

Network Traffic Monitoring and Analysis using Suricata and Kibana

Project Report

Geetansh Chawla

Table of Contents

Introduction.....3

System Setup and Configuration.....4

Data Collection & Log Pipeline.....6

Traffic Observation.....7

Security Recommendation.....9

Conclusion.....9

Other Necessary Screenshots.....10

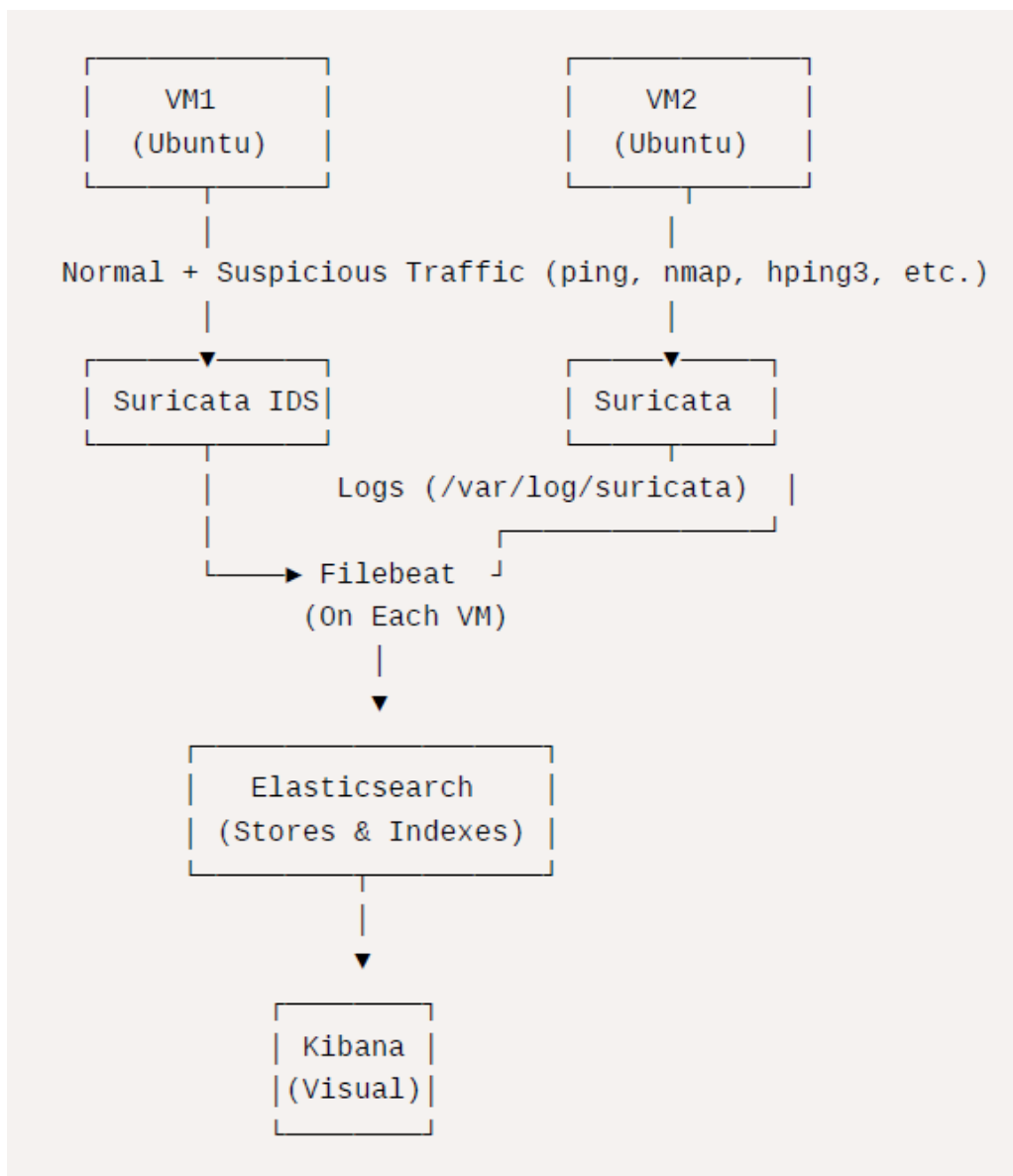
1. Introduction

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.

