# Splunk vs ELK

## The Log Management Tools Decision Making Guide

Alex Zhitnitsky

# Splunk vs the Elastic Stack - Which Tool is Right For You?

Much like promises made by politicians during an election campaign, production environments produce massive files filled with endless lines of text in the form of logs. Unlike election periods, they're doing it all year round, with multiple GBs of unstructured plain text data generated each day.

In most cases, the act of manually going through plain log files, grepping all over the place, severely limits the value you can extract from them. Some might even say that it's… borderline insane.

In this book, and in the light of the new Elastic Stack v5, we're taking a practical look at 2 of the most popular log management solutions: Splunk and ELK (Elasticsearch-Logstash-Kibana) to help guide you through the questions you'll need to ask yourself in order to make the right choice. Let's get started.

# Table of Contents

**Chapter 1**

# The Basics - Who's who in log management?

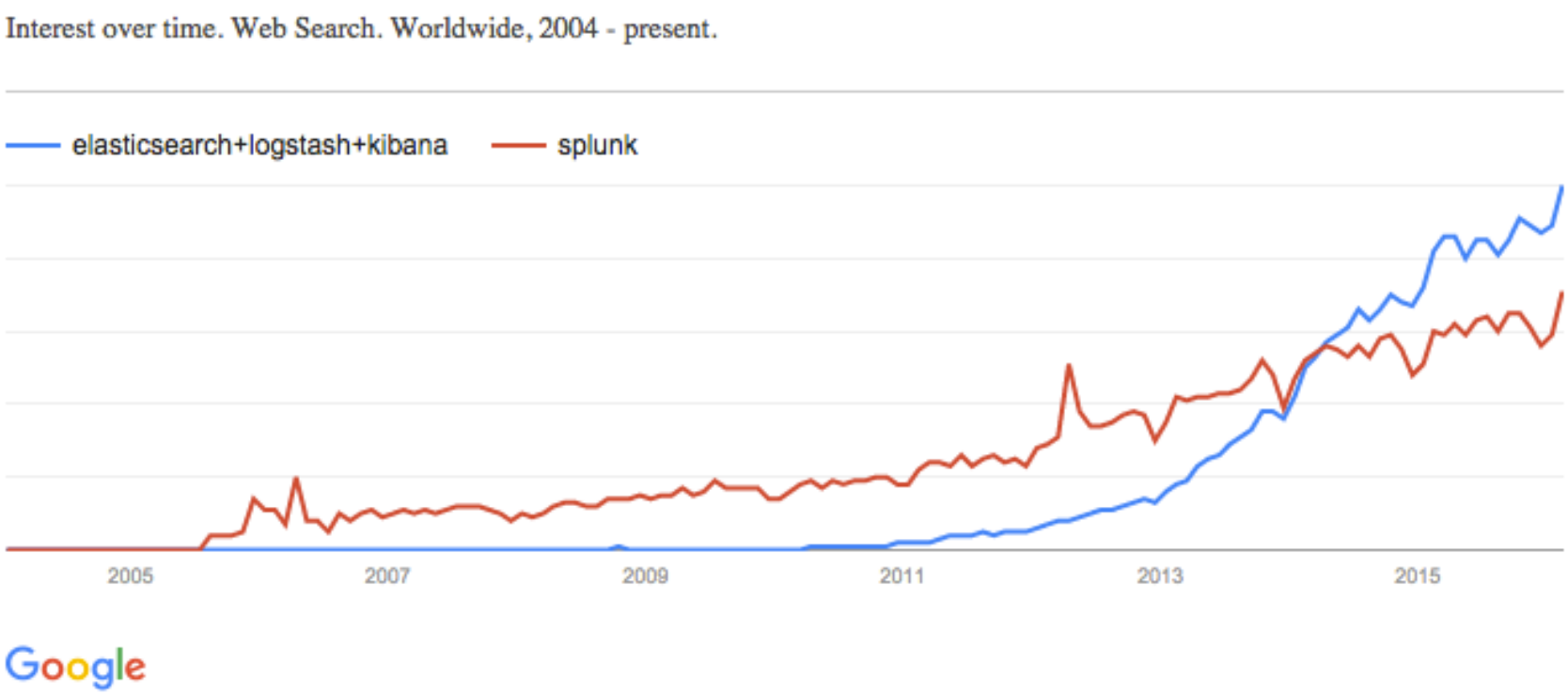
Whether it's log errors and exceptions, business logic, or any other kind of log analytics, Splunk and the ELK / Elastic Stack are the biggest enterprise grade solution approaches in the field. Splunk is a publicly traded company that offers a full commercial solution with a 15 day trial across its different products. ELK is an acronym for ElasticSearch, Logstash and Kibana, a free open source stack for log analytics with commercial support, managed solutions, and additional tools from Elastic. To be more accurate, we can also call it… BELK

