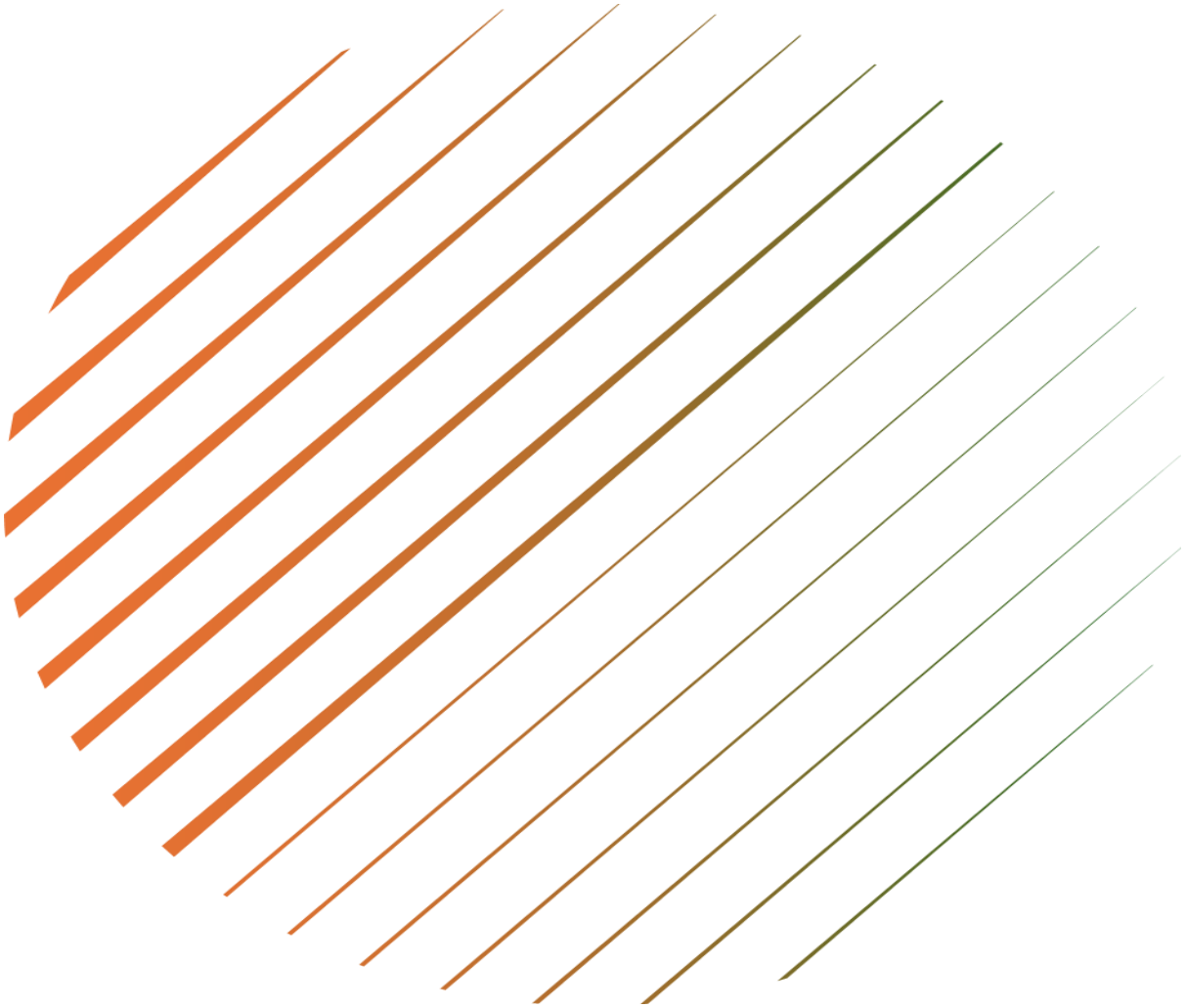
PROJECT REPORT: Log Analysis and Threat Classification using Splunk



Name: Pranit Kalambate

Date: November 18, 2025

Subject: Security Information and Event Management (SIEM) Implementation

TABLE OFN CONTAINT

1.OBJECTIVE

The objective of this project was to configure a Splunk Enterprise environment to ingest Linux authentication logs and implement a classification system for SSH brute-force attacks. The goal was to distinguish between high-severity threats (targeted attacks on valid accounts) and low-severity noise (automated bot scanning).

Environment & Tools

SIEM Platform: Splunk Enterprise (v10.0.1) installed on Windows.

