Final Project
Part 3 – MCP9808_IoT Security Monitoring with Suricata and Kibana

CYT160: Security for Cloud and Internet of Things

Professor Saeed Naghizadeh Qomi

November 24, 2025

Group 2

Jyotpal Singh

Rickie Rihal

Yash Sanjaybhai Patel

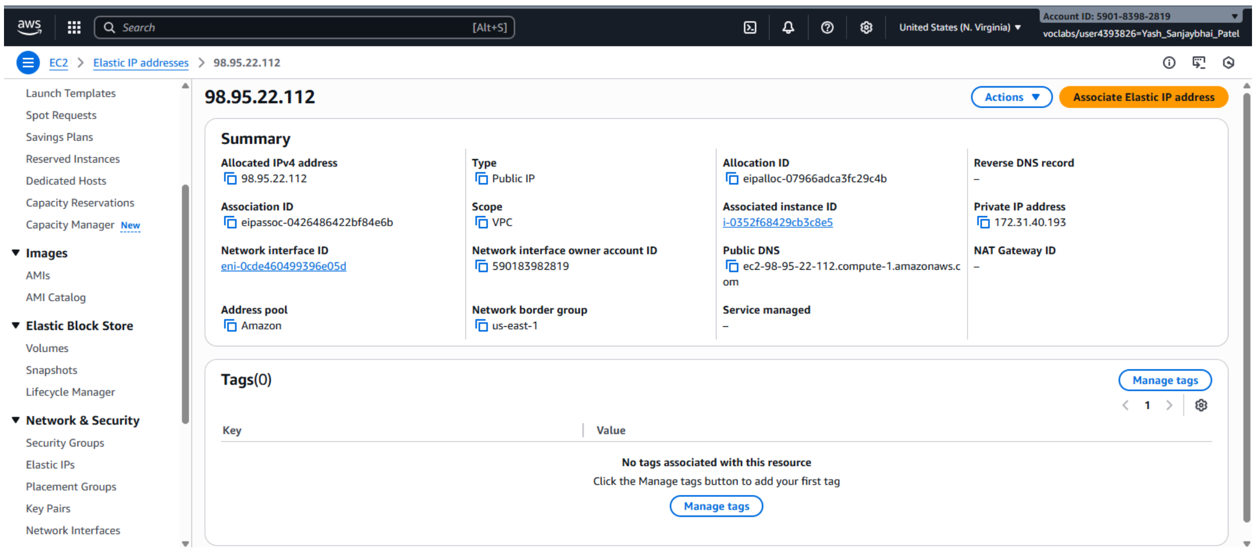
Introduction

In this part of the project, we configured an AWS EC2 Ubuntu server to act as a monitoring system for IoT traffic coming from my Raspberry Pi. We installed and set up Mosquitto, Suricata, Filebeat, Elasticsearch, and Kibana on the EC2 instance. Our goal was to collect mirrored MQTT traffic from my Raspberry Pi, detect different attacks (malformed payloads and DDoS), and visualize these alerts inside Kibana dashboards.

This report explains what we configured, what changes we made, how we simulated attacks, and what results we observed.

EC2 Instance Setup

We launched a t2.medium Ubuntu instance because Suricata + Filebeat + Elasticsearch + Kibana require more memory. Then we created an Elastic IP and attached it to the instance.This fixed IP was later used inside the Raspberry Pi Python script to mirror MQTT traffic.



98.95.22.112

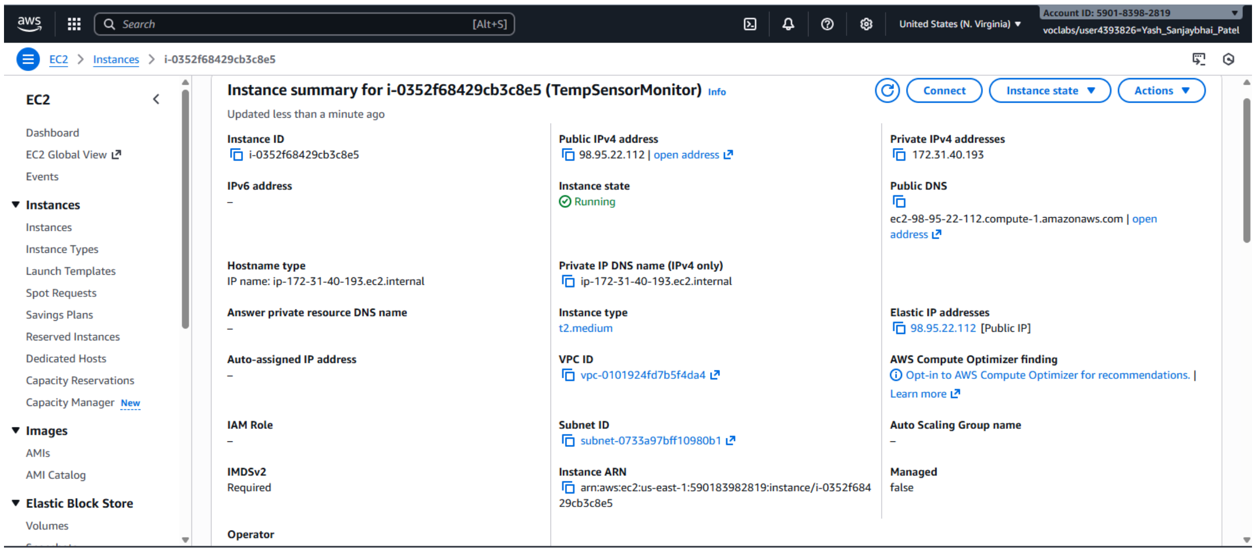
Summary

Allocated IPv4 address: 98.95.22.112	Type: Public IP	Allocation ID: eipalloc-07966adca3fc29c4b	Reverse DNS record: –
Association ID: eipassoc-0426486422bf84e6b	Scope: VPC	Associated instance ID: i-0352f68429cb3c8e5	Private IP address: 172.31.40.193
Network interface ID: eni-0cde460499396e05d	Network interface owner account ID: 590183982819	Public DNS: ec2-98-95-22-112.compute-1.amazonaws.com	NAT Gateway ID: –
Address pool: Amazon	Network border group: us-east-1	Service managed: –	

Tags(0)

Key	Value

No tags associated with this resource
Click the Manage tags button to add your first tag



Instance summary for i-0352f68429cb3c8e5 (TempSensorMonitor) Info

Updated less than a minute ago

Instance ID i-0352f68429cb3c8e5	**Public IPv4 address** 98.95.22.112 | open address	**Private IPv4 addresses** 172.31.40.193
IPv6 address –	**Instance state** Running	**Public DNS** ec2-98-95-22-112.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-40-193.ec2.internal	**Private IP DNS name (IPv4 only)** ip-172-31-40-193.ec2.internal	
Answer private resource DNS name –	**Instance type** t2.medium	**Elastic IP addresses** 98.95.22.112 [Public IP]
Auto-assigned IP address –	**VPC ID** vpc-0101924fd7b5f4da4	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	**Subnet ID** subnet-0733a97bff10980b1	**Auto Scaling Group name** –
IMDSv2 Required	**Instance ARN** arn:aws:ec2:us-east-1:590183982819:instance/i-0352f68429cb3c8e5	**Managed** false
Operator		

We also created a security group that allows:

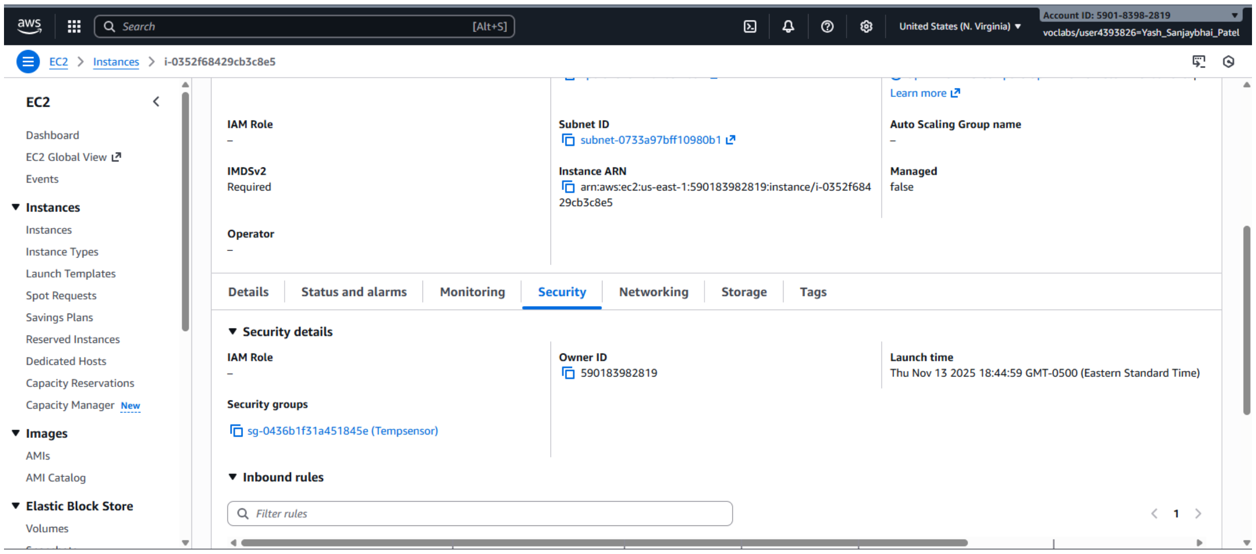
Port 22 → SSH

Port 1883 → MQTT traffic from Raspberry Pi

Port 5601 → Access Kibana

Port 9200 → Elasticsearch (only from local machine)

This allowed proper communication between my Raspberry Pi and the EC2 server.



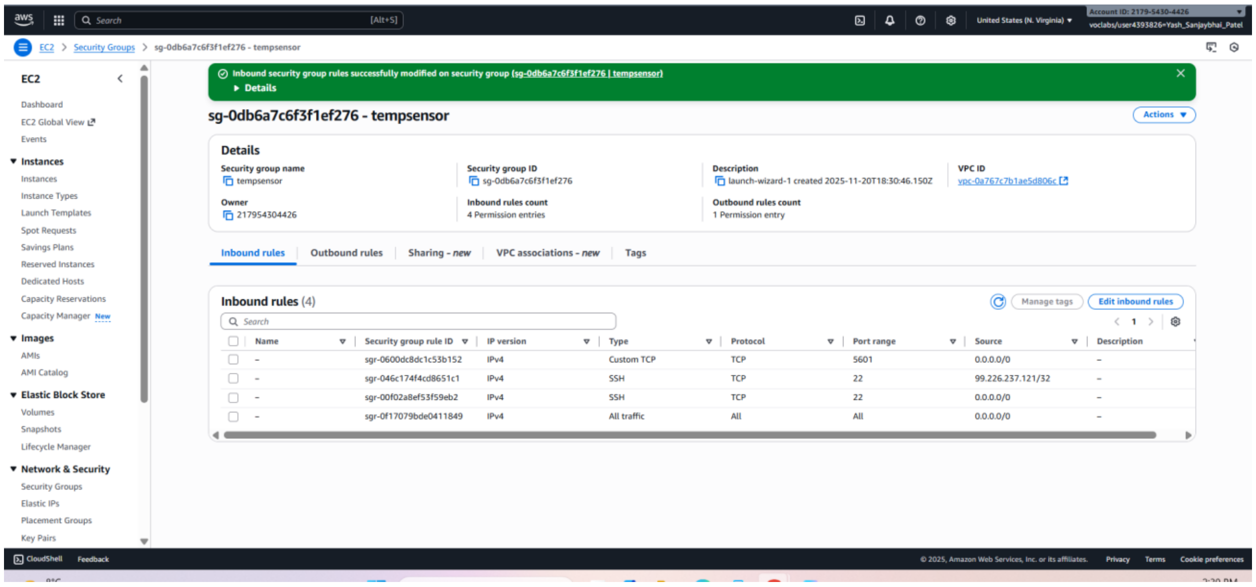
Learn more

IAM Role –	**Subnet ID** subnet-0733a97bff10980b1	**Auto Scaling Group name** –
IMDSv2 Required	**Instance ARN** arn:aws:ec2:us-east-1:590183982819:instance/i-0352f68429cb3c8e5	**Managed** false
Operator –		