Splunk was one of the first companies dealing with the inherent issues in logging and machine data, even before the term big data was coined. Founded in 2003 (Michael Baum, Rob Das and Erik Swan), the origin of its name comes from "spelunking" which is the practice of exploring caves.

On the other side of the pond, ElasticSearch was first released by Shay Banon in 2010 and a company was founded around it, now called Elastic. Joined forces with Logstash (Jordan Sissel) and Kibana (Rashid Khan), and recently PacketBeat (Monica Sarbu and Tudor Golubenco) the tool chain remains free and open source, making place for a variety of ELK-as-a-Service solutions.

When comparing the two through Google Trends we see that both are rising in popularity, with interest in ELK quickly gaining momentum and gradually passing Splunk.

Considering other log management tools in the interest graph, doesn't even come close, while tools like Sumo Logic and Loggly offer feature rich SaaS solutions with competitive pricing.




*View full report in Google Trends*

The biggest criticism against the 2 is that Splunk is expensive; and ELK, while free and open source, is time consuming and includes additional hardware costs that grow exponentially.

**Bottom line:** Splunk and the ELK stack are dominating the interest in the log management space with the most comprehensive and customizable solutions.

**Chapter 2**

# The Problem

# What are you trying to solve?

From a market perspective, Splunk has traditionally been on-prem, targeting big enterprises, and only recently started making an affordable offering for smaller companies. ELK, on the other hand, like open source solutions tend to be, has seen adoption across all types of different companies. With Elastic the company and the growing ELK ecosystem, they're offering premium support and a new set of paid and open source services to accompany it.

Main things to consider:

- On-prem vs cloud vs managed solutions
- Daily GBs consumed and required data retention (Main pricing component)
- Which and how many services you'd like to connect
- Users, are they all developers?
- Specific use cases
- Rate of expected changes to your dashboards

If all you're searching for is advanced grepping capabilities (which have a lot of value on their own) and easy visualizations, then a full Splunk deployment is probably an overkill.

If your use cases are expected to grow fast over time, include complex scenarios with ever changing specs, serving multiple users and departments inside your company, then an ELK deployment is going to take more of your team's time in customizing it to your need. Not speaking of hardware maintenance or cloud storage costs if you're not choosing a managed solution. Determining the Total Cost of Ownership (TCO) for a vanilla ELK installation can be quite tough.

**Bottom line:** Before making a decision, try to understand what would be a sufficient solution. Most important, is it a closed / constant problem or you're expecting it to grow with additional use cases over time.

**Chapter 3**

# Installation modules - What kind of setup do you require?

This breaks down intro 3 decisions:

- Do you require an on-prem solution or prefer a cloud deployment?
- Do you need a specialized solution or a broad log management platform?
- For ELK, would you prefer a managed solution or would like to handle it all by yourself?

Let's have a look at the options:

Splunk's products are split between 2 categories, core and premium specialized solutions. Splunk Enterprise for on-premise installation (that's the one you've probably heard about before), and its younger brother Splunk Cloud. For this post, we've played around with the Splunk Cloud version using some sample log data that we've uploaded to it in plain text files. Enterprise and Cloud provide the same features except for the difference in deployment of course.

On the lighter side (pun intended), there's Splunk Light, available in both hosted and on-prem editions, for smaller scale and a lower budget starting from $124 per month on a monthly retention plan with 1 GB per day. It's the newest addition to Splunk's core offering, here's a comparison matrix between the two.

Apart from that, there's a Splunk product built around Hadoop, if you're already using it and would like to add Splunk's capabilities on top (It's worth noting that you can also ship Hadoop data to ELK through the Logstash Hadoop connector).

For specialized use cases around security, IT services, and user behavior, there are 3 additional specialized products with modules to handle specific types of data. ELK has none of these advanced features built-in.

