Examine alerts, logs, and rules with Suricata

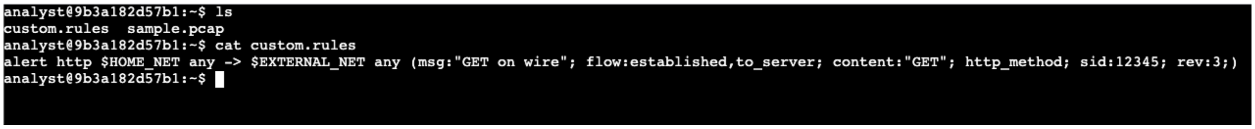
Scenario

In this scenario, you’re a security analyst who must monitor traffic on your employer's network. You’ll be required to configure Suricata and use it to trigger alerts.

Task 1. Examine a custom rule in Suricata

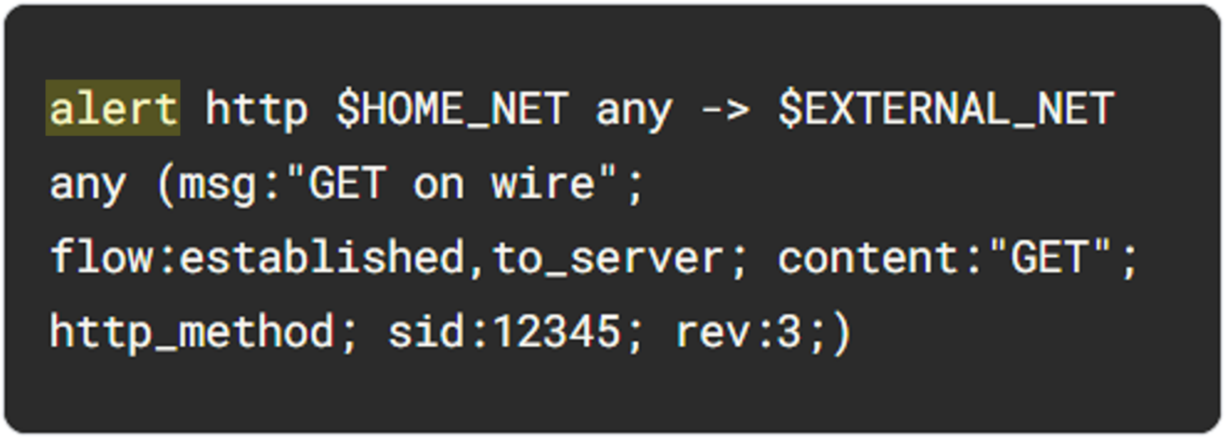
1.1. Use the `cat` command to display the rule in the `custom.rules` file:

Command: cat custom.rules



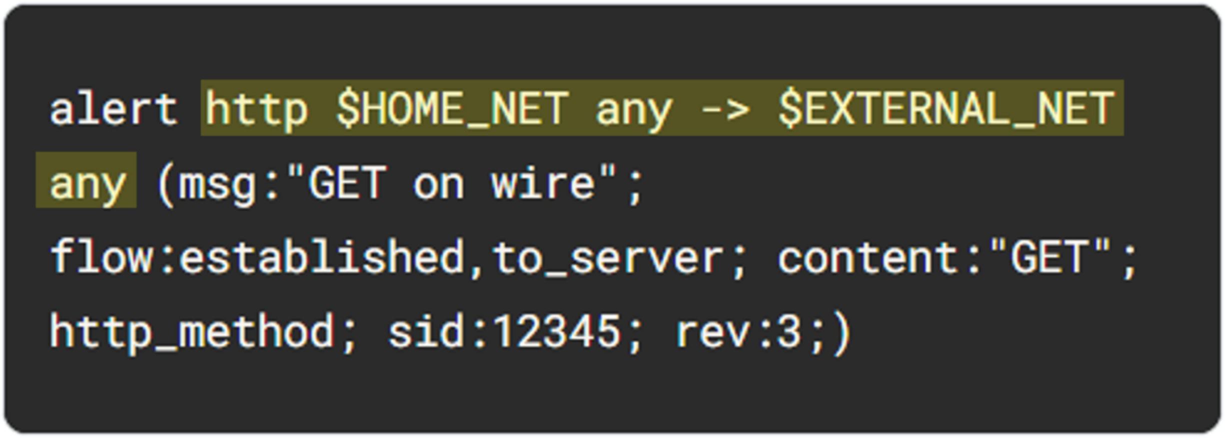
```
analyst@9b3a182d57b1:~$ ls
custom.rules  sample.pcap
analyst@9b3a182d57b1:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:established,to_server; content:"GET"; http_method; sid:12345; rev:3;)
analyst@9b3a182d57b1:~$
```

