

Detection of Brute Force and Port Scanning Attacks Using Splunk SIEM

Prepared By: Rabbi Wasti

Table of Contents

Abstract	3
1. Problem Statement.....	3
2. Objectives	3
3. Architecture &Environment	4
4. Tools & Technologies	4
5.Attack Simulation.....	4
6.Brute Force Detection.....	6
7.Port Scan Detection	9
8.Result.....	11
9.Security Analysis	11
10.Security Impact	11
11.Future Enhacenents	12
Conclusion	12

Abstract

This project illustrates how the Security Operations Center (SOC) simulation could be implemented with the help of Splunk SIEM. The actual cyberattacks like SSH brute-force and network port scanning were simulated with the help of Kali Linux. The detection and analysis of these attacks were performed on centralized log monitoring in Splunk, with alerts being set up to mimic enterprise SOC detection workflows.

1. Problem

Unauthorized access and reconnaissance attacks are constant threats to organizations. These threats are not detected without a SIEM system. This project will attempt to model these attacks and illustrate the detection and notification of security teams by SIEM tools.

2. Objectives

- Simulate brute-force and port scan attacks
- Centralize logs using Splunk Forwarder
- Analyze authentication and system logs
- Identify attacker IP addresses
- Create automated SOC alerts

3. Architecture & Environment

Component	Role
Kali Linux	Attacker
Fedora Linux	Victim
Splunk Enterprise	SIEM
Splunk Forwarder	Log Collector

4. Tools & Technologies

- Splunk Enterprise
- Splunk Universal Forwarder
- Hydra
- Nmap
- Kali Linux
- Fedora Linux

5. Attack Simulation

6.1 SSH Brute Force Attack

A password-guessing attack attempting unauthorized system access.

Tool Used: Hydra

Command executed on Kali:

hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.48

```
(root@kali)-[/home/kali/Desktop]
# hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.48

Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-02-11 06:15:38
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.48:22/
[STATUS] 183.00 tries/min, 183 tries in 00:01h, 14344218 to do in 1306:24h, 14 active
[STATUS] 177.67 tries/min, 533 tries in 00:03h, 14343868 to do in 1345:35h, 14 active
[STATUS] 177.14 tries/min, 1240 tries in 00:07h, 14343161 to do in 1349:30h, 14 active
[STATUS] 159.13 tries/min, 2387 tries in 00:15h, 14342014 to do in 1502:06h, 14 active
```

Figure 1:Kali terminal showing Hydra brute force attack.

(Source:-Self-creation)

6.2 Port Scanning Attack

Reconnaissance activity to identify open services.

Tool Used: Nmap

Command:

nmap -sS -p- 192.168.1.48

```
(root@kali)-[/home/kali/Desktop]
# nmap -sS -p- 192.168.1.48

Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-11 07:10 EST
Nmap scan report for 192.168.1.48
Host is up (0.018s latency).
Not shown: 65373 filtered tcp ports (no-response), 158 filtered tcp ports (admin-prohibited)
PORT      STATE SERVICE
22/tcp    open  ssh
26/tcp    open  rsftp
53/tcp    closed domain
80/tcp    open  http
MAC Address: 08:00:27:0A:2D:45 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 152.73 seconds
```

Figure 2:Kali terminal showing Nmap open ports

(Source:-Self-creation)

6.3 Log Collection & SIEM Integration

Fedora system logs (/var/log/secure, /var/log/messages) were forwarded to Splunk using Universal Forwarder.

Search:

index=* host=fedora

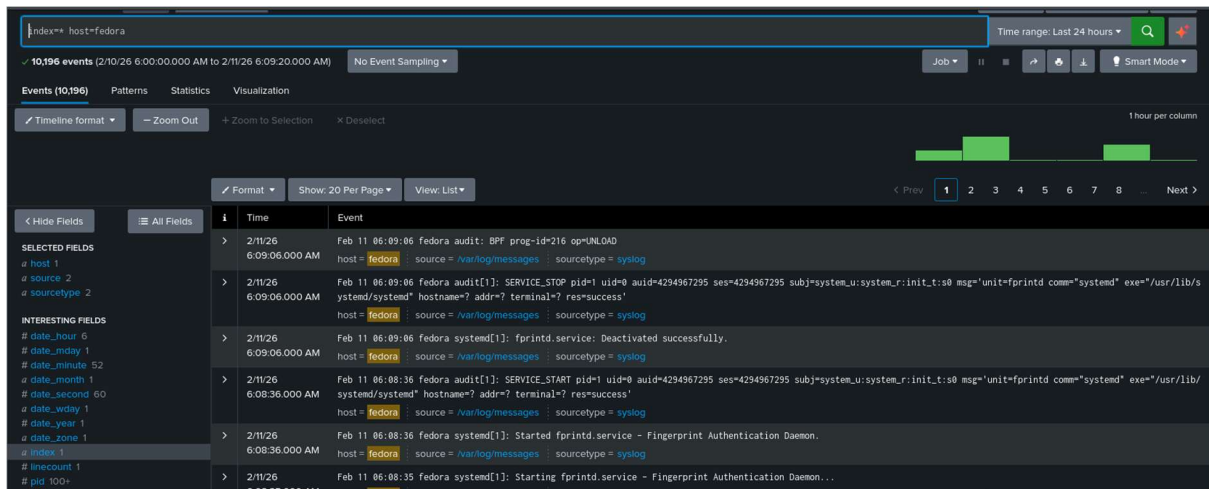


Figure 3: Splunk showing Fedora system logs.

(Source:-Self-creation)

This confirmed successful log ingestion.

6.Brute Force Detection

6.1 Raw Log Identification

"Failed password"

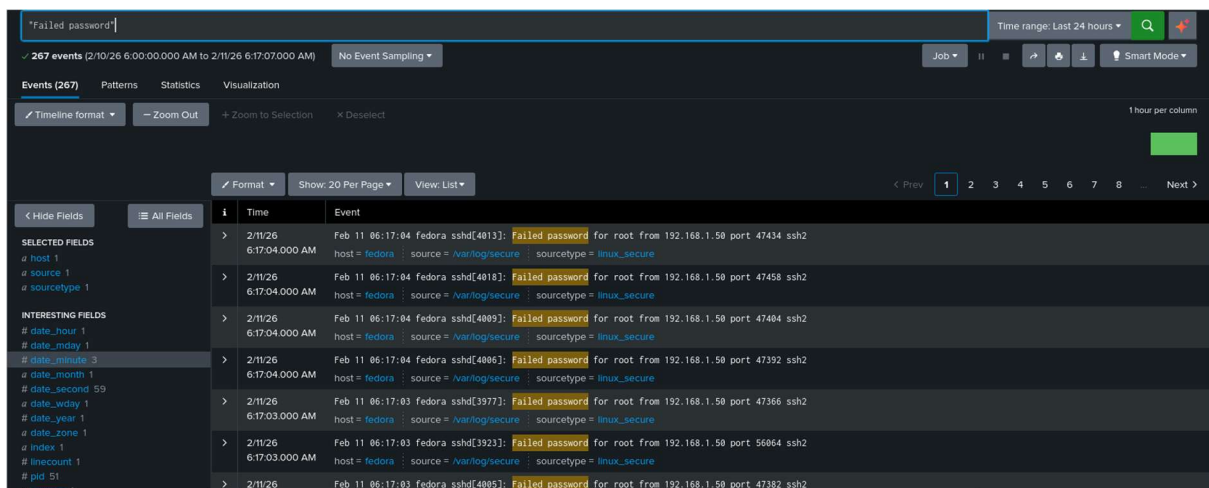


Figure 4: Multiple failed SSH login events.

(Source:-Self-creation)

This filters SSH authentication failures.

6.2 Attacker IP Extraction

"Failed password"

| rex "from (?<attacker_ip>\d+\.\d+\.\d+\.\d+)"