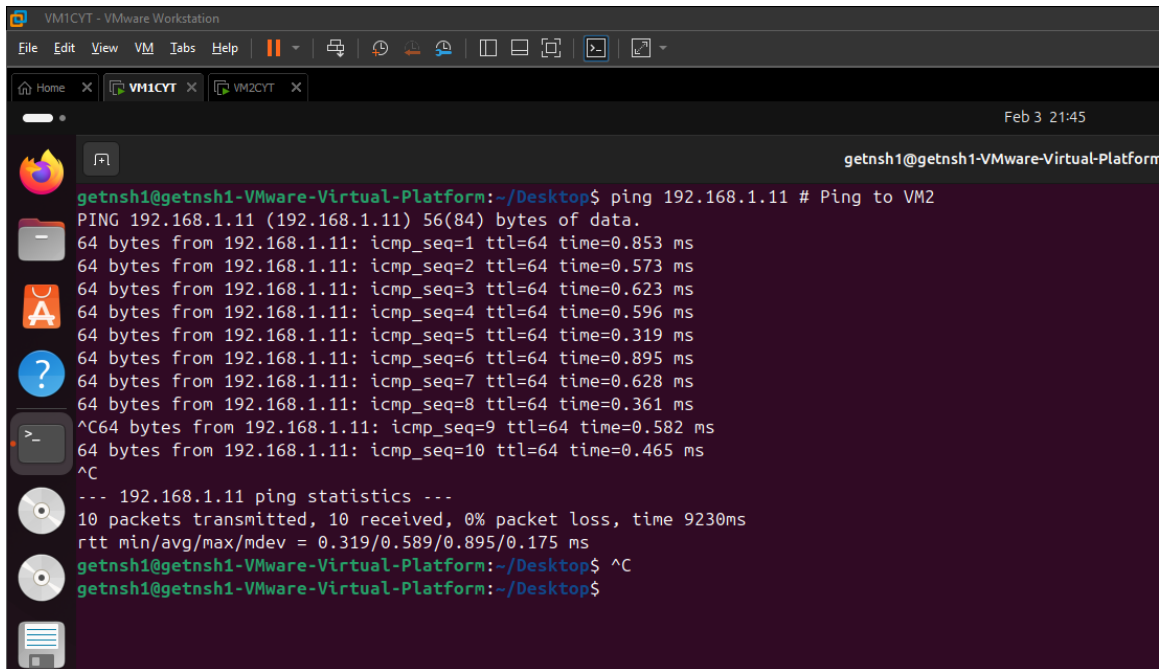
2.3 Network Configuration

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

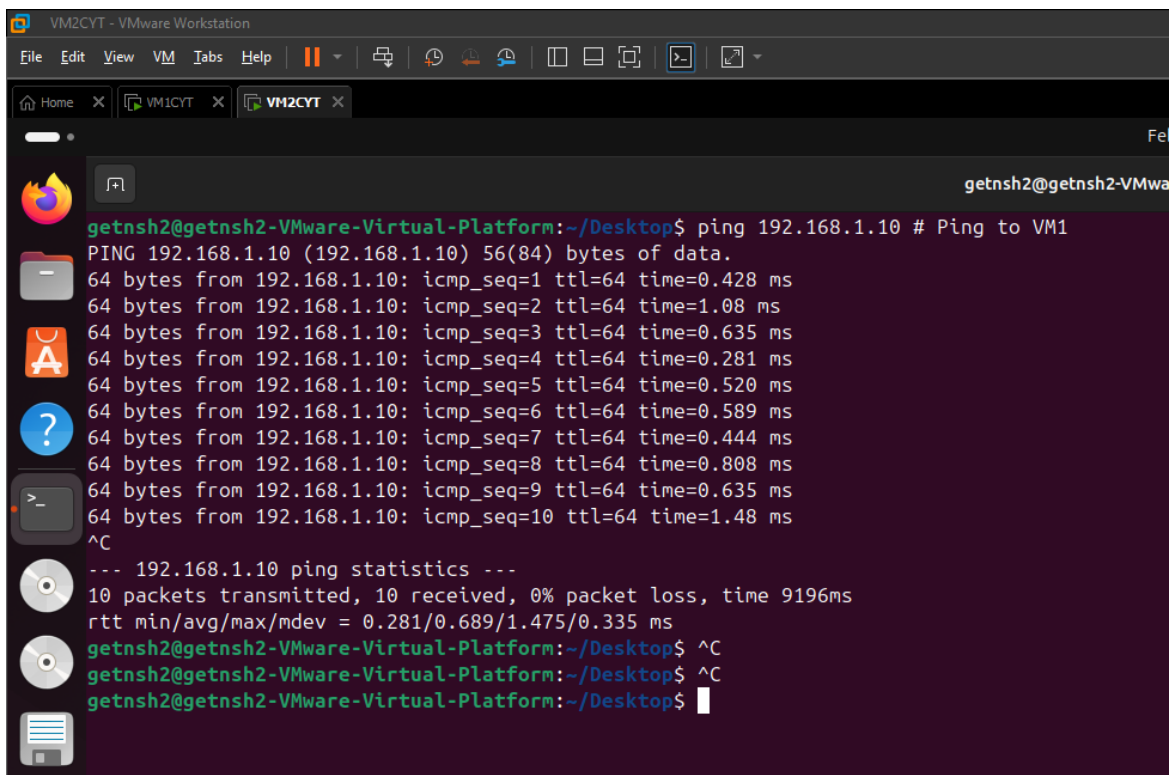
Both tests were successful, confirming that the VMs were correctly networked.



The screenshot shows a terminal window titled 'VM1CYT - VMware Workstation'. The terminal is running a ping command from 'getnsh1@getnsh1-VMware-Virtual-Platform' to '192.168.1.11'. The output shows 10 successful ping requests with varying response times, all receiving 64 bytes of data. The statistics at the bottom indicate 10 packets transmitted, 10 received, 0% packet loss, and a total time of 9230ms.

```
getnsh1@getnsh1-VMware-Virtual-Platform:~/Desktop$ ping 192.168.1.11 # Ping to VM2
PING 192.168.1.11 (192.168.1.11) 56(84) bytes of data.
64 bytes from 192.168.1.11: icmp_seq=1 ttl=64 time=0.853 ms
64 bytes from 192.168.1.11: icmp_seq=2 ttl=64 time=0.573 ms
64 bytes from 192.168.1.11: icmp_seq=3 ttl=64 time=0.623 ms
64 bytes from 192.168.1.11: icmp_seq=4 ttl=64 time=0.596 ms
64 bytes from 192.168.1.11: icmp_seq=5 ttl=64 time=0.319 ms
64 bytes from 192.168.1.11: icmp_seq=6 ttl=64 time=0.895 ms
64 bytes from 192.168.1.11: icmp_seq=7 ttl=64 time=0.628 ms
64 bytes from 192.168.1.11: icmp_seq=8 ttl=64 time=0.361 ms
^C64 bytes from 192.168.1.11: icmp_seq=9 ttl=64 time=0.582 ms
64 bytes from 192.168.1.11: icmp_seq=10 ttl=64 time=0.465 ms
^C
--- 192.168.1.11 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9230ms
rtt min/avg/max/mdev = 0.319/0.589/0.895/0.175 ms
getnsh1@getnsh1-VMware-Virtual-Platform:~/Desktop$ ^C
getnsh1@getnsh1-VMware-Virtual-Platform:~/Desktop$
```

