# SSH Log Analysis using Splunk | This is for SOC Analyst

Instructor: Rajneesh Gupta
Source: https://haxcamp.com/projects/353bc3c9-c9e9-495e-8472-cc80fa5539f6/resources

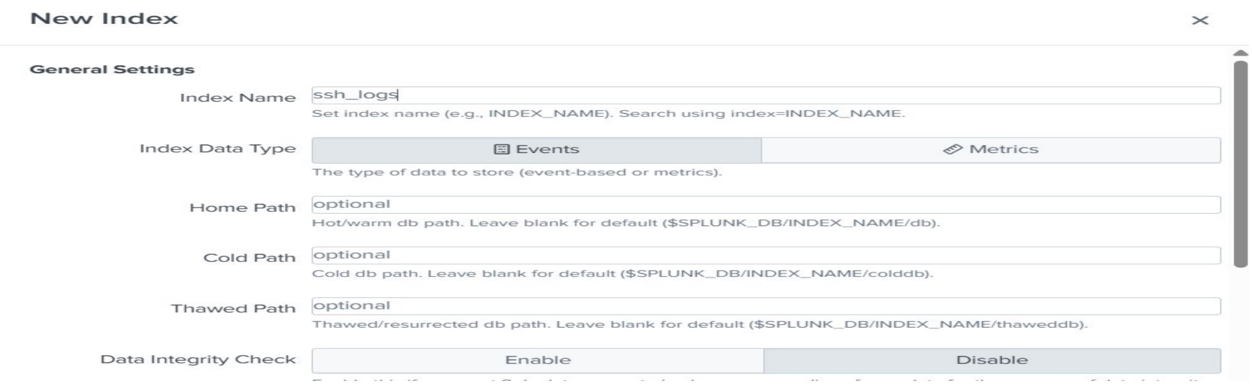
- **Successful logins** (who connected, from where)
- **Failed login attempts** (possible brute-force or password spraying)
- **Multiple failed authentication attempts** (indicators of brute-force)
- **Connections without authentication** (potential scanning or incomplete sessions)

## Lab Setup and Pre-requisite

- Complete the Splunk Installation Project
- Download the SSH Log File

## Preparation

1. Log in to your Splunk instance (Enterprise or Free).
2. Go to Apps > Search & Reporting.
3. Click Add Data → Upload.
4. Select the provided ssh_log.json file and upload it.
5. Click Source_type:_json > Name: ssh_logs, Description: sshlogs, Category: Customs, App Search & Reporting
6. Choose sourcetype = _json so Splunk automatically extracts fields.
7. Index it under a new index, e.g., ssh_logs.


New Index

General Settings

Index Name: ssh_logs
Set index name (e.g., INDEX_NAME). Search using index=INDEX_NAME.

Index Data Type: Events | Metrics
The type of data to store (event-based or metrics).

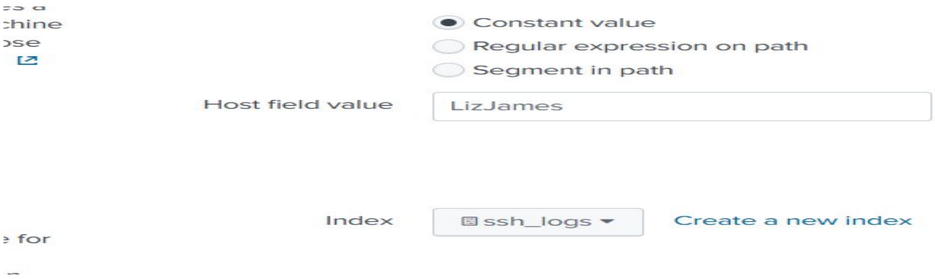
Home Path: optional
Hot/warm db path. Leave blank for default ($SPLUNK_DB/INDEX_NAME/db).

Cold Path: optional
Cold db path. Leave blank for default ($SPLUNK_DB/INDEX_NAME/colddb).

Thawed Path: optional
Thawed/resurrected db path. Leave blank for default ($SPLUNK_DB/INDEX_NAME/thaweddb).

