**Examine alerts, logs, and rules with Suricata**

**Task 1. Examine a custom rule in Suricata**

The /home/analyst directory contains a custom.rules file that defines the network traffic rules, which Suricata captures.

In this task, I’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:

cat custom.rules

The command returns the rule as the output in the shell:

alert http $HOME\_NET any -> $EXTERNAL\_NET any (msg:"GET on wire"; flow:established,to\_server; content:"GET"; http\_method; sid:12345; rev:3;)

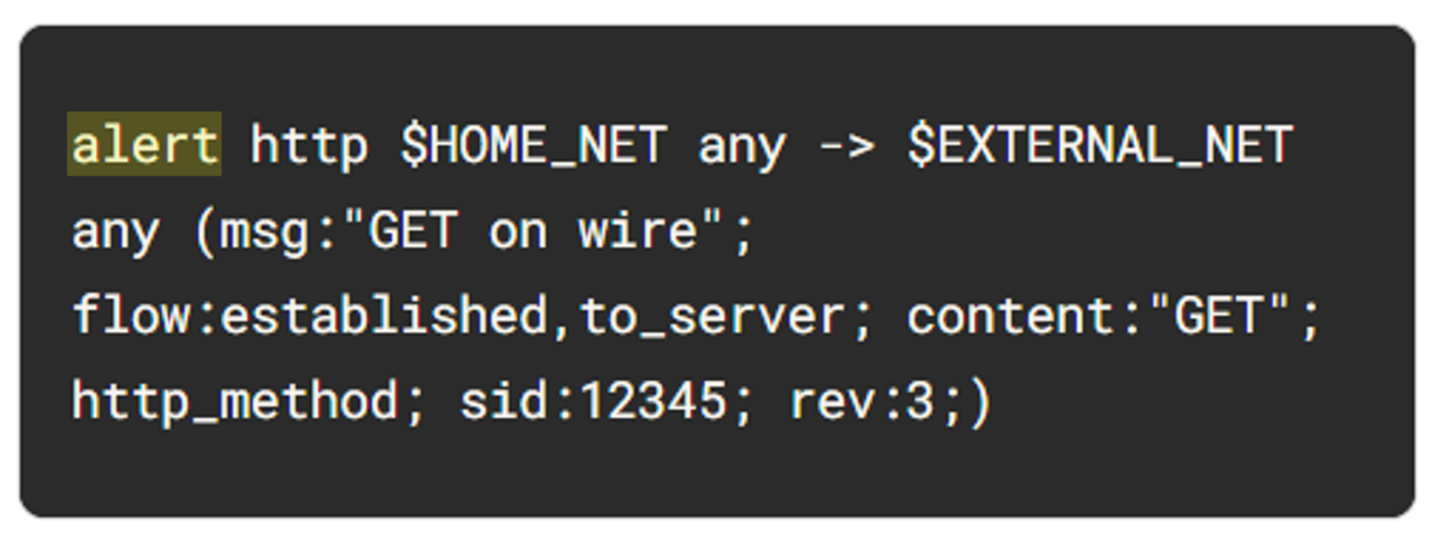
**Terminal Code:**

analyst@c6cfc5a176f1:~$ 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;)

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

Let's examine each component in more detail.

