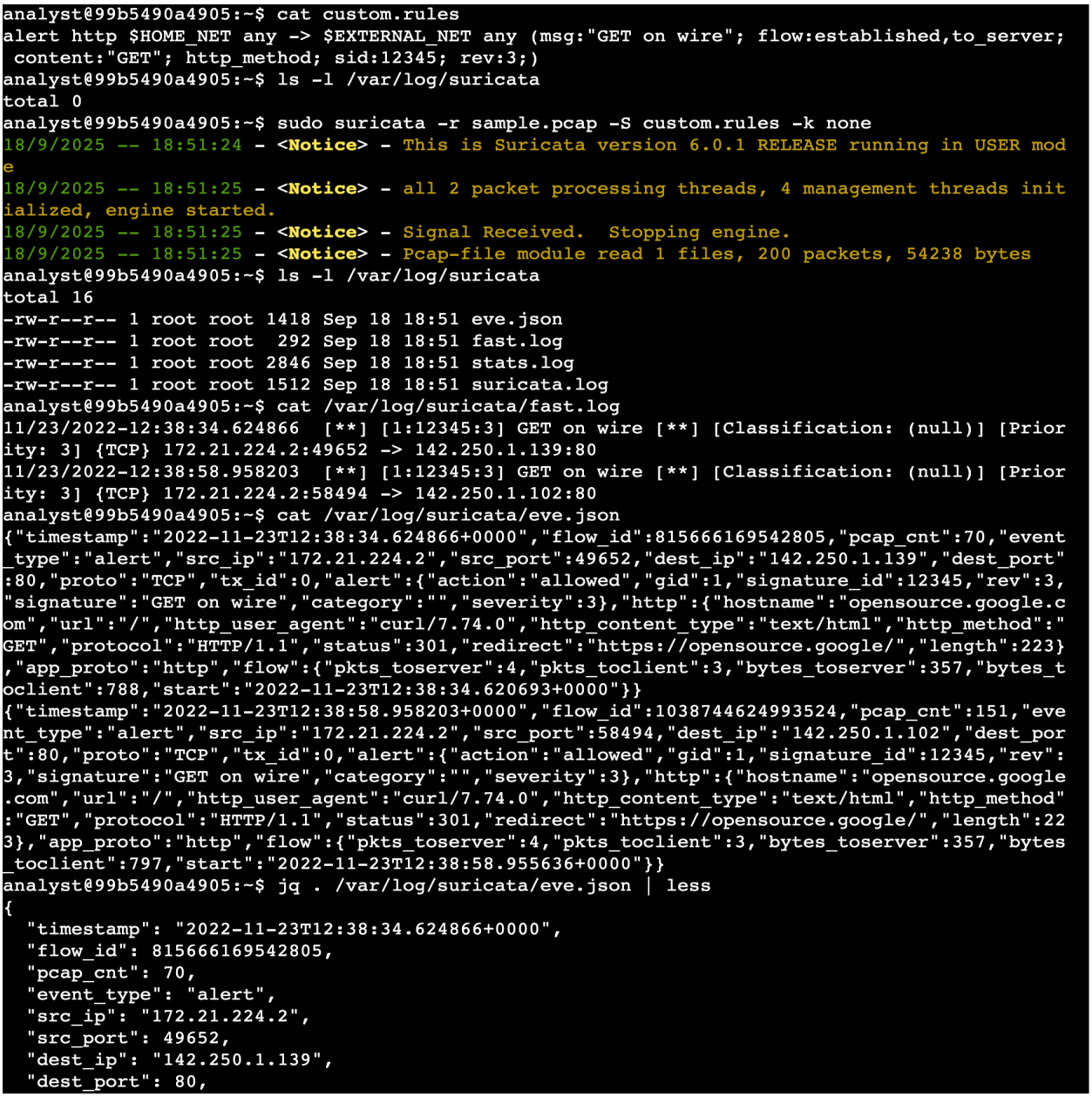
Network Traffic Capture & IDS Analysis

Project Description

I used Suricata to run IDS/alerting on a packet capture, reviewed generated logs and alerts, and parsed the JSON output to extract key fields. The work included creating a custom rule, running Suricata against a pcap, reviewing the produced logs in /var/log/suricata, and attempting to filter records with jq. The screenshots are snapshots of longer outputs.