VM1 to VM2



The screenshot shows a terminal window titled 'VM2CYT - VMware Workstation'. The terminal is running a ping command from 'getnsh2@getnsh2-VMware-Virtual-Platform' to '192.168.1.10'. The output shows 10 successful ping requests with varying response times, all receiving 64 bytes of data. The statistics at the bottom indicate 10 packets transmitted, 10 received, 0% packet loss, and a total time of 9196ms.

```
getnsh2@getnsh2-VMware-Virtual-Platform:~/Desktop$ ping 192.168.1.10 # Ping to VM1
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.428 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=1.08 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.635 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=0.281 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=64 time=0.520 ms
64 bytes from 192.168.1.10: icmp_seq=6 ttl=64 time=0.589 ms
64 bytes from 192.168.1.10: icmp_seq=7 ttl=64 time=0.444 ms
64 bytes from 192.168.1.10: icmp_seq=8 ttl=64 time=0.808 ms
64 bytes from 192.168.1.10: icmp_seq=9 ttl=64 time=0.635 ms
64 bytes from 192.168.1.10: icmp_seq=10 ttl=64 time=1.48 ms
^C
--- 192.168.1.10 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9196ms
rtt min/avg/max/mdev = 0.281/0.689/1.475/0.335 ms
getnsh2@getnsh2-VMware-Virtual-Platform:~/Desktop$ ^C
getnsh2@getnsh2-VMware-Virtual-Platform:~/Desktop$ ^C
getnsh2@getnsh2-VMware-Virtual-Platform:~/Desktop$
```

VM2 to VM1

3. Data Collection & Log Pipeline

3.1 Suricata Configuration (VM1 - 192.168.1.10)

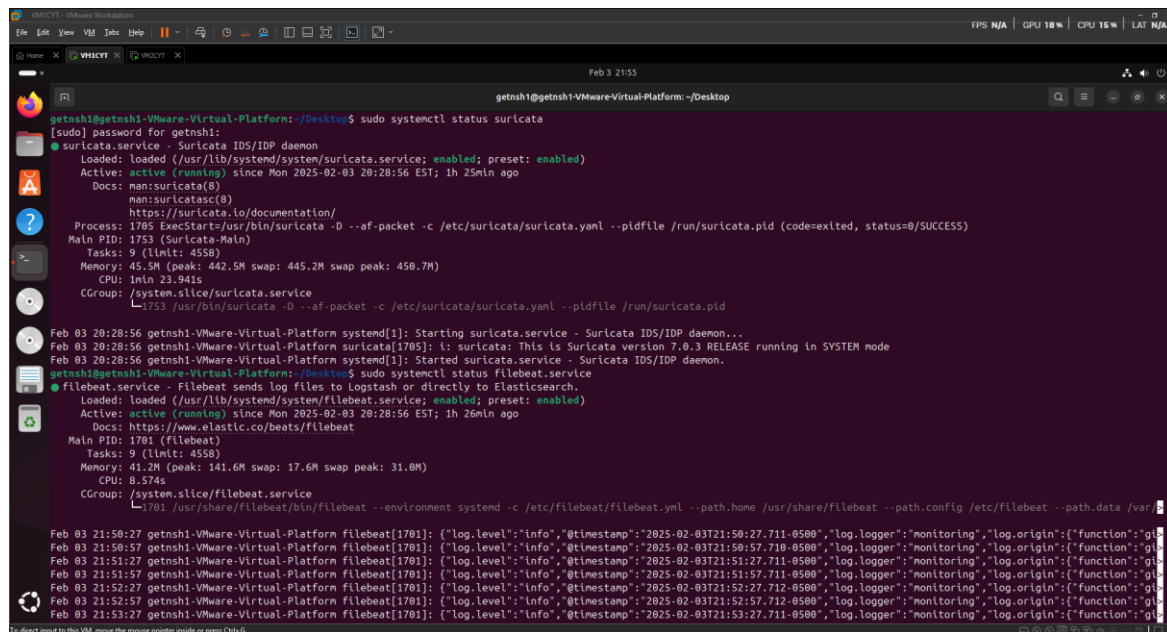
Suricata was configured on VM1 and VM2 to monitor all incoming and outgoing network traffic, with the following key settings:

- 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:

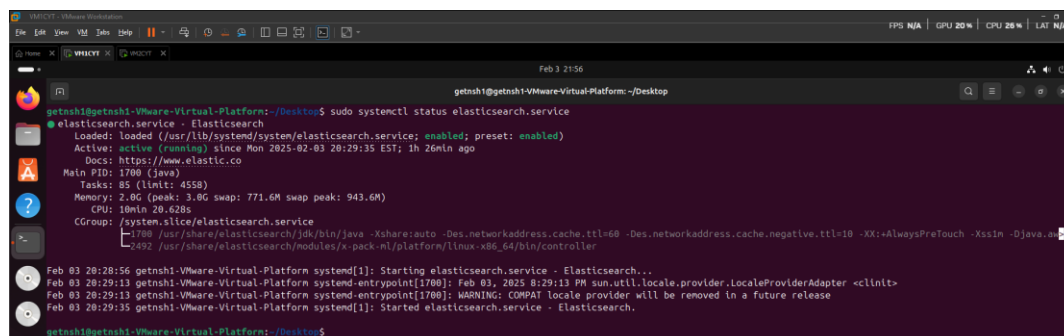


```
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop
[sudo] password for getnsh1:
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop$ sudo systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-02-03 20:28:56 EST; 1h 25min ago
     Docs: man:suricata(8)
           https://suricata.io/documentation/
   Process: 1705 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid (code=exited, status=0/SUCCESS)
   Main PID: 1753 (Suricata-Main)
     Tasks: 9 (Limit: 4558)
   Memory: 45.5M (peak: 442.5M swap: 445.2M swap peak: 450.7M)
     CPU: 1min 23.941s
   CGroup: /system.slice/suricata.service
           └─1753 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Feb 03 20:28:56 getnsh1-Vmware-Virtual-Platform systemd[1]: Starting suricata.service - Suricata IDS/IDP daemon...
Feb 03 20:28:56 getnsh1-Vmware-Virtual-Platform suricata[1705]: lt: suricata: This is Suricata version 7.0.3 RELEASE running in SYSTEM mode
Feb 03 20:28:56 getnsh1-Vmware-Virtual-Platform systemd[1]: Started suricata.service - Suricata IDS/IDP daemon.
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop$ sudo systemctl status filebeat.service
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/usr/lib/systemd/system/filebeat.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-02-03 20:28:56 EST; 1h 26min ago
     Docs: https://www.elastic.co/beats/filebeat
   Main PID: 1701 (filebeat)
     Tasks: 9 (Limit: 4558)
   Memory: 41.2M (peak: 141.6M swap: 17.6M swap peak: 31.0M)
     CPU: 8.574s
   CGroup: /system.slice/filebeat.service
           └─1701 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home /usr/share/filebeat --path.config /etc/filebeat --path.data /var/

Feb 03 21:50:27 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:50:27.711-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:50:57 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:50:57.710-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:51:27 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:51:27.711-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:51:57 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:51:57.711-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:52:27 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:52:27.712-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:52:57 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:52:57.712-0500","log.logger":"monitoring","log.origin":{"function":"gi
Feb 03 21:53:27 getnsh1-Vmware-Virtual-Platform filebeat[1701]: {"log.level":"info","@timestamp":"2025-02-03T21:53:27.711-0500","log.logger":"monitoring","log.origin":{"function":"gi
```

Suricata and Filebeat Status



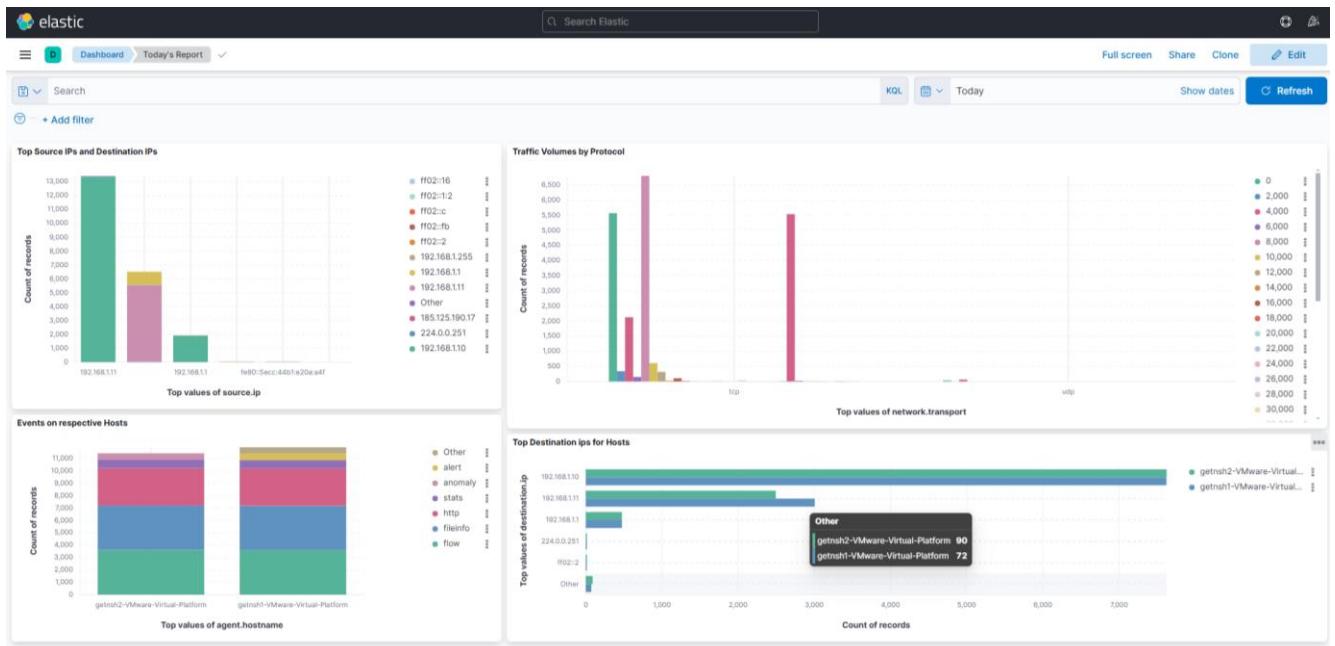
```
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop$ sudo systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-02-03 20:29:35 EST; 1h 26min ago
     Docs: https://www.elastic.co
   Main PID: 1700 (java)
     Tasks: 85 (Limit: 4558)
   Memory: 2.0G (peak: 3.0G swap: 771.6M swap peak: 943.6M)
     CPU: 10min 20.628s
   CGroup: /system.slice/elasticsearch.service
           └─1700 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.o
           └─1692 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Feb 03 20:28:56 getnsh1-Vmware-Virtual-Platform systemd[1]: Starting elasticsearch.service - Elasticsearch...
Feb 03 20:29:11 getnsh1-Vmware-Virtual-Platform systemd-entrypoint[1700]: Feb 03, 2025 8:29:13 PM sun.util.locale.provider.LocaleProviderAdapter <clinit>
Feb 03 20:29:13 getnsh1-Vmware-Virtual-Platform systemd-entrypoint[1700]: WARNING: COMPAT locale provider will be removed in a future release
Feb 03 20:29:35 getnsh1-Vmware-Virtual-Platform systemd[1]: Started elasticsearch.service - Elasticsearch.
getnsh1@getnsh1-Vmware-Virtual-Platform: ~/Desktop$
```

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.



