🛡️ Security Monitoring and Incident Response Report

Project Title: Implementation of Security Monitoring and Incident Response
Environment: Parrot Security OS (Lab Network)
Date: April 24, 2025
Prepared by: [Your Name/Team]

📡 1. Security Monitoring Setup

🎯 Objective:

Establish a basic security monitoring system capable of detecting malicious activities within the lab environment, generate alerts, and support incident response.

🔧 Tools Used:

- **Snort IDS** – Network-based Intrusion Detection System.
- **Syslog + Logrotate** – For centralized log collection and archival.
- **Wireshark** – For traffic capture and packet analysis.
- **OpenCTI** – For correlating detected activities with known IoCs.

🛠️ Setup Procedure:

1. **Install Snort IDS**
   ```
   sql
   CopyEdit


   sudo apt update
   ```
2. `sudo apt install snort`
3.
    - Configuration File: `/etc/snort/snort.conf`

- Interface Monitored: `eth0`

4. **Configure Logging and Alerts**
   - Alert output: `/var/log/snort/alert`
   - Log rotation via `logrotate` to manage size and history.
5. **Update Rules**
   Pulled community rules for web server attacks, port scans, and suspicious SMB traffic.
   bash
   CopyEdit

```
wget https://www.snort.org/downloads/community/
community-rules.tar.gz
```

6. `tar -xvzf community-rules.tar.gz -C /etc/snort/rules`
7.

🔍 2. Use Case Demonstration: Apache Exploit Detection

🎯 Use Case:

Detect remote code execution (RCE) attempts against an outdated Apache web server (v2.4.52) on `192.168.1.100`.

🔐 Detection Rule:

snort
CopyEdit

```
alert tcp any any -> 192.168.1.100 80 (msg:"Possible Apache
RCE attempt"; content:"POST"; http_method; content:"/cgi-
bin/"; content:".sh"; sid:1000002; rev:1;)
```

🧭 Alert Prioritization Process:

Alert Severity	Description	Action

High	Exploitation or system compromise	Immediate response
Medium	Credential harvesting, brute-force	Investigate within 4 hrs
Low	Port scan or recon	Log and monitor

📋 Response Procedure (for High Alert):

1. **Log Review:** Check `/var/log/snort/alert` for match details.
2. **Packet Capture:** Use Wireshark to examine the payload and confirm exploit signature.
3. **Isolate Host:** Disconnect 192.168.1.100 from the network if confirmed.
4. **Mitigation:** Apply Apache patch (if not yet done) and restore from backup if necessary.
5. **IOC Lookup:** Cross-reference attack signature in OpenCTI for APT or malware associations.

🚨 3. Incident Response Scenario

🧠 Incident: RDP Brute Force Attempt

Date Detected: April 20, 2025
Affected Host: `192.168.1.101` (Windows VM)
Source IP: `198.51.100.23`
Method of Detection: Snort alert for multiple failed RDP login attempts

🧾 Incident Classification:

- **Type:** Unauthorized Access Attempt
- **Category:** Brute Force Attack
- **Impact:** Potential account compromise or system breach
- **Severity:** Medium (due to rapid repetition, but blocked by auth limits)

🛠️ Response Steps:

1. **Alert Acknowledgement**

- Received alert via Snort (rule triggered by >5 failed login attempts in 1 minute).

2. **Containment**
   - Added source IP to host-based firewall blocklist.
   - Enabled lockout policy after 3 failed attempts.
3. **Eradication**
   - Scanned 192.168.1.101 for malware (none found).
   - Reviewed Windows Event Logs to confirm no successful access.
4. **Recovery**
   - Reset RDP passwords.
   - Deployed 2FA for administrative access.
5. **Post-Incident Review**
   - Created Snort rule for RDP brute-force detection.
   - Updated documentation and trained team on incident signs.

📘 Lessons Learned:

- **Weak Point Identified:** Open RDP without account lockout or 2FA.
- **Response Speed:** Manual blocking was effective but could be automated.
- **Improvement:** Implemented lockout policy and Snort automation script for dynamic IP blocking.

📊 Summary

Component	Implementation
Monitoring System	Snort IDS, syslog, Wireshark, OpenCTI
Use Case	Apache RCE Detection via Custom Snort Rule
Alert Prioritization	Defined and documented in use case
Incident Response Scenario	RDP brute-force detection and response

Evidence of Functionality	Snort alerts, logs, response logs, IDS rule base

📎 Attachments and Evidence (Provide if submitting as a document)

- Snort rule configuration screenshots
- Snort alert logs (`/var/log/snort/alert`)
- Screenshot: Wireshark showing Apache payload
- Screenshot: Firewall block of malicious IP
- Screenshot: OpenCTI IoC query for Apache CVE