Activity: 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.

Here's how you'll do this task: **First**, you'll explore custom rules in Suricata. **Second**, you'll run Suricata with a custom rule in order to trigger it, and examine the output logs in the fast.log file. **Finally**, you'll examine the additional output that Suricata generates in the standard eve.json log file.

For the purposes of the tests you'll run in this lab activity, you've been supplied with a sample.pcap file and a custom.rules file. These reside in your home folder.

Let's define the files you'll be working with in this lab activity:

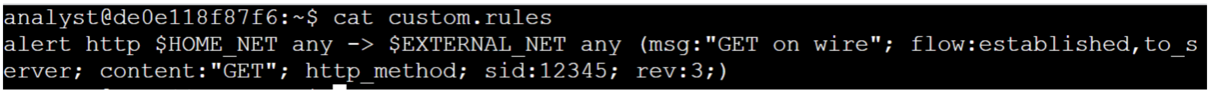
- The **sample.pcap** file is a packet capture file that contains an example of network traffic data, which you'll use to test the Suricata rules. This will allow you to simulate and repeat the exercise of monitoring network traffic.
- The **custom.rules** file contains a custom rule when the lab activity starts. You'll add rules to this file and run them against the network traffic data in the **sample.pcap** file.
- The fast.log file will contain the alerts that Suricata generates. The fast.log file is empty when the lab starts. Each time you test a rule, or set of rules, against the sample network traffic data, Suricata adds a new alert line to the fast.log file when all the conditions in any of the rules are met. The **fast.log file can** be located in the /var/log/suricata directory after Suricata runs. The fast.log file is considered to be a depreciated format and is not recommended for incident response or threat hunting tasks but can be used to perform quick checks or tasks related to quality assurance.
- The **eve.json** file is the main, standard, and default log for events generated by Suricata. It contains detailed information about alerts triggered, as well as other network telemetry events, in JSON format. The **eve.json** file is generated when Suricate runs, and can also be located in the /var/log/suricata directory.

When you create a new rule, you'll need to test the rule to confirm whether or not it worked as expected. You can use the fast.log file to quickly compare the number of alerts generated each time you run Suricata to test a signature against the sample.pcap file.

Task 1. Examine a custom rule in Suricata

In this task, you'll explore the composition of the Suricata rule defined in the custom.rules file.

- Use the **cat** command to display the rule in the custom.rules file:



```
analyst@de0e118f87f6:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:established,to_s
erver; content:"GET"; http_method; sid:12345; rev:3;)
```

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

Action

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

Actions differ across network intrusion detection system (NIDS) rule languages, but some common actions are alert, drop, pass, and reject.

Using our example, the file contains a single alert as the action. The alert keyword instructs to alert on selected network traffic. The IDS will inspect the traffic packets and send out an alert in case it matches.

Note that the drop action also generates an alert, but it drops the traffic. A drop action only occurs when Suricata runs in IPS mode.

The pass action allows the traffic to pass through the network interface. The pass rule can be used to override other rules. An exception to a drop rule can be made with a pass rule.

The reject action does not allow the traffic to pass. Instead, a TCP reset packet will be sent, and Suricata will drop the matching packet. A TCP reset packet tells computers to stop sending messages to each other.

Header

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.

The next field after the action keyword is the protocol field. In our example, the protocol is http, which determines that the rule applies only to HTTP traffic.

The parameters to the protocol http field are $HOME_NET any -> $EXTERNAL_NET any. The arrow indicates the direction of the traffic coming from the $HOME_NET and going to the destination IP address $EXTERNAL_NET.

$HOME_NET is a Suricata variable defined in /etc/suricata/suricata.yaml that you can use in your rule definitions as a placeholder for your local or home network to identify traffic that connects to or from systems within your organization.

In this lab $HOME_NET is defined as the 172.21.224.0/20 subnet.

The word any means that Suricata catches traffic from any port defined in the $HOME_NET network.

Rule options

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.