Security details

IAM Role –	**Owner ID** 590183982819	**Launch time** Thu Nov 13 2025 18:44:59 GMT-0500 (Eastern Standard Time)
Security groups sg-0436b1f31a451845e (Tempsensor)		

Inbound rules


Inbound security group rules successfully modified on security group (sg-0db6a7c6f3f1ef276 | tempsensor)

sg-0db6a7c6f3f1ef276 - tempsensor

Details

Security group name	Security group ID	Description	VPC ID
tempsensor	sg-0db6a7c6f3f1ef276	launch-wizard-1 created 2025-11-20T18:30:46.150Z	vpc-0a767c7b1ae5d806c
Owner	**Inbound rules count**	**Outbound rules count**	
217954304426	4 Permission entries	1 Permission entry	

Inbound rules (4)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
–	sgr-0600dc8dc1c53b152	IPv4	Custom TCP	TCP	5601	0.0.0.0/0	–
–	sgr-046c174f4cd8651c1	IPv4	SSH	TCP	22	99.226.237.121/32	–
–	sgr-00f02a8ef53f59eb2	IPv4	SSH	TCP	22	0.0.0.0/0	–
–	sgr-0f17079bde0411849	IPv4	All traffic	All	All	0.0.0.0/0	–

Mosquitto Installation & Configuration

Mosquitto was installed on the EC2 instance to receive MQTT traffic mirrored from the Raspberry Pi. Changes I made:

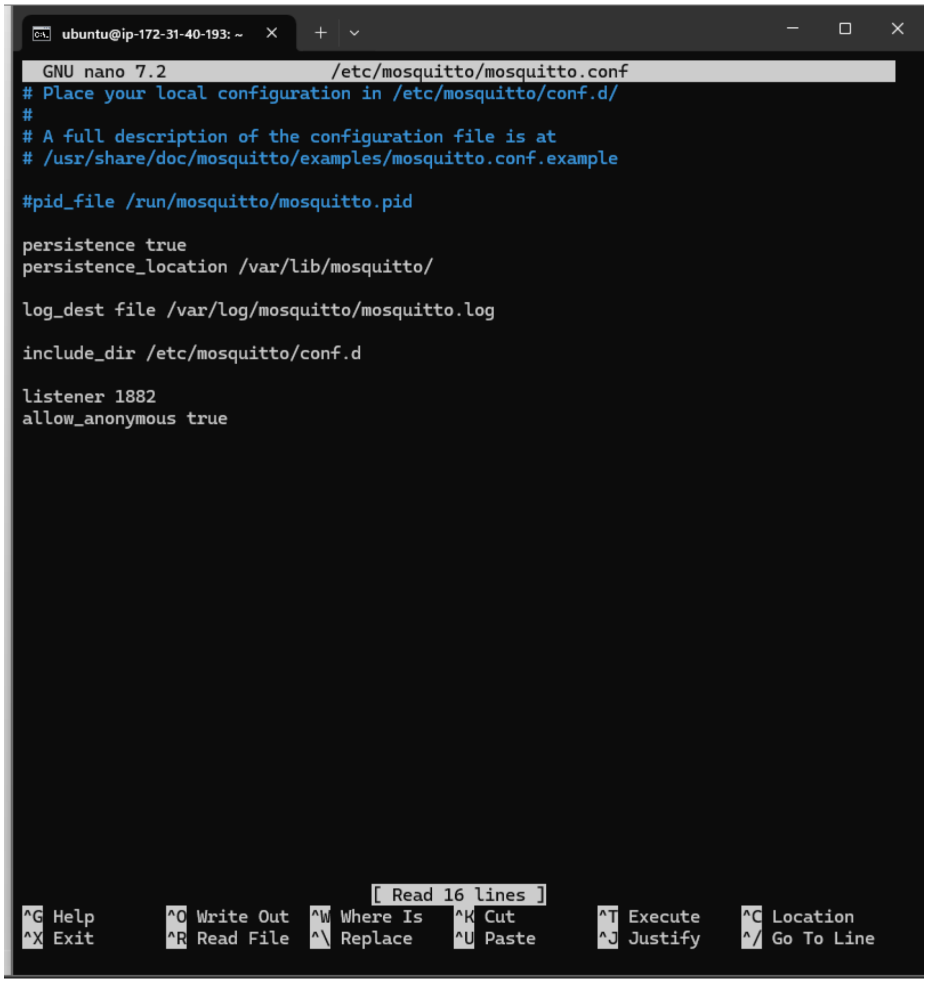
Added a listener on port 1883

Enabled allow_anonymous true for testing

After starting Mosquitto, I verified it with:

sudo systemctl status mosquitto

sudo tail -f /var/log/mosquitto/mosquitto.log

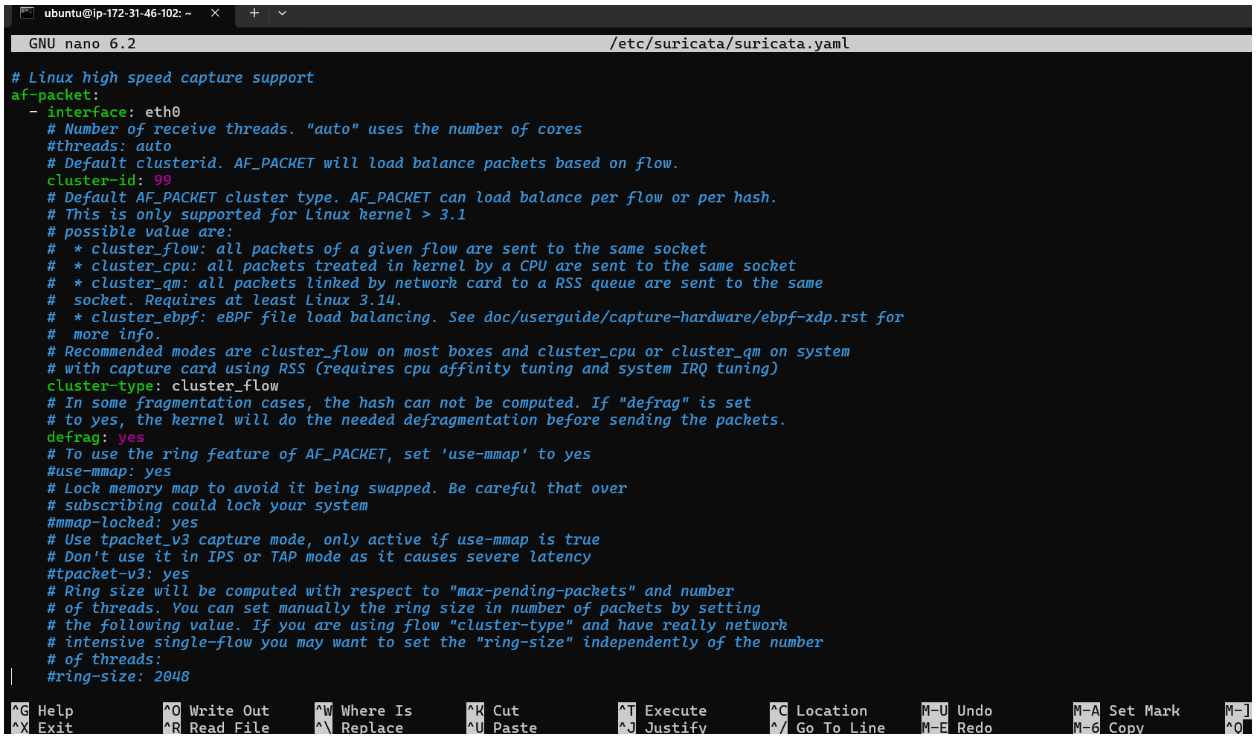


```
  GNU nano 7.2                /etc/mosquitto/mosquitto.conf
# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

#pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1882
allow_anonymous true

                                  [ Read 16 lines ]
^G Help          ^O Write Out  ^W Where Is  ^K Cut         ^T Execute   ^C Location
^X Exit          ^R Read File  ^\ Replace   ^U Paste       ^J Justify   ^/ Go To Line
```

Suricata Installation & Custom Rule Creation

We installed Suricata and updated the configuration file to use my EC2 network interface.

Thenwe created custom Suricata detection rules.



```
ubuntu@ip-172-31-46-102: ~

  GNU nano 6.2                                   /etc/suricata/suricata.yaml

# Linux high speed capture support
af-packet:
  - interface: eth0
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    #  * cluster_flow: all packets of a given flow are sent to the same socket
    #  * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    #  * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #  socket. Requires at least Linux 3.14.
    #  * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
    #  more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
    cluster-type: cluster_flow
    # In some fragmentation cases, the hash can not be computed. If "defrag" is set
    # to yes, the kernel will do the needed defragmentation before sending the packets.
    defrag: yes
    # To use the ring feature of AF_PACKET, set 'use-mmap' to yes
    #use-mmap: yes
    # Lock memory map to avoid it being swapped. Be careful that over
    # subscribing could lock your system
    #mmap-locked: yes
    # Use tpacket_v3 capture mode, only active if use-mmap is true
    # Don't use it in IPS or TAP mode as it causes severe latency
    #tpacket-v3: yes
    # Ring size will be computed with respect to "max-pending-packets" and number
    # of threads. You can set manually the ring size in number of packets by setting
    # the following value. If you are using flow "cluster-type" and have really network
    # intensive single-flow you may want to set the "ring-size" independently of the number
    # of threads:
    #ring-size: 2048

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-]
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q
```

Custom Rules I Added -

High MQTT Traffic (DDoS detection)

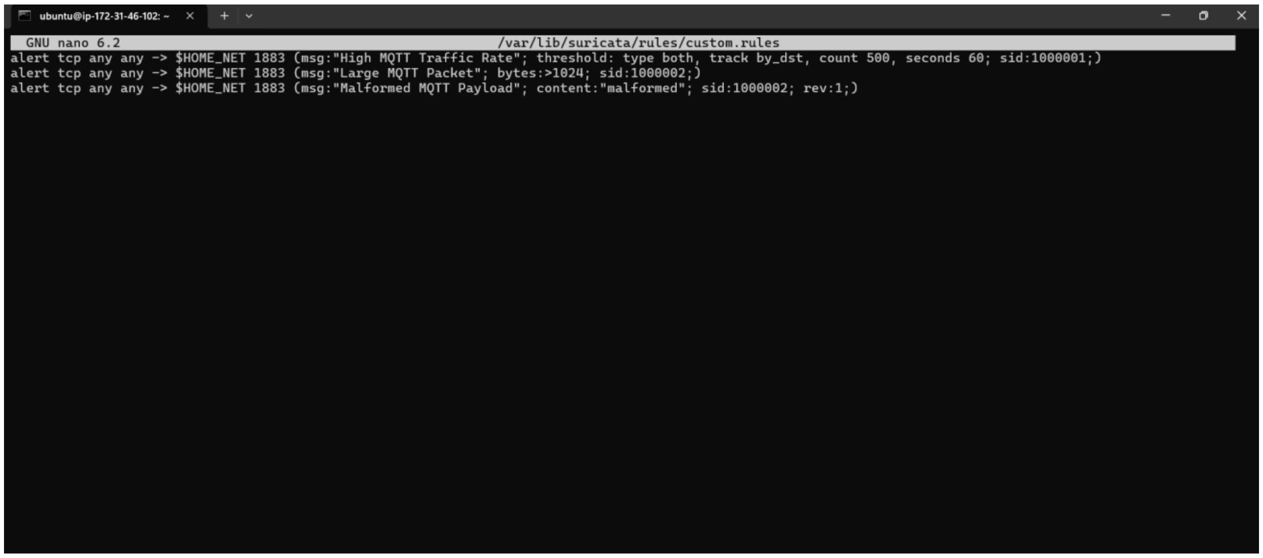
Triggers when many packets hit port 1883 in a short time.