With the ELK stack, we have one central decision: independent deployment or managed. The managed approach is paid of course. The independent free approach, while giving you more flexibility, risks having you or one of the other engineers becoming a full time in-house ELK engineer - Which is probably less than ideal in most cases. Additional paid support is available through Elastic (The company!).

The managed approach is mainly made up of hosted ELK-as-a-Service solutions, and there's also a new option for subscribing to an on-premise ELK solution, currently available by SemaText and also in beta by Elastic. That is, a paid product built upon the ELK stack and deployed on-prem.

**Bottom line:** Splunk has been traditionally on-prem, serving large enterprises, and that's where it puts most of its focus, with easily customized solutions for a big set of use cases. ELK is all over the place, and its success depends on how much effort you'll put in.

**Chapter 4**

# Application Logs vs Business Data - What kind of results do

Splunk and ELK offer log analytics solutions, but that's not the end of the story. Logs traditionally contain machine data that's automatically generated by different services about their live operation. In addition to machine data, there are logs which relate to business metrics. Things like sales, user behavior, and other product specific informations.

However, **the strongest use case for logs is troubleshooting**. In most cases, what goes in your log is the only information you have to understand what went wrong in the execution of your code in production. This includes logged errors, warnings, caught exception and, if you're lucky, notorious uncaught exceptions.

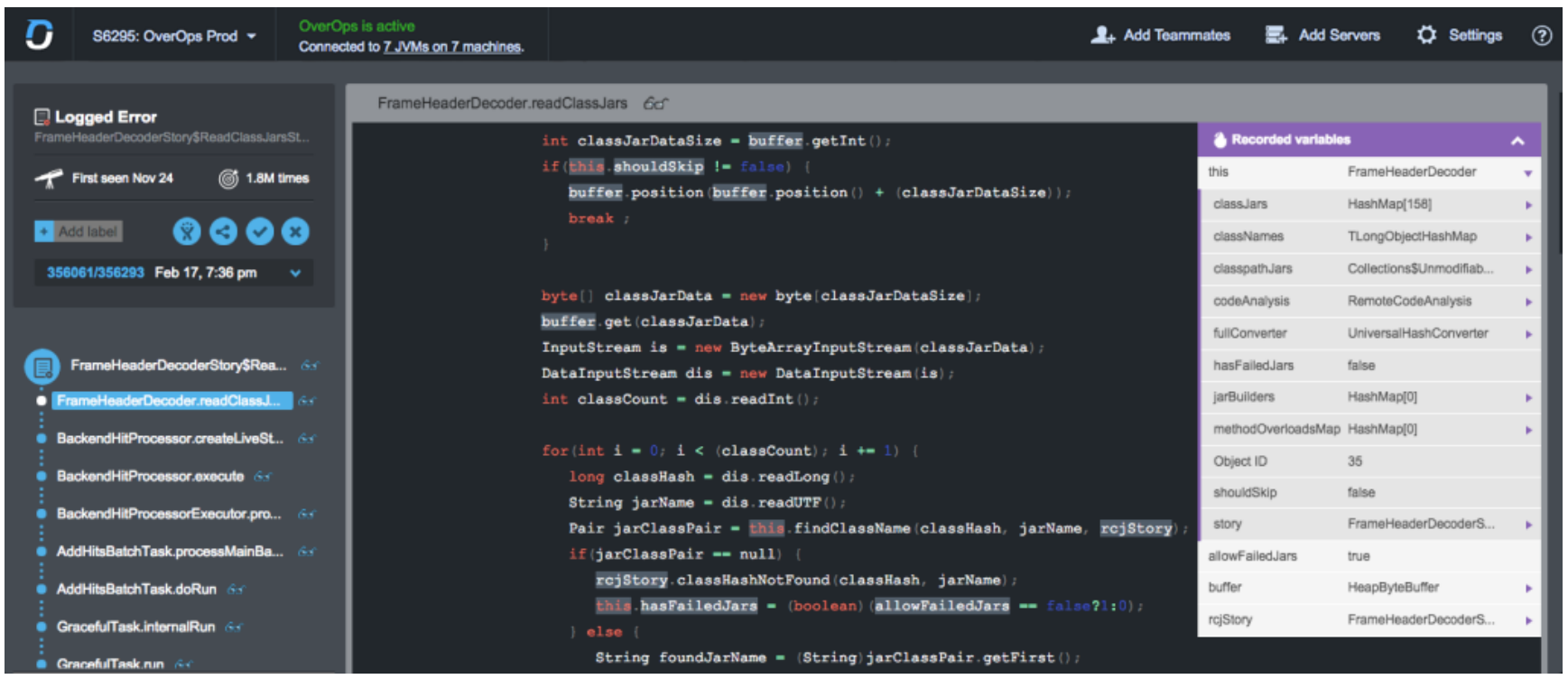
More often than not, the only thing you're left with is the error message you've attached and the stack trace at the moment of error. That's just not enough to understand what's the real root cause of the error and sends you off to an iterative process of trying to replicate the same error, add more info to the log, and so on.