**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.

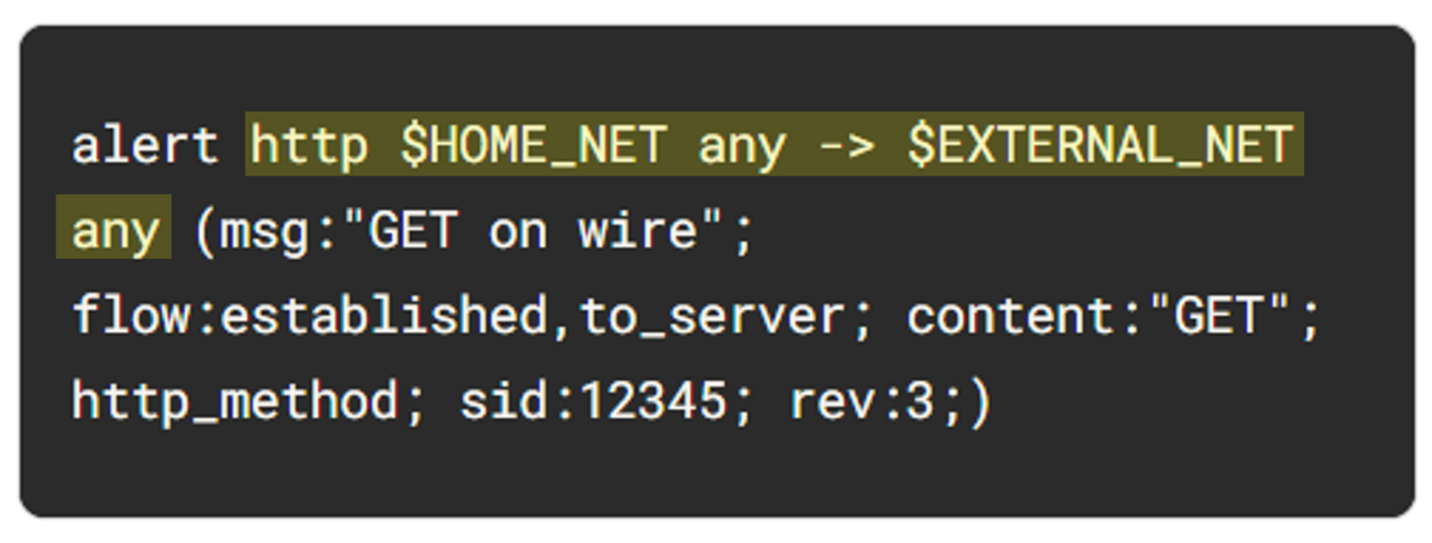
Note that the drop action also generates an alert, but it drops the traffic. A drop action only occurs when Suricata runs in IPS mode.

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. For example, the following rule has an identical signature to the previous example, except that it singles out a specific IP address to allow only traffic from that address to pass:

pass http 172.17.0.77 any -> $EXTERNAL\_NET any (msg:"BAD USER-AGENT";flow:established,to\_server;content:!”Mozilla/5.0”; http\_user\_agent; sid: 12365; rev:1;)

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.

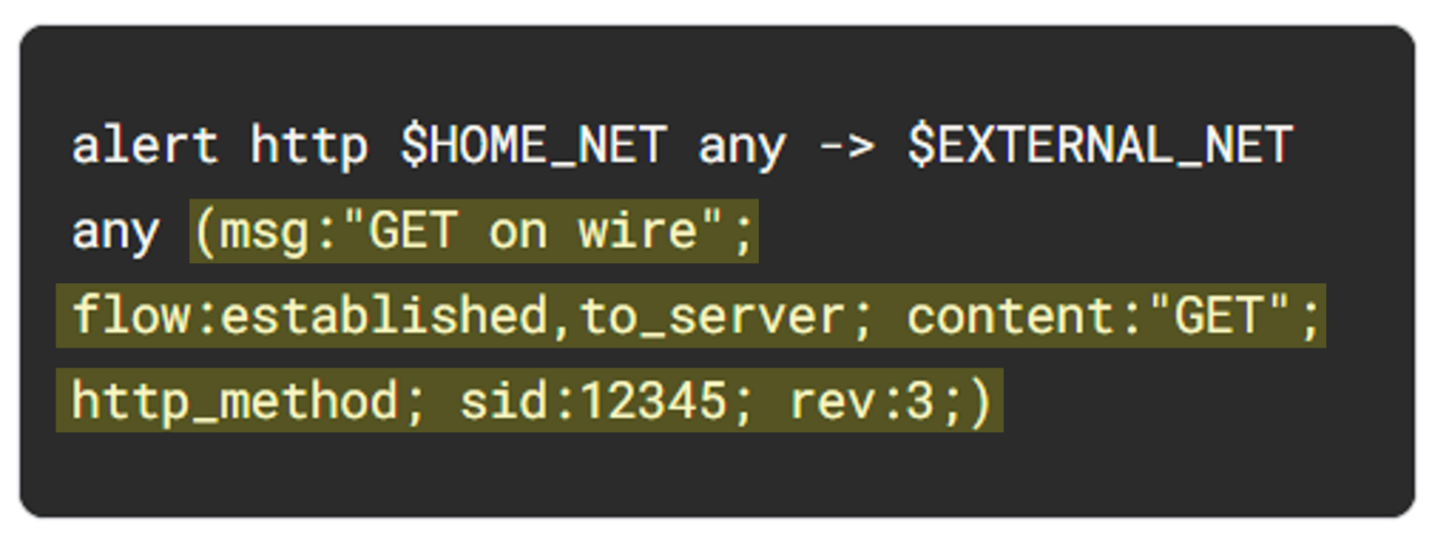
$HOME\_NET is a Suricata variable defined in /etc/suricata/suricata.yaml that you can use in your rule definitions as a placeholder for your local or home network to identify traffic that connects to or from systems within your organization.

