Network Traffic Monitoring and Analysis using Suricata and Kibana

Project Report

Geetansh Chawla

Table of Contents

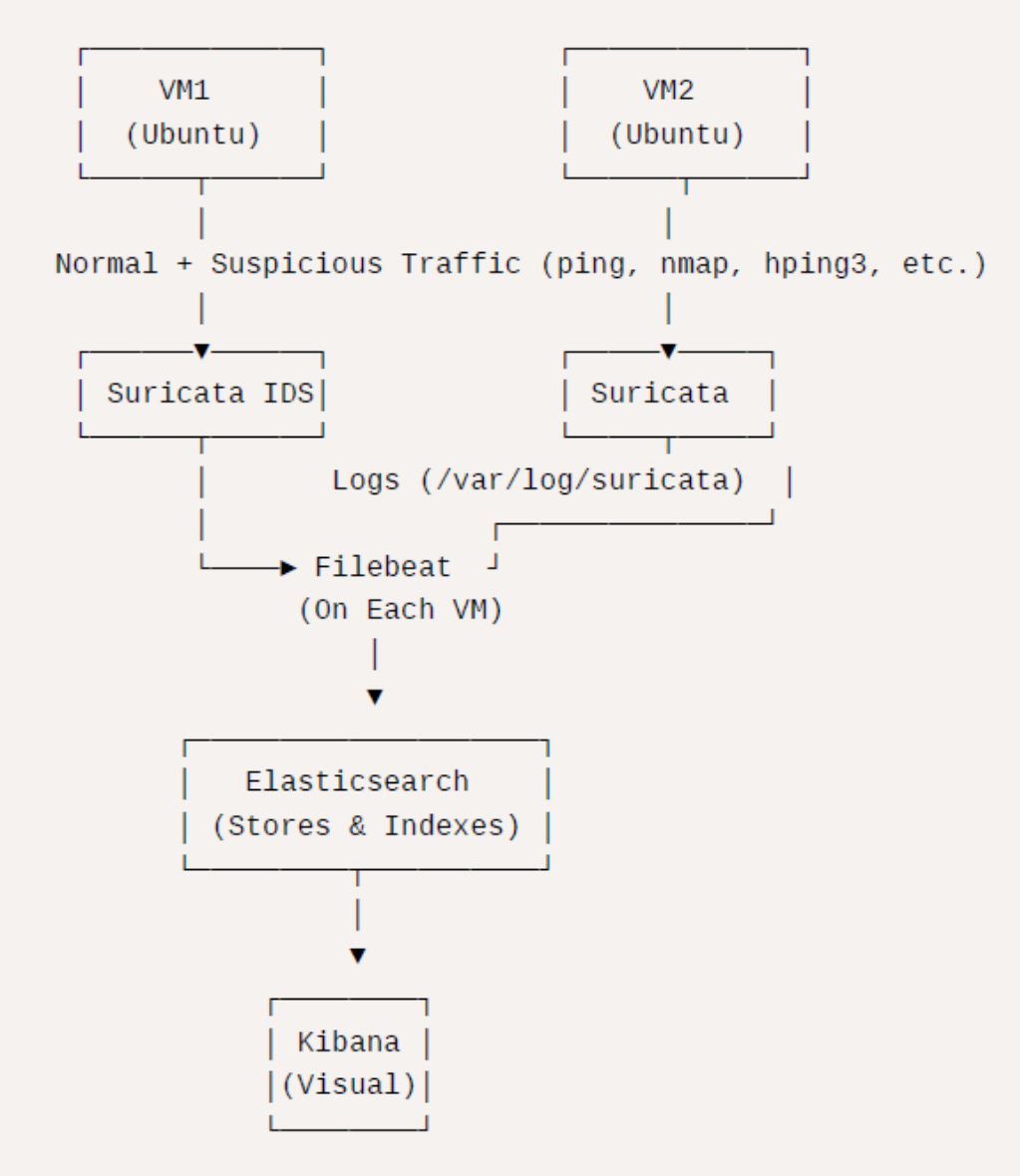
1. Introduction

This project focuses on monitoring and analyzing network traffic between two virtual machines (VMs) using Suricata, Filebeat, Elasticsearch, and Kibana.

The primary objective is to detect and visualize both normal and suspicious activities in the network, identify anomalies, and provide security recommendations based on the observations.

By utilizing Suricata for intrusion detection and Filebeat for log shipping, we aim to analyze and interpret network traffic patterns effectively. Elasticsearch serves as the storage and query engine for logs, while Kibana is employed for visualization, aiding in the identification of potential security threats.