Many of our users add OverOps to their toolbox to help debug those issues. While Splunk and ELK can parse errors that were written into log files, OverOps can help solve them without the need to reproduce them manually.

To overcome that gap and cut down issue resolution times, OverOps's log link feature injects a link that leads to each event's analysis. For example, in Splunk, a OverOps log link would normally show up in an event. In this case, a log error:

| i | Time | Event |
|---|---|---|
| > | 2/17/16 5:46:50.000 PM | [17-02 17:46:50][AH -2838 -S14332] ERROR - Misplaced class hashes: {-2678217155125403057=<, J76500390>}. [ http://tkp.to/qs1CYFuaNTA ] {FrameHeaderDecoderStory} [SQS-prod_taskforce1-Local-Queues-Worker-Executor-1-thread-8 [SID:<br>date_mday = 17 host = splunk_cloud_trial source = tkp sourcetype = tkp-too_small |

When opening the link, you're getting to OverOps's error analysis screen where you can see the code and variable values that led to the error, all across its stack trace:




Unlike reactive log analyzer, OverOps detects exceptions and logged errors in real-time at the JVM level without relying on parsing logs files, so you can fix them fast and keep your users happy.

**Bottom line:** The results you'll be getting are only as good as the data you're sending in. For troubleshooting exceptions and log errors, check out OverOps to enrich your production logs with links to in-depth error analysis.

**Chapter 5**

# Logstash, Beats and Splunk Forwarders - How to ship data to your tool of choice?

The most common operation your'e going to support is shipping real data to the tool you're using. Whether it's Splunk or ELK, the basic principle is similar. Each data source has to be coupled with a data shipper.