Malformed MQTT Payload

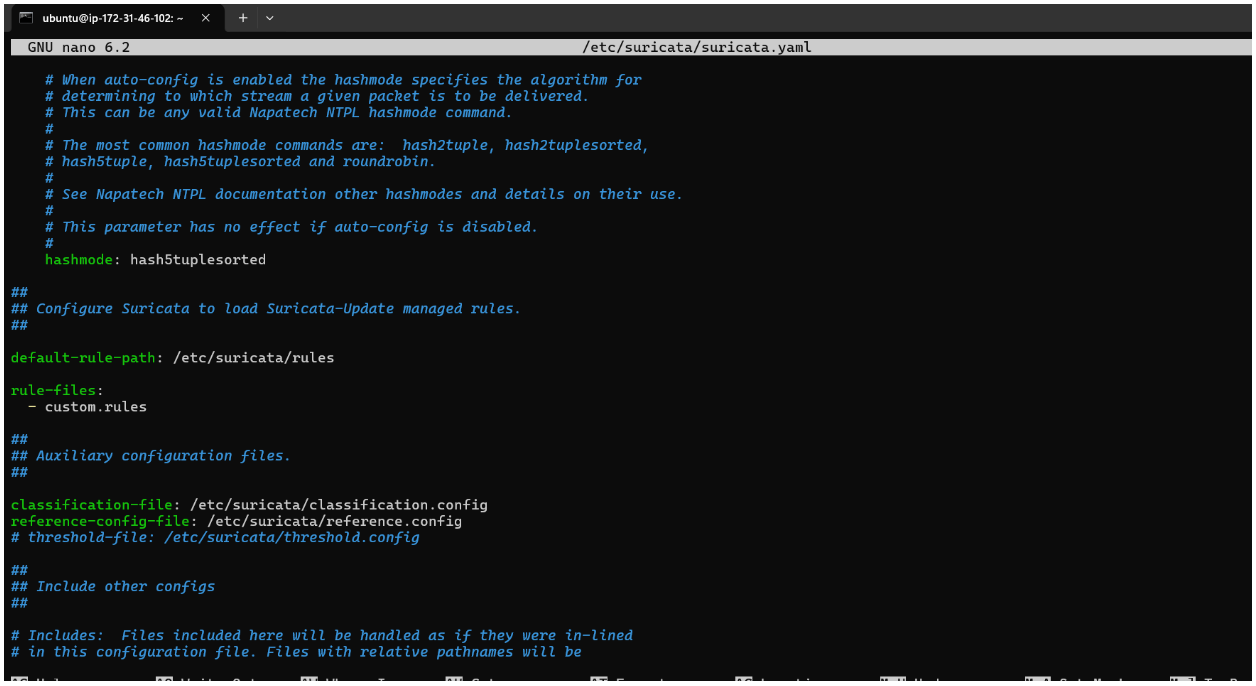
Triggers when invalid or broken JSON is sent.

These rules were saved under:

/var/lib/suricata/rules/custom.rules



```
ubuntu@ip-172-31-46-102: ~
  GNU nano 6.2                                       /var/lib/suricata/rules/custom.rules
alert tcp any any -> $HOME_NET 1883 (msg:"High MQTT Traffic Rate"; threshold: type both, track by_dst, count 500, seconds 60; sid:1000001;)
alert tcp any any -> $HOME_NET 1883 (msg:"Large MQTT Packet"; bytes:>1024; sid:1000002;)
alert tcp any any -> $HOME_NET 1883 (msg:"Malformed MQTT Payload"; content:"malformed"; sid:1000002; rev:1;)
```



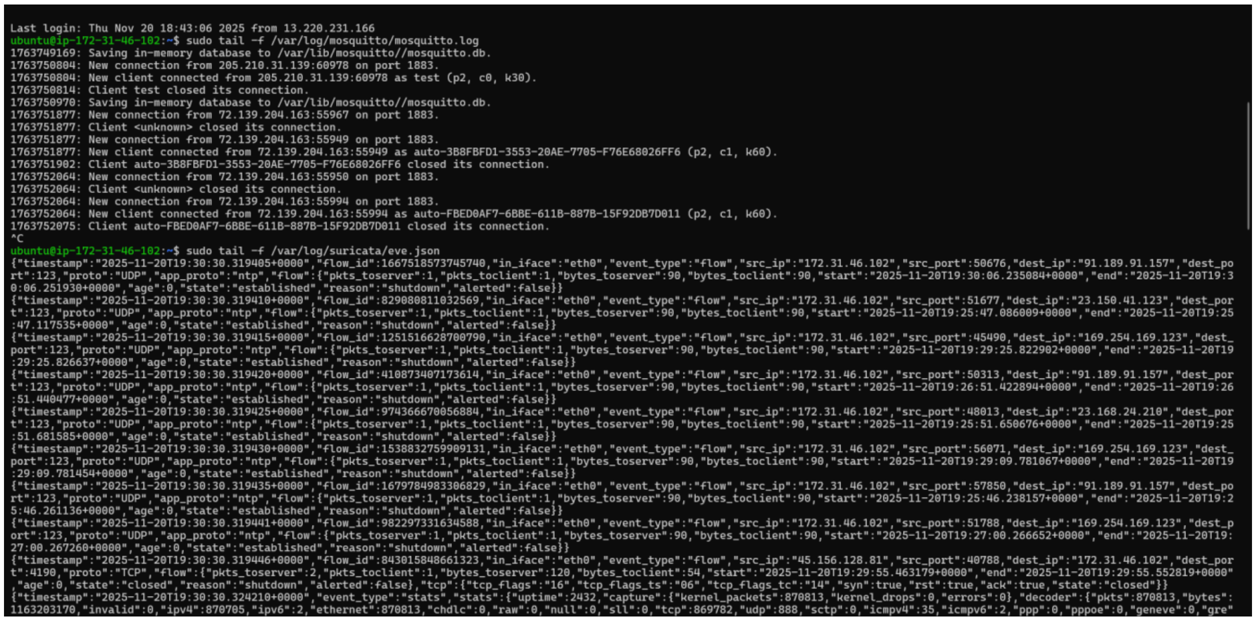
```
ubuntu@ip-172-31-46-102: ~
  GNU nano 6.2                                       /etc/suricata/suricata.yaml

    # When auto-config is enabled the hashmode specifies the algorithm for
    # determining to which stream a given packet is to be delivered.
    # This can be any valid Napatech NTPL hashmode command.
    #
    # The most common hashmode commands are:  hash2tuple, hash2tuplesorted,
    # hash5tuple, hash5tuplesorted and roundrobin.
    #
    # See Napatech NTPL documentation other hashmodes and details on their use.
    #
    # This parameter has no effect if auto-config is disabled.
    #
    hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

default-rule-path: /etc/suricata/rules

rule-files:
  - custom.rules

##
## Auxiliary configuration files.
##

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
## Include other configs
##

# Includes:  Files included here will be handled as if they were in-lined
# in this configuration file. Files with relative pathnames will be
```

Then we restarted Suricata.

After this step, Suricata started generating alerts inside:

/var/log/suricata/eve.json

We verified that the alerts were being logged correctly when we simulated the attacks.



```
Last login: Thu Nov 20 18:43:06 2025 from 13.220.231.166
ubuntu@ip-172-31-46-102:~$ sudo tail -f /var/log/mosquitto/mosquitto.log
1763749169: Saving in-memory database to /var/lib/mosquitto//mosquitto.db.
1763750804: New connection from 205.210.31.139:60978 on port 1883.
1763750804: New client connected from 205.210.31.139:60978 as test (p2, c0, k30).
1763750814: Client test closed its connection.
1763750970: Saving in-memory database to /var/lib/mosquitto//mosquitto.db.
1763751877: New connection from 72.139.204.163:55967 on port 1883.
1763751877: Client <unknown> closed its connection.
1763751877: New connection from 72.139.204.163:55949 on port 1883.
1763751877: New client connected from 72.139.204.163:55949 as auto-3B8FBFD1-3553-20AE-7705-F76E68026FF6 (p2, c1, k60).
1763751902: Client auto-3B8FBFD1-3553-20AE-7705-F76E68026FF6 closed its connection.
1763752064: New connection from 72.139.204.163:55950 on port 1883.
1763752064: Client <unknown> closed its connection.
1763752064: New connection from 72.139.204.163:55994 on port 1883.
1763752064: New client connected from 72.139.204.163:55994 as auto-FBED0AF7-6BBE-611B-887B-15F92DB7D011 (p2, c1, k60).
1763752075: Client auto-FBED0AF7-6BBE-611B-887B-15F92DB7D011 closed its connection.
^C
ubuntu@ip-172-31-46-102:~$ sudo tail -f /var/log/suricata/eve.json
{"timestamp":"2025-11-20T19:30:30.319405+0000","flow_id":1667518573745740,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":50676,"dest_ip":"91.189.91.157","dest_po
rt":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:30:06.235084+0000","end":"2025-11-20T19:3
0:06.251930+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319410+0000","flow_id":829080811032569,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":51677,"dest_ip":"23.150.41.123","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:47.086009+0000","end":"2025-11-20T19:25
:47.117535+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319415+0000","flow_id":1251516628700790,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":45490,"dest_ip":"169.254.169.123","dest_
port":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:29:25.822902+0000","end":"2025-11-20T19
:29:25.826637+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319420+0000","flow_id":410873407173614,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":50313,"dest_ip":"91.189.91.157","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:26:51.422894+0000","end":"2025-11-20T19:26
:51.440477+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319425+0000","flow_id":974366670056884,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":48013,"dest_ip":"23.168.24.210","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:51.650676+0000","end":"2025-11-20T19:25
:51.681585+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319430+0000","flow_id":1538832759909131,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":56071,"dest_ip":"169.254.169.123","dest_
port":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:29:09.781067+0000","end":"2025-11-20T19
:29:09.781454+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319435+0000","flow_id":1679784983306829,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":57850,"dest_ip":"91.189.91.157","dest_po
rt":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:46.238157+0000","end":"2025-11-20T19:2
5:46.261136+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319441+0000","flow_id":982297331634588,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":51788,"dest_ip":"169.254.169.123","dest_p
ort":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:27:00.266652+0000","end":"2025-11-20T19:
27:00.267260+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319446+0000","flow_id":843015848661323,"in_iface":"eth0","event_type":"flow","src_ip":"45.156.128.81","src_port":40788,"dest_ip":"172.31.46.102","dest_por
t":4190,"proto":"TCP","flow":{"pkts_toserver":2,"pkts_toclient":1,"bytes_toserver":120,"bytes_toclient":54,"start":"2025-11-20T19:29:55.463179+0000","end":"2025-11-20T19:29:55.552819+0000"
,"age":0,"state":"closed","reason":"shutdown","alerted":false},"tcp":{"tcp_flags":"16","tcp_flags_ts":"06","tcp_flags_tc":"14","syn":true,"rst":true,"ack":true,"state":"closed"}}
{"timestamp":"2025-11-20T19:30:30.324210+0000","event_type":"stats","stats":{"uptime":2432,"capture":{"kernel_packets":870813,"kernel_drops":0,"errors":0},"decoder":{"pkts":870813,"bytes":
1163203170,"invalid":0,"ipv4":870705,"ipv6":2,"ethernet":870813,"chdlc":0,"raw":0,"null":0,"sll":0,"tcp":869782,"udp":888,"sctp":0,"icmpv4":35,"icmpv6":2,"ppp":0,"pppoe":0,"geneve":0,"gre"
```

Filebeat Setup

Filebeat was installed to ship Suricata's eve.json logs into Elasticsearch.

