Snort Project 2 – False Positive Optimization Using Custom Rules

1. Project Overview

This project is an extension of **Snort Project 1** and focuses specifically on **false positive optimization** in Intrusion Detection Systems (IDS). The objective is to understand how poorly written rules can generate excessive alerts and how rule optimization techniques can significantly reduce false positives while maintaining detection accuracy.

The project uses **Snort 3** on Kali Linux and implements optimized rules for common attack patterns such as SQL Injection, Command Injection, ICMP reconnaissance, and TCP scans.

---

2. Tools & Environment Used

- Operating System: **Kali Linux**
- IDS Tool: **Snort 3**
- Configuration File: snort.lua
- Rules File: local.rules
- Editor: nano

---

3. What Are False Positives?

A false positive occurs when legitimate network traffic is incorrectly flagged as malicious. Excessive false positives can overwhelm security analysts and reduce the effectiveness of an IDS.

This project demonstrates how rule tuning and optimization techniques help reduce unnecessary alerts.

---

4. Optimized Rules Implemented

4.1 Web SQL Injection Detection (Optimized)

```
alert tcp any any -> any 80 (
 msg:"WEB Possible SQL Injection";
 content:"select";
```

```
 sid:1000001;
)
```

Optimization Aspect: - Focuses on HTTP traffic only (port 80) - Content-based inspection instead of generic alerts

---

4.2 Command Injection Detection (Optimized)

```
alert tcp any any -> $HOME_NET any (
 msg:"Possible Command Injection";
 content:"cmd";
 sid:2000009;
)
```

Optimization Aspect: - Restricted to HOME_NET to reduce external noise - Content matching avoids broad pattern alerts

---

4.3 ICMP Reconnaissance Detection with Thresholding

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any \
 (msg:"ICMP Recon Detected"; itype:8; \
 detection_filter:track by_src, count 3, seconds 30; sid:2000001;)
```

Optimization Aspect: - Uses detection_filter to alert only after multiple packets - Prevents alerts on normal single ping requests

---

4.4 TCP SYN Scan Detection

```
alert tcp any any -> any any (msg:"TCP Scan"; flags:S; sid:2000002; rev:1;)
```

Optimization Aspect: - Detects SYN-only packets typical of port scanning - Avoids triggering on established connections

---

5. Execution & Testing

Snort was executed using:

```
sudo snort -c /etc/snort/snort.lua -i <interface> -A alert_fast
```

Configuration validation:

```
sudo snort -c /etc/snort/snort.lua -T
```

---

6. Results & Observations

- Significant reduction in alert noise
- Improved accuracy in detecting suspicious behavior
- Better distinction between normal and malicious traffic

---

7. Learning Outcomes

- Understanding false positives in IDS
- Rule optimization and tuning techniques
- Use of detection filters and content matching
- Practical SOC-level IDS tuning experience

---

8. Conclusion

This project highlights the importance of IDS rule optimization. By tuning Snort rules to reduce false positives, the system becomes more reliable and analyst-friendly, reflecting real-world SOC practices.
