

SURICATA

Intrusion Detection System

Agenda

Introduction to IDS

What is Suricata

Installation & Configuration

Configuration Files

Suricata Rules

Network Intrusion Detection using Suricata

Integrating IDS with SIEM

Intrusion Detection System

- An **Intrusion Detection System** (IDS) is a cybersecurity solution designed to monitor network traffic and devices for **anomalies**, **malicious activities**, and **policy violations** (e.g.: Port scanning or Nmap Scans).
- An IDS can be implemented as software on a device, dedicated hardware, or a cloud-based solution.
- An IDS operates in passive mode. It only detects and reports potential threats, without taking any action, unlike an IPS.
- IDS can be implemented alongside **Honeypots** and **Canaries**.
- **Types of IDS:** Network-based, Host-based, Protocol-based, Application Protocol-based and Hybrid.

Common IDS Types

→ **Network-based Intrusion Detection System (NIDS)**

NIDS monitors traffic across a network by identifying known patterns of suspicious activity. They inspect both sides of network communications and, in IPS mode, can block malicious traffic when a threat is detected.

Typically, NIDS is connected to the network through a SPAN/mirror port or a network tap, allowing them to capture and analyze traffic without affecting network operations (e.g.: *Snort, Suricata*).

→ **Host-based Intrusion Detection System (HIDS)**

HIDS agent is installed on a host device (a server or workstation) to monitor and report system activities, application logs, and system calls.

It focuses on monitoring the device's internal behaviour such as running processes and events, Registry settings and Network traffic (e.g.: *OSSEC, Tripwire*).

IDS/IPS can and can't do



Yes

- Monitor network traffic
- Detect known threats
- Identify Anomalies
- Block malicious traffic (IPS only)
- Provide Forensic Data



No

- Detect zero-day attacks reliably
- Remediate compromised systems
- Replace firewalls
- Prevent insider threats
- Cannot fully eliminate false positives

Suricata

- ❖ **Suricata** is a high performance, open-source **network analysis** and **threat detection** software used by many organizations and embedded by major vendors to protect their assets.
- ❖ Suricata is developed and managed by OISF (Open Information Security Foundation).
- ❖ It provides **Real-time** analysis of network traffic from layer 3 to layer 7.
- ❖ Support for **multi-threading** and **hardware acceleration**, allowing efficient use of hardware.
- ❖ Suricata can integrate with platforms like the **Elastic Stack** for log management and can be used in conjunction with **Wazuh** or **Splunk** for enhanced security monitoring.

Installation & Configuration

→ **Suricata** can be installed on **OS X, Linux and Windows**.

It can be downloaded from its official web site: <https://suricata.io/download/>

Suricata can be configured to operate in two different modes.

- **Active (IPS):** Suricata is deployed in-line to prevent intrusions by blocking or dropping malicious packets in real-time.
- **Passive (IDS):** Suricata monitors network traffic, detects intrusions, and generates alerts without blocking the traffic.

It is placed out of band within the network infrastructure.

Installation & Configuration

We will be using an Ubuntu server (latest version).

To install Suricata we must first add the official OISF repository by this commands:

- *sudo apt-get install software-properties-common*
- *sudo add-apt-repository ppa:oisf/suricata-stable*
- *sudo apt update*
- *sudo apt install suricata jq*

- *sudo systemctl status suricata*

Installation & Configuration

```
root@suricata:/home/suricata# sudo apt install suricata jq -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jq is already the newest version (1.7.1-3build1).
jq set to manually installed.
The following additional packages will be installed:
```

Command
Results

```
root@suricata:/home/suricata# service status suricata
status: unrecognized service
root@suricata:/home/suricata# service suricata status
● suricata.service - LSB: Next Generation IDS/IPS
  Loaded: loaded (/etc/init.d/suricata; generated)
  Active: active (exited) since Tue 2024-11-19 02:11:35 UTC; 35s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 425895 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
  CPU: 81ms

Nov 19 02:11:34 suricata systemd[1]: Starting suricata.service - LSB: Next Generation IDS/IPS...
Nov 19 02:11:35 suricata suricata[425895]: Starting suricata in IDS (af-packet) mode... done.
Nov 19 02:11:35 suricata systemd[1]: Started suricata.service - LSB: Next Generation IDS/IPS.
root@suricata:/home/suricata#
```