Data Integrity Check: Enable | Disable

8. Review and click on start searching


● Constant value
○ Regular expression on path
○ Segment in path

Host field value: LizJames

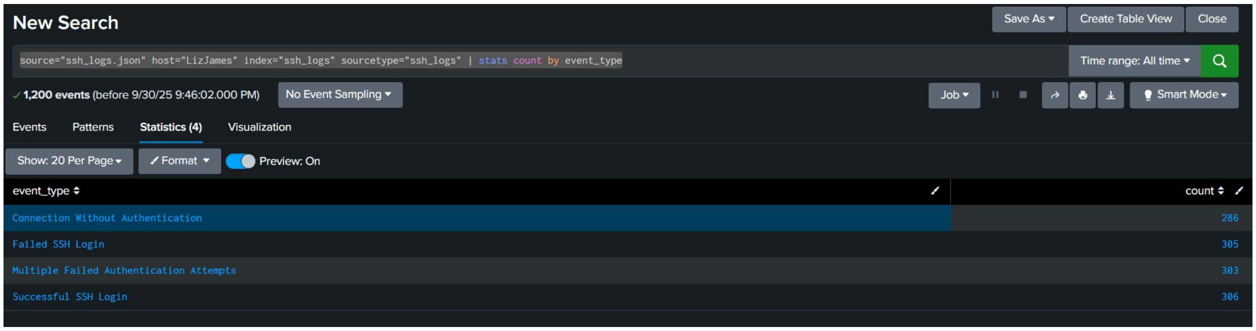
Index: ssh_logs

## Step-by-Step Guide

### Task 1: Ingest and Parse Logs

1. Upload ssh_log.json into Splunk.
2. Ensure the following fields are extracted correctly:
    - event_type (Successful SSH Login, Failed SSH Login, Multiple Failed Authentication Attempts, Connection Without Authentication)
    - auth_success (true/false/null)
    - auth_attempts
    - id.orig_h (source IP)
    - id.resp_h (destination host)
3. Run a validation search:
4. index=ssh_log | stats count by event_type > In new search provide the query below.

**source="ssh_logs.json" host="LizJames" index="ssh_logs" sourcetype="ssh_logs" | stats count by event_type**



```
source="ssh_logs.json" host="LizJames" index="ssh_logs" sourcetype="ssh_logs" | stats count by event_type
```

✓ 1,200 events (before 9/30/25 9:46:02.000 PM)

| event_type | count |
|---|---|
| Connection Without Authentication | 286 |
| Failed SSH Login | 305 |
| Multiple Failed Authentication Attempts | 303 |
| Successful SSH Login | 306 |

### Task 2: Analyze Failed Login Attempts

1. Identify all failed login attempts: > In new search provide the query below.

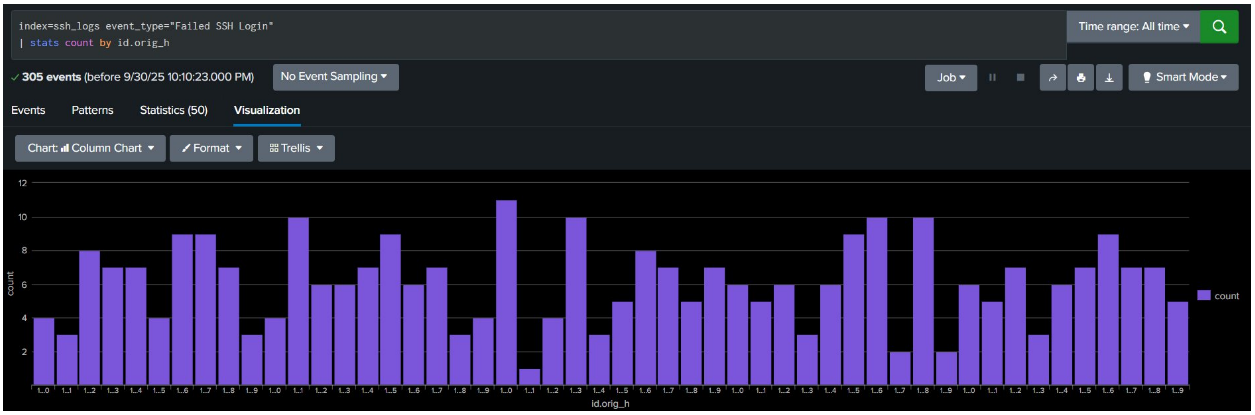
**index=ssh_logs event_type="Failed SSH Login"<br>| stats count by id.orig_h**

