📄 Lab Report – First Splunk Search & Mini Dashboard

🎯 Objective

The goal of this lab was to get hands-on experience with **Splunk**, perform basic searches on built-in data, explore event fields, and create a **live dashboard** that provides quick visibility into internal Splunk activity.

🧰 Tools Used

- **Splunk Cloud Trial** (Splunk Enterprise interface)
- **Web Browser (Chrome)**
- Built-in data source: index=_internal

🧩 Step-by-Step Execution

Step 1: Launch Splunk & Open Search

I logged into Splunk Cloud and navigated to **Apps ▸ Search & Reporting**.
The **time range** was set to *Last 24 hours* to focus on recent internal events.

Step 2: Initial Search – Explore Built-in Data

I started by running:

index=_internal

This displayed internal Splunk logs with fields like _time, host, source, and sourcetype.

To focus on specific types of messages, I refined the search:

index=_internal (error OR warn)

and later filtered by sourcetype:

index=_internal sourcetype=splunkd

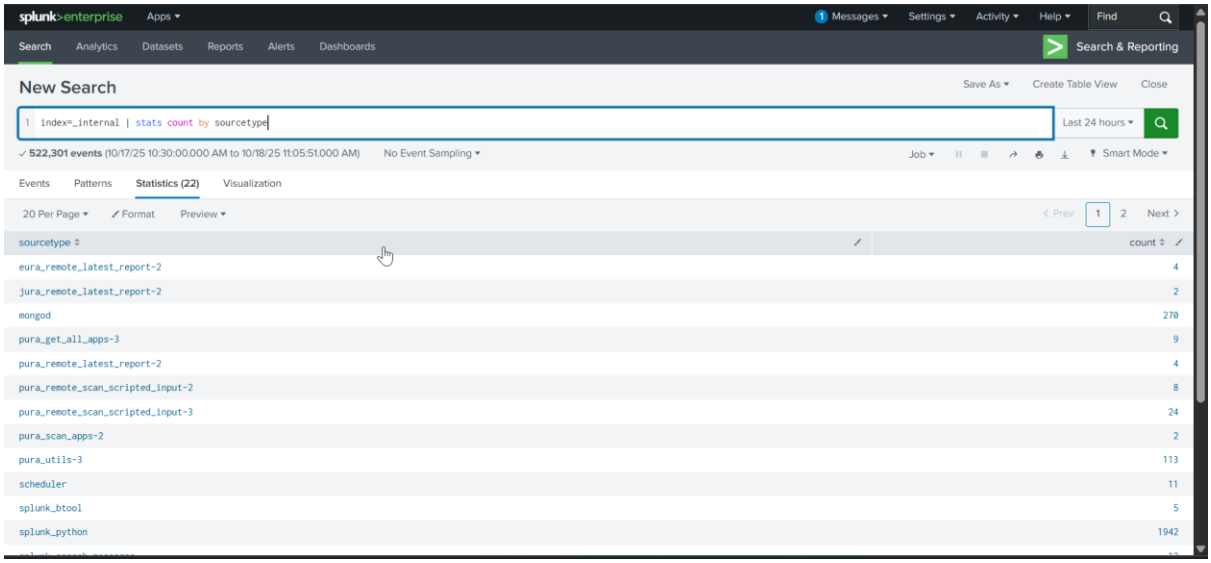
Step 3: Summarizing Data with SPL Commands

(a) Count Events by Sourcetype

I ran:

index=_internal | stats count by sourcetype

This summarized event counts for each sourcetype and provided visibility into which components were most active.



(b) Hourly Timechart of Events

To understand trends over time:

index=_internal | timechart span=1h count

This generated a time-based visualization showing hourly event counts.
I then saved this visualization as a **Dashboard Panel** titled **Internal Events (Hourly)**.