The diagram below illustrates the data flow among the components in this



2. System Setup and Configuration

2.1 Virtual Machine Configuration

The project was implemented using two VMs running on VMware, each configured with a unique static IP address to ensure controlled traffic analysis. The following configurations were made for the VMs:

VM Name	Hostname	IP Address
VM1	getnsh1-VMware-Virtual-Platform	192.168.1.10
VM2	getnsh2-VMware-Virtual-Platform	192.168.1.11

Both VMs are part of the internal network (192.168.1.0/24), ensuring smooth communication without interference from external networks.

2.2 Installed Software and Services

The following components were installed and configured on the respective VMs:

- **VM1 (192.168.1.10 - Data Collector & Suricata Sensor):**
  - **Suricata**: A high-performance Network Intrusion Detection System (NIDS), which inspects network traffic in real-time for potential threats.
  - **Filebeat:** A lightweight log shipper, which forwards Suricata-generated logs to Elasticsearch for indexing and further analysis.
  - **Elasticsearch:** A distributed search and analytics engine, used to index and store Suricata logs for quick querying.
  - **Kibana:** Provides a web interface for visualization of Elasticsearch data, aiding in the creation of dashboards and traffic analysis.
- **VM2 (192.168.1.11 - Traffic Generator & Client):**
  - **The second VM was used to generate various types of traffic**, including normal HTTP requests, network scans, and attack simulations, to assess the effectiveness of the monitoring system.
  - **Suricata**: A high-performance Network Intrusion Detection System (NIDS), which inspects network traffic in real-time for potential threats.
  - **Filebeat:** A lightweight log shipper, which forwards Suricata-generated logs to Elasticsearch for indexing and further analysis.

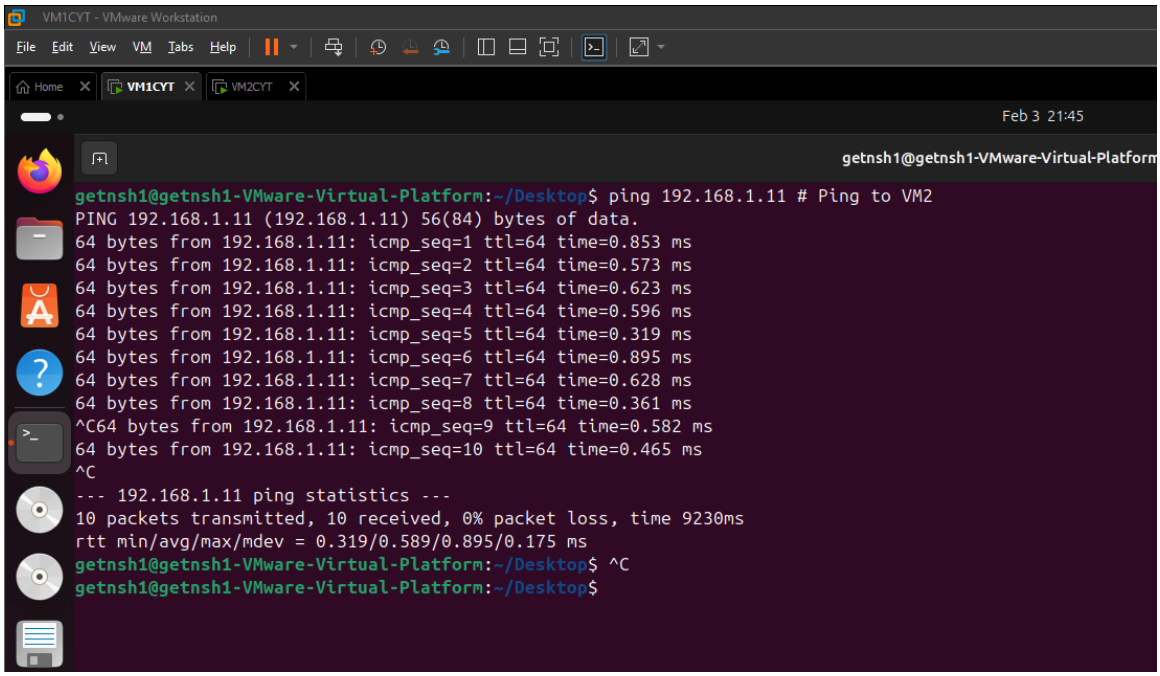
2.3 Network Configuration

To ensure proper communication between the VMs, ping tests were conducted:

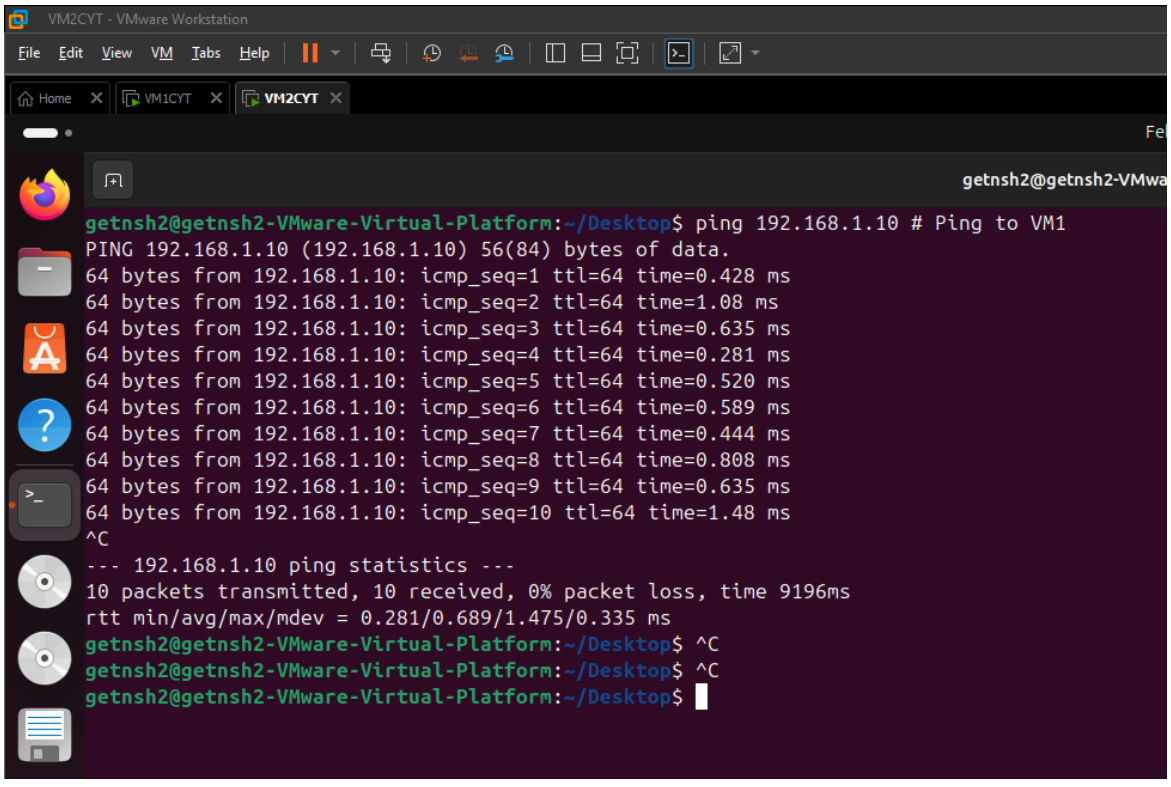
ping 192.168.1.11 # From VM1 to VM2

ping 192.168.1.10 # From VM2 to VM1

Both tests were successful, confirming that the VMs were correctly networked.



VM1 to VM2



VM2 to VM1

3. Data Collection & Log Pipeline

3.1 Suricata Configuration (VM1 - 192.168.1.10)

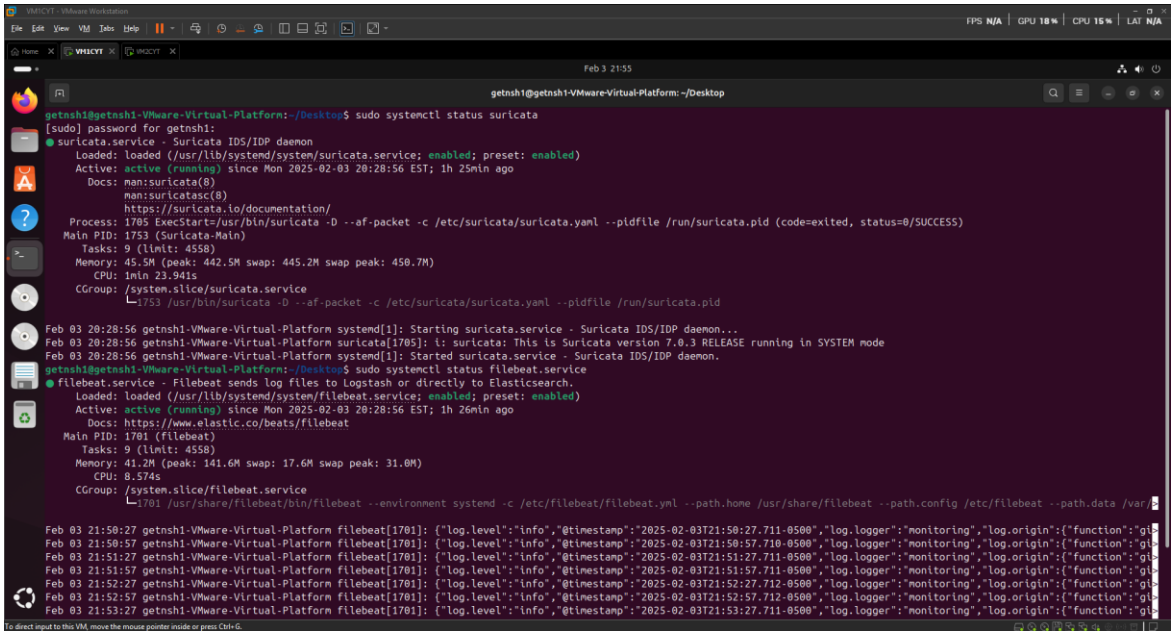
Suricata was configured on VM1 and VM2 to monitor all incoming and outgoing network traffic, with the following key settings:

- Enabled rules for detecting common types of attacks, such as port scans, Nmap scans, and brute-force attempts.
- Suricata logs were stored in JSON format at /var/log/suricata/eve.json to facilitate structured data processing and easy ingestion into Filebeat.

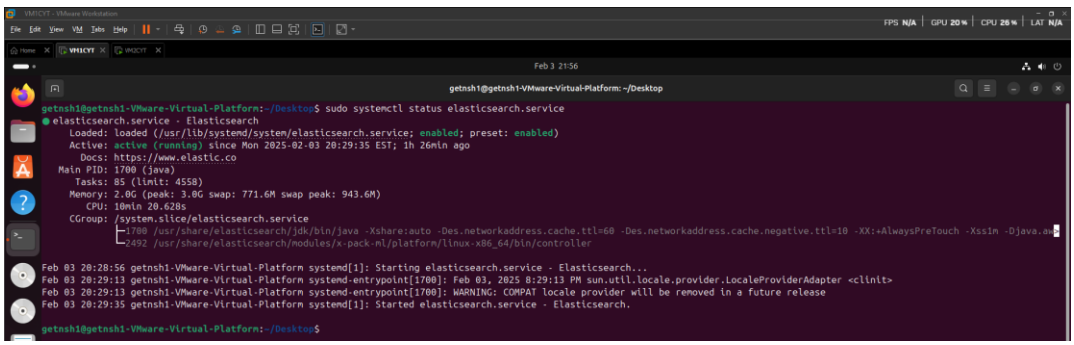
3.2 Filebeat Configuration

Filebeat was installed on VM1 and VM2 and configured to forward Suricata logs to Elasticsearch. The key configuration points include:

- Filebeat was set up to monitor /var/log/suricata/eve.json for new log entries.
- The output for Filebeat was defined to direct the logs to the local Elasticsearch instance, configured in the following manner:



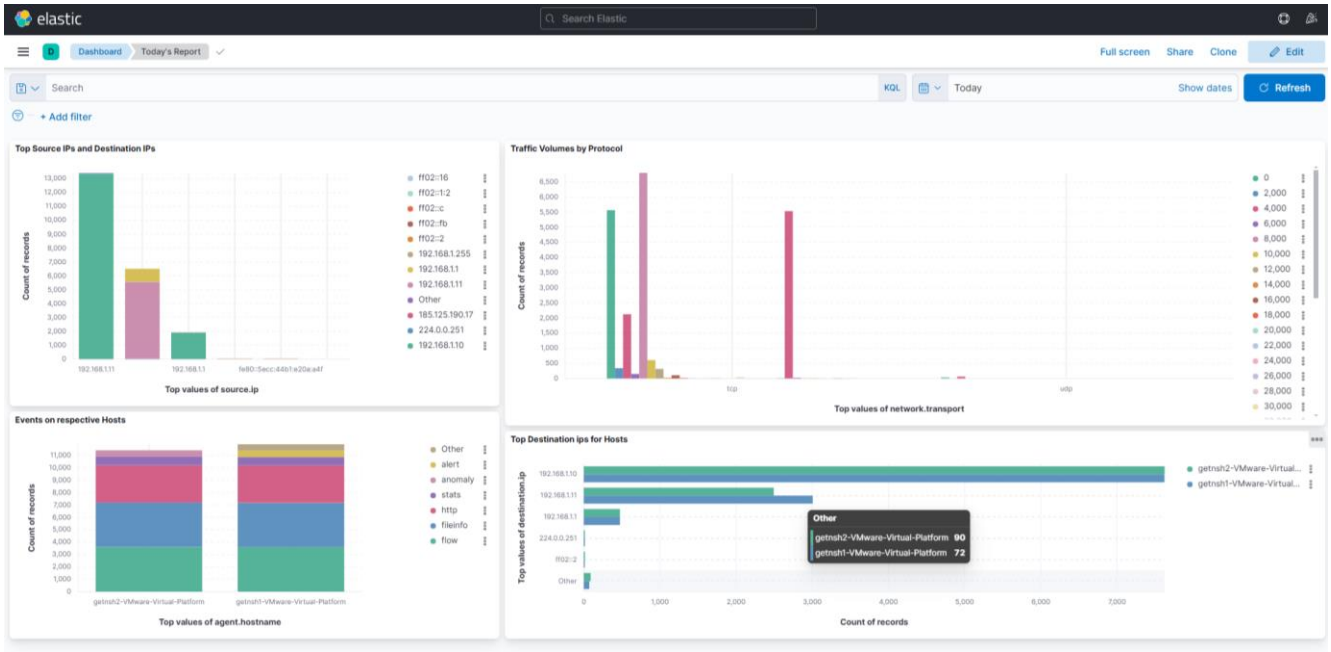
Suricata and Filebeat Status



Elastic Status

3.3 Elasticsearch and Kibana Setup

Once the data was sent from Filebeat, Elasticsearch indexed the logs, enabling efficient querying and search. Kibana was configured to provide a web-based interface for the analysis of traffic patterns. Various visualizations were created to help identify network activity and potential threats.



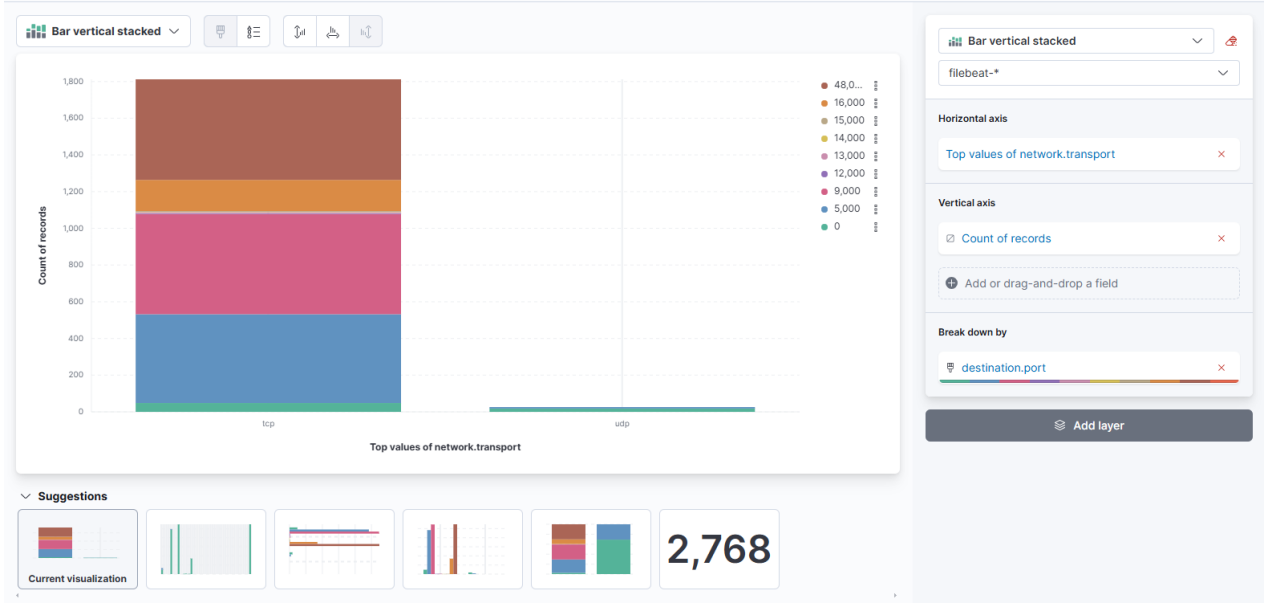
Elastic Dashboard with Specific Visualisation and Graphs

4. Traffic Observations

4.1 Normal Traffic Findings

The following types of traffic were observed as part of the normal behaviour between the VMs:

- HTTP requests accounted for the majority of traffic between the two VMs.
- TCP and UDP protocols were predominantly used for communication.
- Common ports in use included 2000, 4000, 8000, 10000, and 48000.

