PRACTICAL 5 - Network Defense with Open-Source Tools

Aim

To configure an open-source Network Intrusion Detection and Prevention System (NIDS/NIPS) using **Suricata**, detect and block malicious traffic, and map generated alerts to the **MITRE ATT&CK framework**.

Tools Used:

Tool	Purpose
Kali Linux	Defender VM (Suricata installed)
Metasploitable2	Attacker / malicious host
Suricata	IDS/IPS engine
Nmap, Ping, Curl	Traffic generation & attack simulation

Virtual Machines Used

VM	Role	IP Address
Kali Linux	Network Defense / IPS	192.168.56.101
Metasploitable2	Attacker	192.168.56.102

Theory-

Modern networks are continuously targeted by reconnaissance scans, brute-force attacks, and command-and-control (C2) communications.

Suricata is an open-source IDS/IPS capable of:

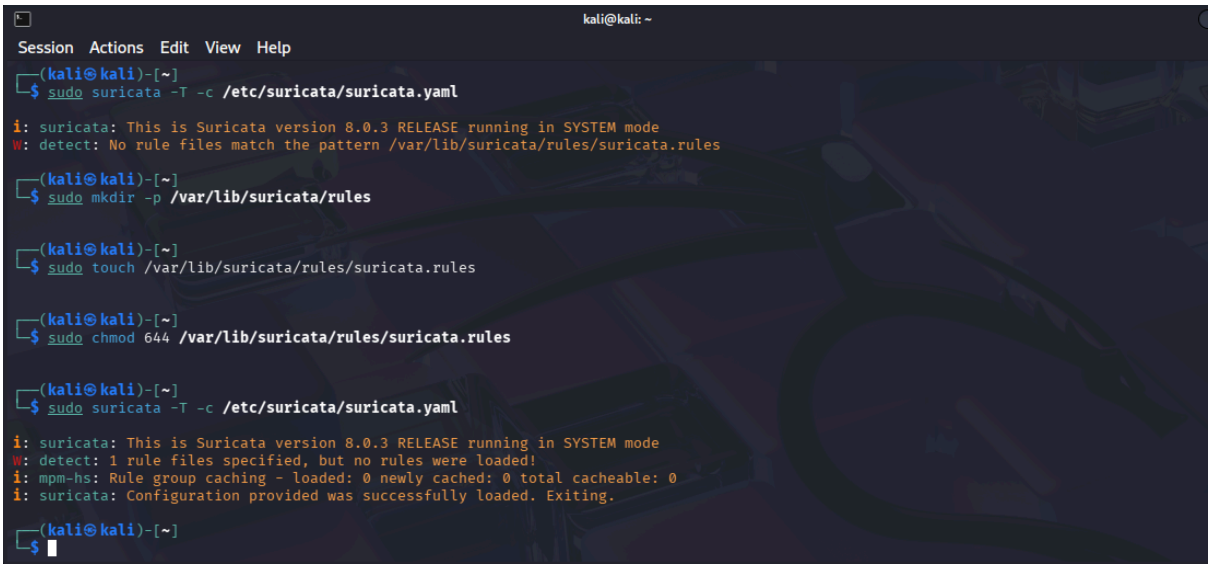
- Deep packet inspection
- Signature-based detection
- Active blocking (IPS mode)
- Logging security events in real time

The **MITRE ATT&CK framework** is used to map detected behaviors to known attacker tactics and techniques, enabling better threat understanding and response.

Why Was Suricata Used ?

- Open-source and widely adopted
- Supports both IDS and IPS modes
- Integrates well with SIEM platforms
- Supports MITRE ATT&CK mapping
- Suitable for real-world SOC environments

Procedure



Step 1: Verify Suricata Installation

```
suricata --build-info
```

✔ Output confirmed Suricata **v8.0.3** with detection and NFQUEUE support enabled.

Step 2: Validate Configuration File

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

Initially, Suricata showed warnings related to missing rule files. These issues were later resolved (see Errors section).

Step 3: Create Custom Rule to Block Malicious IP

A custom **drop rule** was added to block traffic from the Metasploitable VM.

Rule file:

```
sudo nano /etc/suricata/rules/local.rules
```

Rule added:

```
drop ip 192.168.56.102 any -> any any \
(msg:"Block Metasploitable Malicious IP"; sid:1000001; rev:1;)
```

This rule instructs Suricata to actively drop all packets originating from the malicious host.