Configuration Files

- To effectively monitor network traffic, we must specify the correct **interfaces** and the appropriate **network address**.
- Suricata's configuration are stored in the **suricata.yaml** file, which is located in the /etc/suricata/ directory.

```
15 vars:  
16   # more specific is better for alert accuracy and performance  
17   address-groups:  
18     HOME_NET: "[10.15.33.0/24]"  
19     #HOME_NET: "[192.168.0.0/16]"  
20     #HOME_NET: "[10.0.0.0/8]"  
21     #HOME_NET: "[172.16.0.0/12]"  
22     #HOME_NET: "any"  
23  
24     EXTERNAL_NET: "!$HOME_NET"  
25     #EXTERNAL_NET: "any"  
26
```

```
812  
813 # Cross platform libpcap capture support  
814 pcap:  
815   - interface: ens33  
816     # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"  
817     # as total memory used by the ring. So set this to something bigger  
818     # than 1% of your bandwidth.  
819     #buffer-size: 16777216  
820     #bpf-filter: "tcp and port 25"  
821     # Choose checksum verification mode for the interface. At the moment  
822     # of the capture, some packets may have an invalid checksum due to  
823     # the checksum computation being offloaded to the network card.  
824     # Possible values are:  
825     # - yes: checksum validation is forced  
826     # - no: checksum validation is disabled  
827     # - auto: Suricata uses a statistical approach to detect when  
828     # checksum off-loading is used. (default)  
829     # Warning: 'capture.checksum-validation' must be set to yes to have any val
```

Rules

- **Rules** are a set of instructions that tell Suricata how to detect attacks, unusual behaviour, or specific network events.
- For example, a rule might look for a known malware signature, an unauthorized login attempt, or a large data transfer to an unfamiliar server.
- Suricata provides predefined rules for various scenarios, and **custom rules** can be configured to suit our environment.
- It **compares** network packets against predefined **rules**. If a packet matches, the configured action determines whether to drop or log the traffic.

Rules

→ Suricata rules

Not all rules are enabled by default, a pound symbol indicates a **disabled rule**.

```
# alert modbus any any -> any any (msg:"SURICATA Modbus Data mismatch"; flow:to_client; app-layer-event:modbus.value_mismatch;)
# alert modbus any any -> any any (msg:"SURICATA Modbus Request flood detected"; flow:to_server; app-layer-event:modbus.flood;)
alert mqtt any any -> any any (msg:"SURICATA MQTT CONNECT not seen before CONNACK"; app-layer-event:mqtt.missing_connect; count:1, offset:1, duration:10,)
alert mqtt any any -> any any (msg:"SURICATA MQTT PUBLISH not seen before PUBACK/PUBREL/PUBREC/PUBCOMP"; app-layer-event:mqtt.missing_publish; count:1, offset:1, duration:10,)
alert mqtt any any -> any any (msg:"SURICATA MQTT SUBSCRIBE not seen before SUBACK"; app-layer-event:mqtt.missing_subscribe; count:1, offset:1, duration:10,)
```

```
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN nmap TCP"; ack:0; flags:A,12; flow:stateless; reference:arachnids,198; updated_at 2019_07_26;)
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN nmap XMAS"; flow:stateless; flags:FPU,12; reference:arachnids,30; updated_at 2019_07_26;)
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN nmap fingerprint attempt"; flags:SFPU; flow:stateless; reference:arachnids,198; updated_at 2019_09_23, updated_at 2019_07_26;)
# alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"GPL SCAN myscan"; flow:stateless; ack:0; flags:S; ttl:>220; reference:arachnids,198; updated_at 2019_09_23, updated_at 2019_07_26;)
# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"GPL SCAN SYN FIN"; flow:stateless; flags:SF,12; reference:arachnids,198; count:1, offset:1, duration:10,)
```