Or

**source="ssh_logs.json" host="LizJames" index="ssh_logs" sourcetype="ssh_logs" | stats count by id.orig_h**

2. Highlight the **top 10** source IPs generating failed logins.
3. Create a bar chart visualization for failed login attempts per source IP.

**Sample output by Leo:**



```
index=ssh_logs event_type="Failed SSH Login"
| stats count by id.orig_h
```

✓ **305 events** (before 9/30/25 10:10:23.000 PM)

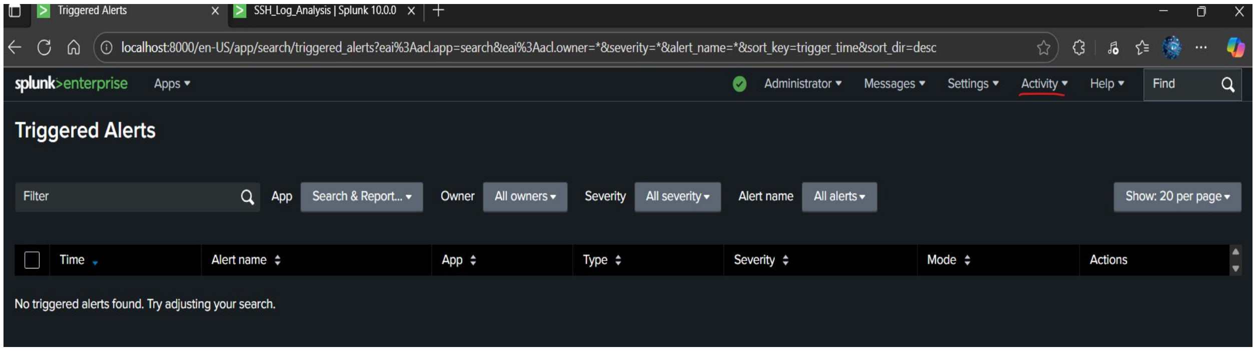
Time range: All time

## Task 3: Detect Multiple Failed Authentication Attempts (Brute Force)

1. Search for multiple failed attempts in logs:

```
index=ssh_logs event_type="Multiple Failed Authentication Attempts"
| stats count by id.orig_h, id.resp_h
```

2. Note if you can't see the result > **Click Time Range: All Time** > Click Detect repeated failures (e.g., more than 5 attempts).
3. **Configure a Splunk alert**: > Click Save As > Alert > Provide the Title > Click Real Time > Click Add Action > Click Add to Triggered Alerts > Click Save
4. If there is a scenario or activity happen this will trigger and alert and you can see this in > Activity > Triggered Alerts



localhost:8000/en-US/app/search/triggered_alerts?eai%3Aacl.app=search&eai%3Aacl.owner=*&severity=*&alert_name=*&sort_key=trigger_time&sort_dir=desc

Triggered Alerts

App: Search & Report... ▾ Owner: All owners ▾ Severity: All severity ▾ Alert name: All alerts ▾ Show: 20 per page ▾

| □ | Time | Alert name | App | Type | Severity | Mode | Actions |
|---|---|---|---|---|---|---|---|

No triggered alerts found. Try adjusting your search.