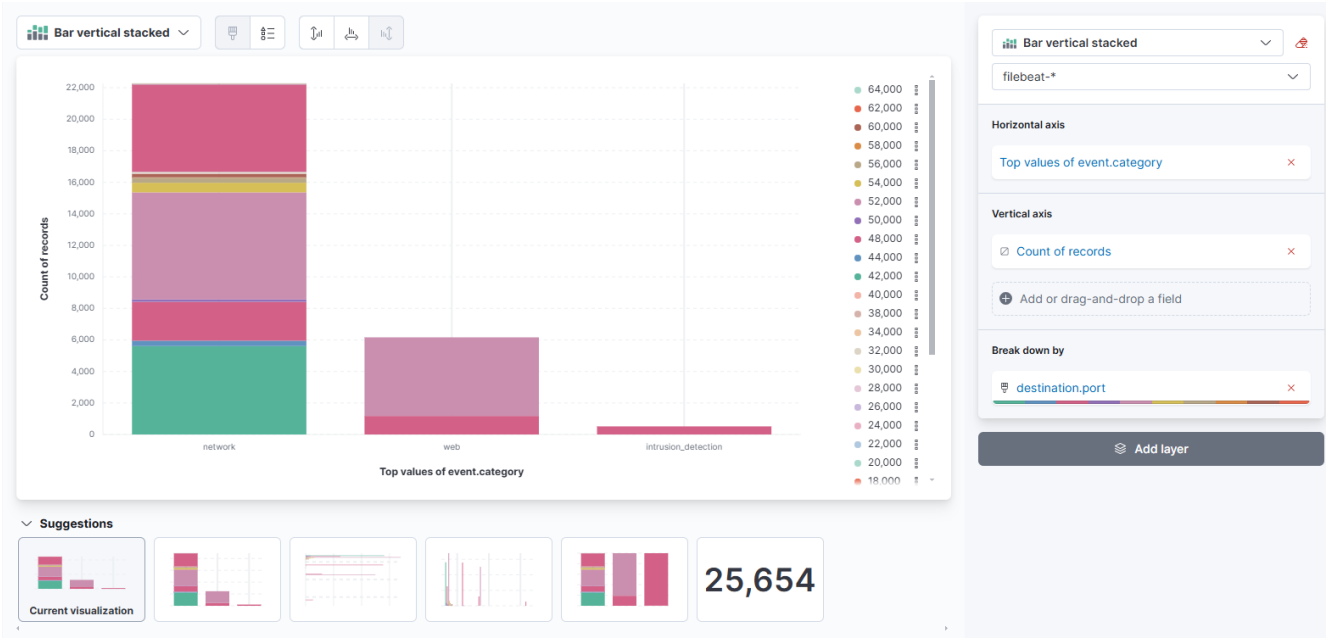


Screenshot showing the "Top Source & Destination ports" for a comprehensive traffic overview.