| stats count by attacker_ip

| sort -count

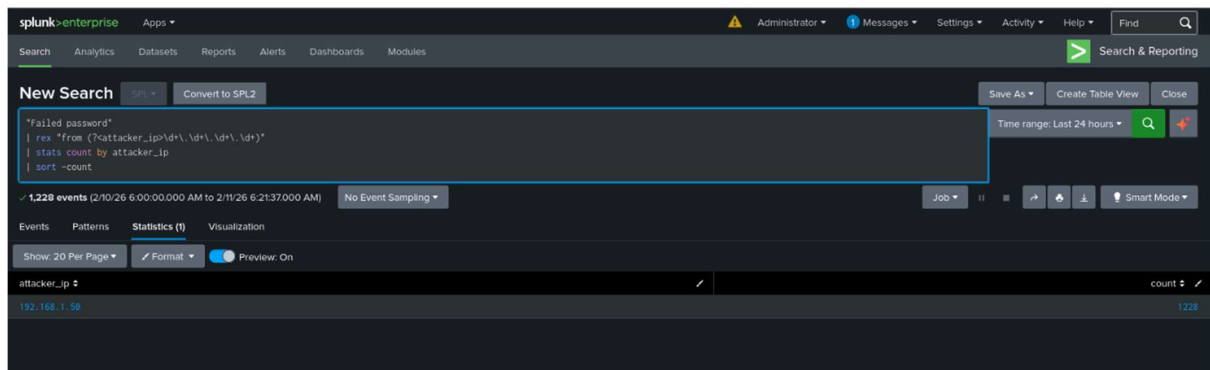


Figure 5: Table showing attacker IP and count

(Source: -Self-created)

This query:

- Filters failed login attempts
- Extracts the attacker IP
- Counts login attempts
- Flags excessive activity