Changes made in filebeat.yml:

Enabled log input

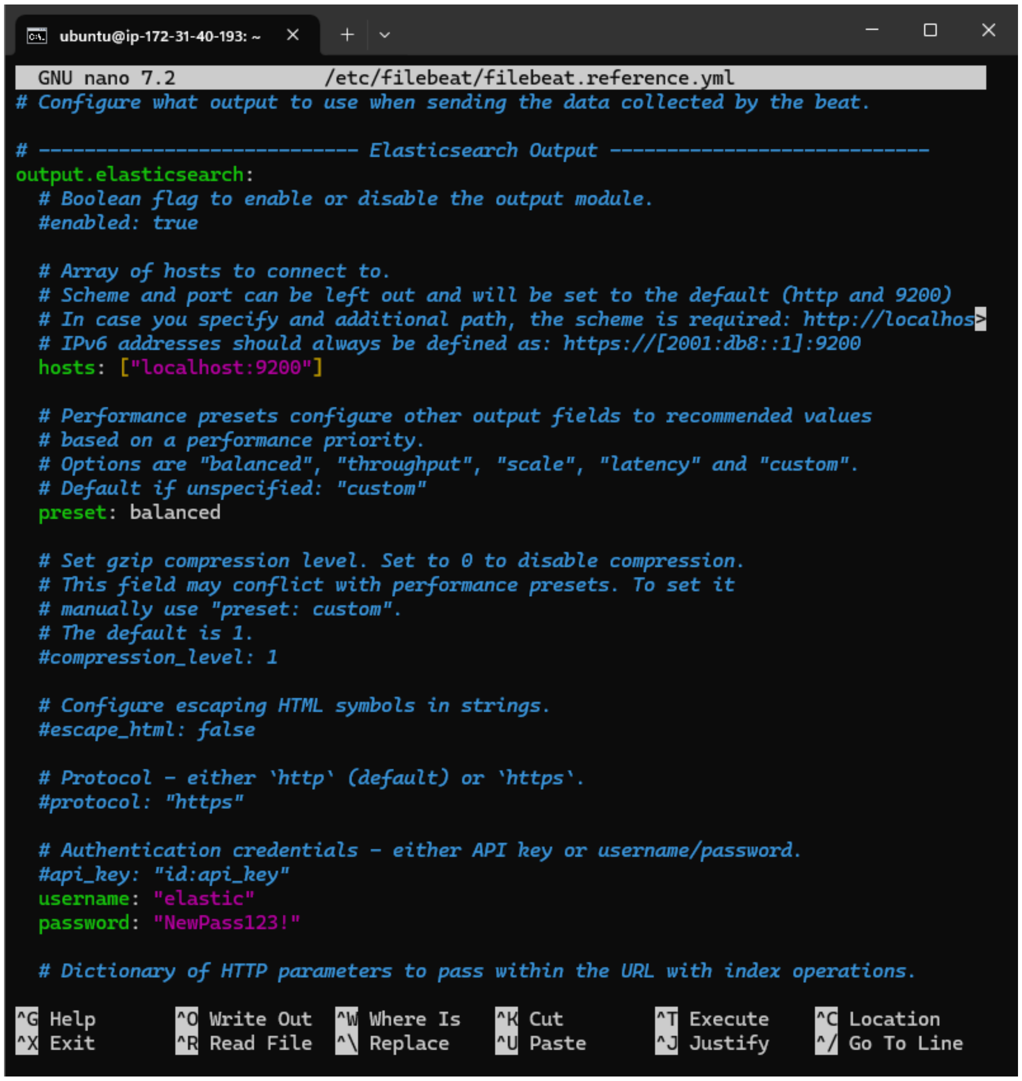
Set Suricata log path

Provided Elasticsearch username & password

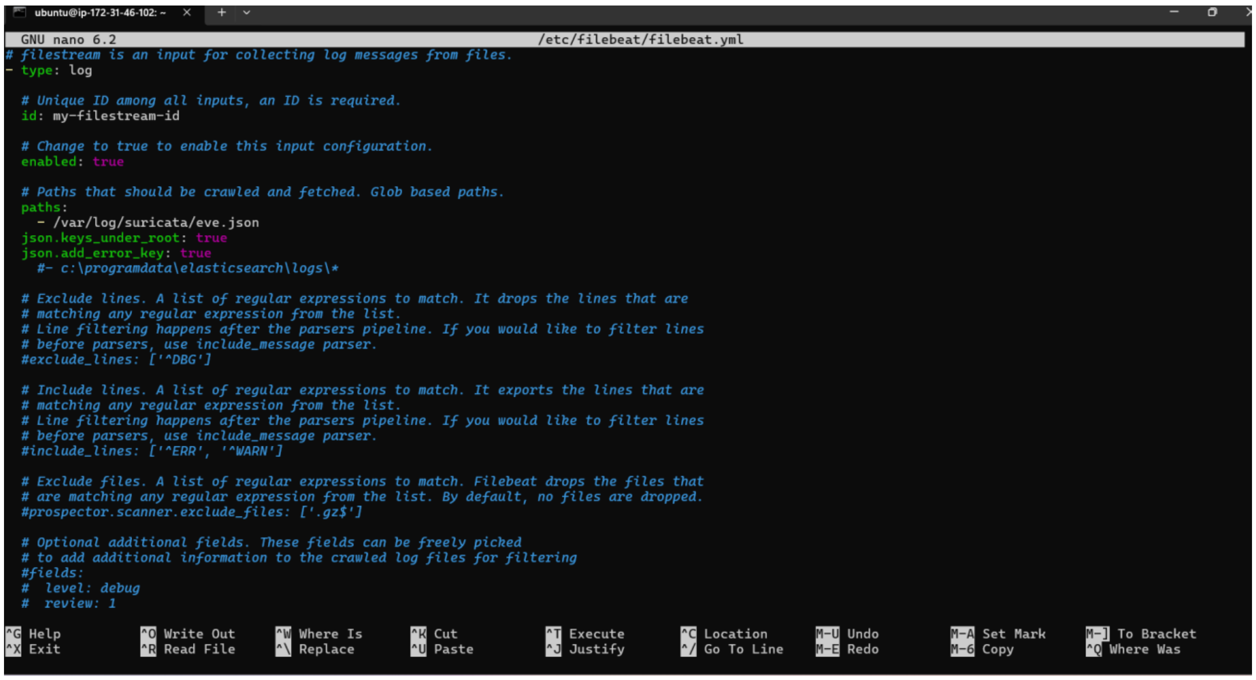
Defined a custom index format

After starting Filebeat, I confirmed it was sending data by checking:

sudo systemctl status filebeat



```
  GNU nano 7.2              /etc/filebeat/filebeat.reference.yml
# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------
output.elasticsearch:
  # Boolean flag to enable or disable the output module.
  #enabled: true

  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to the default (http and 9200)
  # In case you specify and additional path, the scheme is required: http://localhos>
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
  hosts: ["localhost:9200"]

  # Performance presets configure other output fields to recommended values
  # based on a performance priority.
  # Options are "balanced", "throughput", "scale", "latency" and "custom".
  # Default if unspecified: "custom"
  preset: balanced

  # Set gzip compression level. Set to 0 to disable compression.
  # This field may conflict with performance presets. To set it
  # manually use "preset: custom".
  # The default is 1.
  #compression_level: 1

  # Configure escaping HTML symbols in strings.
  #escape_html: false

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "elastic"
  password: "NewPass123!"

  # Dictionary of HTTP parameters to pass within the URL with index operations.

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



```
ubuntu@ip-172-31-46-102: ~
  GNU nano 6.2                                          /etc/filebeat/filebeat.yml
# filestream is an input for collecting log messages from files.
- type: log

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/suricata/eve.json
  json.keys_under_root: true
  json.add_error_key: true
    #- c:\programdata\elasticsearch\logs\*

  # Exclude lines. A list of regular expressions to match. It drops the lines that are
  # matching any regular expression from the list.
  # Line filtering happens after the parsers pipeline. If you would like to filter lines
  # before parsers, use include_message parser.
  #exclude_lines: ['^DBG']

  # Include lines. A list of regular expressions to match. It exports the lines that are
  # matching any regular expression from the list.
  # Line filtering happens after the parsers pipeline. If you would like to filter lines
  # before parsers, use include_message parser.
  #include_lines: ['^ERR', '^WARN']

  # Exclude files. A list of regular expressions to match. Filebeat drops the files that
  # are matching any regular expression from the list. By default, no files are dropped.
  #prospector.scanner.exclude_files: ['.gz$']

  # Optional additional fields. These fields can be freely picked
  # to add additional information to the crawled log files for filtering
  #fields:
  #  level: debug
  #  review: 1

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

After starting Filebeat, we confirmed it was sending data by checking:

sudo systemctl status filebeat

Elasticsearch indices also appeared in Kibana.

Elasticsearch & Kibana Setup

We installed Elasticsearch and Kibana on the same EC2 instance.

Changes made:

Allowed external IP access (0.0.0.0)

Enabled security and created the password “NewPass123!”

Configured Kibana to connect to Elasticsearch

Once Kibana started, we accessed it by visiting:

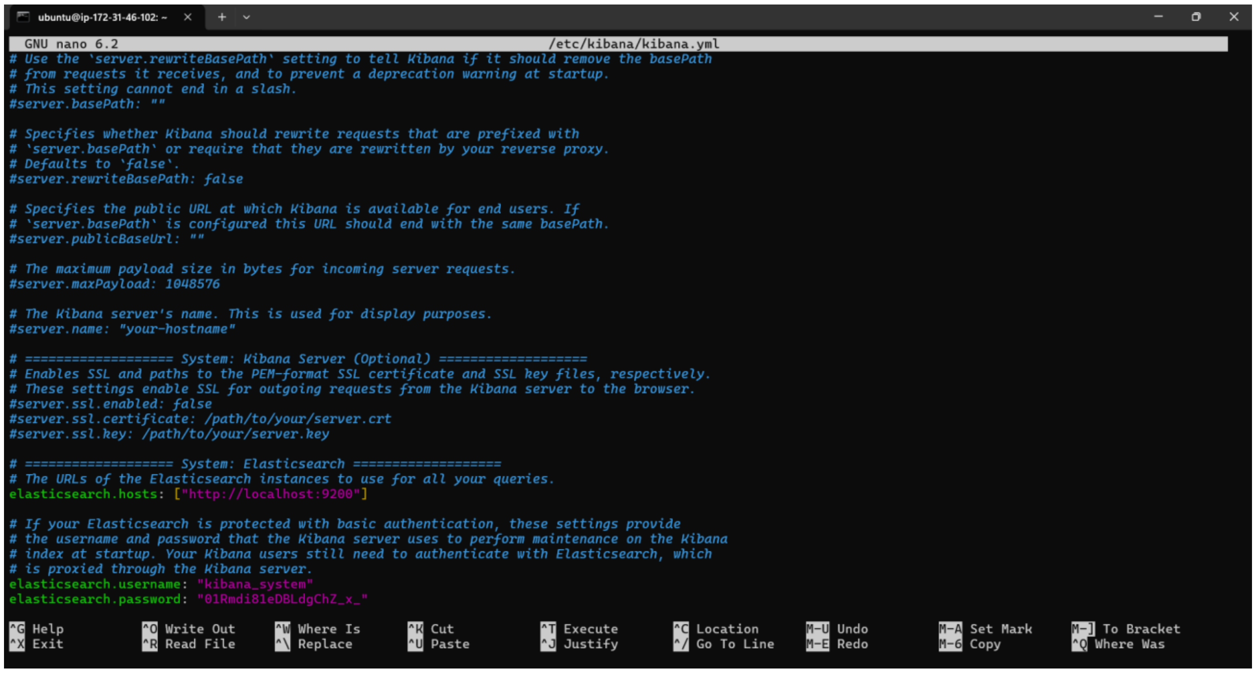
http://<Elastic-IP>:5601

We logged in using:

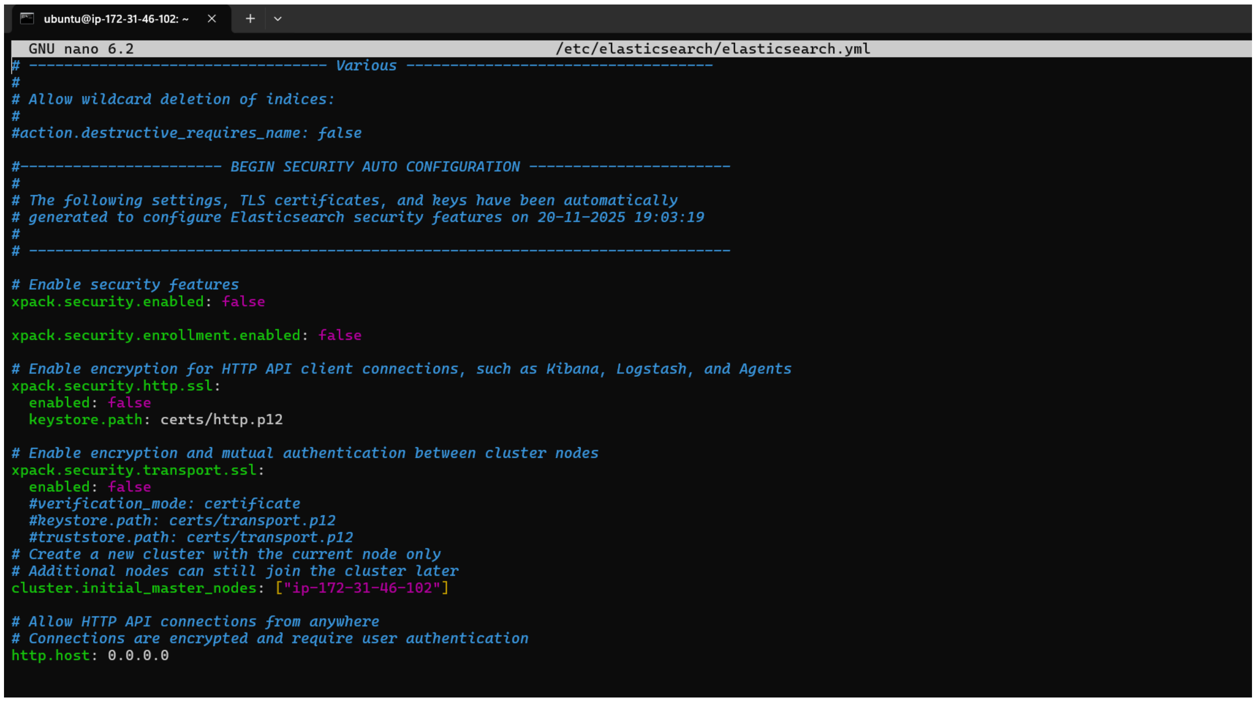
Username: elastic

Password: NewPass123!

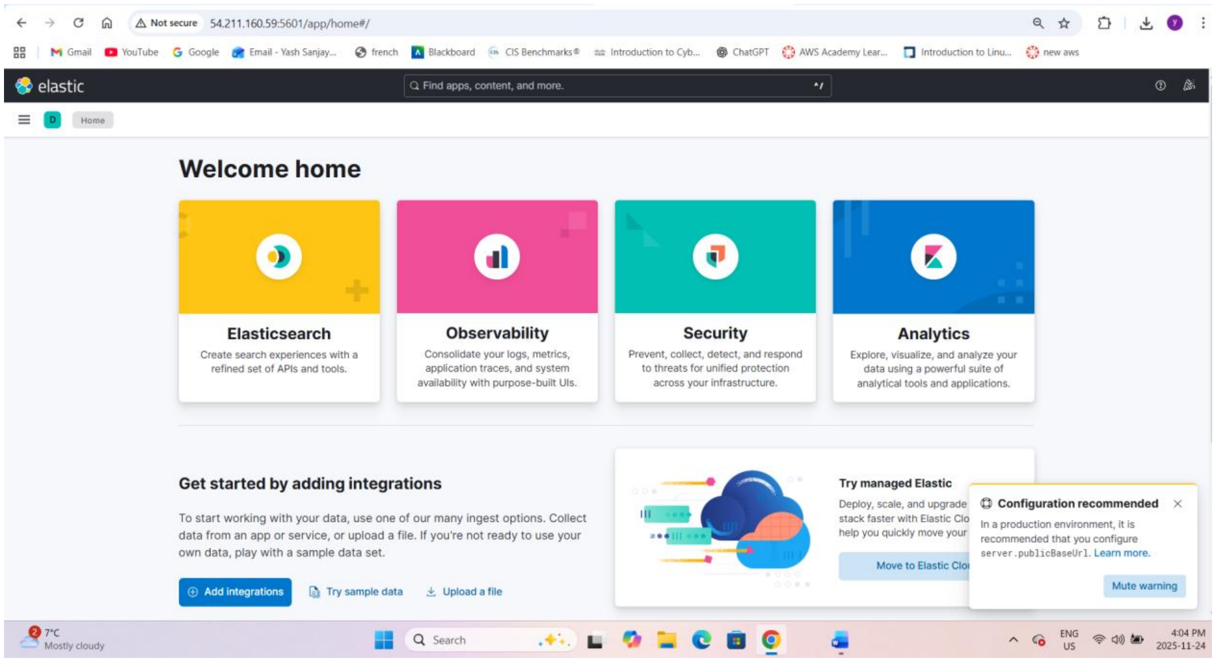
Kibana successfully showed incoming Suricata logs.



```
  GNU nano 6.2                                                    /etc/kibana/kibana.yml
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# Defaults to `false`.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# =================== System: Kibana Server (Optional) ===================
# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# =================== System: Elasticsearch ===================
# The URLs of the Elasticsearch instances to use for all your queries.
elasticsearch.hosts: ["http://localhost:9200"]

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "kibana_system"
elasticsearch.password: "01Rmdi81eDBLdgChZ_x_"

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```



```
  GNU nano 6.2                                          /etc/elasticsearch/elasticsearch.yml
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false

#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 20-11-2025 19:03:19
#
# --------------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: false

xpack.security.enrollment.enabled: false

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: false
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: false
  #verification_mode: certificate
  #keystore.path: certs/transport.p12
  #truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["ip-172-31-46-102"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0
```


Not secure 54.211.160.59:5601/app/home#/
elastic
Find apps, content, and more.
Home

Welcome home

Elasticsearch
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

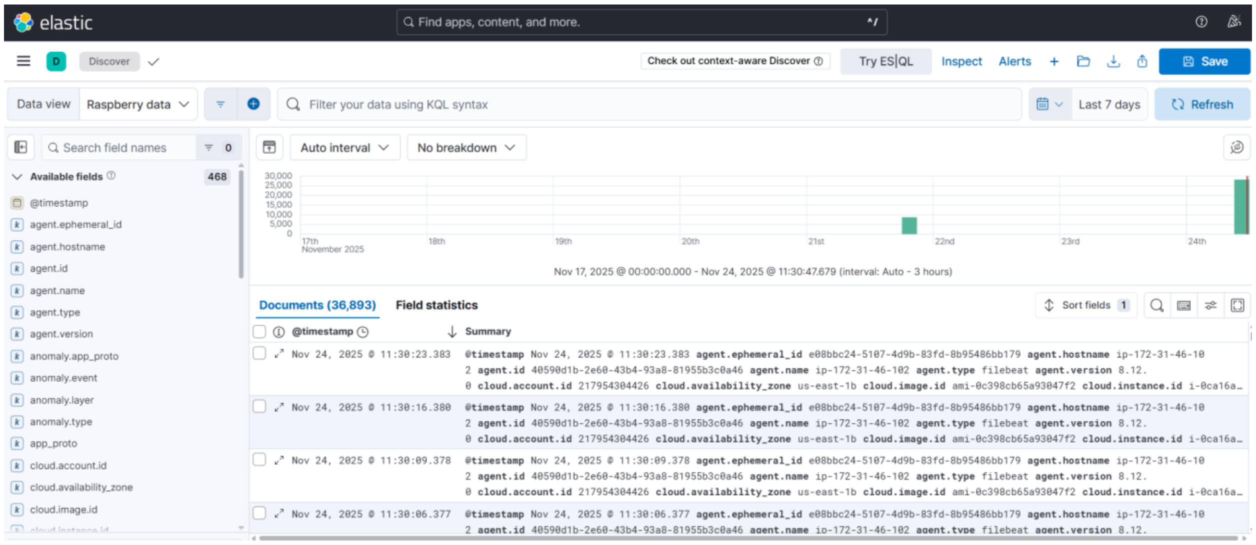
Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, play with a sample data set.

Deploy, scale, and upgrade stack faster with Elastic Clo help you quickly move your

Configuration recommended
In a production environment, it is recommended that you configure `server.publicBaseUrl`. Learn more.



Nov 17, 2025 @ 00:00:00.000 - Nov 24, 2025 @ 11:30:47.679 (interval: Auto - 3 hours)