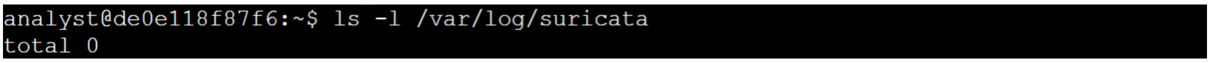
Let's further examine the rule options in our example:

- The msg: option provides the alert text. In this case, the alert will print out the text "GET on wire", which specifies why the alert was triggered.
- The flow:established,to_server option determines that packets from the client to the server should be matched. (In this instance, a server is defined as the device responding to the initial SYN packet with a SYN-ACK packet.)
- The content:"GET" option tells Suricata to look for the word GET in the content of the http.method portion of the packet.
- The sid:12345 (signature ID) option is a unique numerical value that identifies the rule.
- The rev:3 option indicates the signature's revision which is used to identify the signature's version. Here, the revision version is 3.

To summarize, this signature triggers an alert whenever Suricata observes the text GET as the HTTP method in an HTTP packet from the home network going to the external network.

Task 2. Trigger a custom rule in Suricata

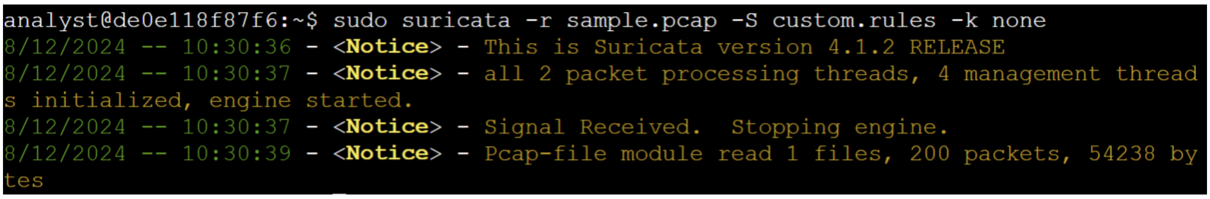
1. List the files in the /var/log/suricata folder:



```
analyst@de0e118f87f6:~$ ls -l /var/log/suricata
total 0
```

Note that before running Suricata, there are no files in the /var/log/suricata directory.

2. Run **suricata** using the **custom.rules** and **sample.pcap** files:

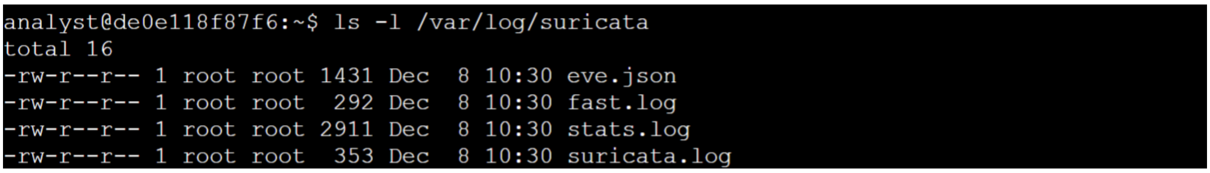


```
analyst@de0e118f87f6:~$ sudo suricata -r sample.pcap -S custom.rules -k none
8/12/2024 -- 10:30:36 - <Notice> - This is Suricata version 4.1.2 RELEASE
8/12/2024 -- 10:30:37 - <Notice> - all 2 packet processing threads, 4 management thread
s initialized, engine started.
8/12/2024 -- 10:30:37 - <Notice> - Signal Received.  Stopping engine.
8/12/2024 -- 10:30:39 - <Notice> - Pcap-file module read 1 files, 200 packets, 54238 by
tes
```

Now you'll further examine the options in the command:

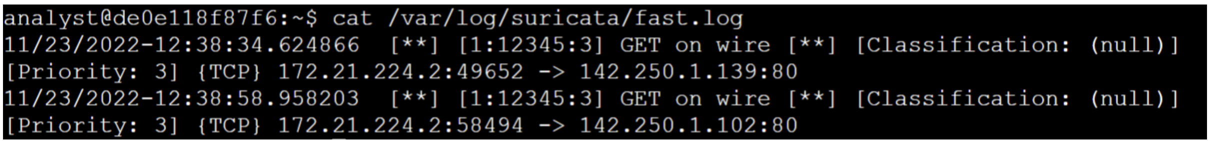
- The -r sample.pcap option specifies an input file to mimic network traffic. In this case, the sample.pcap file.
- The -S custom.rules option instructs Suricata to use the rules defined in the custom.rules file.
- The -k none option instructs Suricata to disable all checksum checks.