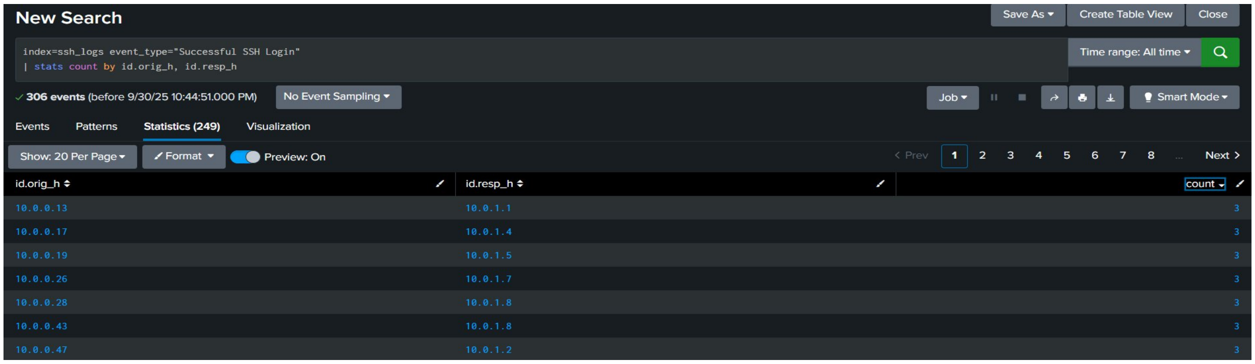
Trigger when any IP attempts more than 5 logins within 10 minutes.

## Task 4: Track Successful Logins

1. Search for successful logins:

**index=ssh_logs event_type="Successful SSH Login"**
**| stats count by id.orig_h, id.resp_h**

2. Note if you can't see the result > **Click Time Range: All Time** > Compare successful logins against prior failed attempts (to detect compromised accounts).
3. Create a dashboard panel showing top source IPs for successful logins.



```
index=ssh_logs event_type="Successful SSH Login"
| stats count by id.orig_h, id.resp_h
```

✓ 306 events (before 9/30/25 10:44:51.000 PM)

| id.orig_h | id.resp_h | count |
|---|---|---|
| 10.0.0.13 | 10.0.1.1 | 3 |
| 10.0.0.17 | 10.0.1.4 | 3 |
| 10.0.0.19 | 10.0.1.5 | 3 |
| 10.0.0.26 | 10.0.1.7 | 3 |
| 10.0.0.28 | 10.0.1.8 | 3 |
| 10.0.0.43 | 10.0.1.8 | 3 |
| 10.0.0.47 | 10.0.1.2 | 3 |

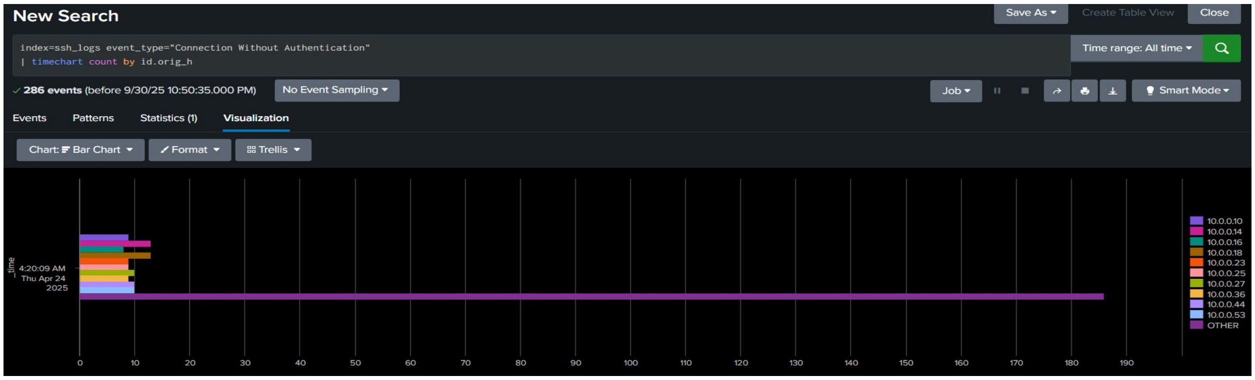
## Task 5: Spot Suspicious Connections Without Authentication