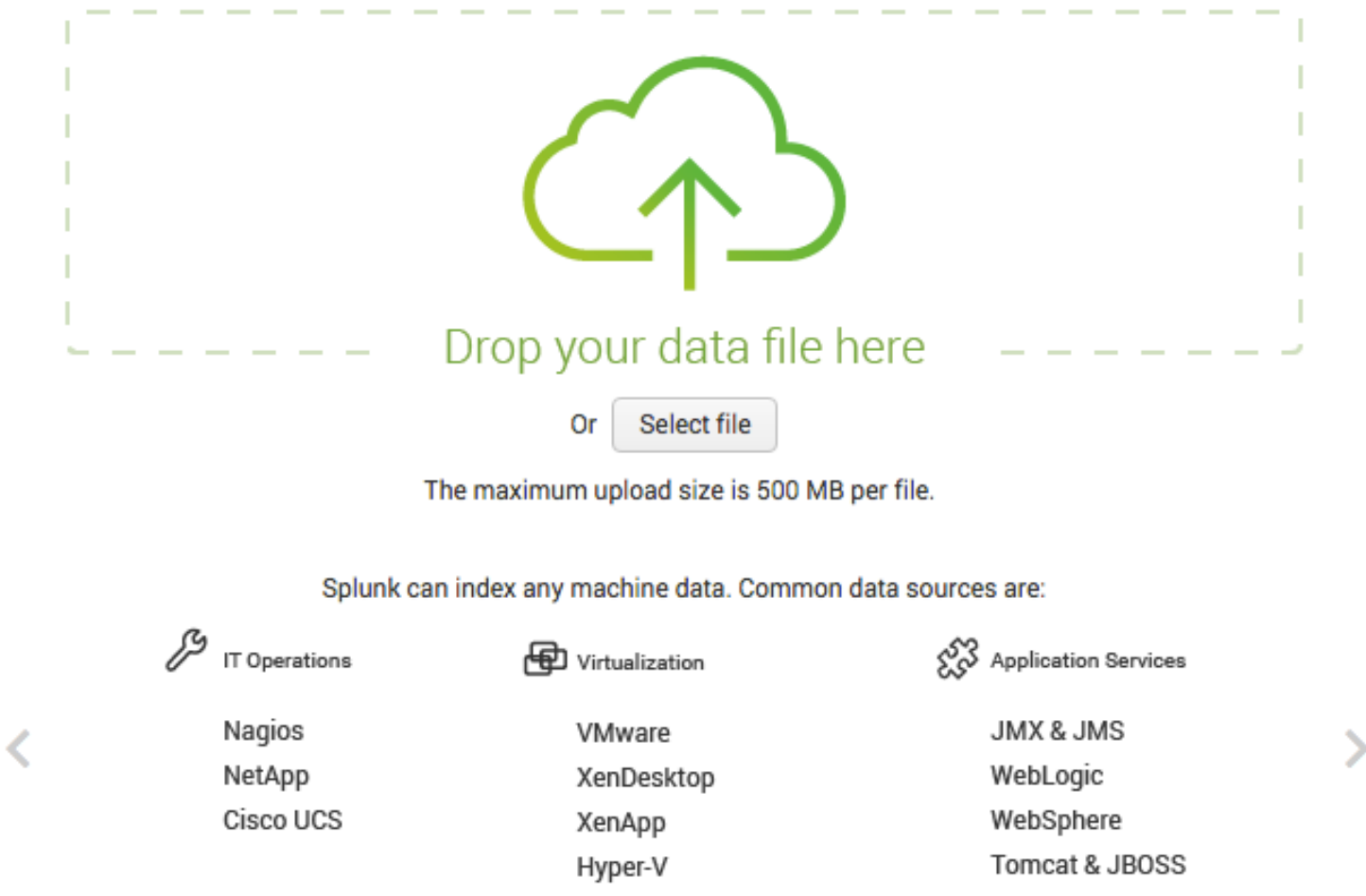
Up until recently we only had one main option for shipping data to ElasticSearch; Logstash, which is notoriously known for its long startup times among other things. Another common rant is that it's hard to debug and that it uses a non-standard configuration language.

Less than a year ago, Elastic launched Beats, based on their acquisition of Packetbeat. Now there's another alternative to format and ship different kinds of data into ElasticSearch, and it's also possible to use Logstash as an intermediary.

On the Splunk front, apart from out of the box support with preconfigured settings for any kind of data source you can think of, it also has centralized forwarder management and straightforward data onboarding wizards.

Another main difference is the way the data is parsed. ELK requires you to identify the data fields BEFORE it's shipped to ElasticSearch, while with Splunk, you can do that after the data is already in the system. This makes data onboarding easier by separating shipping and data classification / field labeling.

**Bottom line:** The Splunk way is smoother, but ELK doesn't limit you in any way.




*Splunk's welcome screen for data onboarding*

**Chapter 6**

# Do you require user management features?

Yes. Something as basic as user management features can be quite a pain with plain ELK and a no brainer with its managed solutions, or a the full service Splunk.

If your internal users are developers from a single team, this might be less important, but If you're serving users from multiple departments who require different access permissions, it becomes more complex. Also, starting off without user permissions can become a problem real quick if you're working in a growing team.

Splunk and managed ELK services offer user management out of the box and also include user auditing. For vanilla ELK you'll have to jump through a few more hoops and add Shield to your ELK stack  or develop a custom solution (soon to be part of the Elastic x-pack and called Security).

**Bottom line:** Missing user management features in basic ELK are a major barrier for larger organizations, and also limit the use cases for smaller companies. Splunk and hosted ELK are providing the biggest benefit here, for vanilla ELK, you'll have to add Shield.