Log Source: Kali Linux machine.

Data Transport: Splunk Universal Forwarder.

Target Log File: `/var/log/auth.log` (Linux Authentication Logs).

2.METHODOLOGY

2.1. Data Ingestion and Initial Analysis

The project began by forwarding the `/var/log/auth.log` file from the Linux machine to the Splunk indexer. An initial search was performed to identify all failed SSH login attempts using the keyword "Failed password".

Search Query:

`index=* "Failed password"`

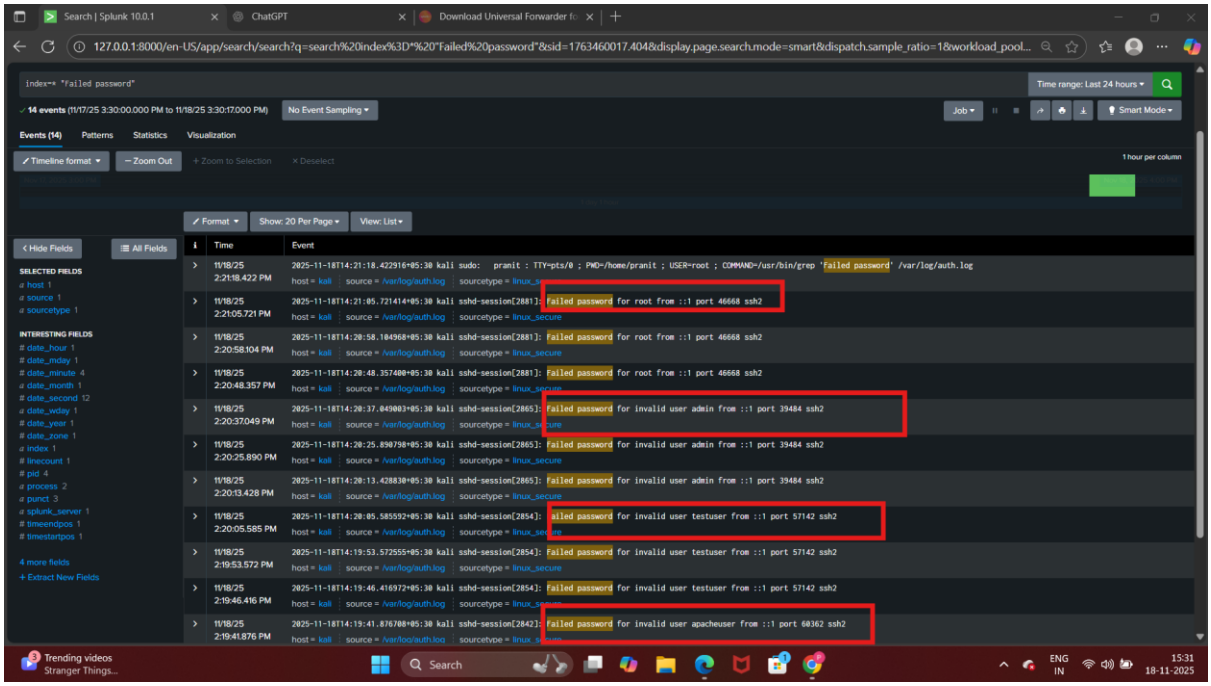


Figure 1: Raw search results showing failed login attempts.

2.2. Developing Threat Logic

Upon analyzing the raw logs, two distinct patterns emerged:

1)Pattern A: Attacks against specific, valid users (e.g., `root`). This indicates a targeted attempt to breach a known account.

2) Pattern B: Attacks against non-existent users (e.g., `admin`, `testuser`, `apacheuser`). This indicates generic "spray-and-pray" scanning.

2.3. Configuring "High Risk" Event Types

To prioritize targeted attacks, a specific search filter was applied to exclude invalid users. This isolates attempts on valid accounts (like root).