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



```
analyst@de0e118f87f6:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1431 Dec  8 10:30 eve.json
-rw-r--r-- 1 root root  292 Dec  8 10:30 fast.log
-rw-r--r-- 1 root root 2911 Dec  8 10:30 stats.log
-rw-r--r-- 1 root root  353 Dec  8 10:30 suricata.log
```

4. Use the **cat** command to display the fast.log file generated by Suricata:



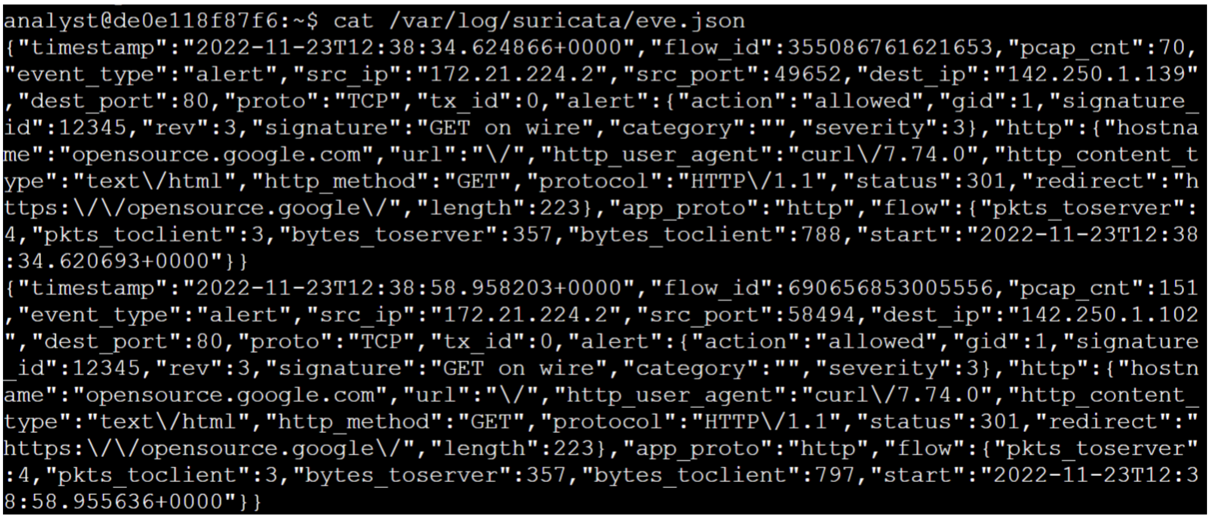
```
analyst@de0e118f87f6:~$ cat /var/log/suricata/fast.log
11/23/2022-12:38:34.624866  [**] [1:12345:3] GET on wire [**] [Classification: (null)]
[Priority: 3] {TCP} 172.21.224.2:49652 -> 142.250.1.139:80
11/23/2022-12:38:58.958203  [**] [1:12345:3] GET on wire [**] [Classification: (null)]
[Priority: 3] {TCP} 172.21.224.2:58494 -> 142.250.1.102:80
```

Each line or entry in the fast.log file corresponds to an alert generated by Suricata when it processes a packet that meets the conditions of an alert generating rule. Each alert line includes the message that identifies the rule that triggered the alert, as well as the source, destination, and direction of the traffic.

Task 3. Examine eve.json output

In this task, you must examine the additional output that Suricata generates in the eve.json file.