This rule consists of three components: an **action**, a **header**, and **rule options**.



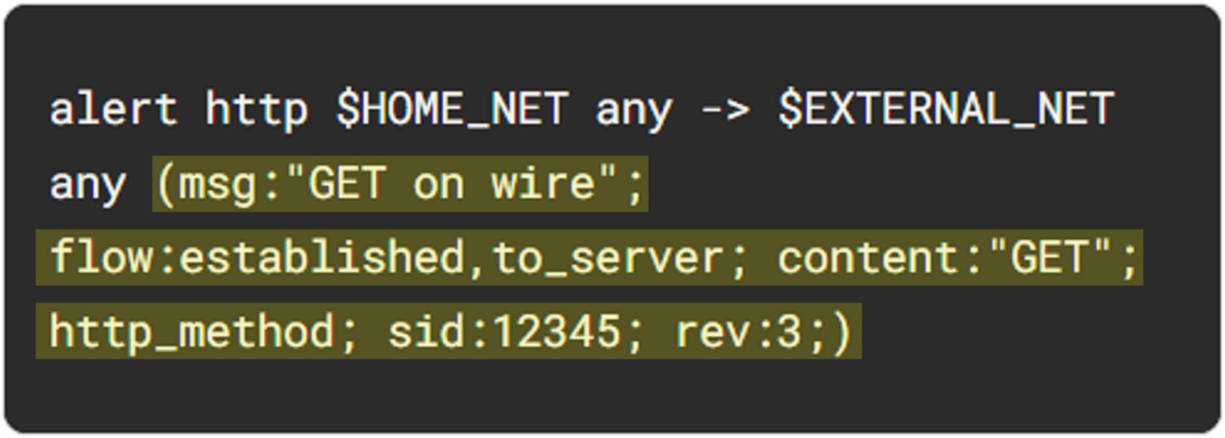
```
alert http $HOME_NET any -> $EXTERNAL_NET
any (msg:"GET on wire";
flow:established,to_server; content:"GET";
http_method; sid:12345; rev:3;)
```

The **action** is the first part of the signature. It determines the action to take if all conditions are met.



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

The next part of the signature is the **header**. The header defines the signature's network traffic, which includes attributes such as protocols, source and destination IP addresses, source and destination ports, and traffic direction.



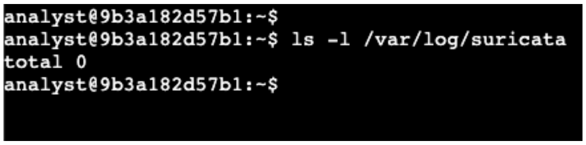
```
alert http $HOME_NET any -> $EXTERNAL_NET
any (msg:"GET on wire";
flow:established,to_server; content:"GET";
http_method; sid:12345; rev:3;)
```

The many available **rule options** allow you to customize signatures with additional parameters. Configuring rule options helps narrow down network traffic so you can find exactly what you're looking for. As in our example, rule options are typically enclosed in a pair of parentheses and separated by semicolons.

