# Assignment 5: Deployment and Testing of Intrusion Detection Systems (Suricata)

Student: Shynbolat Unaibayev

## Objectives

• Understand IDS/IPS principles.  
• Install and configure Suricata on Debian.  
• Generate normal and malicious traffic, analyze logs, and create custom rules.

## Environment

OS: Debian (testing environment)  
Suricata version: 7.0.10  
Tools used: curl, wget, ping, nmap, hping3, python3 (simple http.server), jq

## 1. IDS Installation

Commands run:

sudo apt update && sudo apt upgrade -y  
sudo apt install suricata -y  
suricata -V  
sudo suricata -T -c /etc/suricata/suricata.yaml

## 

## 2. Basic Configuration

Configured Suricata to monitor the loopback interface (lo) and enabled logging outputs (fast.log and eve.json).

Key configuration snippets:

af-packet:  
 - interface: eth3  
 cluster-id: 99  
 cluster-type: cluster\_flow  
 defrag: yes

outputs:  
 - fast:  
 enabled: yes  
 filename: fast.log  
 - eve-log:  
 enabled: yes  
 filetype: regular  
 filename: eve.json  
 types:  
 - alert  
 - dns  
 - http

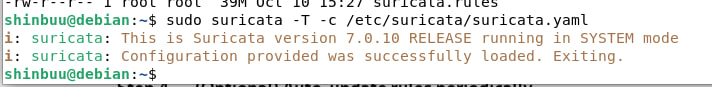
## 3. Normal Traffic Testing

Procedure:

1. Start Suricata on loopback: sudo suricata -c /etc/suricata/suricata.yaml -i lo -v  
2. Generate benign traffic:  
 - ping -c 3 127.0.0.1  
 - curl http://example.com  
 - wget http://example.com -O /tmp/test.html

Observed behavior: Normal traffic produced entries in eve.json (HTTP, DNS, flow events) and typically did not trigger alerts in fast.log.

{"event\_type":"http","src\_ip":"127.0.0.1","dest\_ip":"93.184.216.34","http":{"hostname":"example.com","url":"/"}}



## 4. Attack Simulation

Performed controlled attacks on loopback (127.0.0.1):

- Port scanning with nmap (local): nmap -sS -Pn 127.0.0.1  
- HTTP requests with unusual headers and methods  
- Controlled ICMP rate using hping3 or nping  
- Multiple rapid HTTP GETs to simulate hammering

## 5. IDS Log Analysis

Log files analyzed: /var/log/suricata/eve.json and /var/log/suricata/fast.log

Fields inspected in alerts: event\_type, timestamp, src\_ip, dest\_ip, alert.signature, alert.signature\_id (sid), category, severity.

Sample alert (custom rule example):

{  
 "event\_type": "alert",  
 "timestamp": "2025-10-10T12:34:56.789012",  
 "src\_ip": "127.0.0.1",  
 "dest\_ip": "127.0.0.1",  
 "alert": {  
 "signature": "HTTP TRACE Detected",  
 "signature\_id": 100002  
 }  
}

## 6. Custom Rule Creation

Custom rules added to local.rules (path used during testing: /var/lib/suricata/rules/local.rules):

1) ICMP detection rule:  
alert icmp any any -> any any (msg:"ICMP Detected"; sid:100001; rev:1;)

2) HTTP TRACE detection rule:  
alert tcp any any -> any 8080 (msg:"HTTP TRACE Detected"; flow:to\_server,established; pcre:"/^TRACE\s/"; sid:100002; rev:1;)

3) Long header detection rule:  
alert tcp any any -> any 8080 (msg:"HTTP Long Header X-Long"; flow:to\_server,established; pcre:"/X-Long:\s\*A{500,}/smi"; sid:100003; rev:1;)

Testing steps: restart Suricata or run in foreground on lo, then generate matching traffic (ping, curl TRACE, curl with long X-Long header).

]

## Comparison between Normal and Attack Traffic

Normal traffic produced informational http/dns/flow events without alerts. Attack simulations triggered alert events with associated SIDs and signatures. Custom rules fired for the TRACE method and long header cases during testing.

Limitations observed:  
- Large headers may be fragmented across multiple packets; PCRE matching could fail if pattern spans packets.  
- Running in AF\_PACKET mode requires root; performance considerations apply for high-rate tests.  
- Potential for false positives; tuning required in production.

## Conclusions

Suricata successfully installed and configured on Debian, monitored the loopback interface, and logged both normal and malicious activity. Custom rules were created and validated. The experiment demonstrates Suricata's flexibility and highlights the need for careful tuning and testing in a production environment.

## Appendix: Commands and Outputs

Key commands used during the assignment (copyable):

sudo apt update && sudo apt upgrade -y  
sudo apt install suricata -y  
sudo suricata -T -c /etc/suricata/suricata.yaml  
sudo suricata -c /etc/suricata/suricata.yaml -i lo -v  
sudo suricata-update  
sudo systemctl restart suricata  
ping -c 3 127.0.0.1  
curl -i -X TRACE http://127.0.0.1:8080  
curl -I -H "X-Long: A... (long header)" http://127.0.0.1:8080  
nmap -sS -Pn 127.0.0.1  
sudo hping3 --icmp -c 200 -i u2000 127.0.0.1