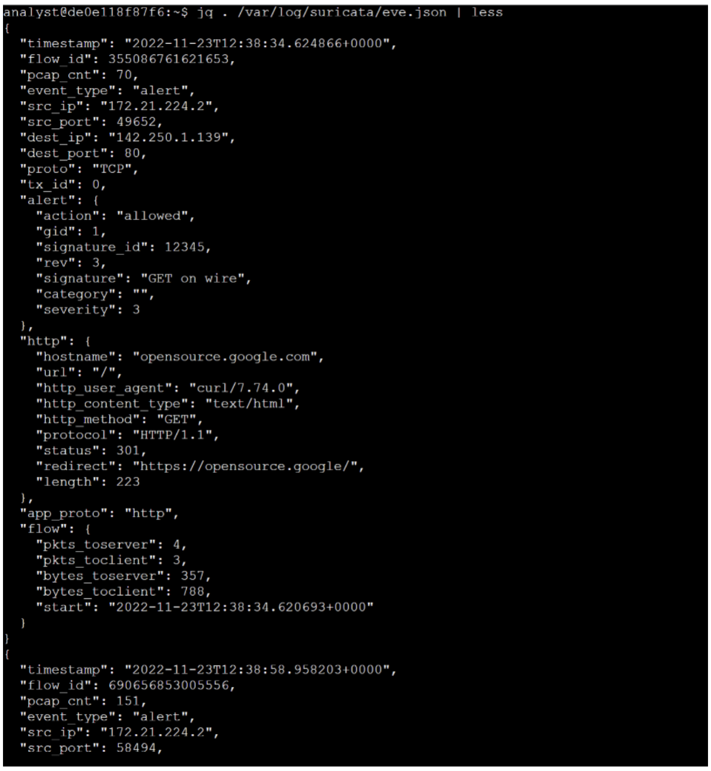
1. Use the **cat** command to display the entries in the eve.json file:



```
analyst@de0e118f87f6:~$ cat /var/log/suricata/eve.json
{"timestamp":"2022-11-23T12:38:34.624866+0000","flow_id":355086761621653,"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":{"hostna
me":"opensource.google.com","url":"\/","http_user_agent":"curl\/7.74.0","http_content_t
ype":"text\/html","http_method":"GET","protocol":"HTTP\/1.1","status":301,"redirect":"h
ttps:\/\/opensource.google\/","length":223},"app_proto":"http","flow":{"pkts_toserver":
4,"pkts_toclient":3,"bytes_toserver":357,"bytes_toclient":788,"start":"2022-11-23T12:38
:34.620693+0000"}}
{"timestamp":"2022-11-23T12:38:58.958203+0000","flow_id":690656853005556,"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":{"hostn
ame":"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":797,"start":"2022-11-23T12:3
8:58.955636+0000"}}
```

The output returns the raw content of the file. You'll notice that there is a lot of data returned that is not easy to understand in this format.

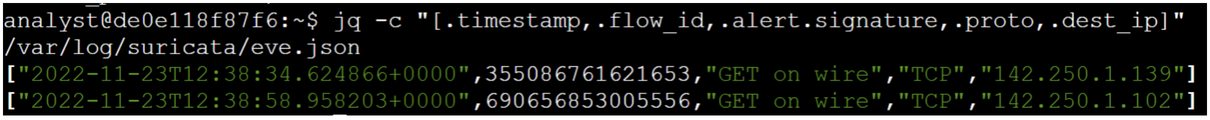
2. Use the **jq** command to display the entries in an improved format:



```
analyst@de0e118f87f6:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 355086761621653,
  "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,
    "start": "2022-11-23T12:38:34.620693+0000"
  }
}
{
  "timestamp": "2022-11-23T12:38:58.958203+0000",
  "flow_id": 690656853005556,
  "pcap_cnt": 151,
  "event_type": "alert",
  "src_ip": "172.21.224.2",
  "src_port": 58494,
```

