Professional Documentation: Exploring Signatures with Suricata

Activity Overview

In this activity, I explored **Suricata**, an open-source **Intrusion Detection System (IDS)**, **Intrusion Prevention System (IPS)**, and **network analysis tool**. The goal was to:

- Examine **custom Suricata rules**, including their **structure and components**.
- **Trigger alerts** by running Suricata against a **sample packet capture (PCAP) file**.
- Analyze **log outputs** (`fast.log` and `eve.json`) to understand how alerts are generated.

This documentation provides a **detailed breakdown** of the tasks performed, commands executed, and key findings.

Scenario

As a **Security Analyst**, I was tasked with:

1. **Monitoring network traffic** using Suricata.
2. **Creating and configuring custom rules** to detect specific traffic patterns.
3. **Analyzing log outputs** to verify the effectiveness of the rules.

Files Used

- **`sample.pcap`**: A packet capture file containing simulated network traffic.
- **`custom.rules`**: A file with **custom Suricata rules** designed to detect specific behaviors.
- **`fast.log`**: A legacy alert log format (located in `/var/log/suricata`).
- **`eve.json`**: Suricata's **primary log file**, in JSON format, containing detailed alert and flow data.

Task 1: Examine a Custom Rule in Suricata

Command Executed

```
cat custom.rules
```

Output

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire";
flow:established,to_server; content:"GET"; http_method; sid:12345; rev:3;)
```

```
analyst@1a33a29b9546:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:es
tablished,to_server; content:"GET"; http_method; sid:12345; rev:3;)
analyst@1a33a29b9546:~$
```

Rule Breakdown

This rule contains **three main components**:

1. Action (`alert`)

- Defines what Suricata should do when the rule conditions are met.
- Possible actions:
  - `alert`: Log the event
  - `drop`: Block traffic and log the alert
  - `pass`: Allow traffic and skip further rule checks
  - `reject`: Block traffic and send a TCP reset

2. Header (`http $HOME_NET any -> $EXTERNAL_NET any`)

- **Protocol**: `http` (applies to HTTP traffic)
- **Source**: `$HOME_NET` (defined in `/etc/suricata/suricata.yaml` as `172.21.224.0/20`)
- **Source Port**: `any`
- **Direction**: `->` (from internal to external network)
- **Destination**: `$EXTERNAL_NET`
- **Destination Port**: `any`

3. Rule Options

- `msg:"GET on wire"`: Custom alert message
- `flow:established,to_server`: Matches client-to-server traffic in established sessions
- `content:"GET"`: Searches for the string “GET”
- `http_method`: Restricts the search to the HTTP method field
- `sid:12345`: Unique Signature ID
- `rev:3`: Revision number of the rule

Summary

This rule **generates an alert** when HTTP traffic from `$HOME_NET` to `$EXTERNAL_NET` contains a `GET` request.

Task 2: Trigger a Custom Rule in Suricata

Step 1: Check `/var/log/suricata` Before Running Suricata

```
ls -l /var/log/suricata
```

Output: No logs yet (directory is empty).

Step 2: Run Suricata Against `sample.pcap`

```
sudo suricata -r sample.pcap -S custom.rules -k none
```

- `-r sample.pcap`: Read packets from the PCAP file
- `-S custom.rules`: Load custom detection rules
- `-k none`: Skip checksum validation (not necessary for offline PCAP analysis)

Output: Suricata processes packets and generates alert logs.

Step 3: Check `/var/log/suricata` After Execution

```
ls -l /var/log/suricata
```

Output:

- `fast.log`: Alerts in legacy format
- `eve.json`: Detailed structured logs in JSON

Step 4: Examine `fast.log`

```
cat /var/log/suricata/fast.log
```

Output:

```
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**]
[Classification: (null)] [Priority: 3] {TCP} 172.21.224.2:49652 ->
142.250.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**]
[Classification: (null)] [Priority: 3] {TCP} 172.21.224.2:58494 ->
142.250.1.139:80
```

Findings:

- **Two alerts** triggered
- Both match the rule `"GET on wire"`
- **Source IP**: `172.21.224.2` (internal)
- **Destination IP**: `142.250.1.139` (external, Google)

```
analyst@1a33a29b9546:~$ ls -l /var/log/suricata
total 0
analyst@1a33a29b9546:~$ sudo suricata -r sample.pcap -S custom.rules -k o
ne
19/7/2025 -- 13:27:03 - <Error> - [ERRCODE: SC_ERR_INITIALIZATION(45)] -
option 'one' invalid for -k
analyst@1a33a29b9546:~$ sudo suricata -r sample.pcap -S custom.rules -k n
one
19/7/2025 -- 13:27:21 - <Notice> - This is Suricata version 6.0.1 RELEASE
 running in USER mode
19/7/2025 -- 13:27:22 - <Notice> - all 2 packet processing threads, 4 man
agement threads initialized, engine started.
19/7/2025 -- 13:27:22 - <Notice> - Signal Received.  Stopping engine.
19/7/2025 -- 13:27:23 - <Notice> - Pcap-file module read 1 files, 200 pac
kets, 54238 bytes
analyst@1a33a29b9546:~$
```

```
analyst@1a33a29b9546:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1417 Jul 19 13:27 eve.json
-rw-r--r-- 1 root root  292 Jul 19 13:27 fast.log
-rw-r--r-- 1 root root 2846 Jul 19 13:27 stats.log
-rw-r--r-- 1 root root 1512 Jul 19 13:27 suricata.log
analyst@1a33a29b9546:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Classifica
tion: (null)] [Priority: 3] {TCP} 172.21.224.2:49652 -> 142.250.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**] [Classifica
tion: (null)] [Priority: 3] {TCP} 172.21.224.2:58494 -> 142.250.1.102:80
analyst@1a33a29b9546:~$
```

Task 3: Analyze `eve.json` Output

Step 1: View Raw JSON

```
cat /var/log/suricata/eve.json
```

Issue: Difficult to interpret due to unformatted JSON

Step 2: Format with `jq`

```
jq . /var/log/suricata/eve.json | less
```

Findings:

- **First alert severity**: `3`
- **Alert signature**: `"GET on wire"`

Step 3: Extract Specific Fields

```
jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]"
/var/log/suricata/eve.json
```

Output:

```
["2022-11-23T12:38:34.624866+0000",14500150016149,"GET on
wire","TCP","142.250.1.139"]
["2022-11-23T12:38:58.958203+0000",1647223379236084,"GET on
wire","TCP","142.250.1.102"]
```

Findings:

- Final event's **destination IP**: `142.250.1.102`

Step 4: Filter Events by `flow_id`

```
jq "select(.flow_id==14500150016149)" /var/log/suricata/eve.json
```

Purpose: Correlate all logs tied to a specific network flow

