

# Project Title: Automated Threat Mitigation with Wazuh and Suricata (IPS)

## Overview

This project demonstrates the implementation of a real-time **Intrusion Prevention System (IPS)**. By integrating **Suricata** (Network IDS) with **Wazuh** (SIEM/XDR), I created a closed-loop security system that detects network-level attacks and automatically executes a firewall-level ban on the attacker.

---

## The Architecture

The system follows a 4-step automated pipeline:

1. **Detection:** Suricata monitors network traffic and identifies threats via custom signatures.
  2. **Analysis:** The Wazuh Manager receives JSON logs from the Suricata agent, processes them via custom rules, and triggers high-severity alerts.
  3. **Active Response:** Upon reaching a severity threshold (Level 12), Wazuh triggers an **Active Response** command.
  4. **Mitigation:** A custom shell script on the agent parses the Suricata-specific `src_ip` field and updates `iptables` to drop all traffic from the malicious source.
- 

## Phase 1: Attack Simulation

To validate the system, an ICMP flood attack was launched from an external attacker VM (192.168.113.129) targeting the Suricata-protected node.

```
root@ubuntu:~# sudo ping -f -c 50 192.168.113.130
PING 192.168.113.130 (192.168.113.130) 56(84) bytes of data.
.....
--- 192.168.113.130 ping statistics ---
50 packets transmitted, 0 received, 100% packet loss, time 543ms

root@ubuntu:~#
```

---

## Phase 2: Detection & Analysis

Suricata identifies the signature match and generates an `eve.json` alert. The Wazuh Manager ingests this log, matches it against **Rule ID 100010**, and escalates it to **Level 12**.

**Suricata Signature (The Trigger)**

Plaintext  
drop icmp any any -> \$HOME\_NET any (msg:"SECURITY-POLICY ICMP Echo Request - Potential Network Discovery"; classtype:policy-violation; sid:1000002; rev:2;)

**Wazuh Custom Rule (The Logic)**

XML  
<rule id="100010" level="12">  
 <if\_sid>86601</if\_sid>  
 <match>1000002</match>  
 <description>IPS ALERT: ICMP Attack/Discovery Blocked</description>  
</rule>

full_log	<pre>{ "timestamp": "2026-01-12T16:24:40.696492+0000", "flow_id": 176664007399043, "in_iface": "eth0", "event_type": "alert", "src_ip": "192.168.113.129", "dest_ip": "192.168.113.130", "proto": "ICMP", "ip_v": 4, "icmp_type": 8, "icmp_code": 0, "pkt_src": "wire/pcap", "alert": { "action": "allowed", "gid": 1, "signature_id": 1000002, "rev": 2, "signature": "SECURITY-POLICY ICMP Echo Request - Potential Network Discovery", "category": "Potential Corporate Privacy Violation", "severity": 1 }, "direction": "to_server", "flow": { "pkts_to_server": 1, "pkts_to_client": 0, "bytes_to_server": 0, "bytes_to_client": 0, "start": "2026-01-12T16:24:40.696492+0000", "end": "2026-01-12T16:24:40.696492+0000" } }</pre>
id	1768235082.2180595
input.type	log
location	/var/log/suricata/eve.json
manager.name	wazuh
rule.description	IPS ALERT: ICMP Attack/Discovery Blocked
rule.firedtimes	1
rule.groups	suricata
rule.id	100010
rule.level	12

Figure 2: Detailed Wazuh alert showing the mapping of Suricata's 'signature\_id: 1000002' to a critical alert level.

**Phase 3: Automated Mitigation**

Upon detection, the Wazuh Dashboard shows the immediate trigger of the Active Response. A custom `suricata-drop.sh` script was developed to overcome JSON parsing differences between Suricata (`src_ip`) and Wazuh's default expectations (`srcip`).

---

## Phase 4: Enforcement Verification

Verification of the `iptables` hierarchy on the Suricata agent confirms that the attacker's IP was moved to the absolute top priority (Line 1) for an immediate drop.

```
root@suricata:~# sudo iptables -L INPUT -n --line-numbers
Chain INPUT (policy ACCEPT)
num  target      prot opt source                destination
1    DROP        0    --  192.168.113.129        0.0.0.0/0
2    NFQUEUE     1    --  0.0.0.0/0              0.0.0.0/0          NFQUEUE num 2
root@suricata:~#
```

Figure 4: System firewall state confirming the automated DROP rule for the attacker IP at the highest priority.

---

## Project Outcomes

- **Reduced MTTR (Mean Time to Respond):** The system moves from detection to mitigation in less than 2 seconds without human intervention.
- **Dynamic Firewall Management:** Utilized `iptables` insertion (`-I INPUT 1`) to ensure malicious traffic is dropped before reaching the application layer.
- **Automated Recovery:** Configured a 10-minute ban timeout to prevent permanent accidental lockouts of legitimate IPs.

## Technical Skills Demonstrated

- **SIEM/XDR:** Wazuh Manager/Agent configuration and Active Response.
- **Network Security:** Suricata IDS/IPS, NFQUEUE, and signature writing.
- **Linux Administration:** Advanced `iptables`, shell scripting (Bash/Regex), and log analysis.
- **Security Automation:** Building end-to-end SOC workflows.