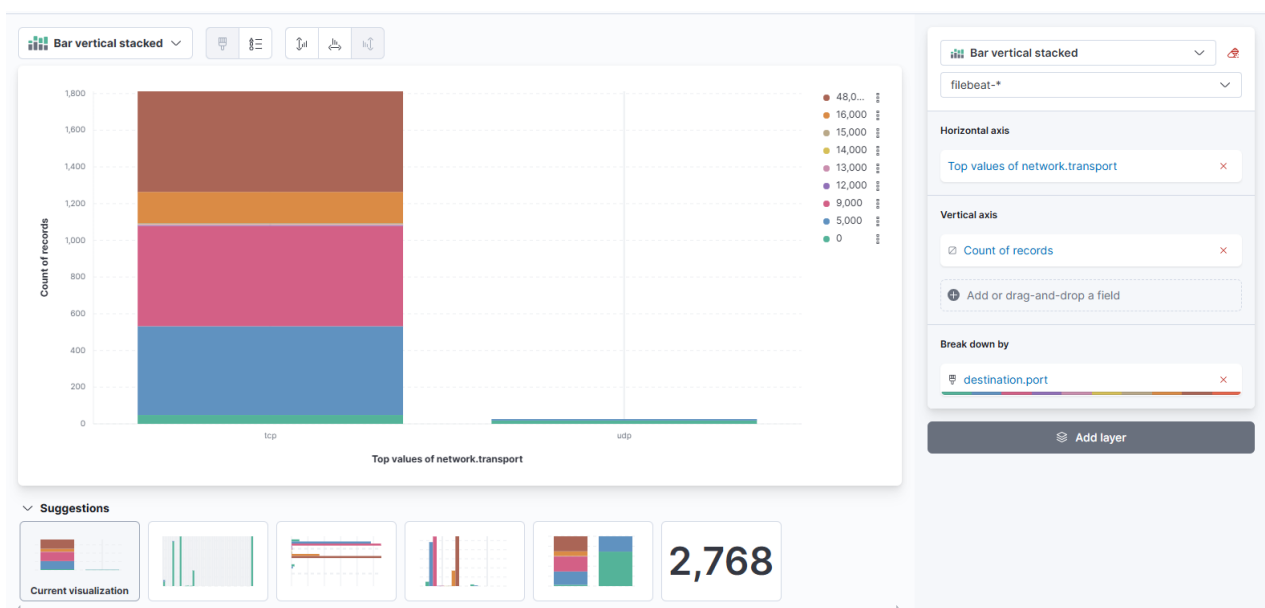
Elastic Dashboard with Specific Visualisation and Graphs

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.

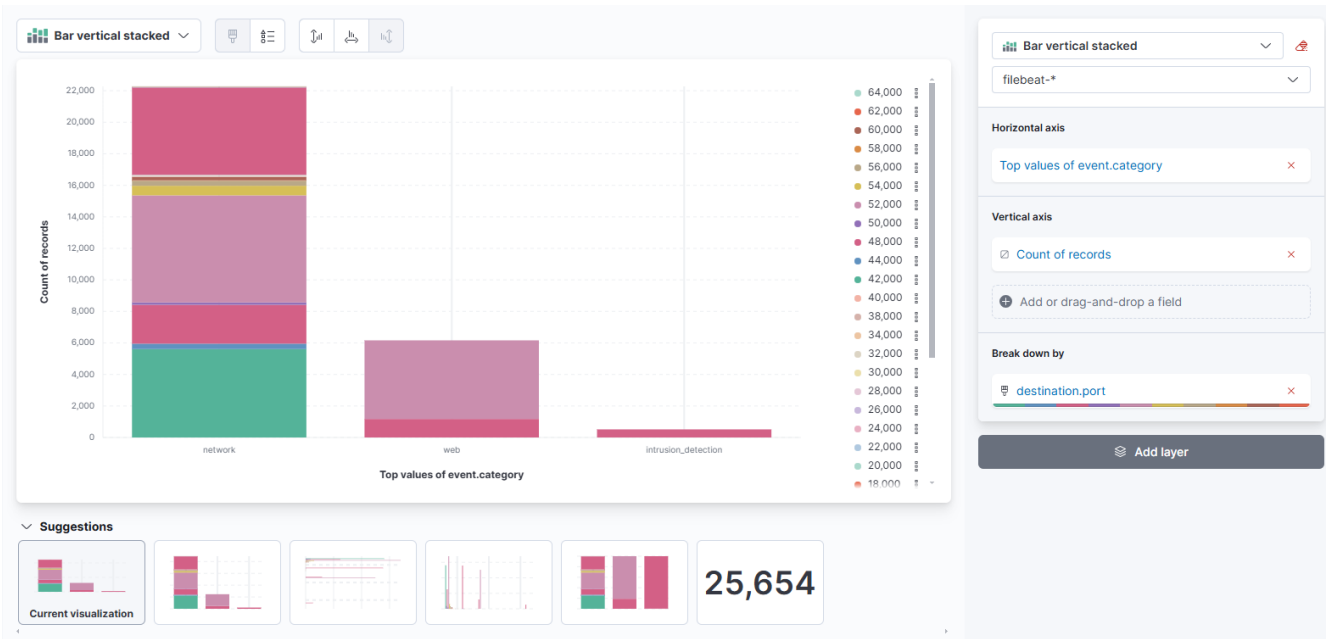


Screenshot showing the “Top Source & Destination ports” for a comprehensive traffic overview.