Step 4: Building the Dashboard

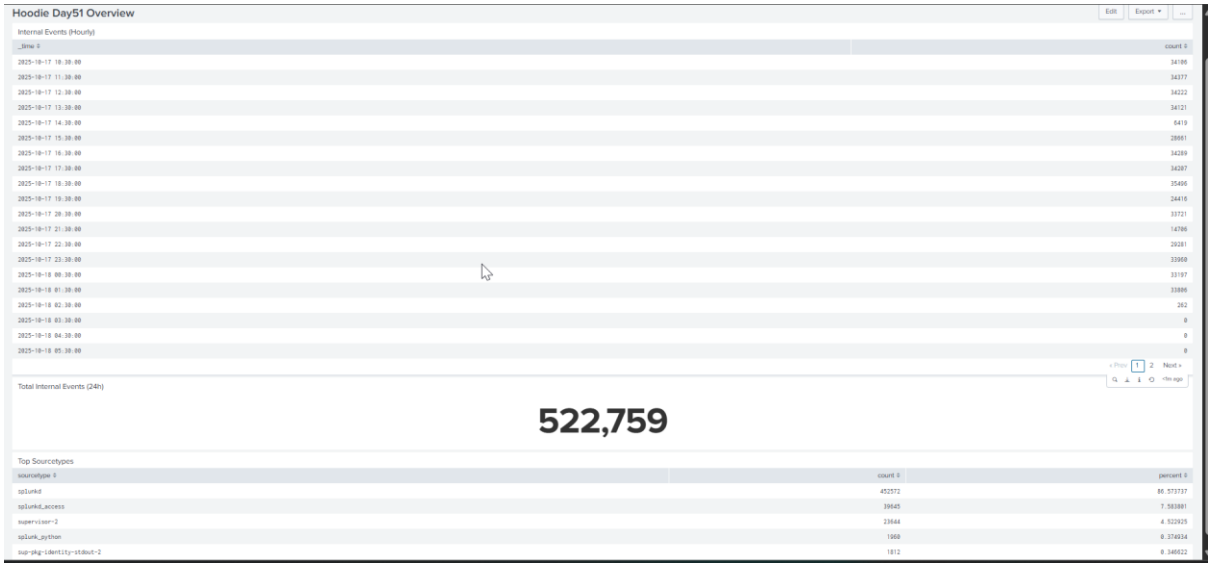
(a) Creating "Hoodie Day51 Overview"

Using **Save As → Dashboard Panel**, I created a dashboard named:

Hoodie Day51 Overview

(b) Adding Panels

1. **Internal Events (Hourly)** – Timechart panel
2. **Total Internal Events (24h)** – Single value KPI using:
3. index=_internal | stats count as events
4. **Top Sourcetypes** – Using:
5. index=_internal | top sourcetype limit=5