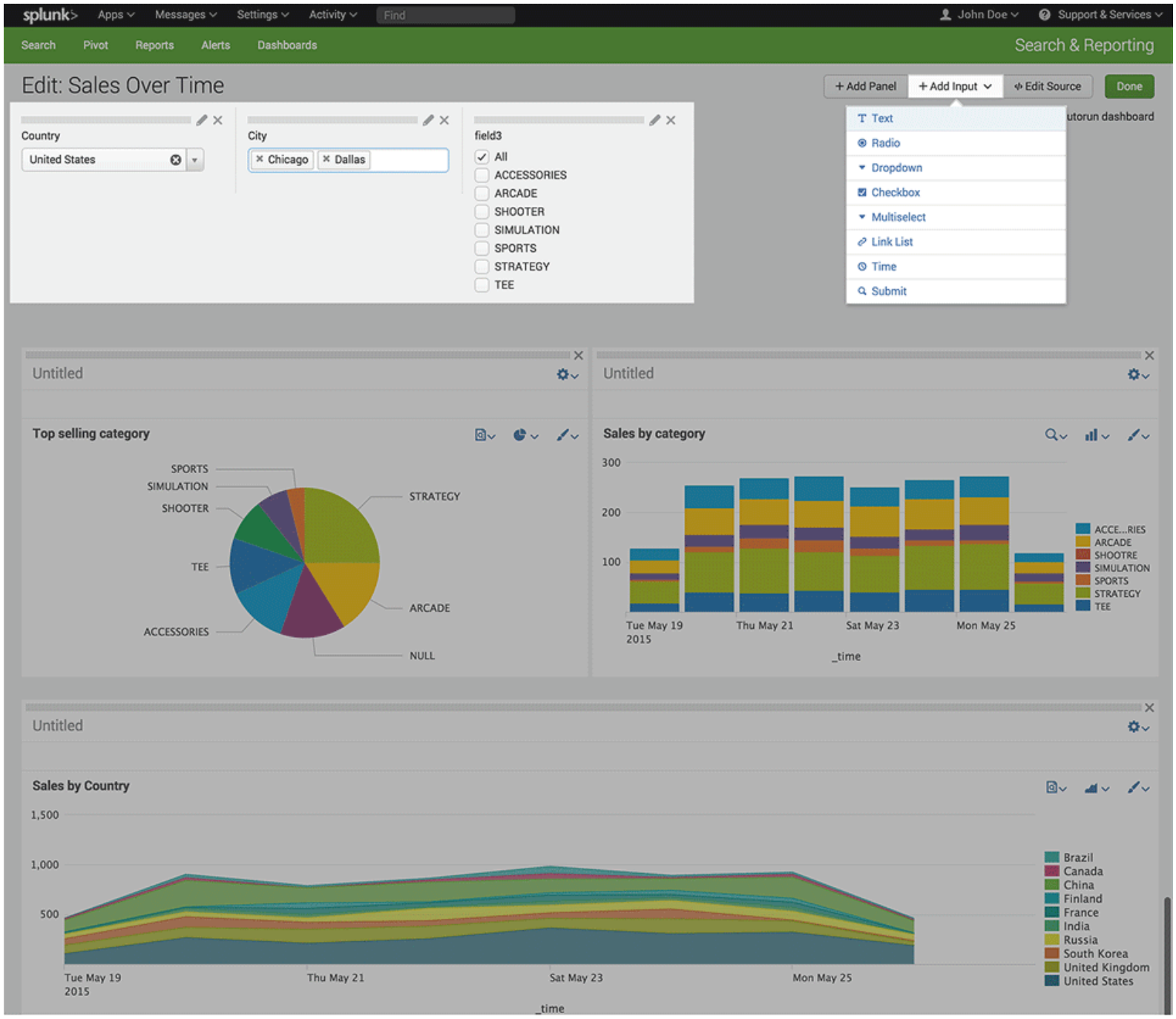
**Chapter 7**

# Usability - What's the difference between the dashboard?

In this section, we'll do a quick comparison between the visualization features of each solutions, that's Splunk's dashboard and ELK's K for Kibana.