```
Nov 24, 2025 @ 11:30:23.383  @timestamp Nov 24, 2025 @ 11:30:23.383 agent.ephemeral_id e08bbc24-5107-4d9b-83fd-8b95486bb179 agent.hostname ip-172-31-46-10
                             2 agent.id 40590d1b-2e60-43b4-93a8-81955b3c0a46 agent.name ip-172-31-46-102 agent.type filebeat agent.version 8.12.
                             0 cloud.account.id 217954304426 cloud.availability_zone us-east-1b cloud.image.id ami-0c398cb65a93047f2 cloud.instance.id i-0ca16a…
Nov 24, 2025 @ 11:30:16.380  @timestamp Nov 24, 2025 @ 11:30:16.380 agent.ephemeral_id e08bbc24-5107-4d9b-83fd-8b95486bb179 agent.hostname ip-172-31-46-10
                             2 agent.id 40590d1b-2e60-43b4-93a8-81955b3c0a46 agent.name ip-172-31-46-102 agent.type filebeat agent.version 8.12.
                             0 cloud.account.id 217954304426 cloud.availability_zone us-east-1b cloud.image.id ami-0c398cb65a93047f2 cloud.instance.id i-0ca16a…
Nov 24, 2025 @ 11:30:09.378  @timestamp Nov 24, 2025 @ 11:30:09.378 agent.ephemeral_id e08bbc24-5107-4d9b-83fd-8b95486bb179 agent.hostname ip-172-31-46-10
                             2 agent.id 40590d1b-2e60-43b4-93a8-81955b3c0a46 agent.name ip-172-31-46-102 agent.type filebeat agent.version 8.12.
                             0 cloud.account.id 217954304426 cloud.availability_zone us-east-1b cloud.image.id ami-0c398cb65a93047f2 cloud.instance.id i-0ca16a…
Nov 24, 2025 @ 11:30:06.377  @timestamp Nov 24, 2025 @ 11:30:06.377 agent.ephemeral_id e08bbc24-5107-4d9b-83fd-8b95486bb179 agent.hostname ip-172-31-46-10
                             2 agent.id 40590d1b-2e60-43b4-93a8-81955b3c0a46 agent.name ip-172-31-46-102 agent.type filebeat agent.version 8.12.
```

Raspberry Pi Code Modification

I updated my Python script on the Raspberry Pi to mirror all MQTT traffic:

The Raspberry Pi now sends:

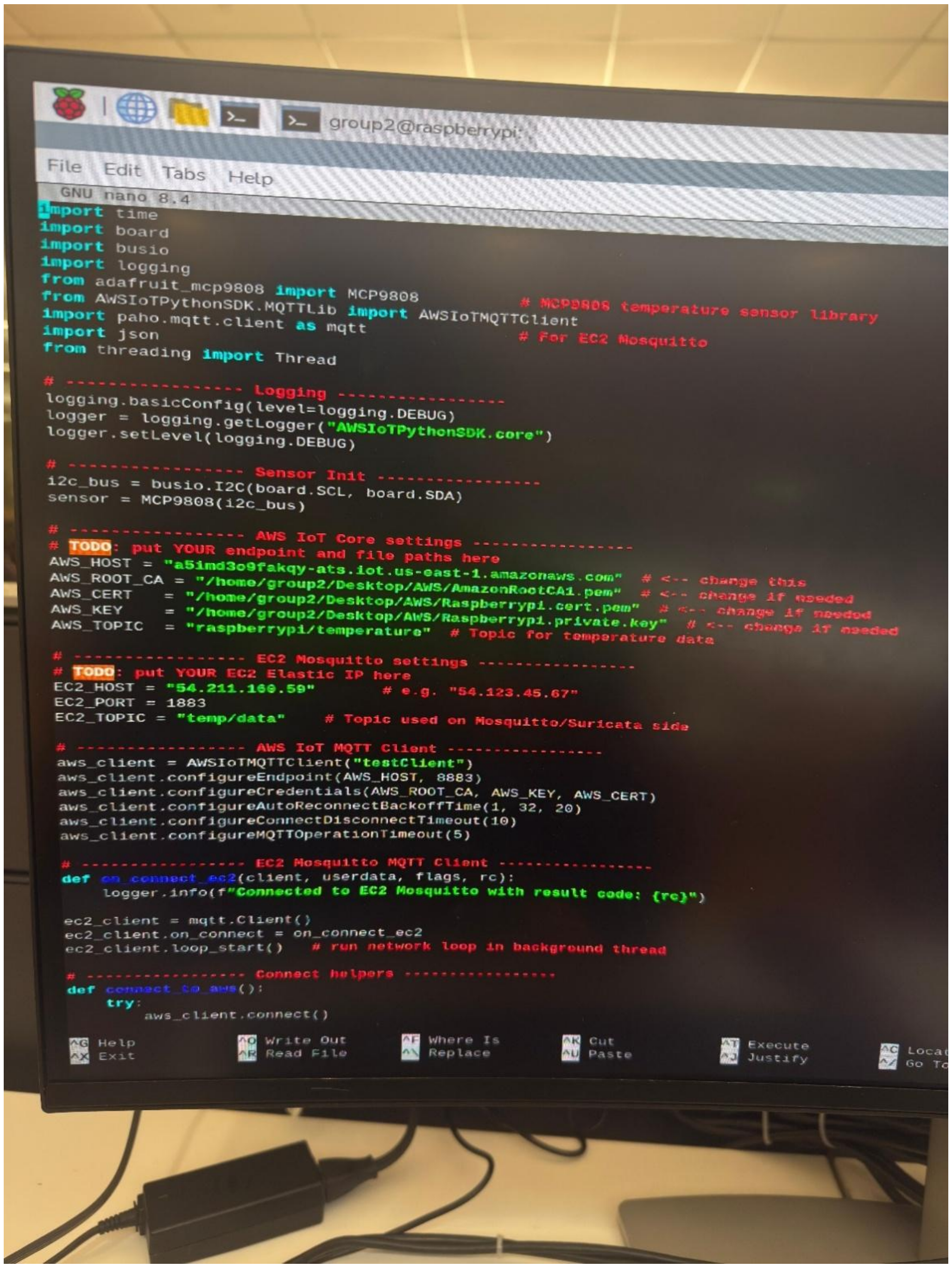
Normal temperature data → AWS IoT Core

Normal temperature data → EC2 Mosquitto

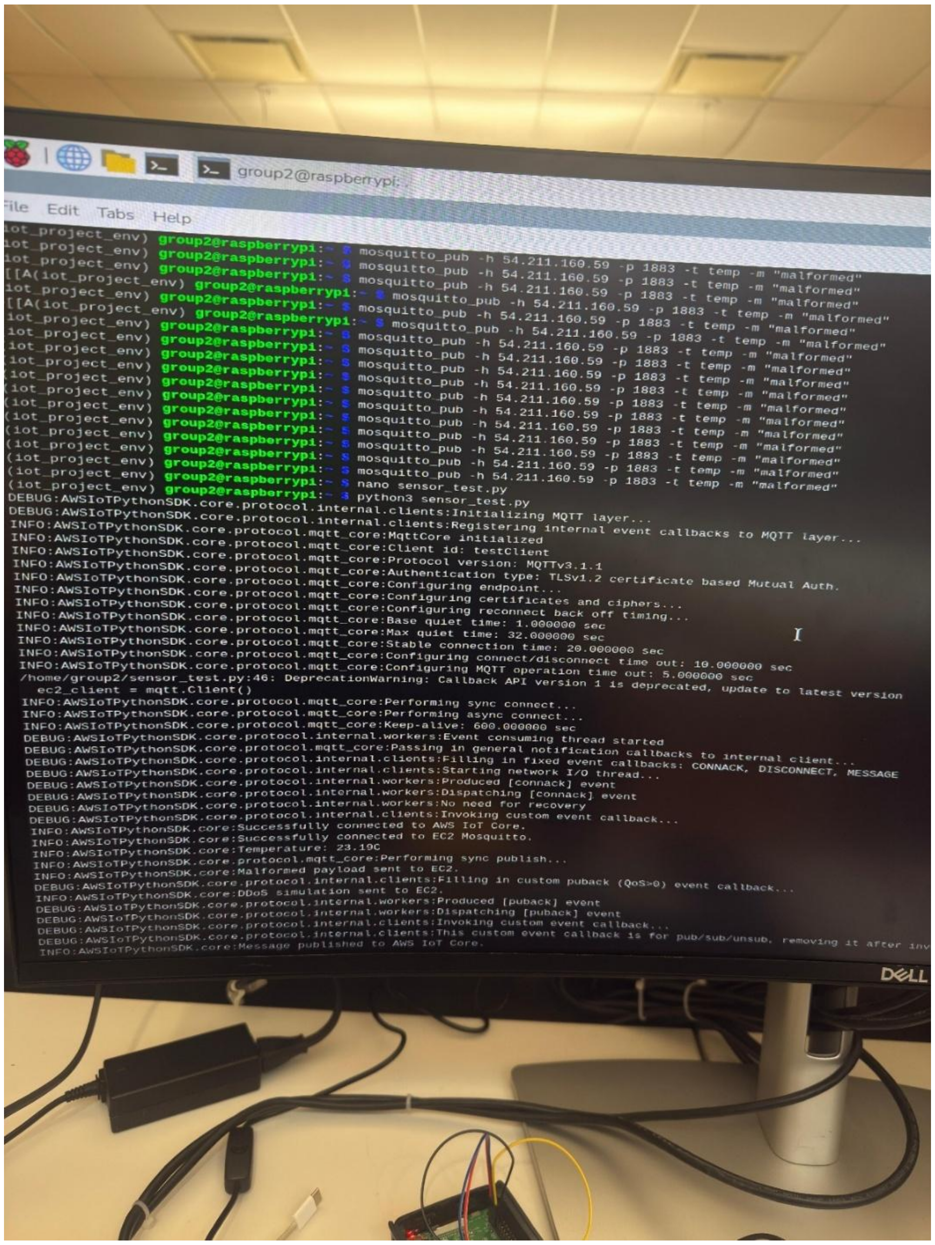
DDoS simulated messages → EC2 Mosquitto

Malformed payloads → EC2 Mosquitto

These changes were required so Suricata could detect attacks.



```
import time
import board
import busio
import logging
from adafruit_mcp9808 import MCP9808
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient              # MCP9808 temperature sensor library
import paho.mqtt.client as mqtt
import json                                                       # For EC2 Mosquitto
from threading import Thread

# ----------------- Logging ----------------
logging.basicConfig(level=logging.DEBUG)
logger = logging.getLogger("AWSIoTPythonSDK.core")
logger.setLevel(logging.DEBUG)

# ----------------- Sensor Init ----------------
i2c_bus = busio.I2C(board.SCL, board.SDA)
sensor = MCP9808(i2c_bus)

# ----------------- AWS IoT Core settings ----------------
# TODO: put YOUR endpoint and file paths here
AWS_HOST = "a5imd3o9fakqy-ats.iot.us-east-1.amazonaws.com"  # <-- change this
AWS_ROOT_CA = "/home/group2/Desktop/AWS/AmazonRootCA1.pem"  # <-- change if needed
AWS_CERT   = "/home/group2/Desktop/AWS/Raspberrypi.cert.pem"  # <-- change if needed
AWS_KEY    = "/home/group2/Desktop/AWS/Raspberrypi.private.key"  # <-- change if needed
AWS_TOPIC  = "raspberrypi/temperature"  # Topic for temperature data

# ----------------- EC2 Mosquitto settings ----------------
# TODO: put YOUR EC2 Elastic IP here
EC2_HOST = "54.211.169.59"        # e.g. "54.123.45.67"
EC2_PORT = 1883
EC2_TOPIC = "temp/data"     # Topic used on Mosquitto/Suricata side

# ----------------- AWS IoT MQTT Client ----------------
aws_client = AWSIoTMQTTClient("testClient")
aws_client.configureEndpoint(AWS_HOST, 8883)
aws_client.configureCredentials(AWS_ROOT_CA, AWS_KEY, AWS_CERT)
aws_client.configureAutoReconnectBackoffTime(1, 32, 20)
aws_client.configureConnectDisconnectTimeout(10)
aws_client.configureMQTTOperationTimeout(5)

# ----------------- EC2 Mosquitto MQTT Client ----------------
def on_connect_ec2(client, userdata, flags, rc):
    logger.info(f"Connected to EC2 Mosquitto with result code: {rc}")

ec2_client = mqtt.Client()
ec2_client.on_connect = on_connect_ec2
ec2_client.loop_start()   # run network loop in background thread