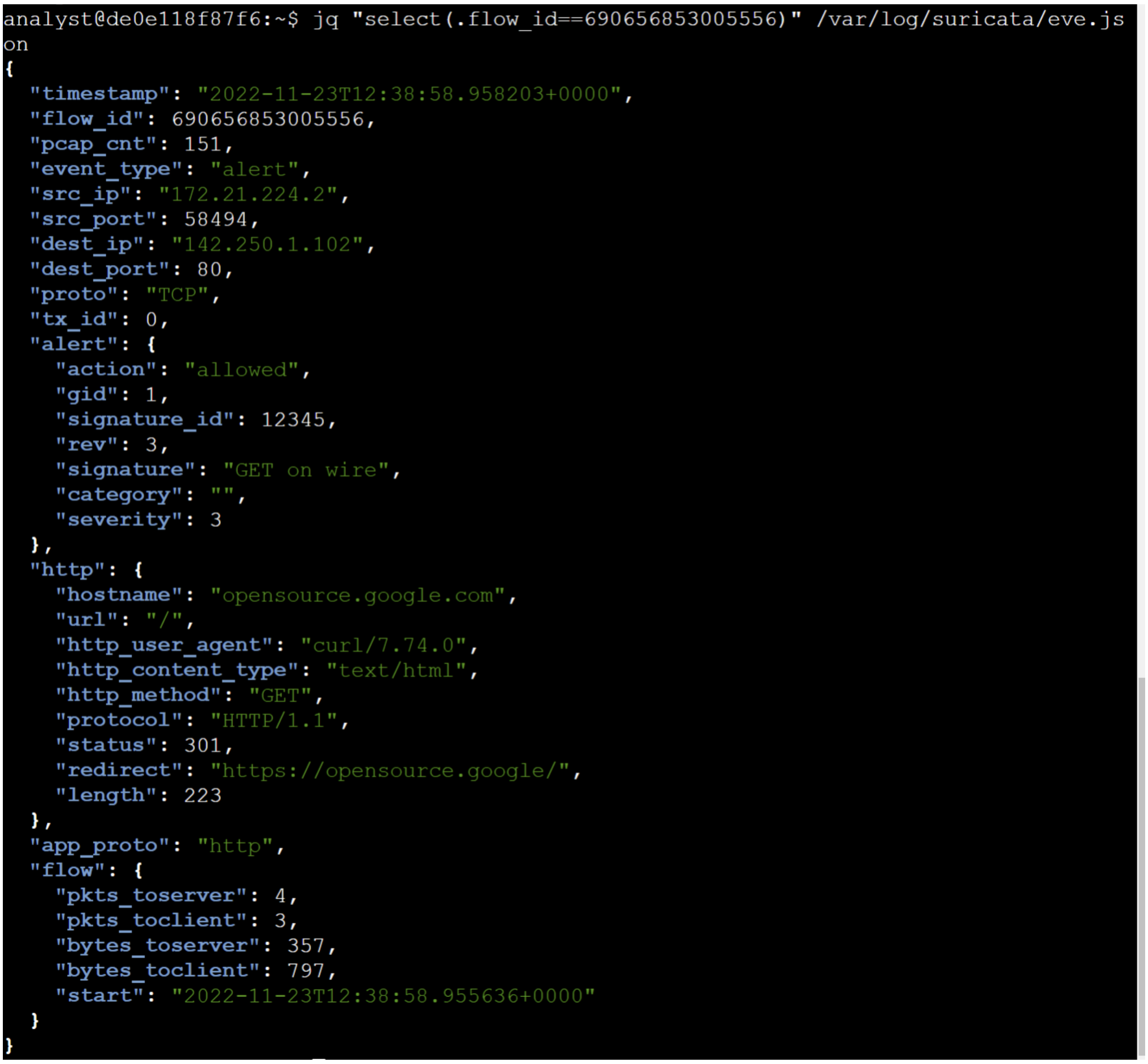
3. Press **Q** to exit the less command and to return to the command-line prompt.

4. Use the **jq** command to extract specific event data from the eve.json file:



```
analyst@de0e118f87f6:~$ jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]"
/var/log/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",355086761621653,"GET on wire","TCP","142.250.1.139"]
["2022-11-23T12:38:58.958203+0000",690656853005556,"GET on wire","TCP","142.250.1.102"]
```

5. Use the **jq** command to display all event logs related to a specific flow_id from the eve.json file. The flow_id value is a 16-digit number and will vary for each of the log entries.



```
analyst@de0e118f87f6:~$ jq "select(.flow_id==690656853005556)" /var/log/suricata/eve.js
on
{
  "timestamp": "2022-11-23T12:38:58.958203+0000",
  "flow_id": 690656853005556,
  "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,
    "bytes_toserver": 357,
    "bytes_toclient": 797,
    "start": "2022-11-23T12:38:58.955636+0000"
  }
}
```