Rules

- Suricata allows us to **download rules** from various sources. You can view the **list of available sources** by entering the command: `suricata-update list-sources`.

```
root@suricata:/etc/suricata# suricata-update list-sources
20/11/2024 -- 19:16:21 - <Info> -- Using data-directory /var/lib/suricata.
20/11/2024 -- 19:16:21 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
20/11/2024 -- 19:16:21 - <Info> -- Using /usr/share/suricata/rules for Suricata provisioning
20/11/2024 -- 19:16:21 - <Info> -- Found Suricata version 7.0.7 at /usr/bin/suricata.
20/11/2024 -- 19:16:21 - <Warning> -- Source index does not exist, will use bundled configuration
20/11/2024 -- 19:16:21 - <Warning> -- Please run suricata-update update-sources.

Name: et/open
  Vendor: Proofpoint
  Summary: Emerging Threats Open Ruleset
  License: MIT

Name: et/pro
  Vendor: Proofpoint
  Summary: Emerging Threats Pro Ruleset
  License: Commercial
  Replaces: et/open
  Parameters: secret-code
  Subscription: https://www.proofpoint.com/us/threat-insight/et-pro-ruleset

Name: etnetera/aggressive
  Vendor: Etnetera a.s.
  Summary: Etnetera aggressive IP blacklist
  License: MIT
```

- Selected rules can be enabled by the command `suricata-update enable <source-name>`
- After enabling the source, you must update the rules by entering `suricata-update`

Rules

Custom rules can be created and added to the Suricata configuration.

Rules Examples;

- A suspicious ping from the Corporate Network to the ICS Network.

```
alert icmp $CORP_NET any -> $ICS_NET any (msg: "Suspicious PING Detected"; sid:1; rev:1;)
```

- Detecting a Brute Force attack

```
alert tcp any any -> any 22 (msg:"Potential SSH Brute-Force Attack"; flags:S; threshold:type both, track by_src, count 5, seconds 60; sid:100002; rev:1;)
```

```
2164
2165 default-rule-path: /var/lib/suricata/rules
2166
2167 rule-files:
2168   - suricata.rules
2169   -----
```



After adding these rules into a .rules file, you must specify the path to that file in suricata.yaml

Configuration Test

Before running Suricata, we can execute the configuration file in the test environment to verify whether it detects the correct interface, utilizes the rules, generates .json logs, and so on.

`suricata -T -c /etc/suricata/suricata.yaml -v`

Options:

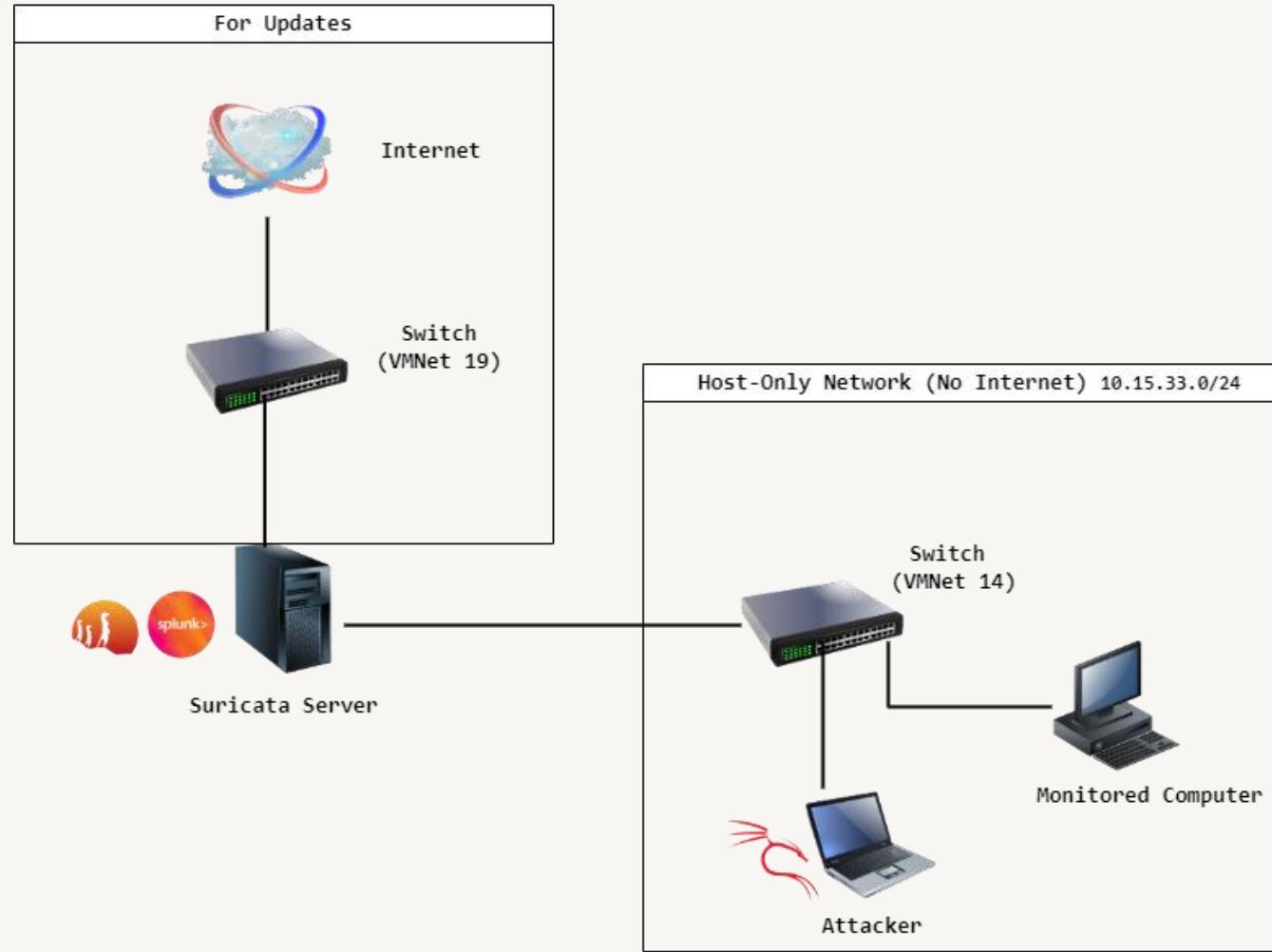
- T for test mode
- c specify the configuration file
- v for verbose output

```
File Edit View Search Terminal Help
root@suricata:/etc/suricata# suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.7 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 4
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 1 rule files processed. 40728 rules successfully loaded, 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 40731 signatures processed. 1191 are IP-only rules, 4261 are inspecting packet payload,
application layer, 108 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
root@suricata:/etc/suricata#
```

Summary

- We learned about **IDS** and the different types, including **NIDS** and **HIDS**.
- We explored the purpose and use of IDSs.
- Next, we studied **Suricata**, including the supported platforms and its different modes of operation, such as **IPS** and **IDS**.
- We went through the installation process of Suricata, starting with adding the **repository** followed by the actual installation.
- After installation, we configured the **suricata.yaml** file and reviewed the **rules** associated with Suricata.

Demonstration Topology



References

Suricata - <https://suricata.io/>

OISF - <https://oisf.net/>

Splunk - <https://www.splunk.com/>

Ubuntu - <https://ubuntu.com/download/server>

Documentation - <https://docs.suricata.io/en/latest/install.html>

IDS/IPS - <https://www.paloaltonetworks.ca/cyberpedia/what-is-an-intrusion-detection-system-ids>

<https://www.ibm.com/topics/intrusion-detection-system>