# ----------------- Connect helpers ----------------
def connect_to_aws():
    try:
        aws_client.connect()
```



```
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
[[A(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
[[A(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ mosquitto_pub -h 54.211.160.59 -p 1883 -t temp -m "malformed"
(iot_project_env) group2@raspberrypi:~ $ nano sensor_test.py
(iot_project_env) group2@raspberrypi:~ $ python3 sensor_test.py
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Initializing MQTT layer...
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Registering internal event callbacks to MQTT layer...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:MqttCore initialized
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Client id: testClient
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Protocol version: MQTTv3.1.1
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Authentication type: TLSv1.2 certificate based Mutual Auth.
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Configuring endpoint...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Configuring certificates and ciphers...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Configuring reconnect back off timing...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Base quiet time: 1.000000 sec
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Max quiet time: 32.000000 sec
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Stable connection time: 20.000000 sec
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Configuring connect/disconnect time out: 10.000000 sec
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Configuring MQTT operation time out: 5.000000 sec
/home/group2/sensor_test.py:46: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  ec2_client = mqtt.Client()
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Performing sync connect...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Performing async connect...
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Keep-alive: 600.000000 sec
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:Event consuming thread started
DEBUG:AWSIoTPythonSDK.core.protocol.mqtt_core:Passing in general notification callbacks to internal client...
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Filling in fixed event callbacks: CONNACK, DISCONNECT, MESSAGE
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Starting network I/O thread...
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:Produced [connack] event
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:Dispatching [connack] event
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:No need for recovery
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Invoking custom event callback...
INFO:AWSIoTPythonSDK.core:Successfully connected to AWS IoT Core.
INFO:AWSIoTPythonSDK.core:Successfully connected to EC2 Mosquitto.
INFO:AWSIoTPythonSDK.core:Temperature: 23.19C
INFO:AWSIoTPythonSDK.core.protocol.mqtt_core:Performing sync publish...
INFO:AWSIoTPythonSDK.core:Malformed payload sent to EC2.
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Filling in custom puback (QoS>0) event callback...
INFO:AWSIoTPythonSDK.core:DDoS simulation sent to EC2.
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:Produced [puback] event
DEBUG:AWSIoTPythonSDK.core.protocol.internal.workers:Dispatching [puback] event
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:Invoking custom event callback...
DEBUG:AWSIoTPythonSDK.core.protocol.internal.clients:This custom event callback is for pub/sub/unsub, removing it after inv
INFO:AWSIoTPythonSDK.core:Message published to AWS IoT Core.
```

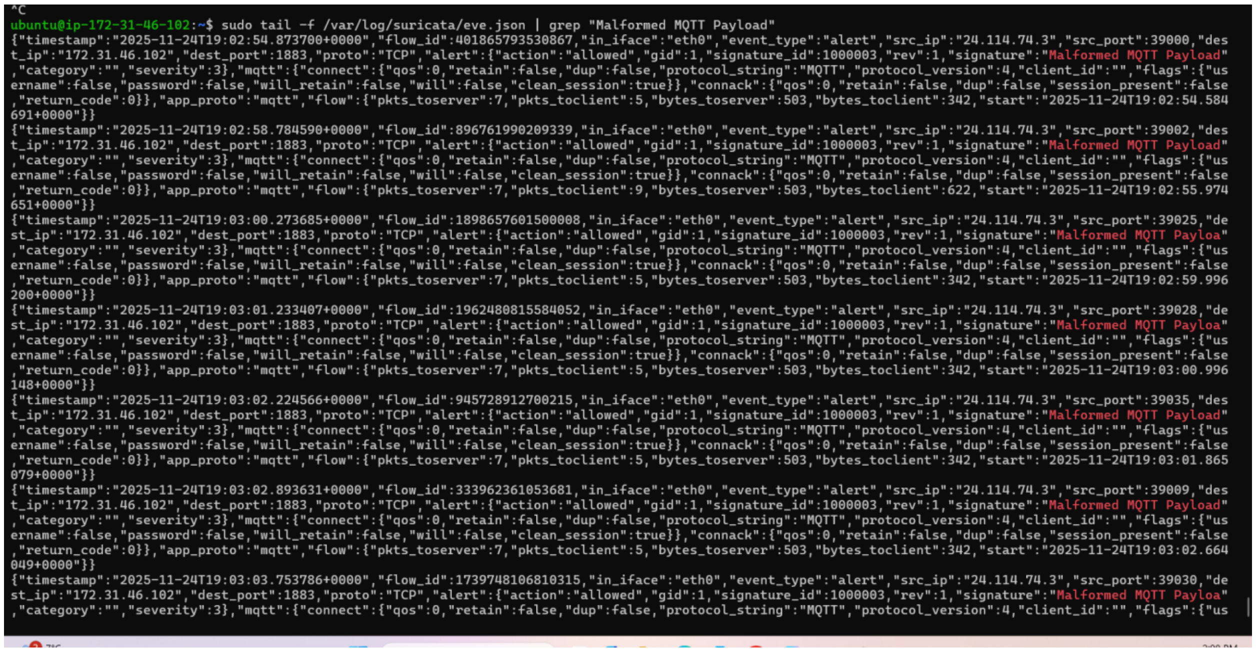
Suricata Log Results

Inside /var/log/suricata/eve.json, I observed:

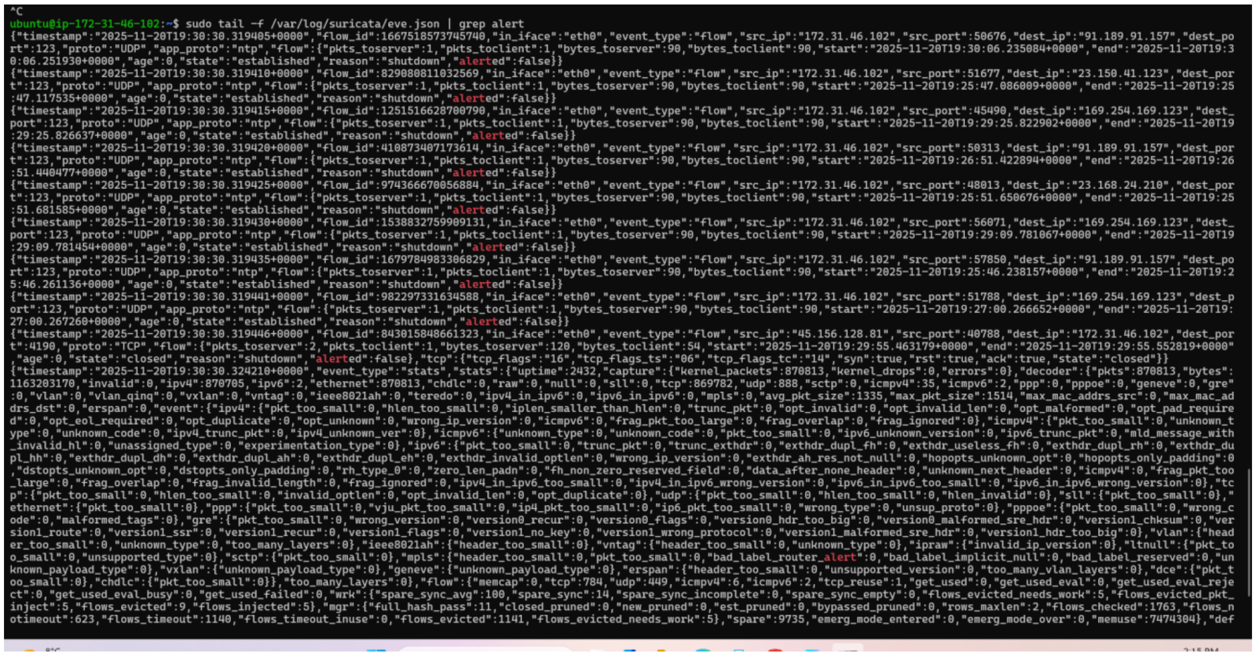
Flow logs showing MQTT communication

Alerts for malformed payloads

Alerts for high-rate traffic (DDoS simulation) Example alerts we saw: “Malformed MQTT Payload” , “High MQTT Traffic Rate”. These confirmed that our rules were working correctly.



```
^C
ubuntu@ip-172-31-46-102:~$ sudo tail -f /var/log/suricata/eve.json | grep "Malformed MQTT Payload"
{"timestamp":"2025-11-24T19:02:54.873700+0000","flow_id":401865793530867,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39000,"des
t_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payload"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":5,"bytes_toserver":503,"bytes_toclient":342,"start":"2025-11-24T19:02:54.584
691+0000"}}
{"timestamp":"2025-11-24T19:02:58.784590+0000","flow_id":896761990209339,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39002,"des
t_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payload"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":9,"bytes_toserver":503,"bytes_toclient":622,"start":"2025-11-24T19:02:55.974
651+0000"}}
{"timestamp":"2025-11-24T19:03:00.273685+0000","flow_id":1898657601500008,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39025,"de
st_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payloa"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":5,"bytes_toserver":503,"bytes_toclient":342,"start":"2025-11-24T19:02:59.996
200+0000"}}
{"timestamp":"2025-11-24T19:03:01.233407+0000","flow_id":1962480815584052,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39028,"de
st_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payloa"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":5,"bytes_toserver":503,"bytes_toclient":342,"start":"2025-11-24T19:03:00.996
148+0000"}}
{"timestamp":"2025-11-24T19:03:02.224566+0000","flow_id":945728912700215,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39035,"des
t_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payload"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":5,"bytes_toserver":503,"bytes_toclient":342,"start":"2025-11-24T19:03:01.865
079+0000"}}
{"timestamp":"2025-11-24T19:03:02.893631+0000","flow_id":333962361053681,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39009,"des
t_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payload"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
ername":false,"password":false,"will_retain":false,"will":false,"clean_session":true}},"connack":{"qos":0,"retain":false,"dup":false,"session_present":false
,"return_code":0}},"app_proto":"mqtt","flow":{"pkts_toserver":7,"pkts_toclient":5,"bytes_toserver":503,"bytes_toclient":342,"start":"2025-11-24T19:03:02.664
049+0000"}}
{"timestamp":"2025-11-24T19:03:03.753786+0000","flow_id":1739748106810315,"in_iface":"eth0","event_type":"alert","src_ip":"24.114.74.3","src_port":39030,"de
st_ip":"172.31.46.102","dest_port":1883,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature_id":1000003,"rev":1,"signature":"Malformed MQTT Payloa"
,"category":"","severity":3},"mqtt":{"connect":{"qos":0,"retain":false,"dup":false,"protocol_string":"MQTT","protocol_version":4,"client_id":"","flags":{"us
```