Search Query:

`index=* "Failed password"* NOT invalid`

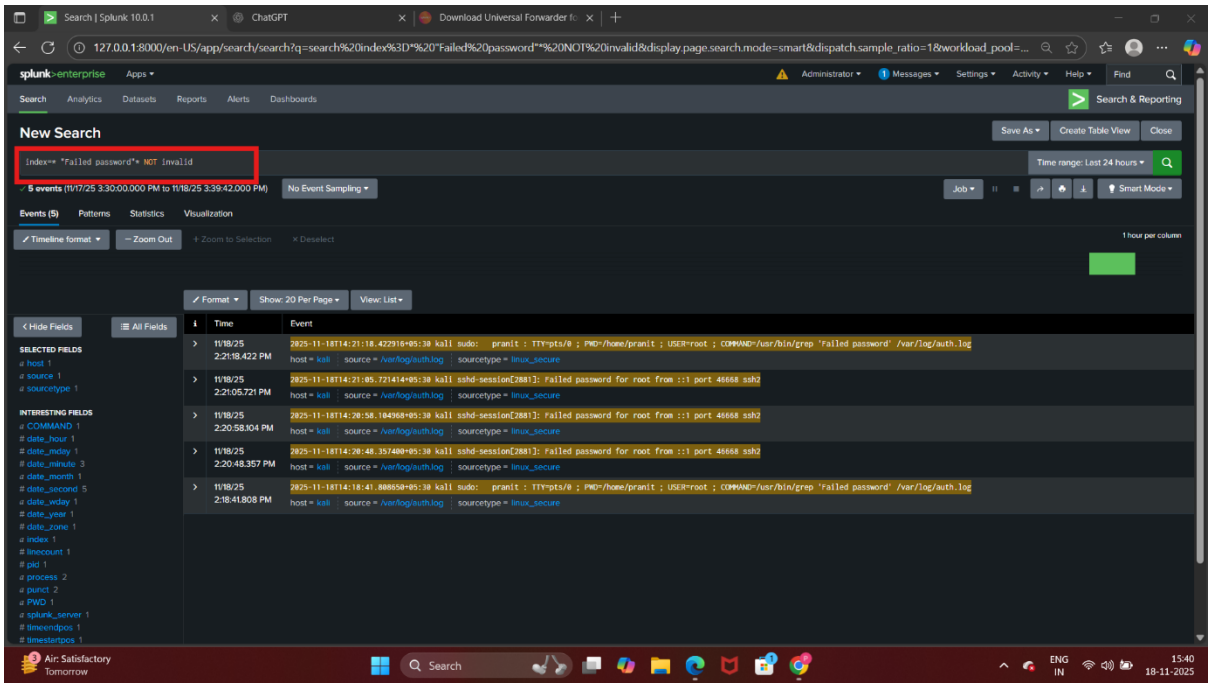


Figure 2: Filtering logs to show only potential targeted attacks.

This search logic was saved as a new Event Type named "highrisky".

Priority: 3 (High)

Color: Red

Rationale: High priority is assigned because the attacker knows the username is valid, significantly increasing the risk of a breach.