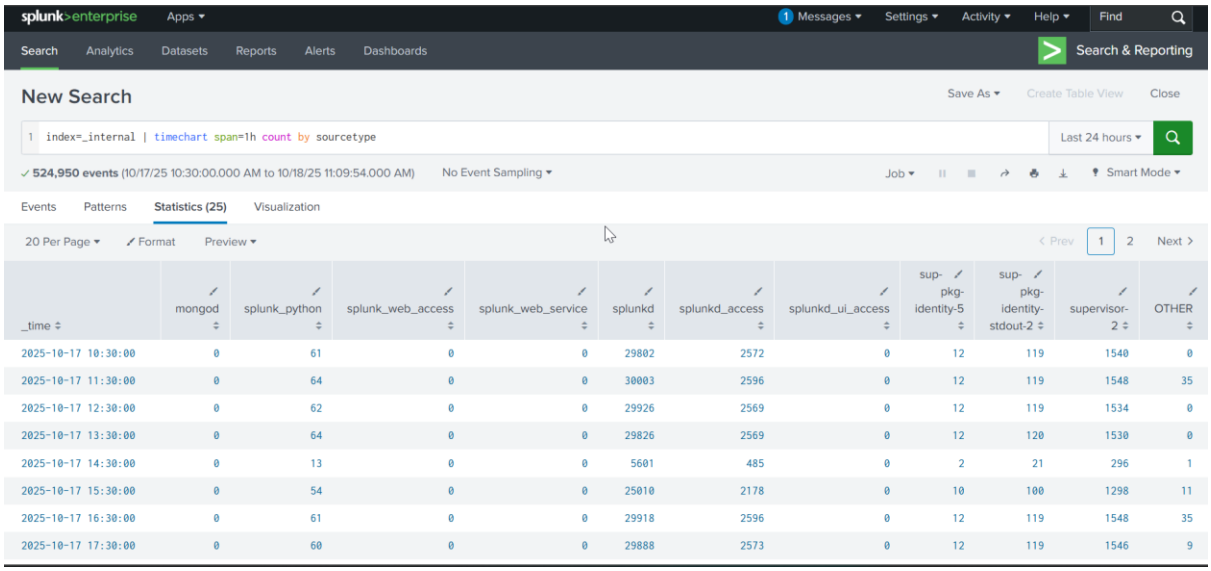


Step 5: Advanced Visualization

I extended my query with:

index=_internal | timechart span=1h count by sourcetype

This split the timechart by sourcetype, allowing me to see event distribution for each type over time.



_time	mongod	splunk_python	splunk_web_access	splunk_web_service	splunkd	splunkd_access	splunkd_ui_access	sup-pkg-identity-5	sup-pkg-identity-stdout-2	supervisor-2	OTHER
2025-10-17 10:30:00	0	61	0	0	29802	2572	0	12	119	1540	0
2025-10-17 11:30:00	0	64	0	0	30003	2596	0	12	119	1548	35
2025-10-17 12:30:00	0	62	0	0	29926	2569	0	12	119	1534	0
2025-10-17 13:30:00	0	64	0	0	29826	2569	0	12	120	1530	0
2025-10-17 14:30:00	0	13	0	0	5601	485	0	2	21	296	1
2025-10-17 15:30:00	0	54	0	0	25010	2178	0	10	100	1298	11
2025-10-17 16:30:00	0	61	0	0	29918	2596	0	12	119	1548	35
2025-10-17 17:30:00	0	60	0	0	29888	2573	0	12	119	1546	9

Step 6: Making It Live

I edited the dashboard and enabled **Auto-Refresh every 60 seconds** for real-time updates.
This ensures the dashboard reflects the latest system activity continuously.

✅ **Deliverables**

- **Screenshot 1:** Event count by sourcetype (stats count by sourcetype)

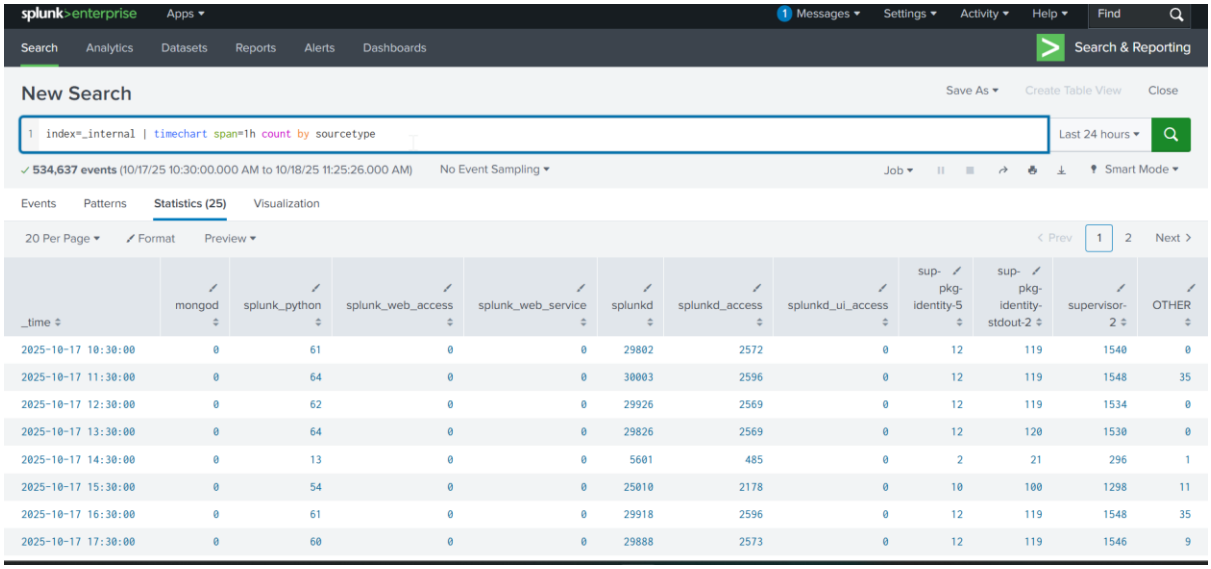
- **Screenshot 2:** Hoodie Day51 Overview dashboard with 3 panels
- **Screenshot 3:** Hourly timechart split by sourcetype