```
analyst@1a33a29b9546:~$ cat /var/log/suricata/eve.json
{"timestamp":"2022-11-23T12:38:34.624866+0000","flow_id":549376049707157,
"pcap_cnt":70,"event_type":"alert","src_ip":"172.21.224.2","src_port":496
52,"dest_ip":"142.250.1.139","dest_port":80,"proto":"TCP","tx_id":0,"aler
t":{"action":"allowed","gid":1,"signature_id":12345,"rev":3,"signature":"
GET on wire","category":"","severity":3},"http":{"hostname":"opensource.g
oogle.com","url":"/","http_user_agent":"curl/7.74.0","http_content_type":
"text/html","http_method":"GET","protocol":"HTTP/1.1","status":301,"redir
ect":"https://opensource.google/","length":223},"app_proto":"http","flow"
:{"pkts_toserver":4,"pkts_toclient":3,"bytes_toserver":357,"bytes_toclien
t":788,"start":"2022-11-23T12:38:34.620693+0000"}}
{"timestamp":"2022-11-23T12:38:58.958203+0000","flow_id":614865712616692,
"pcap_cnt":151,"event_type":"alert","src_ip":"172.21.224.2","src_port":58
494,"dest_ip":"142.250.1.102","dest_port":80,"proto":"TCP","tx_id":0,"ale
rt":{"action":"allowed","gid":1,"signature_id":12345,"rev":3,"signature":
"GET on wire","category":"","severity":3},"http":{"hostname":"opensource.
google.com","url":"/","http_user_agent":"curl/7.74.0","http_content_type"
:"text/html","http_method":"GET","protocol":"HTTP/1.1","status":301,"redi
rect":"https://opensource.google/","length":223},"app_proto":"http","flow
":{"pkts_toserver":4,"pkts_toclient":3,"bytes_toserver":357,"bytes_toclie
nt":797,"start":"2022-11-23T12:38:58.955636+0000"}}
analyst@1a33a29b9546:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 549376049707157,
  "pcap_cnt": 70,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 49652,
  "dest_ip": "142.250.1.139",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 12345,
    "rev": 3,
```

```
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 549376049707157,
  "pcap_cnt": 70,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 49652,
  "dest_ip": "142.250.1.139",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 12345,
    "rev": 3,
    "signature": "GET on wire",
    "category": "",
    "severity": 3
  },
  "http": {
    "hostname": "opensource.google.com",
    "url": "/",
    "http_user_agent": "curl/7.74.0",
    "http_content_type": "text/html",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 301,
    "redirect": "https://opensource.google/",
    "length": 223
  },
  "app_proto": "http",
  "flow": {
    "pkts_toserver": 4,
    "pkts_toclient": 3,
    "bytes_toserver": 357,
    "bytes_toclient": 788,
:
```

```
analyst@1a33a29b9546:~$ jq -c "[.timestamp,.flow_id,.alert.signature,.pro
to,.dest_ip]" /var/log/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",549376049707157,"GET on wire","TCP","1
42.250.1.139"]
["2022-11-23T12:38:58.958203+0000",614865712616692,"GET on wire","TCP","1
42.250.1.102"]
analyst@1a33a29b9546:~$
```

```
analyst@1a33a29b9546:~$ jq "select(.flow_id==614865712616692)" /var/log/s
uricata/eve.json
{
  "timestamp": "2022-11-23T12:38:58.958203+0000",
  "flow_id": 614865712616692,
  "pcap_cnt": 151,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 58494,
  "dest_ip": "142.250.1.102",
  "dest_port": 80,
  "proto": "TCP",
  "tx_id": 0,
  "alert": {
    "action": "allowed",
    "gid": 1,
    "signature_id": 12345,
    "rev": 3,
    "signature": "GET on wire",
    "category": "",
    "severity": 3
  },
  "http": {
    "hostname": "opensource.google.com",
    "url": "/",
    "http_user_agent": "curl/7.74.0",
    "http_content_type": "text/html",
    "http_method": "GET",
    "protocol": "HTTP/1.1",
    "status": 301,
    "redirect": "https://opensource.google/",
    "length": 223
  },
  "app_proto": "http",
  "flow": {
    "pkts_toserver": 4,
    "pkts_toclient": 3,
```

Conclusion

Key Takeaways

✅ **Analyzed a Suricata rule** (action, header, and options)
✅ **Successfully triggered alerts** using a **custom rule** on a **PCAP file**
✅ **Verified rule effectiveness** via Suricata logs (`fast.log` and `eve.json`)
✅ **Utilized `jq`** to query and format Suricata's structured JSON logs

Next Steps

- Write **more advanced rules**, e.g., to detect malware C2 activity
- Deploy Suricata in **IPS mode** to actively block threats
- Automate alert analysis and forwarding with **SIEM integration**

This activity offered valuable **hands-on experience** in **IDS rule crafting, alert generation, log inspection, and network traffic monitoring**—all critical for **threat detection and incident response** roles.