Suricata Installation and Integration with Wazuh

1. Installing and Configuring Suricata on the Gateway

Suricata Installation (IPS Mode)

*Suricata was installed and configured to run in **Intrusion Prevention System (IPS)** mode.*

```
root@vbox:/etc/suricata/rules# suricata --build-info
This is Suricata version 6.0.10 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HAVE_LUA HAVE_LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST
SIMD support: none
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 12.2.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.42, linked against LibHTP v0.5.42

Suricata Configuration:
  AF_PACKET support:                       yes
  eBPF support:                            yes
  XDP support:                             yes
  PF_RING support:                         no
  NFQueue support:                         yes
  NFLOG support:                           yes
```

⚙️ Configuration (`/etc/suricata/suricata.yaml`)

➤ Define Networks

```
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.100.1/24]"
    EXTERNAL_NET: "!$HOME_NET"
```

➤ Configure Network Interfaces

```
af-packet:
  - interface: enp0s3
    cluster-id: 99
    cluster-type: cluster_flow
    defrag: yes
  - interface: default
```

➤ **Configure Rule Sources**

Custom rules were added to the `custom.rules` file. Additionally, community rules from Emerging Threats (ET) were integrated:

```
rule-files:
  - custom.rules
  - emerging-web_server.rules
  - emerging-sql.rules
  - emerging-scan.rules
```

emerging-web_server.rules

- **Source: From the Emerging Threats (ET) community rule set.**
- **Purpose: Detects attacks against web servers**

emerging-sql.rules

- **Purpose: Focused on detecting SQL injection attacks, one of the OWASP Top 10 threats.**

emerging-scan.rules

- **Purpose: Detects network scanning behavior**

custom.rules

Purpose: A user-defined rule file:

Rule 1: SYN Scan Detection

```
drop tcp any any -> any any (msg:"Possible Syn Scan Technique attempted";
flow:to_server, stateless; flags:S; window:1024; detection_filter:track
by_src, count 100, seconds 8;
```

```
classtype:attempted_port_scan_from_the_internet; sid:40000003; rev:1;
priority:6;)
```

🔍 **What it does:**

- **Purpose**: Detects a **TCP SYN scan** (commonly used by Nmap).
- **Mechanism**: Watches for 100 SYN packets from a single source in 8 seconds.
- **Trigger**: Matches TCP packets with only the SYN flag and a specific window size (`1024`), often indicative of Nmap behavior.
- **Action**: **Drops** the packet to prevent reconnaissance activity.

Rule 2: Hydra Brute-force Login Attempt

```
drop http $EXTERNAL_NET any -> $HOME_NET 80 (msg:"WEB Attack - Hydra
Brute-force attempt on DVWA login"; flow:to_server,established;
content:"/dvwa/login.php"; http_uri; content:"Hydra"; http_user_agent;
classtype:web-application-attack; sid:40000004; rev:1;)
```

🔍 **What it does:**

- **Purpose**: Detects brute-force login attempts targeting **DVWA** using **Hydra**.
- **Mechanism**:
  - Matches HTTP requests to `/dvwa/login.php`
  - Looks for the **User-Agent** string containing `"Hydra"`
- **Trigger**: When Hydra is actively brute-forcing the login page.
- **Action**: **Drops** the malicious HTTP request.

2. Configuring Wazuh Agent to Send Suricata Logs

```
root@vbox:/etc/suricata/rules# sudo systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-06-08 22:23:57 EEST; 1h 36min ago
    Process: 29410 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
      Tasks: 29 (limit: 3484)
     Memory: 44.9M
        CPU: 58.287s
     CGroup: /system.slice/wazuh-agent.service
             ├─29433 /var/ossec/bin/wazuh-execd
             ├─29444 /var/ossec/bin/wazuh-agentd
             ├─29458 /var/ossec/bin/wazuh-syscheckd
             ├─29471 /var/ossec/bin/wazuh-logcollector
             └─29484 /var/ossec/bin/wazuh-modulesd
```

🔧 Edit Agent Configuration File

File: `/var/ossec/etc/ossec.conf`

Add the following block to monitor the Suricata alert file (`eve.json`):

```
<ossec_config>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/suricata/eve.json</location>
  </localfile>
  <logging>
    <log_format>json</log_format>
  </logging>
```

This configuration allows the **Wazuh agent** to collect Suricata alerts and send them to the **Wazuh manager**, where they can be analyzed and correlated in the Dashboard.