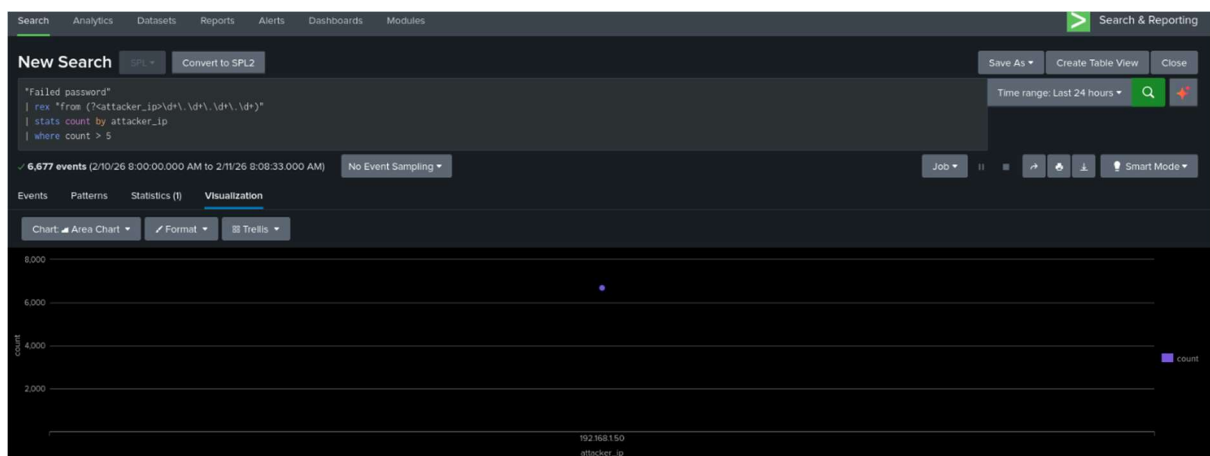


Figure 6: Visualization showing attacker_ip vs count

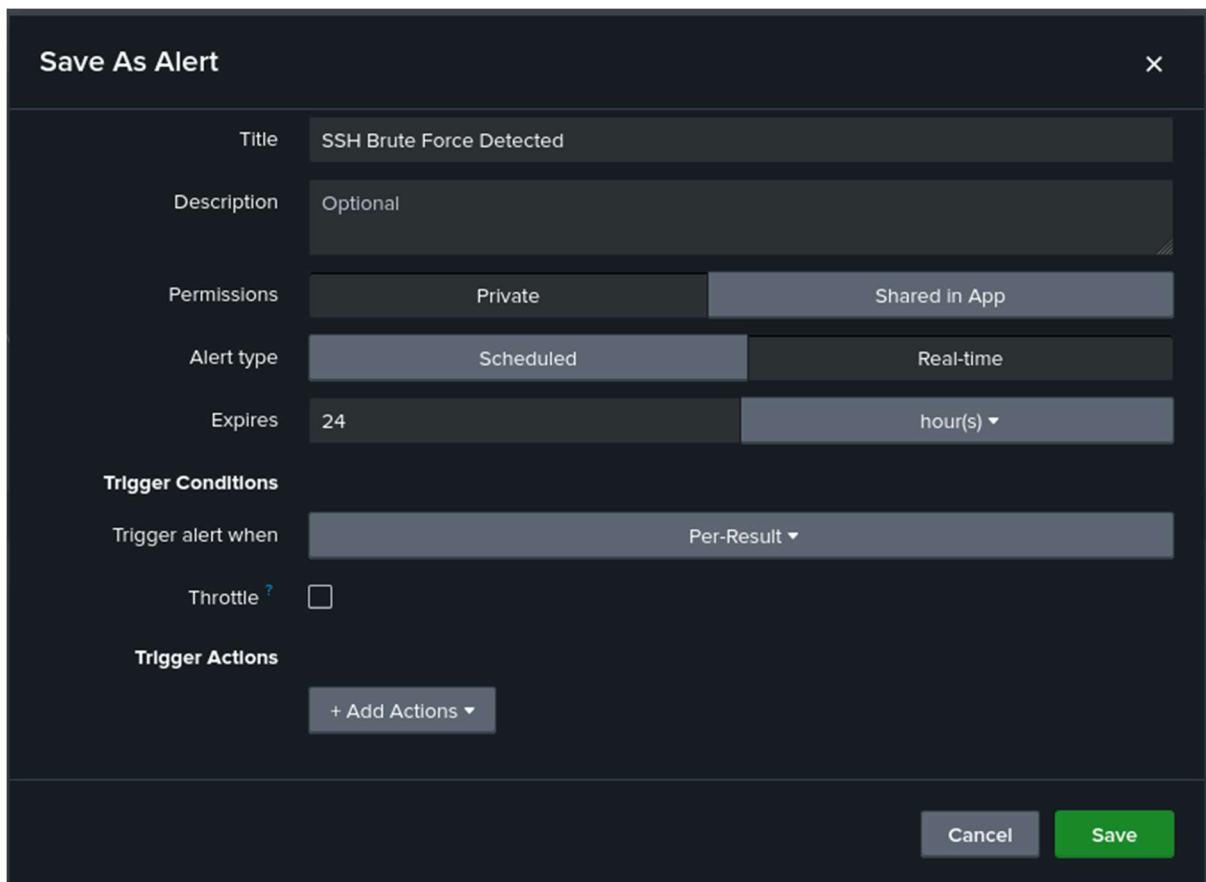
(Source:Self-created)

The visualization clearly shows a spike in login attempts from IP 192.168.1.50, indicating brute-force behavior.

6.3. Brute Force Alert

Alert Name: **SSH Brute Force Detected**

Trigger: Per result

A screenshot of a 'Save As Alert' configuration window. The window has a dark theme and a close button (X) in the top right corner. It contains several sections: 'Title' with the text 'SSH Brute Force Detected'; 'Description' with the text 'Optional'; 'Permissions' with two radio buttons, 'Private' (selected) and 'Shared in App'; 'Alert type' with two radio buttons, 'Scheduled' (selected) and 'Real-time'; 'Expires' with a text input '24' and a dropdown menu 'hour(s)'; 'Trigger Conditions' with a dropdown menu 'Per-Result'; 'Throttle' with a checkbox and a question mark icon; and 'Trigger Actions' with a button '+ Add Actions'. At the bottom right, there are 'Cancel' and 'Save' buttons.

Save As Alert ×

Title: SSH Brute Force Detected

Description: Optional

Permissions: Private (selected) Shared in App

Alert type: Scheduled (selected) Real-time

Expires: 24 hour(s) ▼

Trigger Conditions

Trigger alert when: Per-Result ▼

Throttle ? ☐

Trigger Actions

+ Add Actions ▼

Cancel Save

Figure 7:Alert creation page

(Source:Self-created)

