Automated Incident Response System

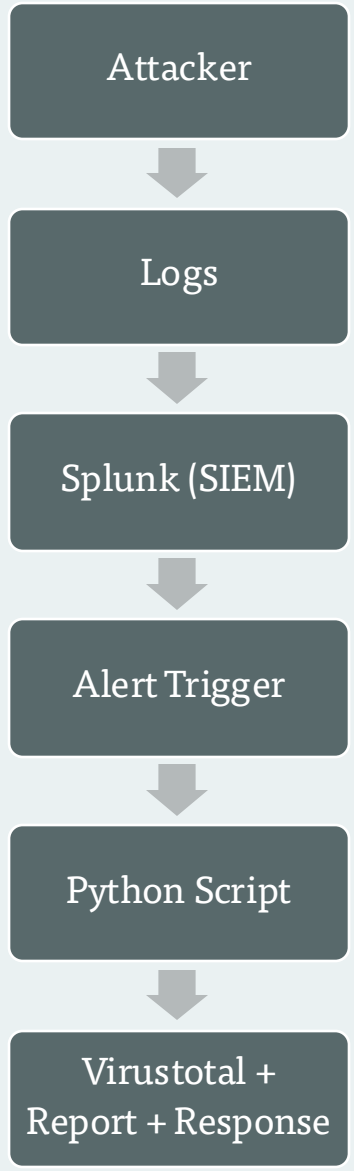
A SOC automation project using
Splunk Python and Virus total API

Sangram Rajput

Project Objective

- To automate detection, enrichment, reporting and response for security incidents in a SOC environment

Architecture Overview



Technologies Used

1. Splunk (SIEM)
2. Python
3. Virustotal API
4. CSV and Reports

Detection Logic

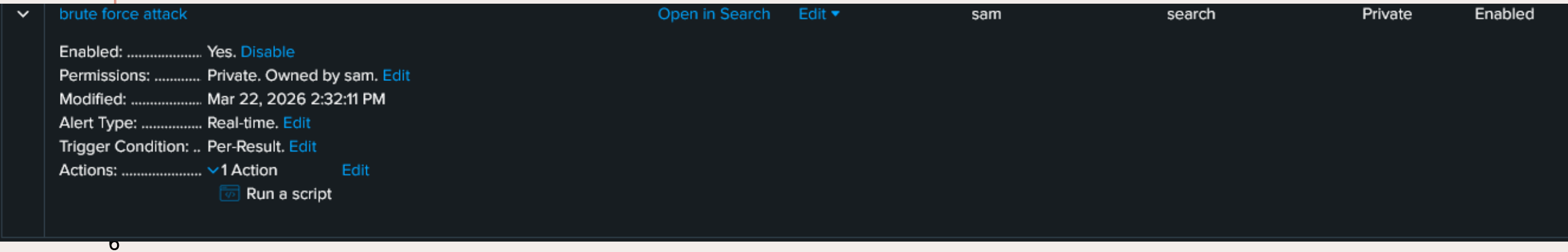
New Search

```
index="main"  EventCode=4625
| stats count by src_ip, user
| where count > 5
```

- Detects multiple failed login attempts (Event ID 4625)
- Groups by source IP and counts attempts
- Flags IPs with more than 5 failures as brute force attack

Automated Alert Configuration (Splunk)

- Real-time alert triggers on detection
- Per-result alert ensures each event is processed
- Executes Python script automatically
- Enables automated incident response



python automation script (incident response engine)

- The Python script is triggered automatically by a Splunk alert when suspicious activity is detected.
- It extracts the source IP address passed from Splunk as an argument.
- The script queries the VirusTotal API to check the reputation of the IP address.
- Based on the response, it classifies the threat as **Malicious, Suspicious, or Safe**.
- It generates a detailed incident report in both **text and CSV formats**.
- If the IP is marked as malicious, it simulates a response action by blocking the IP.
- This script automates the investigation and response process, reducing manual effort.

Key Capabilities:

- API Integration
- Automation
- Threat Enrichment
- Incident Reporting
- Response Simulation

Python Automation Logic

1. Input from Splunk

```python
ip = sys.argv[1]
```

This receives the IP from Splunk alert

2. VirusTotal API Call

```python
url = f"https://www.virustotal.com/api/v3/ip_addresses/{ip}"
response = requests.get(url, headers=headers)
```