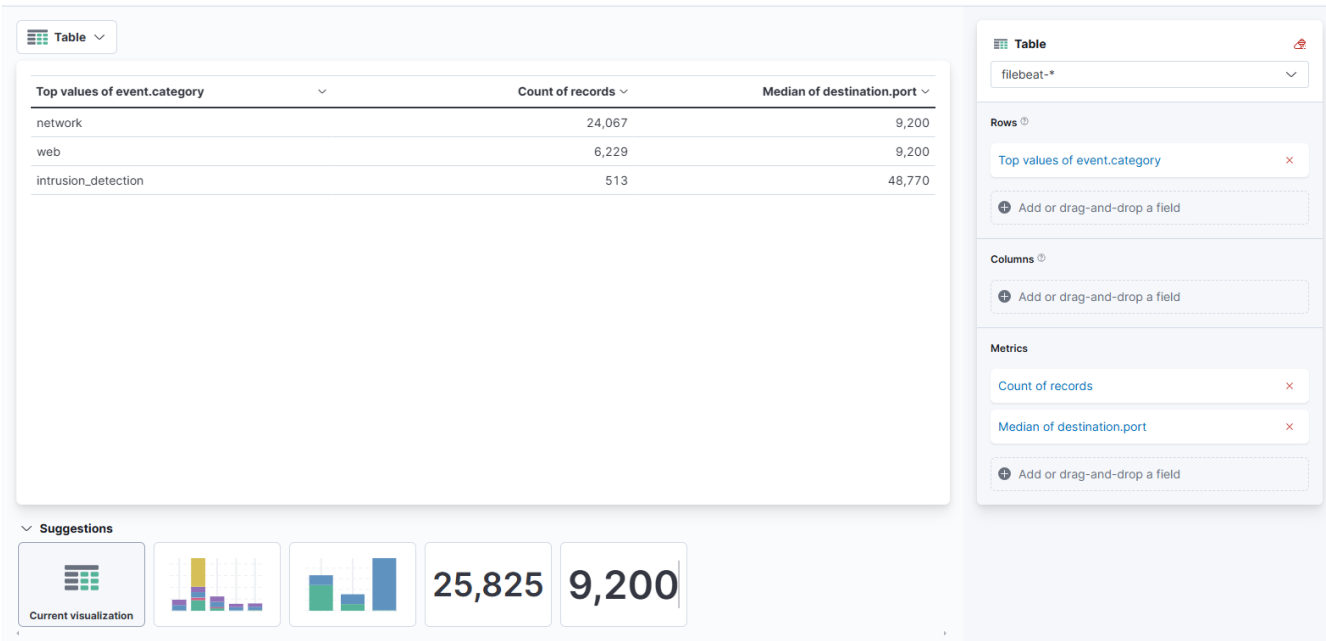
4.2 Suspicious Activity & Anomalies

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

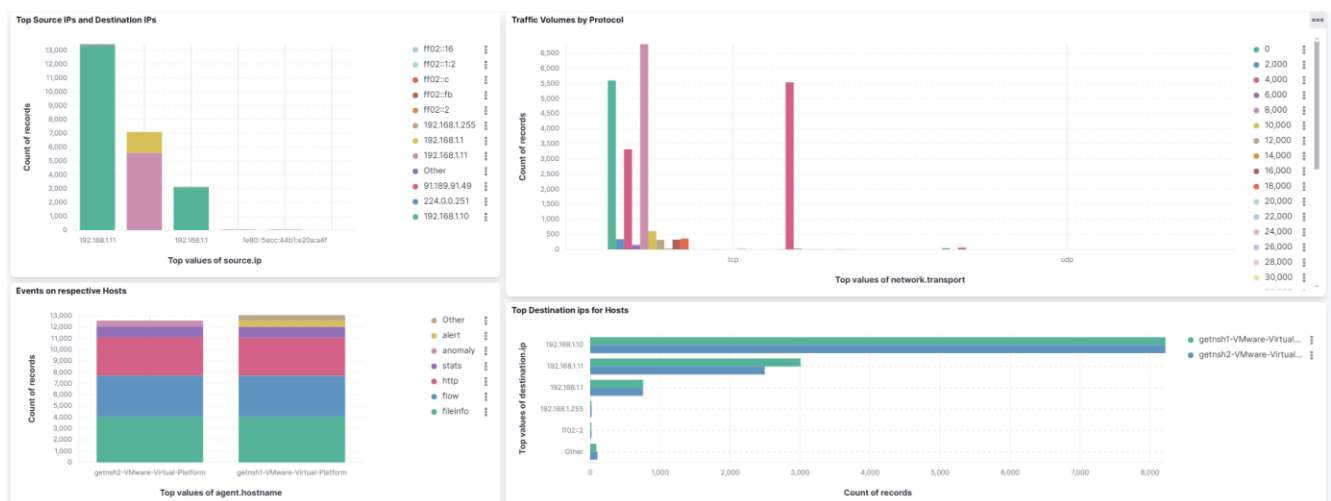
Implement network segmentation by utilizing VLANs or separate subnets for different parts of the infrastructure. This will help prevent lateral movement within the network in case of a breach.