7.Port Scan Detection

Search:

index=* host=fedora (ssh OR failed OR refused OR nmap OR scan OR denied OR rejected)

| rex "from (?<attacker_ip>\d+\.\d+\.\d+\.\d+)"

| stats count by attacker_ip

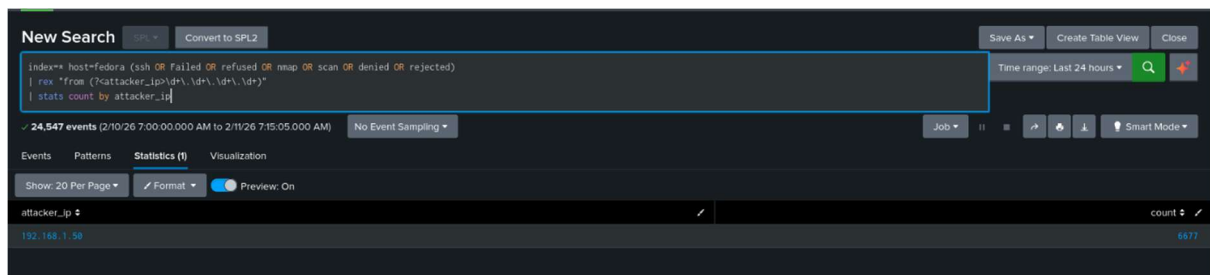


Figure 8:Attacker IP detected with high event count.

(Source:Self-created)

This query:

- Correlates suspicious connection attempts
- Extracts attacker IP
- Aggregates event counts

