Best Practices and Better Practices for Users

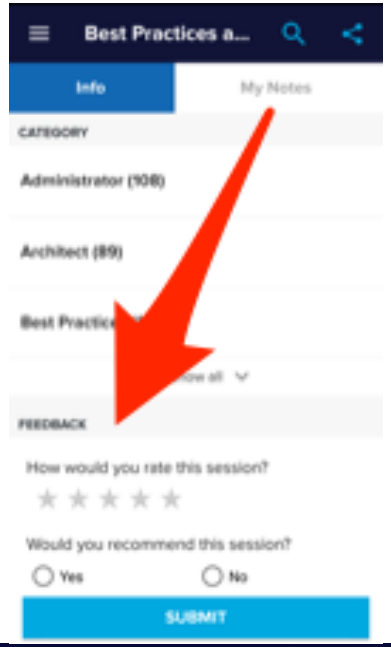
While you get settled…

Download Latest Slides:

https://splunk.box.com/v/burch-practices-user

or ask a neighbor with flash drive

Load Feedback:



Best Practices and Better Practices for Users

Burch

Sales Engineer @ Splunk

Disclaimer

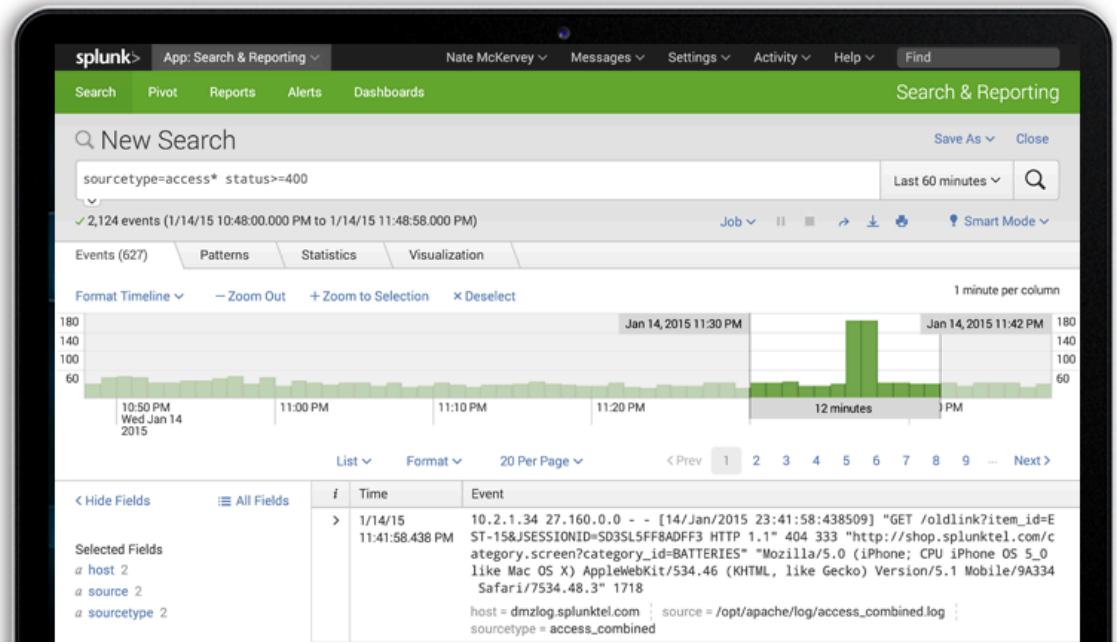
During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Burch's Goal

Learn from my (our) mistakes

Agenda

- Who are we?
- Resources
- Treat Yo’ Self
- Searching
- Next Steps



Best Practices

Who are we?

What's a Burch?

- Senior Sales Engineer in Boston
- Education
  - CS @ Boston University
  - MBA @ Northeastern University
- Splunk Customer
  - Middleware for 8 years (+splunk)
  - Splunk Admin for 1.5 years (splunk 4.3+)
- Certs: Knowledge, Admin, Architect
- @Splunk since Dec 14
- Splunkbase apps

About you

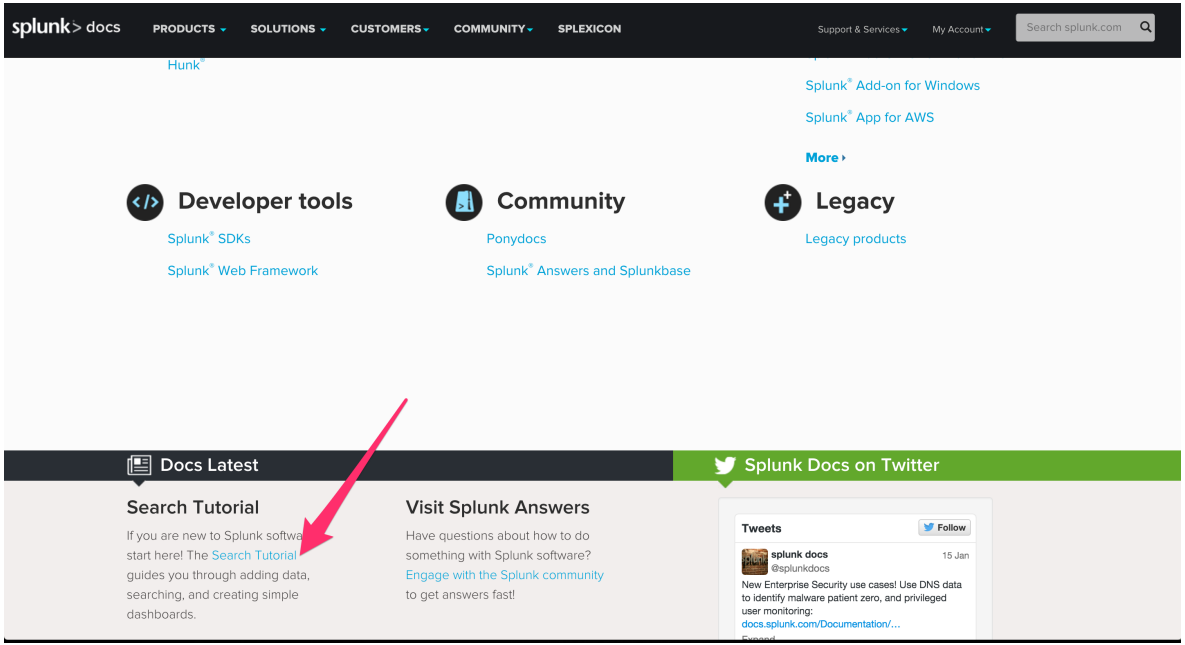
- Name
- User?
- Power User?
- Admin?
- Groupie?