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

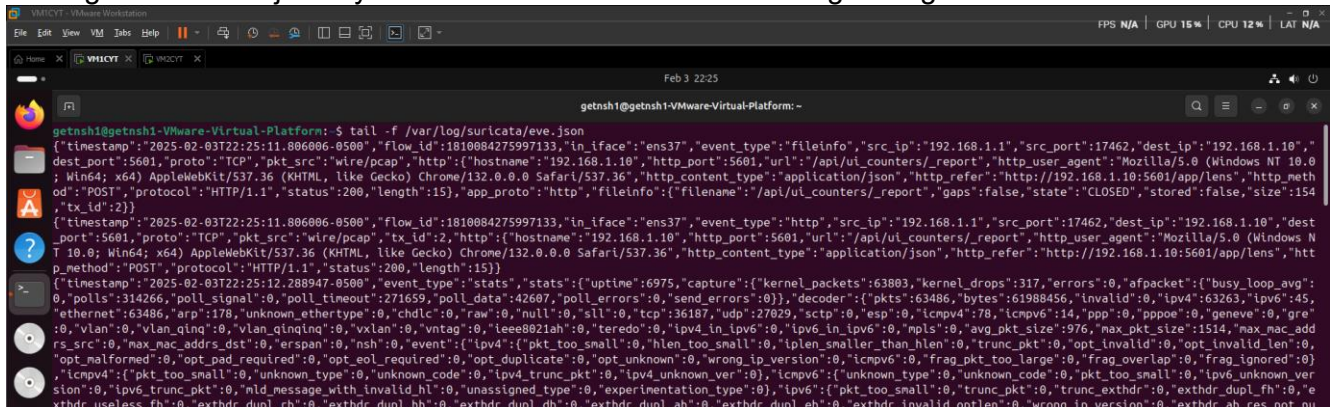
This project successfully established a network monitoring and analysis system utilizing Suricata, Filebeat, Elasticsearch, and Kibana. By analyzing network traffic and detecting anomalies, we identified areas for improvement in security. The next steps will involve refining Suricata rules, optimizing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



Kibana Dashboard

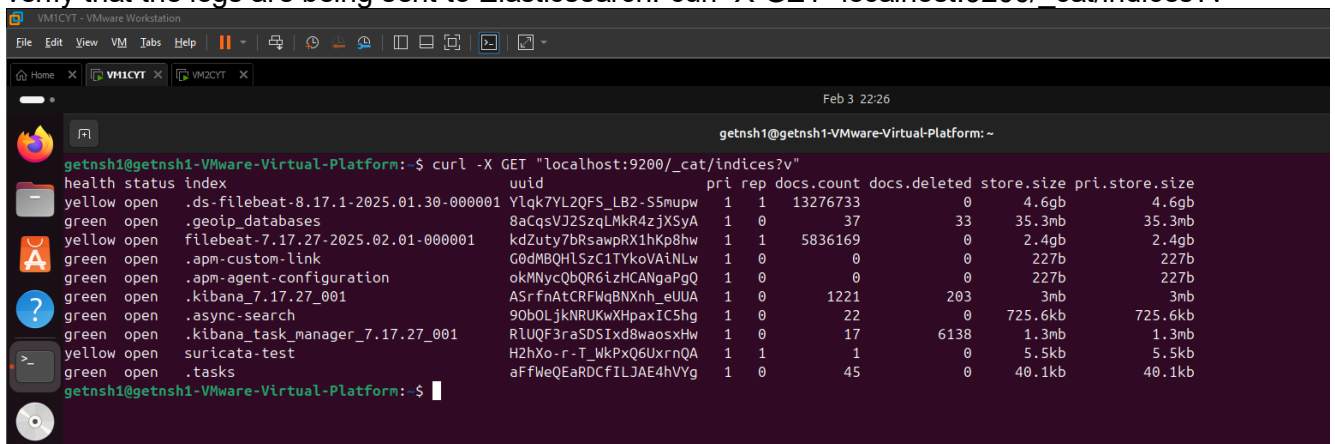
Other necessary Screenshots:

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:



```
getnsh1@getnsh1-VMware-Virtual-Platform: ~  
getnsh1@getnsh1-VMware-Virtual-Platform: $ tail -f /var/log/suricata/eve.json  
{  
  "timestamp": "2025-02-03T22:25:11.806006-0500",  
  "flow_id": 1810084275997133,  
  "in_iface": "ens37",  
  "event_type": "fileinfo",  
  "src_ip": "192.168.1.1",  
  "src_port": 17462,  
  "dest_ip": "192.168.1.10",  
  "dest_port": 5601,  
  "proto": "TCP",  
  "pkt_src": "wire/pcap",  
  "tx_id": 2,  
  "http": {  
    "hostname": "192.168.1.10",  
    "http_port": 5601,  
    "url": "/api/ui_counters/report",  
    "http_user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36",  
    "http_content_type": "application/json",  
    "http_refer": "http://192.168.1.10:5601/app/lens",  
    "http_method": "POST",  
    "protocol": "HTTP/1.1",  
    "status": 200,  
    "length": 15,  
    "app_proto": "http",  
    "fileinfo": {  
      "filename": "/api/ui_counters/report",  
      "gaps": false,  
      "state": "CLOSED",  
      "stored": false,  
      "size": 154,  
      "tx_id": 2  
    }  
  }  
},  
  "timestamp": "2025-02-03T22:25:11.806006-0500",  
  "flow_id": 1810084275997133,  
  "in_iface": "ens37",  
  "event_type": "http",  
  "src_ip": "192.168.1.1",  
  "src_port": 17462,  
  "dest_ip": "192.168.1.10",  
  "dest_port": 5601,  
  "proto": "TCP",  
  "pkt_src": "wire/pcap",  
  "tx_id": 2,  
  "http": {  
    "hostname": "192.168.1.10",  
    "http_port": 5601,  
    "url": "/api/ui_counters/report",  
    "http_user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36",  
    "http_content_type": "application/json",  
    "http_refer": "http://192.168.1.10:5601/app/lens",  
    "http_method": "POST",  
    "protocol": "HTTP/1.1",  
    "status": 200,  
    "length": 15  
  }  
},  
  "timestamp": "2025-02-03T22:25:12.288947-0500",  
  "event_type": "stats",  
  "stats": {  
    "uptime": 6975,  
    "capture": {  
      "kernel_packets": 63803,  
      "kernel_drops": 317,  
      "errors": 0,  
      "afpacket": {  
        "busy_loop_avg": 0,  
        "polls": 314266,  
        "poll_signal": 0,  
        "poll_timeout": 271659,  
        "poll_data": 42607,  
        "poll_errors": 0,  
        "send_errors": 0  
      },  
      "decoder": {  
        "pkts": 63486,  
        "bytes": 61988456,  
        "invalid": 0,  
        "ipv4": 63263,  
        "ipv6": 45,  
        "ethernet": 63486,  
        "arp": 178,  
        "unknown_ethertype": 0,  
        "chdlc": 0,  
        "raw": 0,  
        "null": 0,  
        "sl": 0,  
        "tcp": 36187,  
        "udp": 27829,  
        "sctp": 0,  
        "esp": 0,  
        "icmpv4": 78,  
        "icmpv6": 14,  
        "ppp": 0,  
        "pppoe": 0,  
        "geneve": 0,  
        "gre": 0,  
        "vlan": 0,  
        "vlan_qinq": 0,  
        "vxlan": 0,  
        "vntag": 0,  
        "teredo": 0,  
        "ipv4_in_ipv6": 0,  
        "ipv6_in_ipv6": 0,  
        "mpls": 0,  
        "avg_pkt_size": 976,  
        "max_pkt_size": 1514,  
        "max_mac_addr": 0,  
        "max_mac_addr_src": 0,  
        "max_mac_addr_dst": 0,  
        "nsh": 0,  
        "event": {  
          "ipv4": {  
            "pkt_too_small": 0,  
            "hlen_too_small": 0,  
            "iplen_smaller_than_hlen": 0,  
            "trunc_pkt": 0,  
            "opt_invalid": 0,  
            "opt_invalid_len": 0,  
            "opt_malformed": 0,  
            "opt_pad_required": 0,  
            "opt_duplicate": 0,  
            "opt_unknown": 0,  
            "wrong_ip_version": 0,  
            "icmpv6": 0,  
            "frag_pkt_too_large": 0,  
            "frag_overlap": 0,  
            "frag_ignored": 0,  
            "icmpv4": {  
              "pkt_too_small": 0,  
              "unknown_type": 0,  
              "unknown_code": 0,  
              "ipv4_trunc_pkt": 0,  
              "ipv4_unknown_ver": 0  
            },  
            "icmpv6": {  
              "unknown_type": 0,  
              "unknown_code": 0,  
              "pkt_too_small": 0,  
              "trunc_pkt": 0,  
              "trunc_exthdr": 0,  
              "exthdr_dupl_fh": 0,  
              "exthdr_useless_fh": 0,  
              "exthdr_dupl_rh": 0,  
              "exthdr_dupl_hh": 0,  
              "exthdr_dupl_dh": 0,  
              "exthdr_dupl_ah": 0,  
              "exthdr_dupl_eh": 0,  
              "exthdr_invalid_optlen": 0,  
              "wrong_ip_version": 0,  
              "exthdr_ah_res_not_nu"  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

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"`



```
getnsh1@getnsh1-VMware-Virtual-Platform: ~  
getnsh1@getnsh1-VMware-Virtual-Platform: $ curl -X GET "localhost:9200/_cat/indices?v"  
health status index      uid      pri rep docs.count docs.deleted store.size pri.store.size  
yellow open   .ds-filebeat-8.17.1-2025.01.30-000001 YLqk7YL2QF5_LB2-S5mupw 1 1 13276733 0 4.6gb 4.6gb  
green open   .geop_databases 8aCqsVJ2SzlMkR4zjXsYa 1 0 37 33 35.3mb 35.3mb  
yellow open   filebeat-7.17.27-2025.02.01-000001 kdZuty7BrSawpRX1hKp8hw 1 1 5836169 0 2.4gb 2.4gb  
green open   .apm-custom-link G0dMBQHLSzC1TYkoVAiNLw 1 0 0 0 227b 227b  
green open   .apm-agent-configuration okMNYcQb0R6izHCANgaPgQ 1 0 0 0 227b 227b  
green open   .kibana_7.17.27_001 ASrFnAtCRFwqBNXnh_eUUA 1 0 1221 203 3mb 3mb  
green open   .async-search 90b0LjKNRUKwXHpaxIC5hg 1 0 22 0 725.6kb 725.6kb  
green open   .kibana_task_manager_7.17.27_001 RUQOf3raSDSIXd8WaosxHw 1 0 17 6138 1.3mb 1.3mb  
yellow open   suricata-test H2hXo-r-T_WkPx06UxrnQA 1 1 1 0 5.5kb 5.5kb  
green open   .tasks aFFWeQeARdcFLJAE4hVYg 1 0 45 0 40.1kb 40.1kb  
getnsh1@getnsh1-VMware-Virtual-Platform: $
```

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.