Task 2. Trigger a custom rule in Suricata

2.1. List the files in the `/var/log/suricata` folder:

Command: ls -l /var/log/suricata

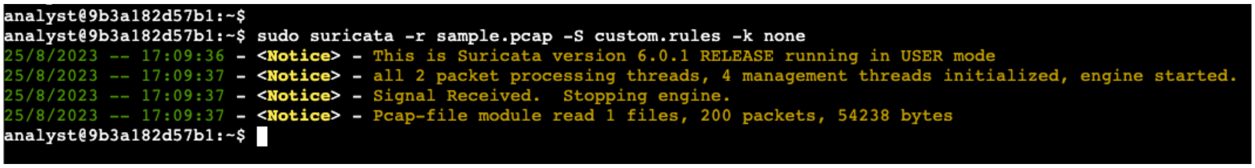


```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ ls -l /var/log/suricata
total 0
analyst@9b3a182d57b1:~$
```

2.2. Run `suricata` using the `custom.rules` and `sample.pcap` files:

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

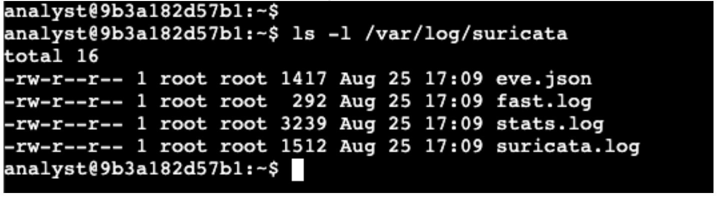
- The -r sample.pcap option specifies an input file to mimic network traffic. In this case, the sample.pcap file.
- The -S custom.rules option instructs Suricata to use the rules defined in the custom.rules file.
- The -k none option instructs Suricata to disable all checksum checks.



```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ sudo suricata -r sample.pcap -S custom.rules -k none
25/8/2023 -- 17:09:36 - <Notice> - This is Suricata version 6.0.1 RELEASE running in USER mode
25/8/2023 -- 17:09:37 - <Notice> - all 2 packet processing threads, 4 management threads initialized, engine started.
25/8/2023 -- 17:09:37 - <Notice> - Signal Received.  Stopping engine.
25/8/2023 -- 17:09:37 - <Notice> - Pcap-file module read 1 files, 200 packets, 54238 bytes
analyst@9b3a182d57b1:~$ 
```

2.3. List the files in the `/var/log/suricata` folder again:

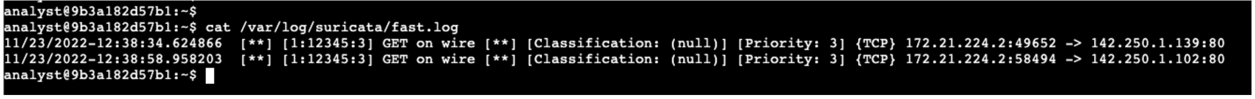
Command: ls -l /var/log/suricata



```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1417 Aug 25 17:09 eve.json
-rw-r--r-- 1 root root  292 Aug 25 17:09 fast.log
-rw-r--r-- 1 root root 3239 Aug 25 17:09 stats.log
-rw-r--r-- 1 root root 1512 Aug 25 17:09 suricata.log
analyst@9b3a182d57b1:~$
```

2.4. Use the `cat` command to display the `fast.log` file generated by Suricata:

Command: cat /var/log/suricata/fast.log

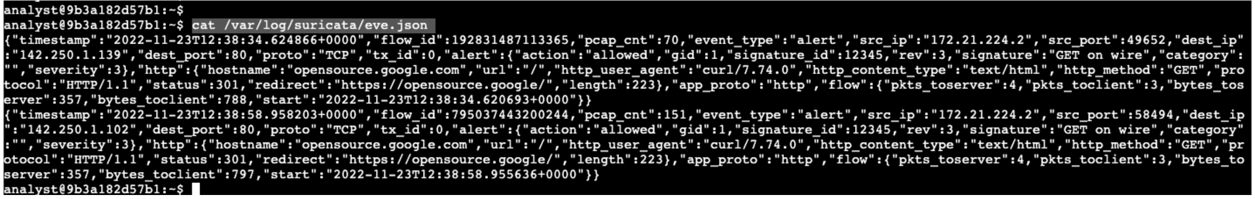


```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Classification: (null)] [Priority: 3] {TCP} 172.21.224.2:49652 -> 142.250.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**] [Classification: (null)] [Priority: 3] {TCP} 172.21.224.2:58494 -> 142.250.1.102:80
analyst@9b3a182d57b1:~$
```