Observation:

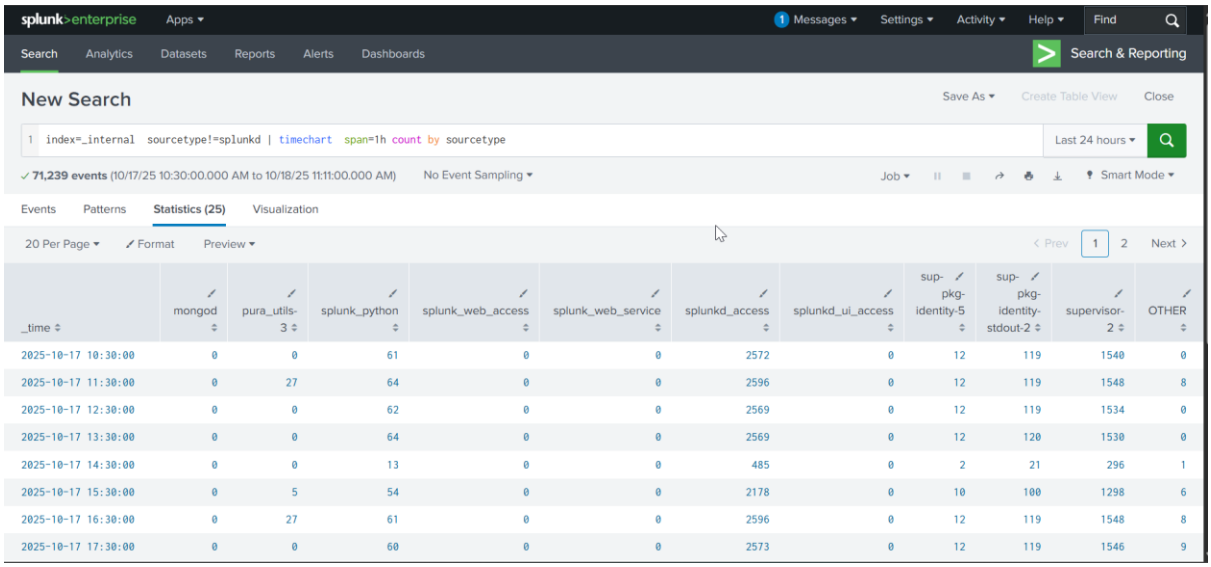
Transitioning from raw logs to summary views revealed patterns hidden in plain text — showing counts, trends, and top sources at a glance.

Stretch Goals (Optional)