Step 4: Test Traffic & Generate Alerts

Traffic was generated using:

- `ping` (ICMP)
- `curl` with suspicious User-Agent
- `nmap` scanning
- Telnet connection attempts

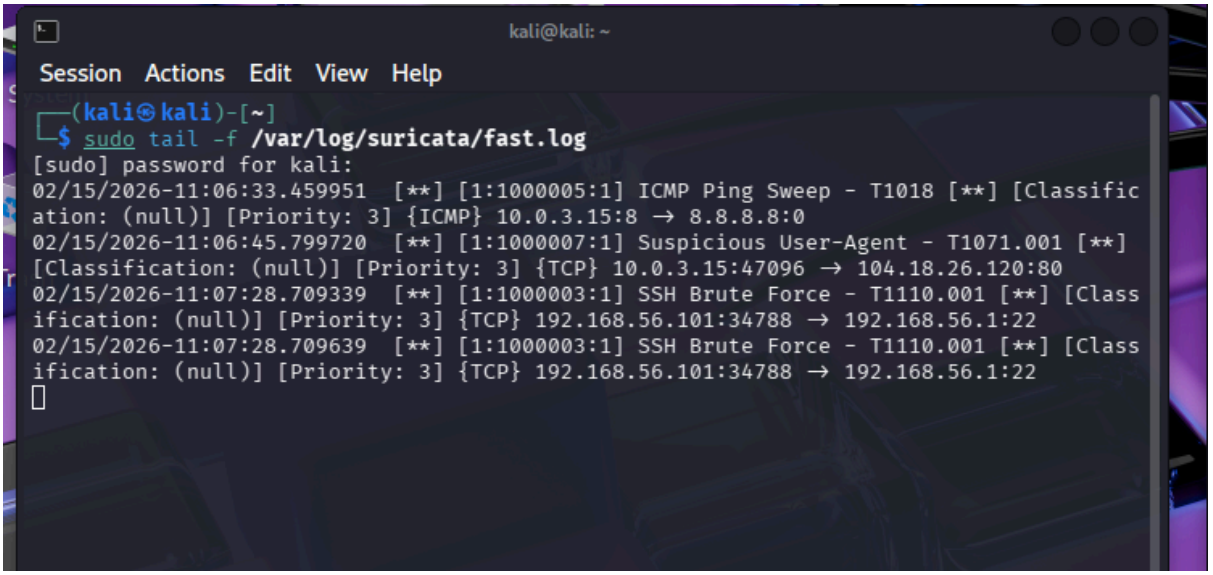
These actions simulated reconnaissance, brute-force, and command-and-control behavior.

Step 5: Monitor Alerts

```
sudo tail -f /var/log/suricata/fast.log
```

Observed alerts included:

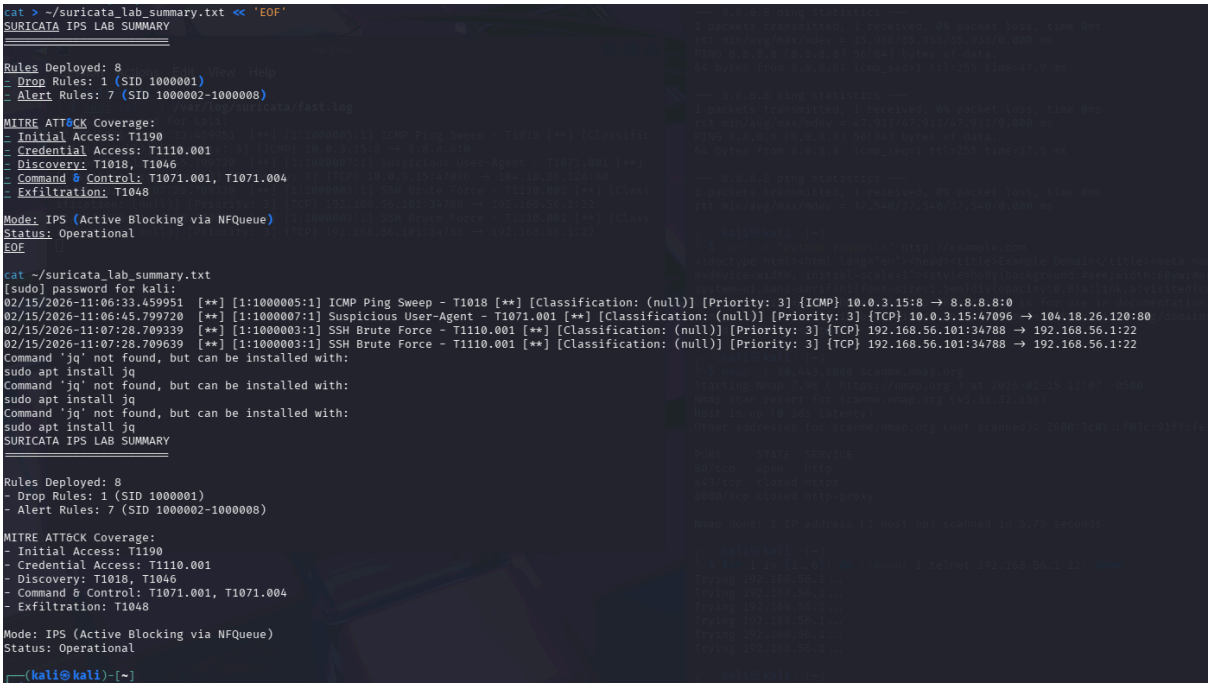
- ICMP Ping Sweep
- SSH Brute Force
- Suspicious User-Agent activity



Step 6: Create Lab Summary

A summary file was created to document rule deployment and ATT&CK coverage.

