Suricata

Introduction

Suricata is a high performance, open source network analysis and threat detection software used by most private and public organizations, and embedded by major vendors to protect their assets.

Suricata is far more than an IDS/IPS.

- Home - Suricata
- ??Suricata???? ??????????? | ??? (netadmin.com.tw)

Suricata features

There are three main ways Suricata can be used:

- **Intrusion detection system** (**IDS**): As a network-based IDS, Suricata can monitor network traffic and alert on suspicious activities and intrusions. Suricata can also be set up as a host-based IDS to monitor the system and network activities of a single host like a computer.
- **Intrusion prevention system** (**IPS**): Suricata can also function as an intrusion prevention system (IPS) to detect and block malicious activity and traffic. Running Suricata in IPS mode requires additional configuration such as enabling IPS mode.
- **Network security monitoring** (**NSM**): In this mode, Suricata helps keep networks safe by producing and saving relevant network logs. Suricata can analyze live network traffic, existing packet capture files, and create and save full or conditional packet captures. This can be useful for forensics, incident response, and for testing signatures. For example, you can trigger an alert and capture the live network traffic to generate traffic logs, which you can then analyze to refine detection signatures.

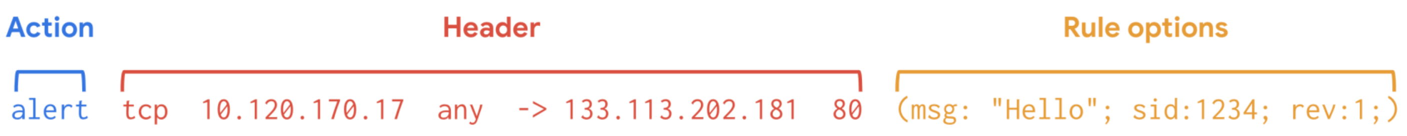
Signatures (Rules)

Suricata uses **signatures analysis**, which is a detection method used to find events of interest. Signatures consist of three components:

- **Action**: The first component of a signature. It describes the action to take if network or system activity matches the signature. Examples include: alert, pass, drop, or reject.

- **Header**: The header includes network traffic information like source and destination IP addresses, source and destination ports, protocol, and traffic direction.
- **Rule options:** The rule options provide you with different options to customize signatures.

Here's an example of a Suricata signature:



```
alert tcp 10.120.170.17 any -> 133.113.202.181 80 (msg: "Hello"; sid:1234; rev:1;)
```

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

Action

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.

> **Note:** Rule order refers to the order in which rules are evaluated by Suricata. Rules are loaded in the order in which they are defined in the configuration file. However, Suricata processes rules in a different default order: pass, drop, reject, and alert. Rule order affects the final verdict of a packet.

Header

`$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.

Rule options

- 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.

Confiuration file

Configuration files let you customize exactly how you want your IDS to interact with the rest of your environment.

Suricata's configuration file is `suricata.yaml`, which uses the YAML file format for syntax and structure.

Log files

There are two log files that Suricata generates when alerts are triggered:

- **eve.json**: The eve.json file is the standard Suricata log file. This file contains detailed information and metadata about the events and alerts generated by Suricata stored in JSON format. For example, events in this file contain a unique identifier called flow_id which is used to correlate related logs or alerts to a single network flow, making it easier to analyze network traffic. The eve.json file is used for more detailed analysis and is considered to be a better file format for log parsing and SIEM log ingestion.
- **fast.log**: The fast.log file is used to record minimal alert information including basic IP address and port details about the network traffic. The fast.log file is used for basic logging and alerting and is considered a legacy file format and is not suitable for incident response or threat hunting tasks.

The main difference between the eve.json file and the fast.log file is the level of detail that is recorded in each. The fast.log file records basic information, whereas the eve.json file contains additional verbose information.

Trigger a custom rule

With a packet capture file

- 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.

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

Check the logs

```
# For fast.log
cat /var/log/suricata/fast.log


# For eve.log, using jq command to display the JSON format
jq . /var/log/suricata/eve.json | less
jq -c "[.timestamp,.flow_id,.alert.signature,.proto,.dest_ip]" /var/log/suricata/eve.json
jq "select(.flow_id==1200997752018164)" /var/log/suricata/eve.json
```

Resources

- Suricata user guide
- Suricata features
- Rule management
- Rule performance analysis
- Suricata threat hunting webinar
- Introduction to writing Suricata rules
- Eve.json jq examples

Revision #21
Created 26 August 2024 11:24:07 by Admin
Updated 16 September 2024 12:16:31 by Admin