```
^C
ubuntu@ip-172-31-46-102:~$ sudo tail -f /var/log/suricata/eve.json | grep alert
{"timestamp":"2025-11-20T19:30:30.319405+0000","flow_id":1667518573745740,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":50676,"dest_ip":"91.189.91.157","dest_po
rt":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:30:06.235084+0000","end":"2025-11-20T19:3
0:06.251930+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319410+0000","flow_id":829080811032569,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":51677,"dest_ip":"23.150.41.123","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:47.086009+0000","end":"2025-11-20T19:25
:47.117535+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319415+0000","flow_id":1251516628700790,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":45490,"dest_ip":"169.254.169.123","dest_
port":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:29:25.822902+0000","end":"2025-11-20T19
:29:25.826637+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319420+0000","flow_id":410873407173614,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":50313,"dest_ip":"91.189.91.157","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:26:51.422894+0000","end":"2025-11-20T19:26
:51.440477+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319425+0000","flow_id":974366670056884,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":48013,"dest_ip":"23.168.24.210","dest_por
t":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:51.650676+0000","end":"2025-11-20T19:25
:51.681585+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319430+0000","flow_id":1538832759909131,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":56071,"dest_ip":"169.254.169.123","dest_
port":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:29:09.781067+0000","end":"2025-11-20T19
:29:09.781454+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319435+0000","flow_id":1679784983306829,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":57850,"dest_ip":"91.189.91.157","dest_po
rt":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:25:46.238157+0000","end":"2025-11-20T19:2
5:46.261136+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319441+0000","flow_id":982297331634588,"in_iface":"eth0","event_type":"flow","src_ip":"172.31.46.102","src_port":51788,"dest_ip":"169.254.169.123","dest_p
ort":123,"proto":"UDP","app_proto":"ntp","flow":{"pkts_toserver":1,"pkts_toclient":1,"bytes_toserver":90,"bytes_toclient":90,"start":"2025-11-20T19:27:00.266652+0000","end":"2025-11-20T19:
27:00.267260+0000","age":0,"state":"established","reason":"shutdown","alerted":false}}
{"timestamp":"2025-11-20T19:30:30.319446+0000","flow_id":843015848661323,"in_iface":"eth0","event_type":"flow","src_ip":"45.156.128.81","src_port":40788,"dest_ip":"172.31.46.102","dest_por
t":4190,"proto":"TCP","flow":{"pkts_toserver":2,"pkts_toclient":1,"bytes_toserver":120,"bytes_toclient":54,"start":"2025-11-20T19:29:55.463179+0000","end":"2025-11-20T19:29:55.552819+0000"
,"age":0,"state":"closed","reason":"shutdown","alerted":false},"tcp":{"tcp_flags":"16","tcp_flags_ts":"06","tcp_flags_tc":"14","syn":true,"rst":true,"ack":true,"state":"closed"}}
{"timestamp":"2025-11-20T19:30:30.324210+0000","event_type":"stats","stats":{"uptime":2432,"capture":{"kernel_packets":870813,"kernel_drops":0,"errors":0},"decoder":{"pkts":870813,"bytes":
1163203170,"invalid":0,"ipv4":870705,"ipv6":2,"ethernet":870813,"chdlc":0,"raw":0,"null":0,"sll":0,"tcp":869782,"udp":888,"sctp":0,"icmpv4":35,"icmpv6":2,"ppp":0,"pppoe":0,"geneve":0,"gre"
:0,"vlan":0,"vlan_qinq":0,"vxlan":0,"vntag":0,"ieee8021ah":0,"teredo":0,"ipv4_in_ipv6":0,"ipv6_in_ipv6":0,"mpls":0,"avg_pkt_size":1335,"max_pkt_size":1514,"max_mac_addrs_src":0,"max_mac_ad
drs_dst":0,"erspan":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_require
d":0,"opt_eol_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_t
ype":0,"unknown_code":0,"ipv4_trunc_pkt":0,"ipv4_unknown_ver":0},"icmpv6":{"unknown_type":0,"unknown_code":0,"pkt_too_small":0,"ipv6_unknown_version":0,"ipv6_trunc_pkt":0,"mld_message_with
_invalid_hl":0,"unassigned_type":0,"experimentation_type":0},"ipv6":{"pkt_too_small":0,"trunc_pkt":0,"trunc_exthdr":0,"exthdr_dupl_fh":0,"exthdr_useless_fh":0,"exthdr_dupl_rh":0,"exthdr_du
pl_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_null":0,"hopopts_unknown_opt":0,"hopopts_only_padding":0
,"dstopts_unknown_opt":0,"dstopts_only_padding":0,"rh_type_0":0,"zero_len_padn":0,"fh_non_zero_reserved_field":0,"data_after_none_header":0,"unknown_next_header":0,"icmpv4":0,"frag_pkt_too
_large":0,"frag_overlap":0,"frag_invalid_length":0,"frag_ignored":0,"ipv4_in_ipv6_too_small":0,"ipv4_in_ipv6_wrong_version":0,"ipv6_in_ipv6_too_small":0,"ipv6_in_ipv6_wrong_version":0},"tc
p":{"pkt_too_small":0,"hlen_too_small":0,"invalid_optlen":0,"opt_invalid_len":0,"opt_duplicate":0},"udp":{"pkt_too_small":0,"hlen_too_small":0,"hlen_invalid":0},"sll":{"pkt_too_small":0},"
ethernet":{"pkt_too_small":0},"ppp":{"pkt_too_small":0,"vju_pkt_too_small":0,"ip4_pkt_too_small":0,"ip6_pkt_too_small":0,"wrong_type":0,"unsup_proto":0},"pppoe":{"pkt_too_small":0,"wrong_c
ode":0,"malformed_tags":0},"gre":{"pkt_too_small":0,"wrong_version":0,"version0_recur":0,"version0_flags":0,"version0_hdr_too_big":0,"version0_malformed_sre_hdr":0,"version1_chksum":0,"ver
sion1_route":0,"version1_ssr":0,"version1_recur":0,"version1_flags":0,"version1_no_key":0,"version1_wrong_protocol":0,"version1_malformed_sre_hdr":0,"version1_hdr_too_big":0},"vlan":{"head
er_too_small":0,"unknown_type":0,"too_many_layers":0},"ieee8021ah":{"header_too_small":0},"vntag":{"header_too_small":0,"unknown_type":0},"ipraw":{"invalid_ip_version":0},"ltnull":{"pkt_to
o_small":0,"unsupported_type":0},"sctp":{"pkt_too_small":0},"mpls":{"header_too_small":0,"pkt_too_small":0,"bad_label_router_alert":0,"bad_label_implicit_null":0,"bad_label_reserved":0,"un
known_payload_type":0},"vxlan":{"unknown_payload_type":0},"geneve":{"unknown_payload_type":0},"erspan":{"header_too_small":0,"unsupported_version":0,"too_many_vlan_layers":0},"dce":{"pkt_t
oo_small":0},"chdlc":{"pkt_too_small":0}},"too_many_layers":0},"flow":{"memcap":0,"tcp":784,"udp":449,"icmpv4":6,"icmpv6":2,"tcp_reuse":1,"get_used":0,"get_used_eval":0,"get_used_eval_reje
ct":0,"get_used_eval_busy":0,"get_used_failed":0,"wrk":{"spare_sync_avg":100,"spare_sync":14,"spare_sync_incomplete":0,"spare_sync_empty":0,"flows_evicted_needs_work":5,"flows_evicted_pkt_
inject":5,"flows_evicted":9,"flows_injected":5},"mgr":{"full_hash_pass":11,"closed_pruned":0,"new_pruned":0,"est_pruned":0,"bypassed_pruned":0,"rows_maxlen":2,"flows_checked":1763,"flows_n
otimeout":623,"flows_timeout":1140,"flows_timeout_inuse":0,"flows_evicted":1141,"flows_evicted_needs_work":5},"spare":9735,"emerg_mode_entered":0,"emerg_mode_over":0,"memuse":7474304},"def
```

Kibana Visualization

After Suricata → Filebeat → Elasticsearch pipeline started working, all events were visible inside Kibana.

We created two main visualizations:

1. Malformed MQTT Payload Table

Showed count of malformed messages

Displayed timestamps, Displayed source IPs

2. DDoS Detection Line Graph

Counted how many packets per 30 minutes

Clearly showed spikes when I ran the DDoS attack

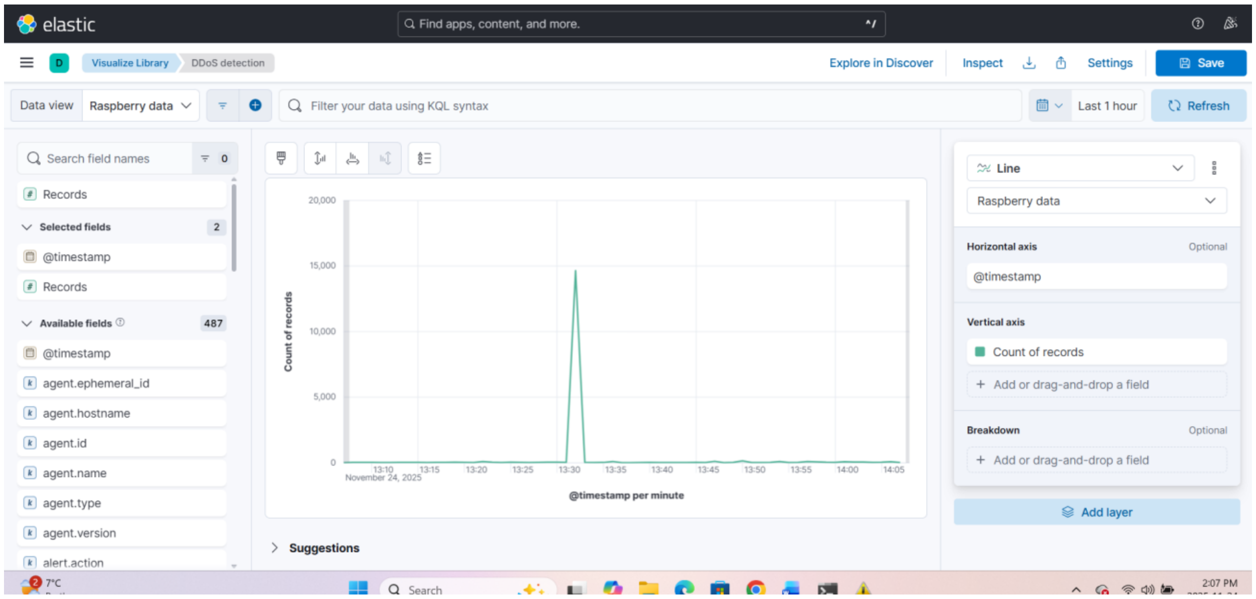
Confirmed traffic rate alerts

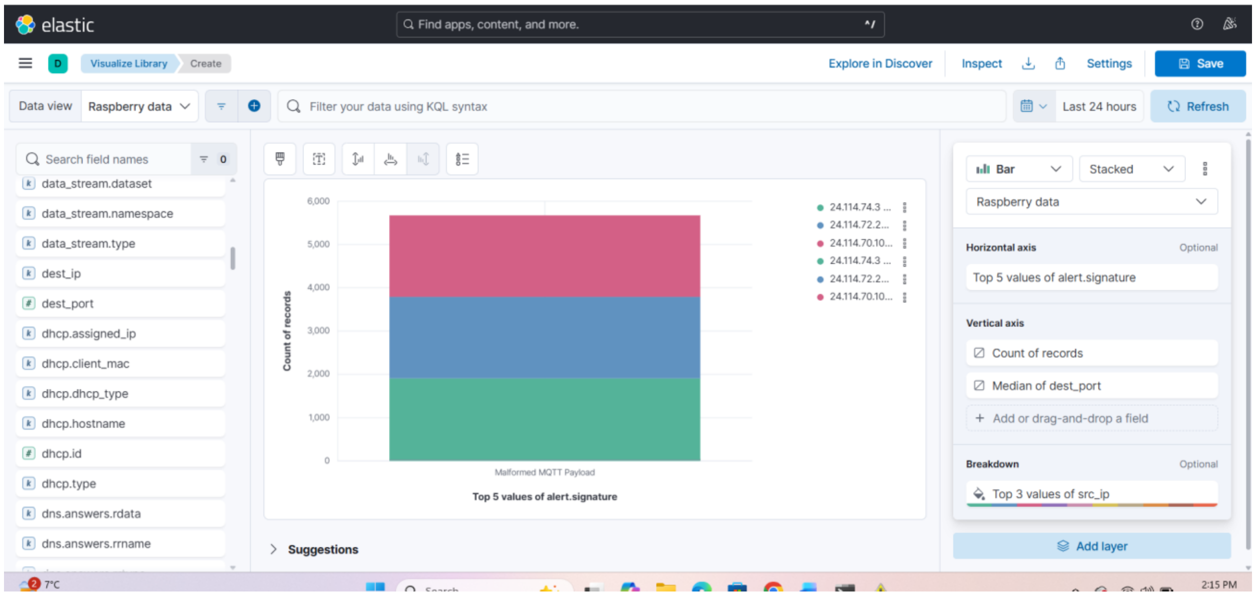
After creating both, I added them into a single dashboard called:

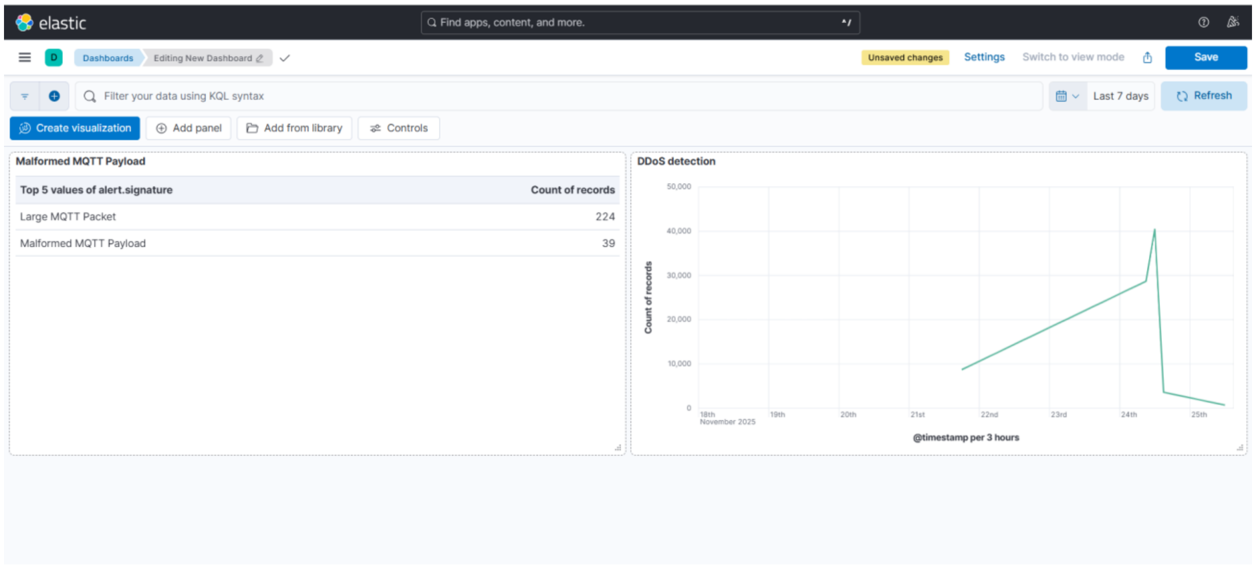
"MQTT Security Dashboard"

This dashboard clearly showed:

Normal traffic, Attack traffic, Alert counts, Time-based attack spikes







Malformed MQTT Payload

Top 5 values of alert.signature	Count of records
Large MQTT Packet	224
Malformed MQTT Payload	39

Conclusion

In this part of the project, we successfully built a complete IoT security monitoring system using AWS EC2 and open-source tools. Our Raspberry Pi sent both normal and attack traffic to the EC2 instance. Suricata detected the attacks, Filebeat forwarded logs to Elasticsearch, and Kibana visualized everything in real time.

We learned:

How to mirror IoT MQTT traffic for security monitoring

How to configure and tune Suricata for custom rule detection

How to send Suricata logs to Elasticsearch

How to build visual dashboards in Kibana

How DDoS and malformed MQTT traffic look in a real log environment

This part of the project helped me understand hands-on IoT security, network monitoring, and alert visualization.