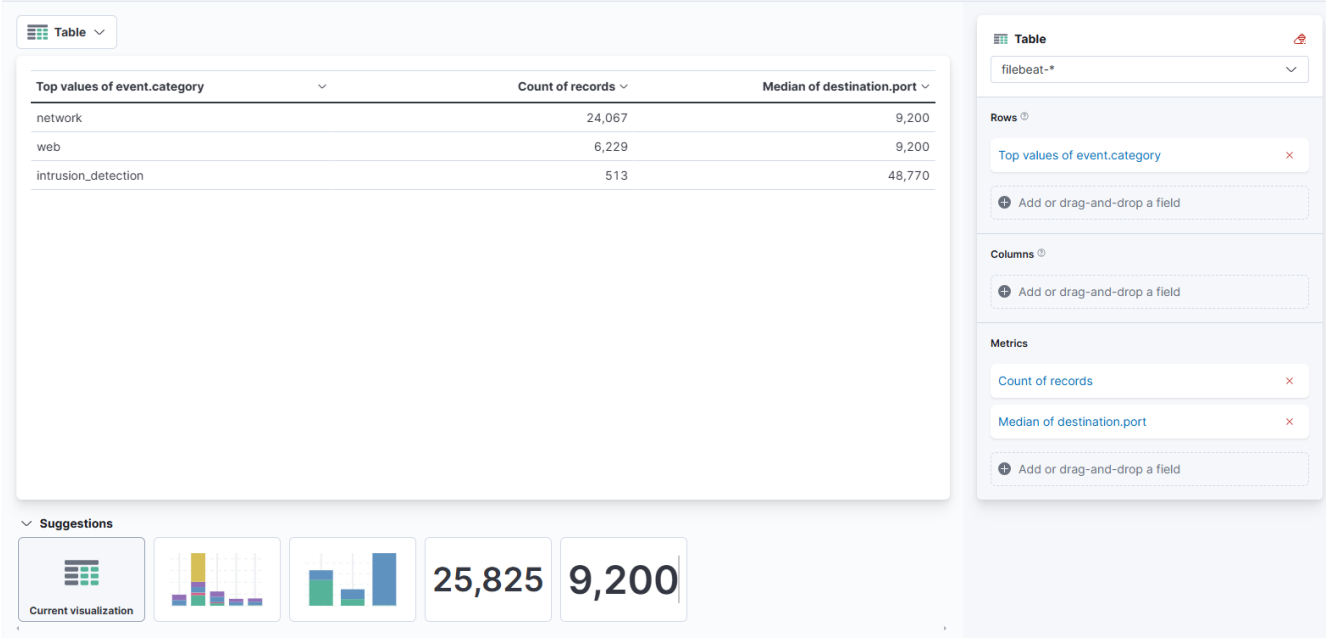
4.2 Suspicious Activity & Anomalies

While most traffic was normal, some anomalies were detected, which could be indicative of potential security threats:

- Repeated connections on high-numbered ports (e.g., 48000, 32000) were observed, potentially signaling port scanning or attempts to exploit open ports.
- Traffic from multicast addresses (224.0.0.251, ff02::2) was detected, which could indicate service discovery attempts or scanning activities.
- Suricata flagged 513 events (Intrusion Detection) , highlighting various types of suspicious traffic.



"Event Categories Breakdown" showing network, web, and intrusion detection categories.



"Event Categories Breakdown" showing network, web, and intrusion detection categories.

5. Security Recommendations

Based on the findings from the traffic analysis, the following security improvements are recommended:

5.1 Implement Firewall Rules

To minimize the risk of unauthorized access, firewall rules should be configured to restrict traffic to only the necessary ports. For example:

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT # Allow SSH

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT # Allow HTTP

sudo iptables -A INPUT -p tcp --dport 8000:9000 -j DROP # Block high ports

5.2 Improve Suricata Rule Coverage

To enhance detection capabilities, it is crucial to keep Suricata's rule set updated:

This will enable additional detection rules for brute-force attacks, network scanning, and suspicious user agents.

5.3 Segment Network Traffic

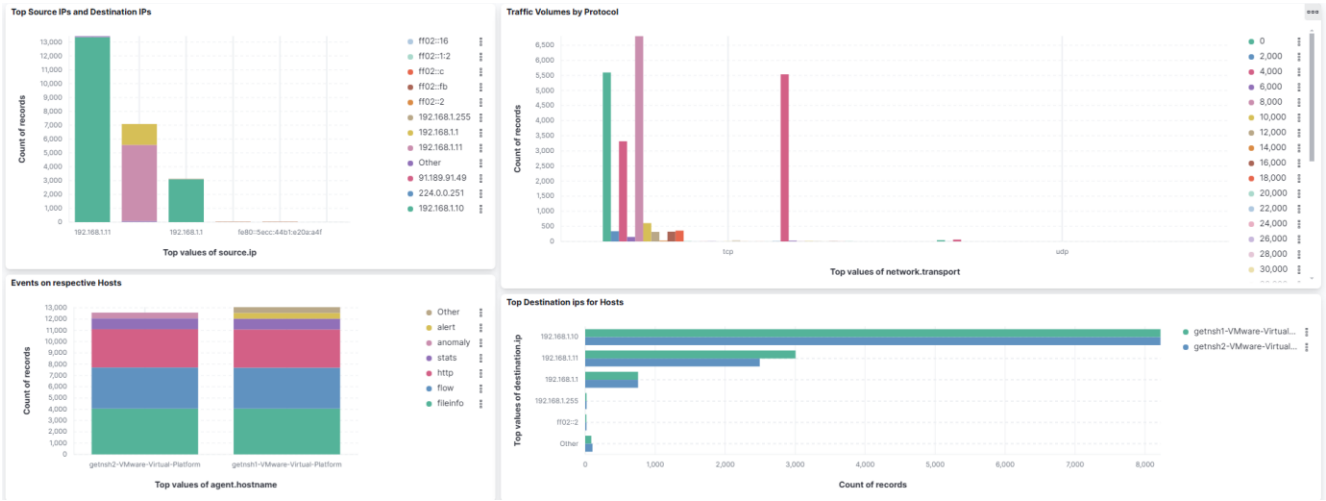
Implement network segmentation by utilizing VLANs or separate subnets for different parts of the infrastructure. This will help prevent lateral movement within the network in case of a breach.

5.4 Encrypt Sensitive Traffic

All sensitive communication should be encrypted using SSL/TLS protocols to ensure that traffic between the VMs cannot be intercepted by malicious actors.