This checks IP reputation using VirusTotal API

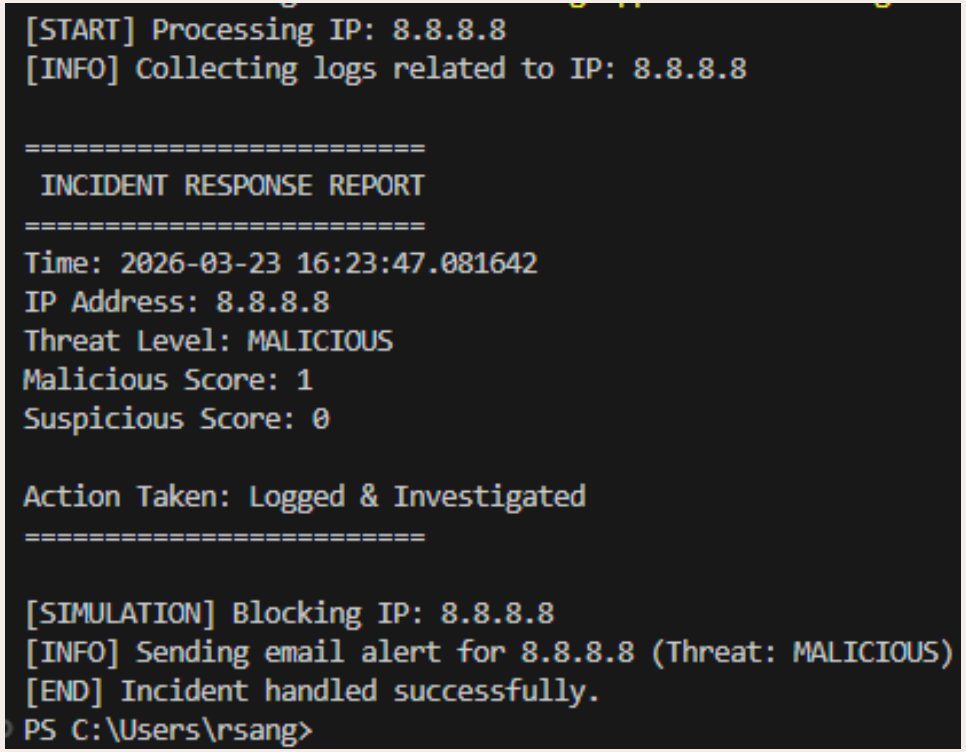
3. Decision + Response Logic

```python
if malicious > 0:
    block_ip(ip)
```

If IP is malicious, response action is triggered

Python Script Execution (Automation In Action)

- This screenshot shows the execution of the Python automation script triggered by a Splunk alert.
- The script processes the suspicious IP received from Splunk.
- It performs threat enrichment by querying the VirusTotal API.
- Based on the analysis, it classifies the IP and generates an incident report.
- If the IP is identified as malicious, it simulates a response action by blocking the IP.
- This demonstrates a complete automated incident response workflow.



Incident Report Output (TXT & CSV)

- The system automatically generates incident reports in both TXT and CSV formats.
- The TXT report provides a detailed, human-readable summary of the incident, including IP address, timestamp, threat level, and analysis results.
- The CSV report stores structured data, enabling easy analysis, filtering, and integration with dashboards.
- Threat classification is based on VirusTotal enrichment results.
- This dual-format reporting ensures both readability and scalability for SOC operations.

```
==========================
 INCIDENT RESPONSE REPORT
==========================
Time: 2026-03-22 14:39:54.931163
IP Address:103.78.3.2
Threat Level: MALICIOUS
Malicious Score: 1
Suspicious Score: 0

Action Taken: Logged & Investigated
==========================
```

TXT

	A	B	C	D
1	Time	IP	Malicious	Suspicious
2	39:54.9	103.78.3.2	1	0

CSV

Use Case & Impact

- Automates detection and response to brute force attacks in a SOC environment
- Reduces manual investigation effort by integrating SIEM with automated scripts
- Enables faster threat identification using real-time enrichment (VirusTotal)
- Improves incident response time and operational efficiency
- Helps SOC analysts focus on critical threats instead of repetitive tasks

Future Enhancements

- Integration with email/SMS alerts for real-time notification
- Implementation of actual IP blocking using firewall APIs
- Integration with Splunk REST API for advanced log collection
- Development of a dashboard for visualizing incidents and trends
- Integration with SOAR platforms for fully automated response workflows

Conclusion

- Successfully designed and implemented an automated incident response system
- Demonstrated integration of SIEM (Splunk) with Python-based automation
- Achieved real-time detection, enrichment, reporting, and response simulation
- Showcased practical SOC skills including threat detection, automation, and analysis
- This project reflects real-world SOC workflows and readiness for security operations roles

This project can be extended into a full SOAR-based security automation system

Thank you

Sangram Rajput

Sangramrajput1436@gmail.com

https://sangramrajput.netlify.app/