Snort Challenge - Live Attacks

# 📝 ****Full Incident Report – Snort Challenge: Live Attacks****

**Lab Name:** Snort Challenge – Live Attacks  
**Category:** Network Security & Traffic Analysis – Report

## **1. Executive Summary**

This lab simulated **real-world malicious network traffic** and required the creation and deployment of **Snort IDS rules** to detect live attacks in progress.  
Scenarios included web application exploitation, malware file transfers, and C2 communications.  
The goal was to **simulate SOC detection engineering and alert tuning in real time**.

## **2. Incident Timeline**

| **Time (UTC)** | **Event** |
| --- | --- |
| 10:00 | Suspicious HTTP traffic detected from external IP. |
| 10:05 | Snort deployed in IDS mode to monitor live attack traffic. |
| 10:10 | Rule triggered on suspicious file download over HTTP. |
| 10:15 | Second alert triggered on SQL injection payload in HTTP request. |
| 10:20 | C2 beaconing detected over non-standard TCP port. |
| 10:25 | Containment recommendations issued. |

## **3. Technical Investigation**

### **3.1 Tools Used**

* **Snort** – IDS mode for real-time packet inspection.
* **Wireshark** – Traffic verification & payload inspection.
* **Custom Snort Rules** – Written specifically for live attack signatures.

### **3.2 Attack Scenarios Detected**

#### **a. Malicious File Download**

* HTTP GET request for malware.exe from suspicious domain.
* Snort Rule Example:

snort

CopyEdit

alert tcp any any -> any 80 (msg:"Malware File Download Detected"; content:"malware.exe"; nocase; sid:2000001; rev:1;)

#### **b. SQL Injection Attempt**

* Detected UNION SELECT and ' OR 1=1 -- patterns in HTTP POST request.
* Rule Example:

snort

CopyEdit

alert tcp any any -> any 80 (msg:"SQL Injection Detected"; content:"UNION SELECT"; nocase; sid:2000002; rev:1;)

#### **c. Command & Control Communication**

* TCP traffic to external IP 198.51.100.34 on port 8080 at regular intervals (beaconing).
* Rule Example:

snort

CopyEdit

alert tcp any any -> 198.51.100.34 8080 (msg:"Potential C2 Traffic"; flags:S; sid:2000003; rev:1;)

## **4. MITRE ATT&CK Mapping**

| **Technique ID** | **Technique Name** | **Observation** |
| --- | --- | --- |
| T1071.001 | Application Layer Protocol: Web Traffic | Malware file download & SQLi over HTTP |
| T1190 | Exploit Public-Facing Application | SQL injection |
| T1105 | Ingress Tool Transfer | malware.exe download |
| T1071.004 | Application Layer Protocol: WebSockets | C2 beaconing on port 8080 |

## **5. Containment & Mitigation Recommendations**

**Immediate Actions:**

* Block malicious IPs at perimeter firewall.
* Terminate active C2 sessions.
* Isolate infected hosts for forensic analysis.

**Long-Term Measures:**

* Deploy Snort rules to production with auto-updates.
* Implement WAF rules for SQL injection prevention.
* Enhance proxy filtering for suspicious file downloads.

## **6. Lessons Learned**

* Real-time attack detection requires **precise rule tuning** to minimize false positives.
* Combining **custom rules** with **threat intel feeds** increases detection speed.
* Detecting **C2 beaconing** often requires **time-based traffic analysis** in addition to Snort signatures.