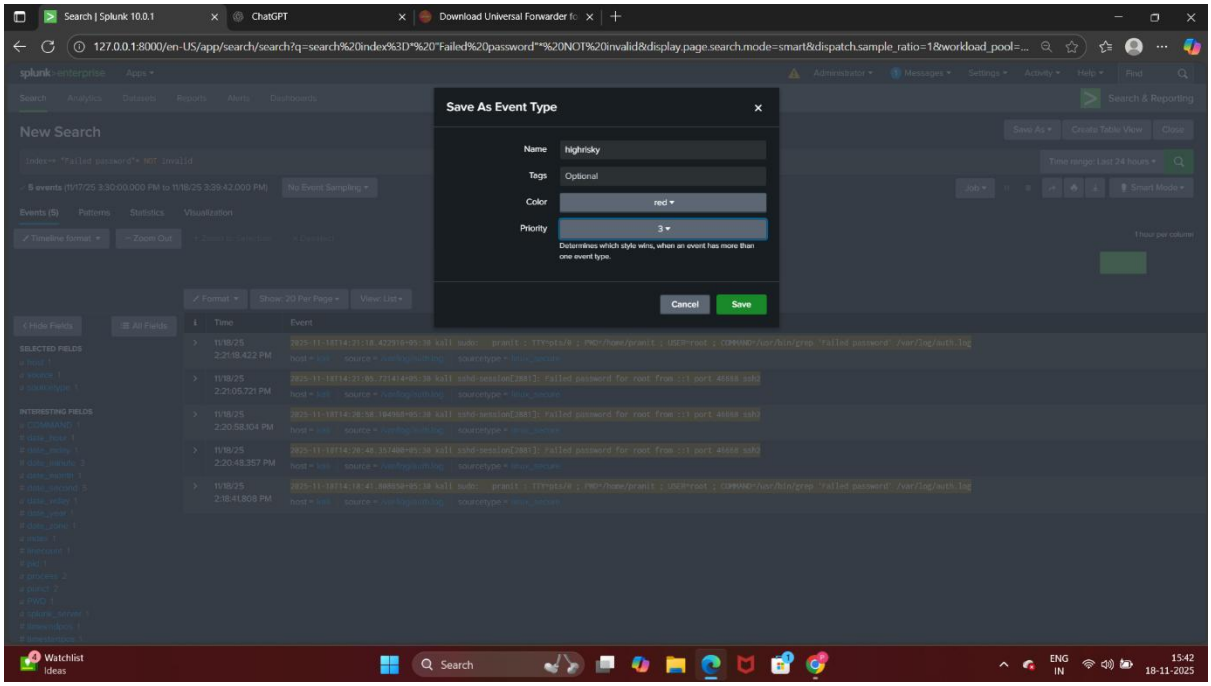


Figure 3: Configuring the High Risk Event Type.

2.4. Configuring "Less Risky" Event Types

Next, a filter was created to identify the noise generated by bots guessing random usernames.

Search Query:

`index=* "Failed password"* * for invalid user *`

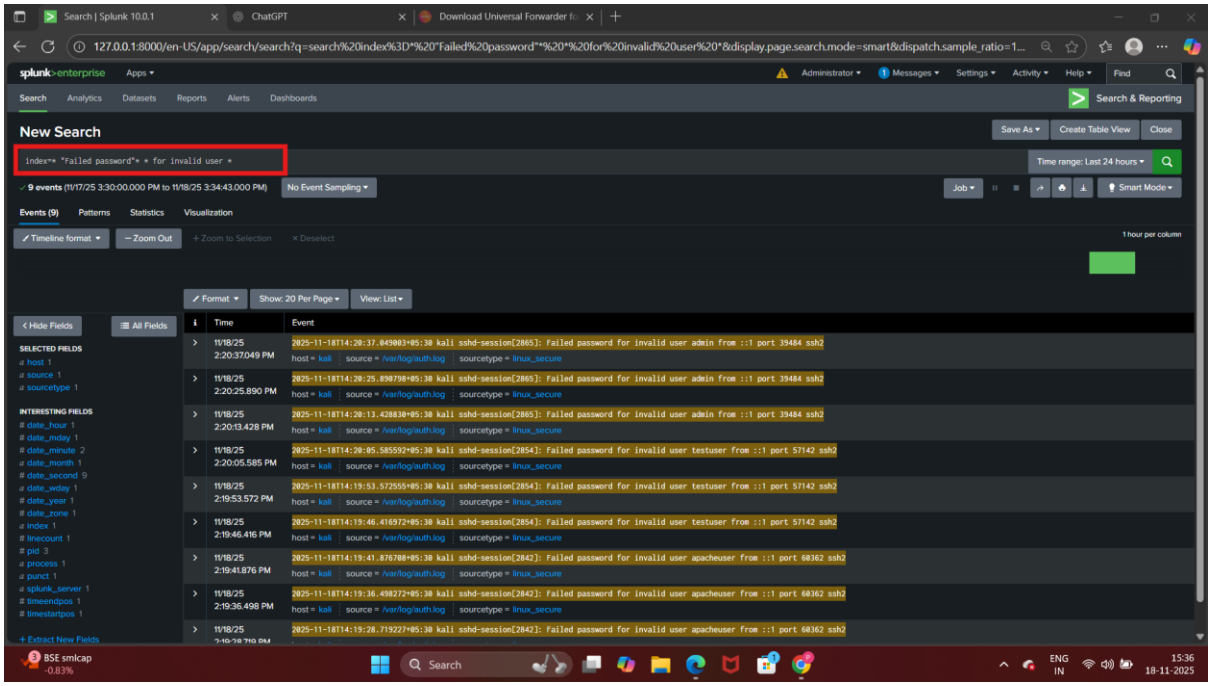


Figure 4: Filtering logs to show generic bot activity.

This search logic was saved as an Event Type named "lessrisky".

Priority: 8 (Low)

Color: Blue

Rationale: While these events are malicious, they are usually automated and less likely to succeed compared to targeted attacks.