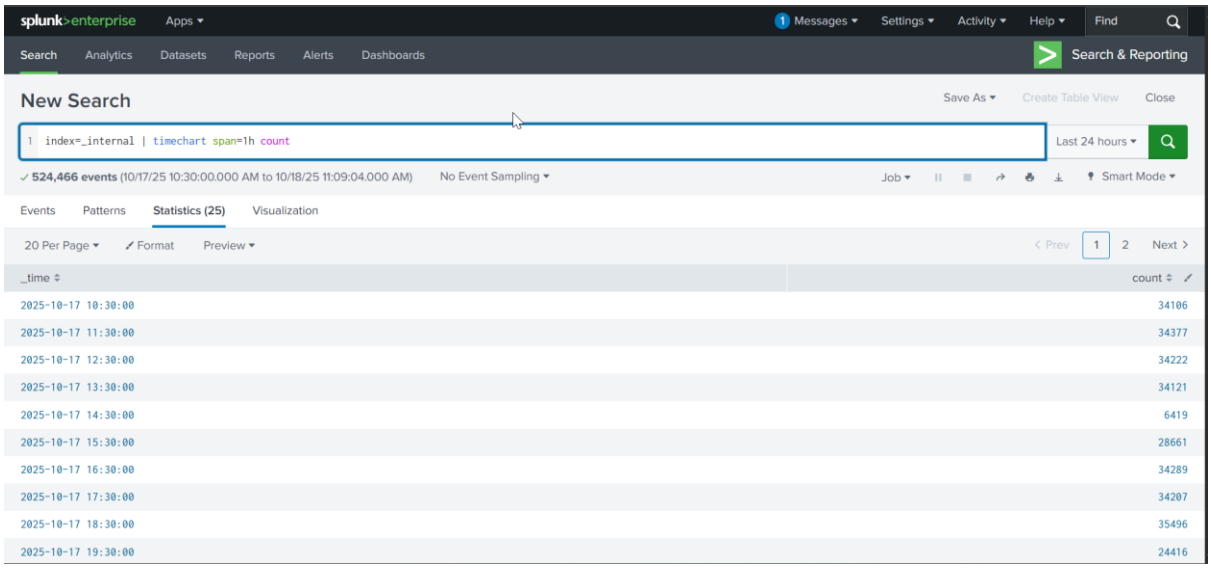
- Add this split-by status timechart:
  index=_internal | timechart span=1h count by sourcetype



- Make a drilldown: clicking a sourcetype in the table opens a new search filtered by that value (use tokens).



- Create a second Single Value panel with a sparkline:
  index=_internal | timechart span=1h count (then convert to Single Value with sparkline)



🧠 Reflection

1️⃣ What did the stats/top/timechart commands reveal that raw events didn't?

They highlighted **aggregate insights** — which sourcetypes were most active and when spikes occurred — providing situational awareness that isn't visible in individual event lines.

2️⃣ How would this dashboard help your SOC or DevOps team during an incident?

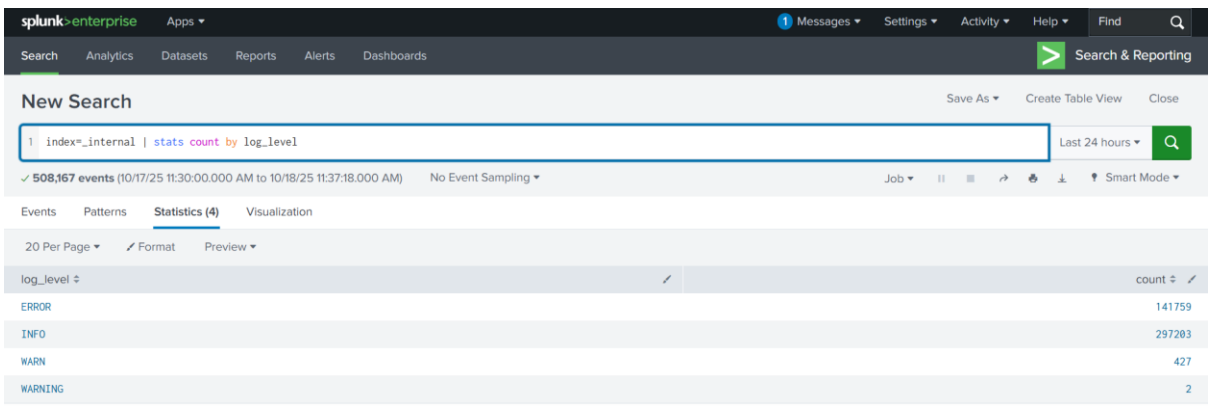
This dashboard acts as a **real-time monitoring tool**, helping teams detect abnormal activity, ingestion delays, or internal system errors immediately.

3️⃣ What panel would you add next and why?

I would add a **log_level summary**:

index=_internal | stats count by log_level

This would display error, warning, and info distributions to help quickly identify spikes in critical errors.



📜 Lab Summary

In this lab, I successfully:

- Ran foundational SPL searches on built-in data
- Explored event fields and filters
- Created **summary statistics**, **time-based visualizations**, and **top-value tables**
- Built a **real-time auto-refreshing dashboard**

This marks the first step toward mastering **Splunk-based data analytics**, enabling better monitoring, incident detection, and operational visibility for SOC or DevOps teams.