Task 3. Examine eve.json output

3.1. Use the `cat` command to display the entries in the `eve.json` file:

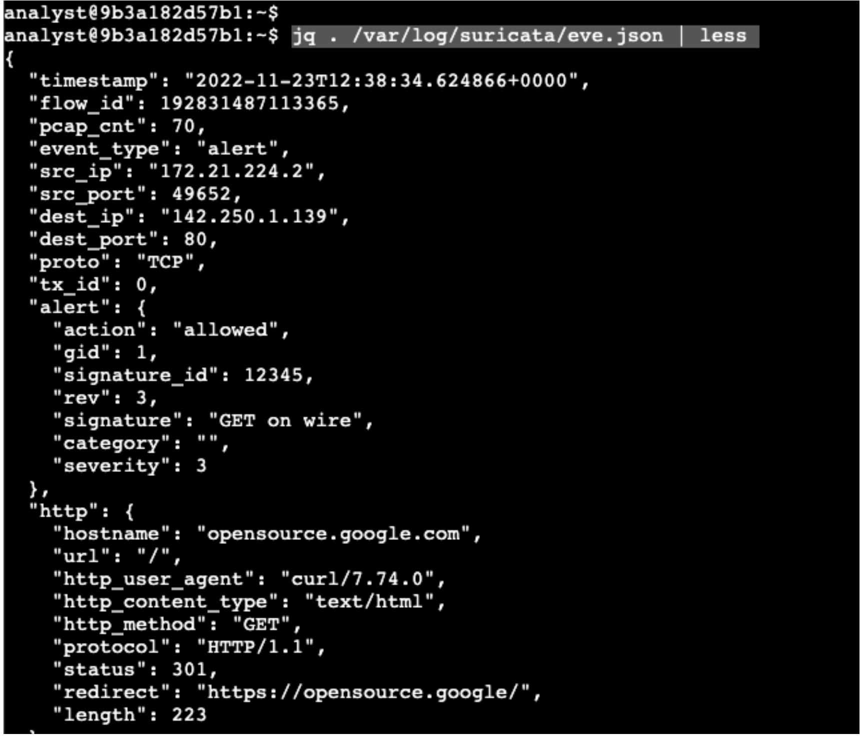
Command: cat /var/log/suricata/eve.json



```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ cat /var/log/suricata/eve.json
{"timestamp":"2022-11-23T12:38:34.624866+0000","flow_id":192831487113365,"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","pro
tocol":"HTTP/1.1","status":301,"redirect":"https://opensource.google/","length":223},"app_proto":"http","flow":{"pkts_toserver":4,"pkts_toclient":3,"bytes_tos
erver":357,"bytes_toclient":788,"start":"2022-11-23T12:38:34.620693+0000"}}
{"timestamp":"2022-11-23T12:38:58.958203+0000","flow_id":795037443200244,"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","pr
otocol":"HTTP/1.1","status":301,"redirect":"https://opensource.google/","length":223},"app_proto":"http","flow":{"pkts_toserver":4,"pkts_toclient":3,"bytes_to
server":357,"bytes_toclient":797,"start":"2022-11-23T12:38:58.955636+0000"}}
analyst@9b3a182d57b1:~$
```

3.2. Use the `jq` command to display the entries in an improved format:

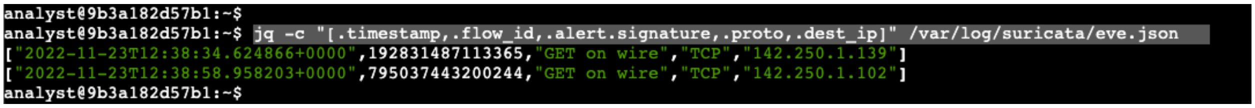
Command: jq . /var/log/suricata/eve.json | less



```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 192831487113365,
  "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
```

3.3. Use the `jq` command to extract specific event data from the `eve.json` file:

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



```
analyst@9b3a182d57b1:~$
analyst@9b3a182d57b1:~$ jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]" /var/log/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",192831487113365,"GET on wire","TCP","142.250.1.139"]
["2022-11-23T12:38:58.958203+0000",795037443200244,"GET on wire","TCP","142.250.1.102"]
analyst@9b3a182d57b1:~$
```