Custom rule, Suricata run & log generation

This screenshot shows the custom Suricata rule (an alert matching HTTP GET traffic) and the command used to run Suricata against a PCAP with that rule loaded. The Suricata engine initialized, processed the 'pcap', and generated log files under /var/log/suricata. The view also shows the directory listing of /var/log/suricata and snippets from 'fast.log' and 'eve.json', demonstrating that alerts were recorded with metadata like timestamps, source/destination IPs, ports, and HTTP fields.*(Output is long - this image is a snapshot of the full logs.)*



```
analyst@99b5490a4905:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:established,to_server;
 content:"GET"; http_method; sid:12345; rev:3;)
analyst@99b5490a4905:~$ ls -l /var/log/suricata
total 0
analyst@99b5490a4905:~$ sudo suricata -r sample.pcap -S custom.rules -k none
18/9/2025 -- 18:51:24 - <Notice> - This is Suricata version 6.0.1 RELEASE running in USER mod
e
18/9/2025 -- 18:51:25 - <Notice> - all 2 packet processing threads, 4 management threads init
ialized, engine started.
18/9/2025 -- 18:51:25 - <Notice> - Signal Received.  Stopping engine.
18/9/2025 -- 18:51:25 - <Notice> - Pcap-file module read 1 files, 200 packets, 54238 bytes
analyst@99b5490a4905:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1418 Sep 18 18:51 eve.json
-rw-r--r-- 1 root root  292 Sep 18 18:51 fast.log
-rw-r--r-- 1 root root 2846 Sep 18 18:51 stats.log
-rw-r--r-- 1 root root 1512 Sep 18 18:51 suricata.log
analyst@99b5490a4905:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Classification: (null)] [Prior
ity: 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)] [Prior
ity: 3] {TCP} 172.21.224.2:58494 -> 142.250.1.102:80
analyst@99b5490a4905:~$ cat /var/log/suricata/eve.json
{"timestamp":"2022-11-23T12:38:34.624866+0000","flow_id":815666169542805,"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.c
om","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_t
oclient":788,"start":"2022-11-23T12:38:34.620693+0000"}}
{"timestamp":"2022-11-23T12:38:58.958203+0000","flow_id":1038744624993524,"pcap_cnt":151,"eve
nt_type":"alert","src_ip":"172.21.224.2","src_port":58494,"dest_ip":"142.250.1.102","dest_por
t":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":22
3},"app_proto":"http","flow":{"pkts_toserver":4,"pkts_toclient":3,"bytes_toserver":357,"bytes
_toclient":797,"start":"2022-11-23T12:38:58.955636+0000"}}
analyst@99b5490a4905:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 815666169542805,
  "pcap_cnt": 70,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 49652,
  "dest_ip": "142.250.1.139",
  "dest_port": 80,
```

Network Traffic Capture & IDS Analysis

Extracting and filtering JSON events with ‘jq’

This screenshot shows attempts to parse ‘eve.json’ with ‘jq’ to extract specific fields and produce compact, readable output (timestamp, flow_id, signature, proto, dest_ip). The screenshot also captures an attempted ‘jq’ filter using a variable (select(.flow_id==X)) that returned an error because the variable X was not defined in that invocation. The screenshot demonstrates both successful JSON extraction and a quick troubleshooting moment when a ‘jq’ variable was referenced incorrectly. (The ‘eve.json’ content and queries can be long — the screenshot shows only a snippet.)

```
analyst@99b5490a4905:~$ jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]" /var/l
og/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",815666169542805,"GET on wire","TCP","142.250.1.139"]
["2022-11-23T12:38:58.958203+0000",1038744624993524,"GET on wire","TCP","142.250.1.102"]
analyst@99b5490a4905:~$ jq "select(.flow_id==X)" /var/log/suricata/eve.json
jq: error: X/0 is not defined at <top-level>, line 1:
select(.flow_id==X)
jq: 1 compile error
analyst@99b5490a4905:~$
```

Summary

I created and tested a custom Suricata rule, ran the IDS engine against packet captures, inspected the generated alert logs (fast.log and eve.json), and parsed findings with ‘jq’ to extract relevant fields. This workflow illustrates end-to-end detection and analysis: rule authoring → packet scanning → alert logging → structured JSON parsing for downstream analysis or reporting.