Suricata

Suricata is an open-source intrusion detection system, intrusion prevention system, and network analysis tool.

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.

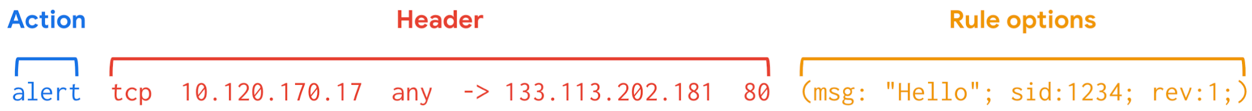
Rules

Rules or signatures are used to identify specific patterns, behavior, and conditions of network traffic that might indicate malicious activity. The terms rule and signature are often used interchangeably in Suricata. Security analysts use **signatures**, or patterns associated with malicious activity, to detect and alert on specific malicious activity. Rules can also be used to provide additional context and visibility into systems and networks, helping to identify potential security threats or vulnerabilities.

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;)
```

Rule options have a specific ordering and changing their order would change the meaning of the rule.

Note: The terms rule and signature are synonymous.

Note: Rule order refers to the order in which rules are evaluated by Suricata. Rules are processed 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 especially when conflicting actions such as a drop rule and an alert rule both match on the same packet.

Custom rules

Although Suricata comes with pre-written rules, it is highly recommended that you modify or customize the existing rules to meet your specific security requirements.

There is no one-size-fits-all approach to creating and modifying rules. This is because each organization's IT infrastructure differs. Security teams must extensively test and modify detection signatures according to their needs.

Creating custom rules helps to tailor detection and monitoring. Custom rules help to minimize the amount of false positive alerts that security teams receive. It's important to develop the ability to write effective and customized signatures so that you can fully leverage the power of detection technologies.

Configuration file

Before detection tools are deployed and can begin monitoring systems and networks, you must properly configure their settings so that they know what to do. A **configuration file** is a file used to configure the settings of an application. 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.