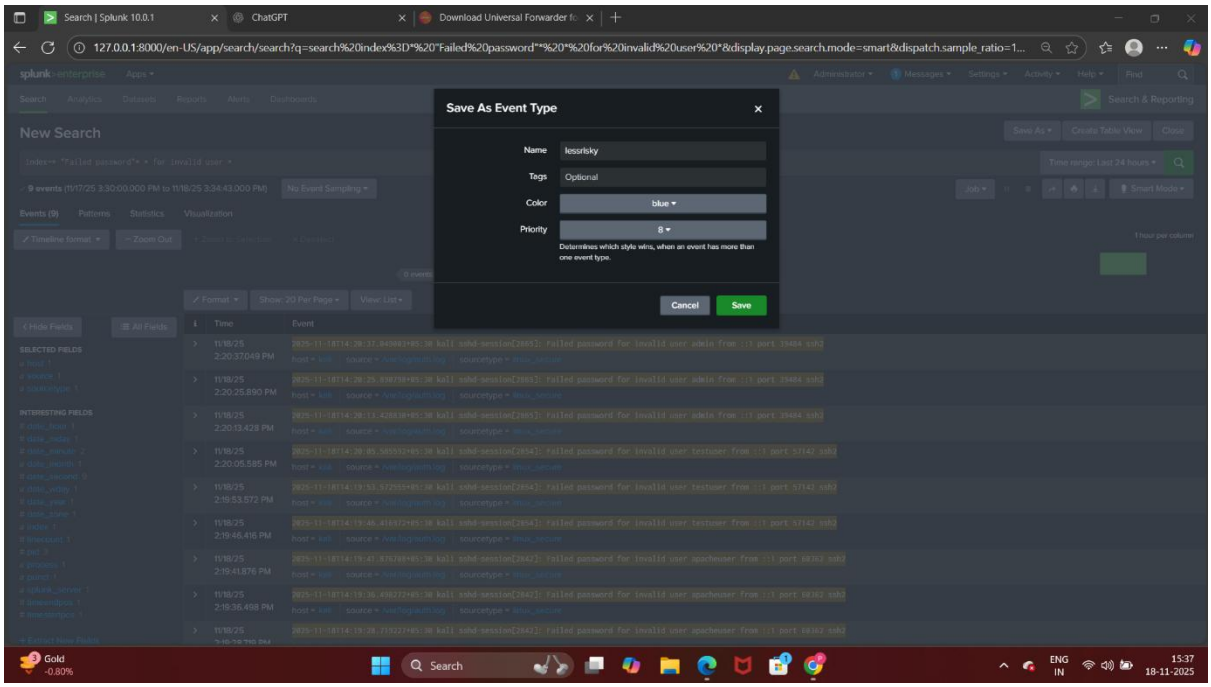


Figure 5: Configuring the Less Risky Event Type.

3.RESULTS AND VISUALIZATION

3.1. Enhanced Log Visibility

After applying the Event Types, the Splunk search interface now automatically tags and color-codes the incoming logs. This allows a security analyst to immediately spot critical alerts (Red) amidst the background noise (Blue).

