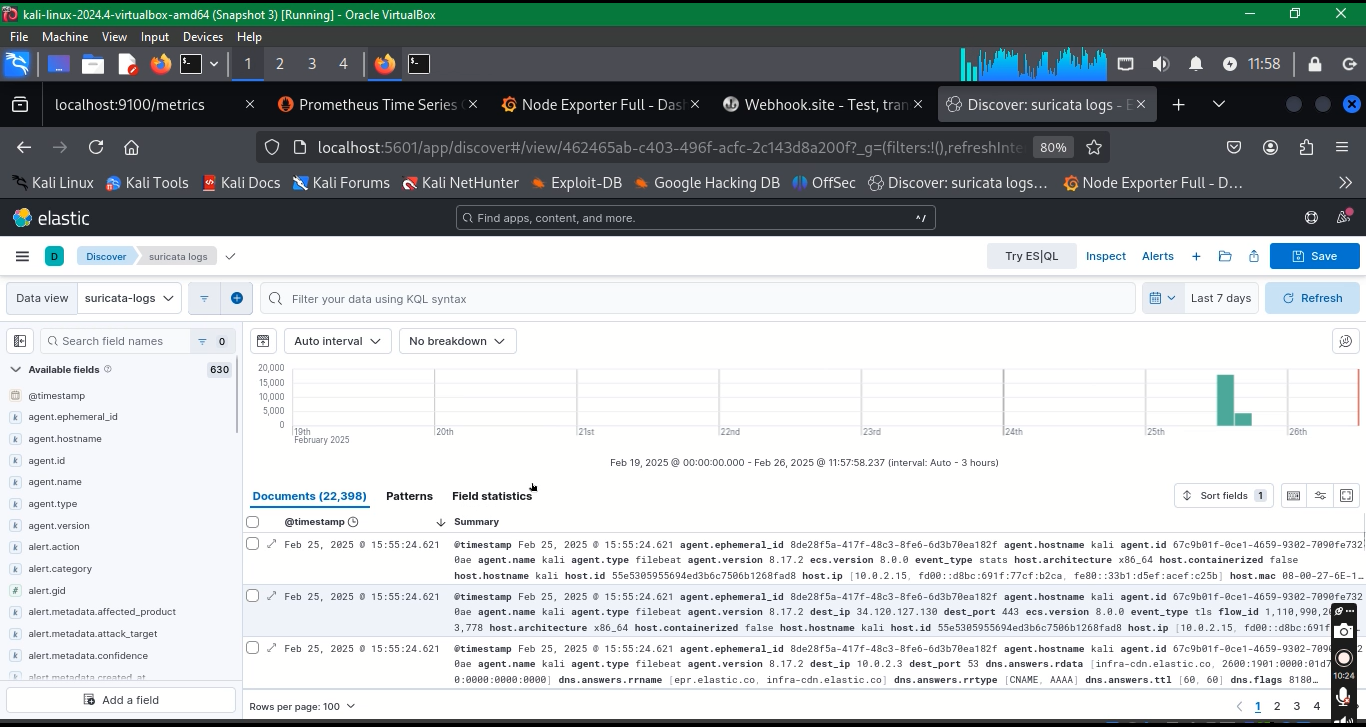
setup.kibana:

host: "http://localhost:5601"

#### ****Step 4: Start Filebeat****

sudo systemctl start filebeat

sudo systemctl enable filebeat



### ****4.3 Log Storage and Visualization (Elasticsearch & Kibana)****

### 

### ****4.4 Performance Metrics Analysis (Prometheus, Node Exporter, Grafana)****

#### ****Step 1: Install Prometheus****

wget https://github.com/prometheus/prometheus/releases/latest/download/prometheus-linux-amd64.tar.gz

tar xvf prometheus-linux-amd64.tar.gz

cd prometheus-linux-amd64

#### ****Step 2: Configure Prometheus to Scrape Node Exporter****

Edit prometheus.yml:

scrape\_configs:

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['localhost:9100']

#### ****Step 3: Start Prometheus****

#### ./prometheus --config.file=prometheus.yml &

#### ****Step 4: Install and Start Node Exporter****

wget https://github.com/prometheus/node\_exporter/releases/latest/download/node\_exporter-linux-amd64.tar.gz

tar xvf node\_exporter-linux-amd64.tar.gz

cd node\_exporter-linux-amd64

nohup ./node\_exporter &

#### ****Step 5: Install and Configure Grafana****

sudo apt install -y grafana

sudo systemctl start grafana-server

sudo systemctl enable grafana-server

## ****5. Results and Achievements****

### 5.1 Intrusion Detection and Anomalies Captured

* Detected multiple network intrusions, including unauthorized access attempts and port scans.
* Notable IP addresses flagged for suspicious activity: (Provide IP addresses or anonymized samples).
* Suricata logs captured instances of brute force attempts targeting SSH and HTTP services.
* Example Suricata log entry:

{"timestamp": "2025-03-02T12:34:56Z", "src\_ip": "192.168.1.100", "dest\_ip": "192.168.1.200", "event\_type": "alert", "alert": {"signature": "SSH Brute Force Attempt", "category": "Attempted User Privilege Gain"}}