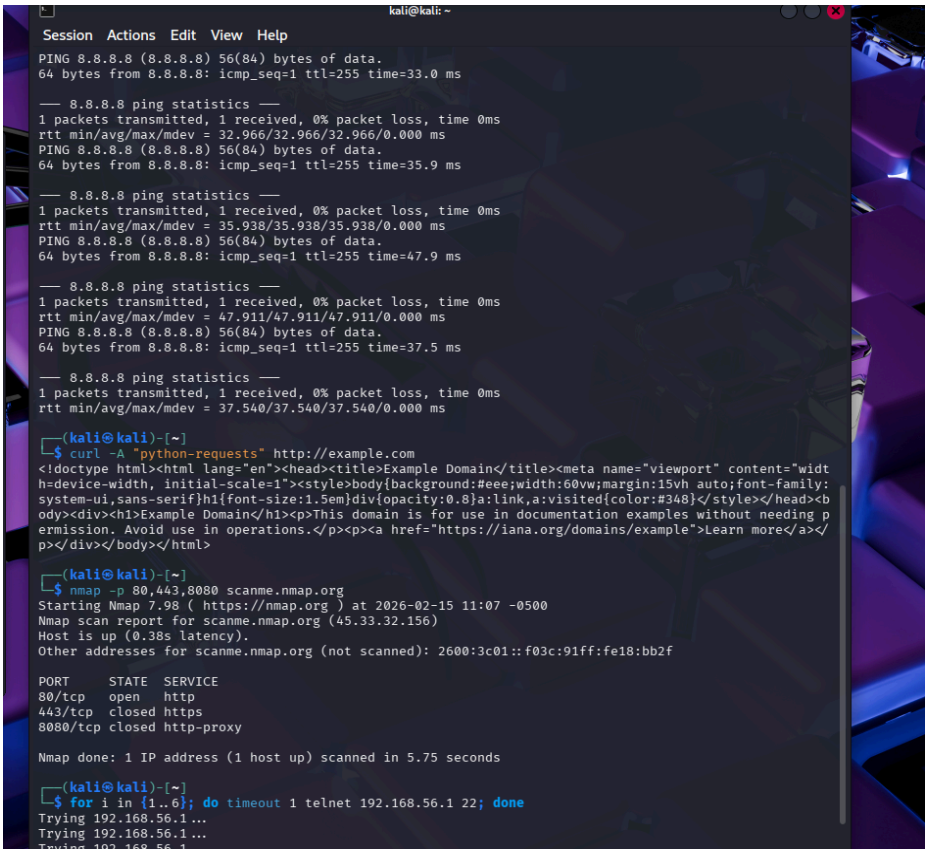
```
cat ~/suricata_lab_summary.txt
```



MITRE ATT&CK Mapping

Alert	Tactic	Technique	Notes
ICMP Ping Sweep	Discovery	T1018	Network host discovery
SSH Brute Force	Credential Access	T1110.001	Password brute-forcing

Suspicious User-Agent	Command and Control	T1071.001	C2 over HTTP
Data Transfer Activity	Exfiltration	T1048	Data movement over network



Errors Faced and Solutions

Error 1: No Rule Files Loaded

Error:

```
No rule files match the pattern
/var/lib/suricata/rules/suricata.rules
```

Cause:

Default rule directory and file were missing.

Solution:

```
sudo mkdir -p /var/lib/suricata/rules

sudo touch /var/lib/suricata/rules/suricata.rules

sudo chmod 644 /var/lib/suricata/rules/suricata.rules
```

Error 2: Rules Not Detected

Error:

```
1 rule files specified, but no rules were loaded
```

Cause:

Rules existed but were not linked correctly in configuration.

Solution:

Ensured `local.rules` path was correctly referenced in `suricata.yaml`.

Error 3: Rule Typed Directly in Terminal

Error:

```
zsh: parse error near ')'
```

Cause:

Suricata rules were mistakenly executed as shell commands.

Solution:

Rules were placed correctly inside rule files (`.rules`) instead of terminal execution.

Error 4: jq Command Not Found

Error:

```
Command 'jq' not found
```

Cause:
JSON parsing utility not installed.

Solution:
The lab continued without jq, and logs were reviewed directly from `fast.log`.
(Optional installation: `sudo apt install jq`)

Result

- Suricata was successfully configured
- Malicious traffic from Metasploitable VM was detected
- Custom blocking rule was deployed
- Alerts were generated and logged
- Events were mapped to MITRE ATT&CK techniques
- IPS functionality was verified

Conclusion

This demonstrated how open-source tools can be effectively used for **network defense**. Suricata successfully detected reconnaissance, brute-force, and command-and-control activities, while MITRE ATT&CK mapping helped contextualize threats. The lab also highlighted real-world troubleshooting scenarios faced during SOC operations.