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.

So far, we know that this signature triggers an alert when it detects any http traffic leaving the home network and going to the external 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**

Now I am familiar with the composition of the custom Suricata rule, I must trigger this rule and examine the alert logs that Suricata generates.

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

ls -l /var/log/suricata

**Terminal Code:**

analyst@c6cfc5a176f1:~$ ls -l /var/log/suricata

total 0

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

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

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

This command starts the Suricata application and processes the sample.pcap file using the rules in the custom.rules file. It returns an output stating how many packets were processed by Suricata.

**Terminal Code:**

analyst@c6cfc5a176f1:~$ sudo suricata -r sample.pcap -S custom.rules -k none

18/2/2025 -- 06:32:14 - <Notice> - This is Suricata version 4.1.2 RELEASE

18/2/2025 -- 06:32:15 - <Notice> - all 2 packet processing threads, 4 management threads initialized, engine started.

18/2/2025 -- 06:32:15 - <Notice> - Signal Received. Stopping engine.

18/2/2025 -- 06:32:17 - <Notice> - Pcap-file module read 1 files, 200 packets, 54238 bytes

Now I’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.

Suricata adds a new alert line to the /var/log/suricata/fast.log file when all the conditions in any of the rules are met.

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

ls -l /var/log/suricata

**Terminal Code:**

analyst@c6cfc5a176f1:~$ ls -l /var/log/suricata

total 16

-rw-r--r-- 1 root root 1430 Feb 18 06:32 eve.json

-rw-r--r-- 1 root root 292 Feb 18 06:32 fast.log

-rw-r--r-- 1 root root 2911 Feb 18 06:32 stats.log

-rw-r--r-- 1 root root 353 Feb 18 06:32 suricata.log

Note that after running Suricata, there are now four files in the /var/log/suricata directory, including the fast.log and eve.json files. You'll examine these files in more detail.

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

cat /var/log/suricata/fast.log

The output returns alert entries in the 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 -> 142.250.1.139:80

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.

**Terminal Code:**

analyst@c6cfc5a176f1:~$ 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

**Task 3. Examine eve.json output**

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

As previously mentioned, this file is located in the /var/log/suricata/ directory.

The eve.json file is the standard and main Suricata log file and contains a lot more data than the fast.log file. This data is stored in a JSON format, which makes it much more useful for analysis and processing by other applications.

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

cat /var/log/suricata/eve.json

The output returns the raw content of the file. I have noticed that there is a lot of data returned that is not easy to understand in this format.

**Terminal Code:**

analyst@c6cfc5a176f1:~$ cat /var/log/suricata/eve.json

{"timestamp":"2022-11-23T12:38:34.624866+0000","flow\_id":7862278060181,"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":1529124663497972,"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"}}

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

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

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

Note how much easier it is to read the output now as opposed to the cat command output.

**Terminal Code:**

analyst@c6cfc5a176f1:~$ jq . /var/log/suricata/eve.json | less

{

"timestamp": "2022-11-23T12:38:34.624866+0000",

"flow\_id": 7862278060181,

"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",

**Conclusion**

I’ve completed this activity to use Suricata to trigger alerts on network traffic.

Which is based on some practical experience in running Suricata to

* create custom rules and run them in Suricata,
* monitor traffic captured in a packet capture file, and
* examine the fast.log and eve.json output.