6. Conclusion

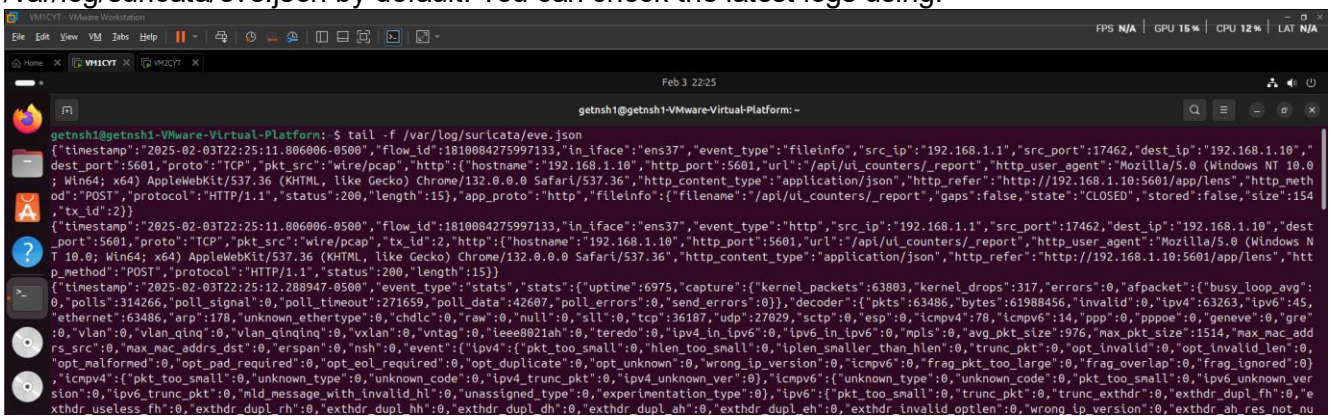
This project successfully established a network monitoring and analysis system utilizing Suricata, Filebeat, Elasticsearch, and Kibana. By analyzing network traffic and detecting anomalies, we identified areas for improvement in security. The next steps will involve refining Suricata rules, optimizing firewall policies, and further tuning the monitoring system to enhance its detection capabilities.



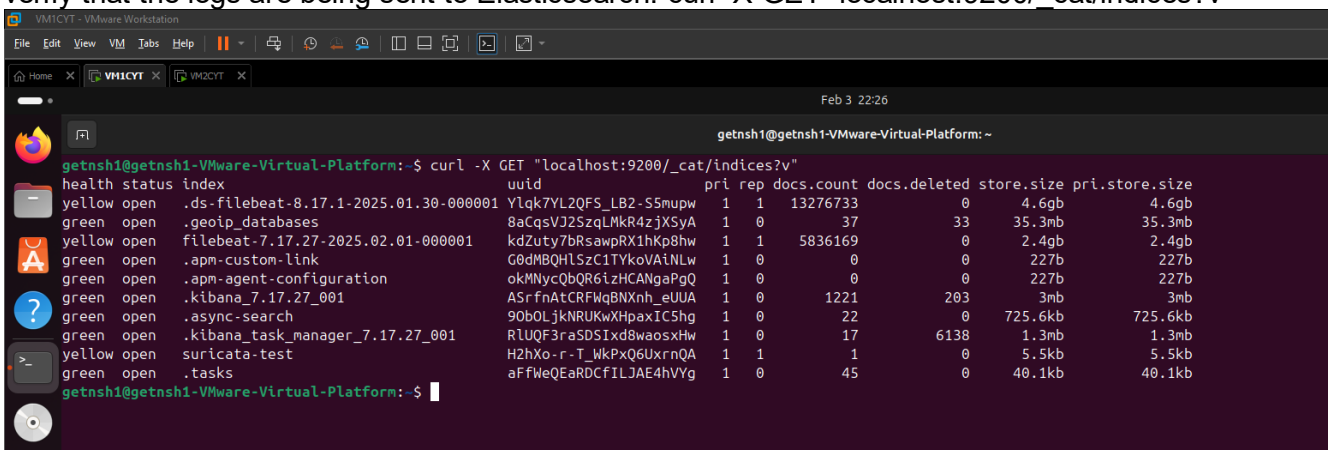
Kibana Dashboard

Other necessary Screenshots:

Check Suricata logs to confirm it is logging traffic: Suricata logs are typically stored in /var/log/suricata/eve.json by default. You can check the latest logs using:



Verify Filebeat is shipping Suricata logs to Elasticsearch: You can use the following command to verify that the logs are being sent to Elasticsearch: curl -X GET "localhost:9200/_cat/indices?v"



The query event.dataset: "suricata.eve" and http.request.method : "POST" shows network traffic logs from Suricata where the HTTP request method is POST, indicating that the captured events are HTTP POST requests logged by Suricata.