Examine Alerts, Logs, and Rules Using Suricata

1. Introduction

Suricata is an advanced, open-source network threat detection engine capable of intrusion detection (IDS), intrusion prevention (IPS), and network security monitoring (NSM). This project focuses on creating and analyzing custom rules within Suricata, examining the alerts and logs generated, and gaining practical experience in network traffic analysis.

2. Project Objectives

- To understand the structure of Suricata rules and customize them.
- To generate network traffic and trigger alerts based on custom rules.
- To analyze the logs generated by Suricata in both fast.log and eve.json formats.

3. Tools and Technologies

- **Suricata**: The primary tool used for network traffic monitoring and analysis.
- **Packet Capture Files (PCAP)**: Sample.pcap file to simulate network traffic.
- **jq**: A command-line JSON processor to format and query JSON output.
- **Linux Command Line**: For executing commands and managing files.

4. Methodology

Task 1: Examine a Custom Rule in Suricata

1. **Display the Custom Rule**:

Command:

```
cat custom.rules
```

Output:

```
analyst@e4ffb3c23add:~$ ls
custom.rules  sample.pcap
analyst@e4ffb3c23add:~$ cat custom.rules
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GET on wire"; flow:establi
shed,to_server; content:"GET"; http_method; sid:12345; rev:3;)
```

Rule Components:

- **Action**: `alert` - triggers an alert when the conditions are met.
- **Header**: Specifies the protocol (`http`), source and destination networks, and the direction of traffic.
- **Rule Options**: Additional parameters like `msg`, `flow`, `content`, `sid`, and `rev`.

Task 2: Trigger a Custom Rule in Suricata

1. **List Suricata Log Files**:

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

```
analyst@e4ffb3c23add:~$ ls -l /var/log/suricata
total 0
```

2. **Run Suricata with Custom Rules**:

Command:
`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.

```
analyst@e4ffb3c23add:~$ sudo suricata -r sample.pcap -S custom.rules -k none
20/10/2024 -- 09:19:40 - <Notice> - This is Suricata version 4.1.2 RELEASE
20/10/2024 -- 09:19:41 - <Notice> - all 2 packet processing threads, 4 managem
ent threads initialized, engine started.
20/10/2024 -- 09:19:41 - <Notice> - Signal Received.  Stopping engine.
20/10/2024 -- 09:19:43 - <Notice> - Pcap-file module read 1 files, 200 packets
, 54238 bytes
```

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 Log Files Again**:

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

- After running Suricata, additional files appear, including `fast.log` and `eve.json`.

```
analyst@e4ffb3c23add:~$ ls -l /var/log/suricata
total 16
-rw-r--r-- 1 root root 1433 Oct 20 09:19 eve.json
-rw-r--r-- 1 root root  292 Oct 20 09:19 fast.log
-rw-r--r-- 1 root root 2912 Oct 20 09:19 stats.log
-rw-r--r-- 1 root root  357 Oct 20 09:19 suricata.log
```

4. **Display the Fast Log**:

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

```
analyst@e4ffb3c23add:~$ 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

1. **Display the eve.json Content**:

Command:

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

The output returns the raw content of the file. A lot of data returned are not easy to understand in this format.

2. **Format JSON Output Using jq**:

   `jq` command display the entries in an improved format

Command:

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

This command improves readability of the JSON output.

```
analyst@e4ffb3c23add:~$ jq . /var/log/suricata/eve.json | less
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 1570764369852565,
  "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": "/",
```

3. **Extract Specific Data**:

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

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

```
analyst@e4ffb3c23add:~$ jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.d
est_ip]" /var/log/suricata/eve.json
["2022-11-23T12:38:34.624866+0000",1570764369852565,"GET on wire","TCP","142.2
50.1.139"]
["2022-11-23T12:38:58.958203+0000",1845015213151476,"GET on wire","TCP","142.2
50.1.102"]
```

4. **Display Logs for a Specific Flow ID**:

Command:

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

The flow_id value is a 16-digit number and will vary for each of the log entries. Replace X with any of the flow_id values returned by the previous query.

```
analyst@e4ffb3c23add:~$ jq "select(.flow_id==1570764369852565)" /var/log/suric
ata/eve.json
{
  "timestamp": "2022-11-23T12:38:34.624866+0000",
  "flow_id": 1570764369852565,
  "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",
```

5. Results

- Successfully triggered a custom rule in Suricata and examined the output logs.
- Gained insights into the types of network traffic being monitored and the alerts generated by Suricata based on custom rules.

6. Conclusion

This project provided hands-on experience in using Suricata for network traffic analysis. By examining custom rules, triggering alerts, and analyzing log files, significant skills were developed that are essential for a career in cybersecurity, particularly in the role of a security analyst.