1. Search for unauthenticated SSH connections: Gain an access without authentication

**index=ssh_logs event_type="Connection Without Authentication"**
**| stats count by id.orig_h**

2. Create a **timechart** visualization to monitor such events over time:

**index=ssh_logs event_type="Connection Without Authentication"**
**| timechart count by id.orig_h**



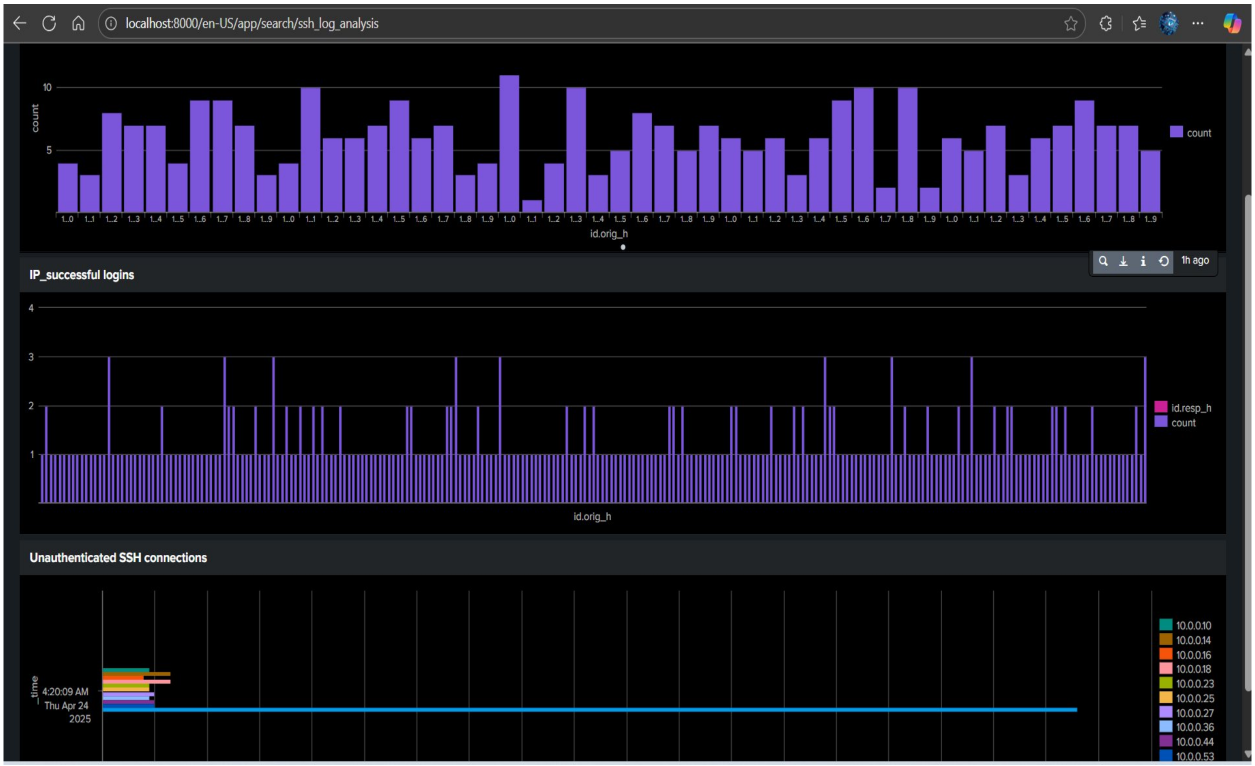
```
index=ssh_logs event_type="Connection Without Authentication"
| timechart count by id.orig_h
```

✓ 286 events (before 9/30/25 10:50:35.000 PM)

3. Identify repeated unauthenticated attempts — potential indicators of port scanning or SSH probing.

**Conclusion:**

**Leo final dashboard for SSH Log analysis:**



By the end of this project, you will have:

- Built dashboards to monitor SSH activity.
- Identified brute-force login attempts and suspicious access attempts.
- Configured Splunk alerts for high-risk behavior.
- Learned how to parse, search, visualize, and alert on SSH logs in Splunk.

This project provides practical SOC Analyst, level log analysis skills and strengthens your cybersecurity portfolio.