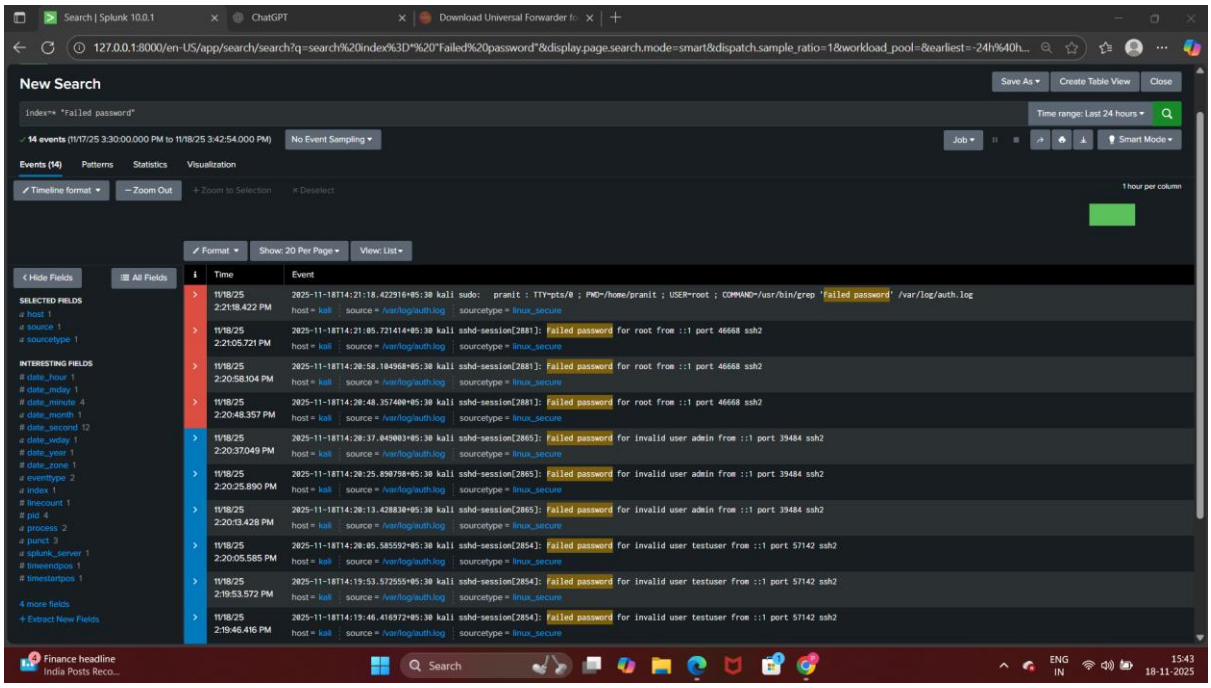


Figure 6: The final output showing color-coded events based on risk priority.

3.2. Statistical Analysis

A statistical chart was generated to compare the volume of high-risk events versus low-risk events. This visualization helps in understanding the nature of traffic hitting the server.

SPL for Visualization:

`index=* "Failed password" | stats count by eventtype`

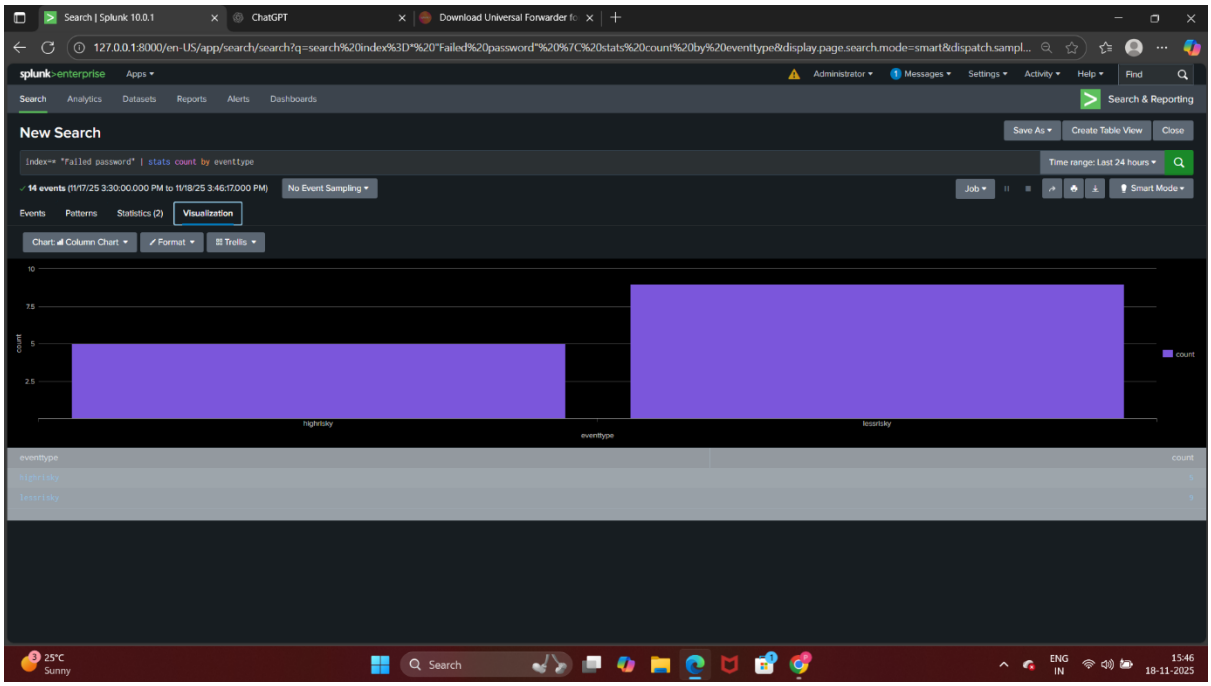


Figure 7: Bar chart comparing the count of High Risk vs. Less Risky events.

4.CONCLUSION

In this project, I successfully transformed raw text logs into actionable intelligence. By using Splunk's Event Types, I created a system that automatically prioritizes security incidents. This implementation demonstrates the critical role of a SIEM in reducing "alert fatigue" by visually distinguishing between generic automated scanning and targeted, high-severity intrusion attempts.