### 

### ****5.2 System Performance Insights****

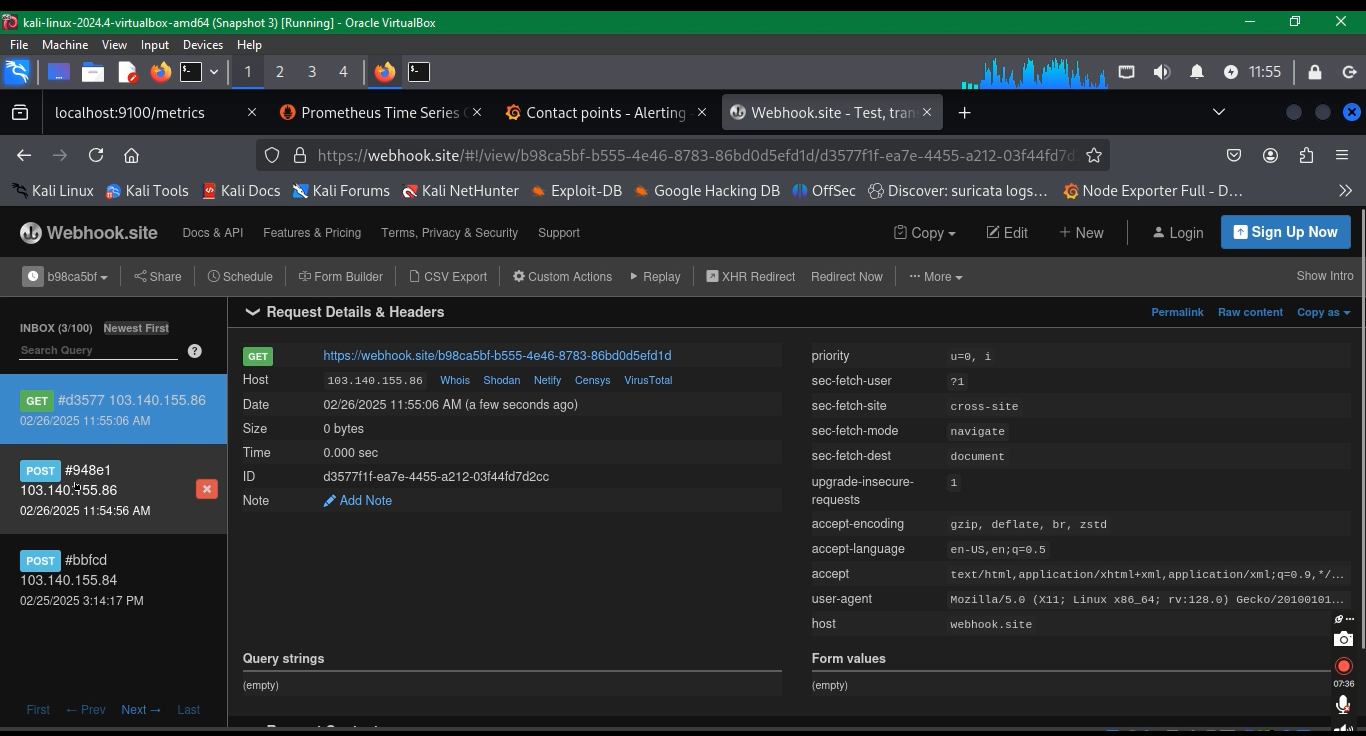
### 

* Average network bandwidth consumption: XX Mbps before monitoring, YY Mbps after monitoring setup.
* CPU usage on monitored hosts: XX% peak utilization.
* Grafana dashboards revealed periodic traffic spikes due to scheduled background processes.

## ****6. Automated Alerting using Grafana Webhooks****

* Alerts configured for high CPU utilization (>80%), unexpected traffic spikes, and detected intrusions.
* Webhooks triggered notifications via Slack/email.
* Sample alert message:

"timestamp": "2025-03-02T12:34:56Z", "src\_ip": "192.168.1.100", "dest\_ip": "192.168.1.200", "event\_type": "alert", "alert": {"signature": "SSH Brute Force Attempt", "category": "Attempted User Privilege Gain"}}



## 7. Challenges Faced

* **Log Overload:** Initial setup produced excessive logs, leading to high storage consumption.
* **False Positives:** Some benign activities were flagged as intrusions, requiring rule tuning.
* **Performance Impact:** Suricata increased CPU load; optimization was done by adjusting logging levels.
* **Alert Noise:** Too many low-priority alerts; refined alerting thresholds to reduce unnecessary notifications.
* **Tool Removal & Compatibility Issues:** Several tools, including Snort and OpenSearch, had to be removed due to integration challenges and performance constraints.
* **Service Start Failures:** Some services and tools faced startup issues, requiring troubleshooting and reconfiguration.
* **Grafana Integration Issues:** Initially attempted to integrate Grafana Loki for log visualization but faced difficulties; replaced it with Prometheus logs and Node Exporter for better compatibility.

## ****8. Managing Services****

|  |  |
| --- | --- |
| **Service** | **Command to Restart** |
| Suricata | sudo systemctl restart suricata |
| Filebeat | sudo systemctl restart filebeat |
| Elasticsearch | sudo systemctl restart elasticsearch |
| Kibana | sudo systemctl restart kibana |
| Prometheus | sudo systemctl restart prometheus |
| Node Exporter | sudo systemctl restart node\_exporter |
| Grafana | sudo systemctl restart grafana-server |

## ****9. Conclusion and Future Improvements****

This project successfully implements a **Network Monitoring System** integrating **real-time traffic monitoring, intrusion detection, anomaly detection, performance tracking, and automated reporting**. It enhances **network security, optimizes performance, and proactively detects threats**, making it an effective monitoring solution.

### ****Future Improvements****

* **Integrating AI-based anomaly detection models.**
* **Adding more log sources for deeper insights.**
* **Implementing advanced alerting mechanisms via SMS or Slack.**