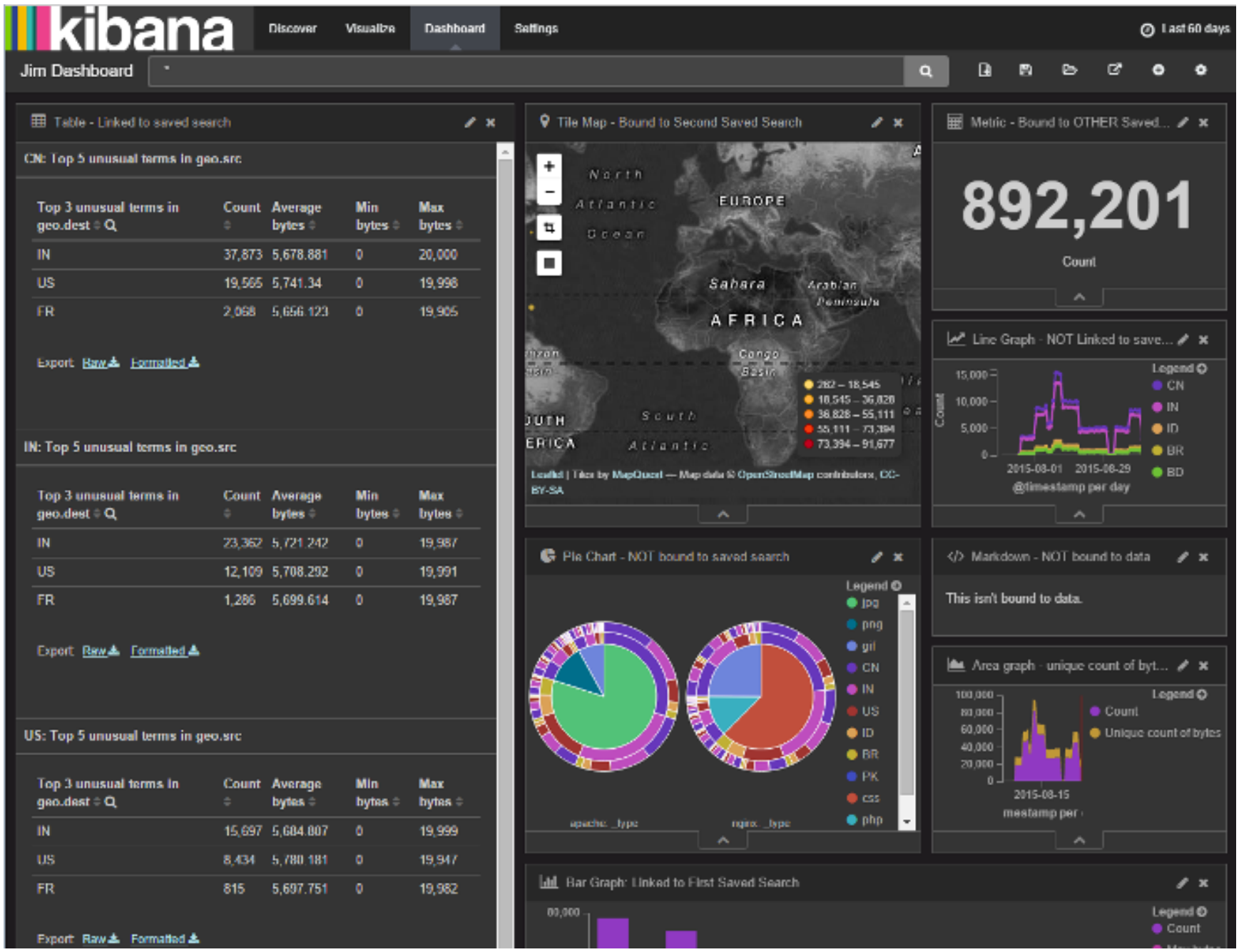
Compared to Kibana, some of the unique features in Splunk include in-dashboard controls that let you easily change the visualization, and dashboard management / user controls, allowing you to assign different dashboards to different users or departments.

Even something that seems as simple as exporting dashboards to PDF, which exists in Splunk for a while now, is an open issue in Kibana (Since 2013!)




*Splunk's dashboard, including visualization controls*

Taking a look at Kibana, personally, I really like the dark themes. Also available in Splunk but a bit harder to customize. The look and feel of Kibana seems more natural, but that just might be related to personal taste.




*The Kibana dashboard with a dark theme*

Here's a quick video walkthrough of the Kibana UI.

Usability wise, in addition to the desktop dashboards, Splunk also has mobile applications to support its offering.

**Bottom line:** Both dashboard provide a good experience, although Splunk's dashboard has more features and suitable for enterprise clients.

# Final Thoughts

A head to head comparison is always a tough call, especially when there's no clear winner and the tool you choose can potentially have a huge impact on the business. Before making a decision, try to better understand your own requirements and keep in mind that it's not only about log analytics – It's also what goes in them.

Also, if you haven't already, be sure to check out OverOps and see how you can enhance your logs with rich data about each error.

We hope you've found this guide useful and would be happy to hear your feedback on twitter @overops and over email: hello@overops.com