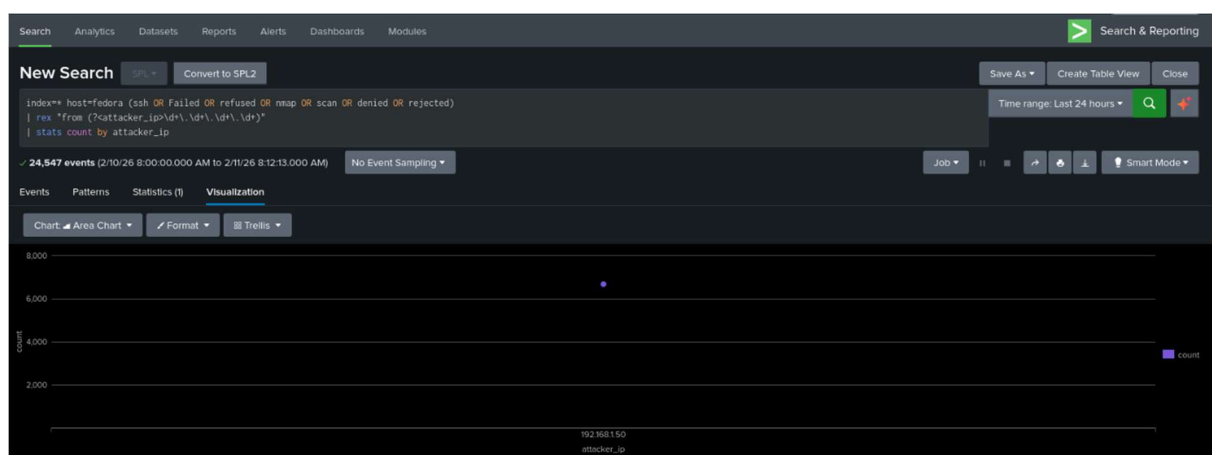


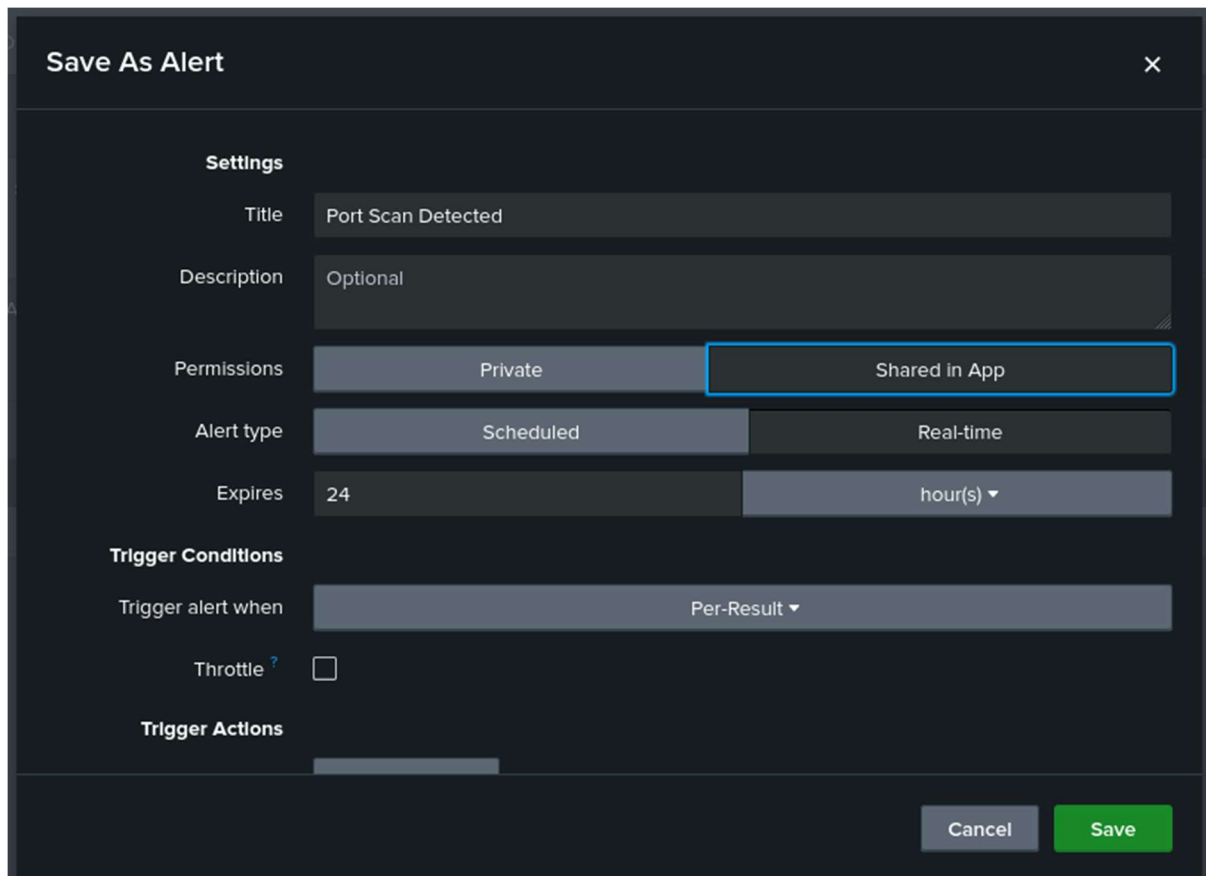
Figure 9:Visualization graph of port scan activity

(Source:Self-created)

The graph highlights a significant volume of suspicious activity originating from the attacker IP, confirming reconnaissance behavior.

7.1 Port Scan Alert

Alert Name: **Port Scan Detected**



The screenshot shows a 'Save As Alert' dialog box with a dark theme. It contains the following fields and options:

- Title:** Port Scan Detected
- Description:** Optional
- Permissions:** Private (selected), Shared in App (highlighted with a blue border)
- Alert type:** Scheduled (selected), Real-time
- Expires:** 24 hour(s) (dropdown menu)
- Trigger Conditions:**
 - Trigger alert when: Per-Result (dropdown menu)
 - Throttle: ☐ (checkbox)
- Trigger Actions:** (empty field)

At the bottom right, there are 'Cancel' and 'Save' buttons.

Figure 10:Port Scan alert creation screen

(Source:Self-created)

8.Result

Attack	Attacker IP	Events
Brute Force	192.168.1.50	1228+
Port Scan	192.168.1.50	6677+

9.Security Analysis

The following threat patterns were successfully detected:

Attack	MITRE Technique	Result
SSH Brute Force	T1110	Identified attacker IP
Port Scan	T1046	Detected reconnaissance

10.Security Impact

- Early detection of unauthorized access
- Identification of threat actor IP
- Automated alerting for SOC teams

11.Future Enhancements

- Geo-IP correlation
- Threat intelligence integration
- SOAR automation
- Dashboards

Conclusion

This SOC simulation successfully demonstrated how SIEM tools like Splunk can detect and alert on real-world cyber threats.

The system effectively identified attacker behavior and generated security alerts, simulating an enterprise SOC workflow.