Best Practices

Resources

Search Tutorial

- Free Search Tutorial -> docs.splunk.com -> Search Tutorial



Benefits:

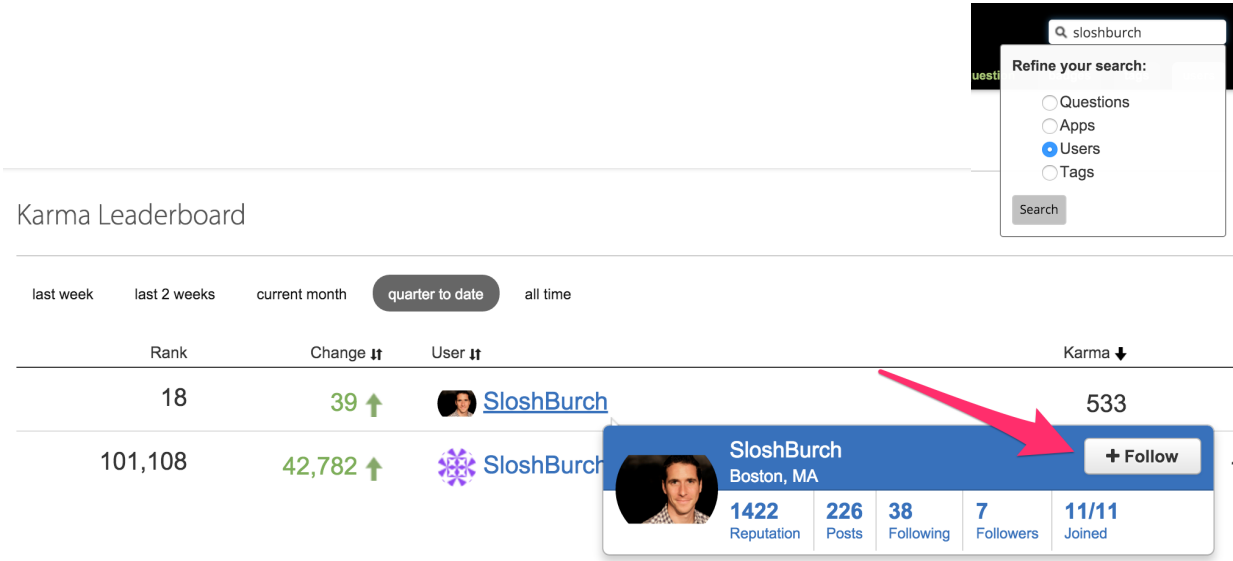
- Downloads & Installs Splunk
- Local sandbox
- Add real data

Community Q&A

- answers.splunk.com

Benefits:

- E-mail notifications
- Fast answers
- Larger distribution



Splunk! The Book

- http://www.splunk.com/goto/book

Exploring Splunk

SEARCH PROCESSING LANGUAGE (SPL)
PRIMER AND COOKBOOK

Splunk is probably the single most powerful tool for searching and exploring data you will ever encounter. Exploring Splunk provides an introduction to Splunk -- a basic understanding of Splunk's most important parts, combined with solutions to real-world problems.

Part I: Exploring Splunk

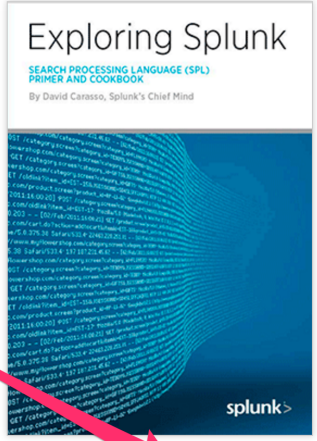
- Chapter 1 tells you what Splunk is and how it can help you.
- Chapter 2 discusses how to download Splunk and get started.
- Chapter 3 discusses the search user interface and searching with Splunk.
- Chapter 4 covers the most commonly used search commands.
- Chapter 5 explains how to visualize and enrich your data with knowledge.

Part II: Solution Recipes

- Chapter 6 covers the most common monitoring and alerting solutions.
- Chapter 7 covers the most common transaction solutions.
- Chapter 8 covers the most common lookup table solutions.

About the Author

David Carasso, Splunk's Chief Mind, was the third Splunk employee. He has been responsible for innovating and prototyping a class of hard problems at the Splunk core, including developing the Search Processing Language (SPL), dynamic event and source tagging, automatic field extraction, transaction grouping, event aggregation, and timestamping. He holds two patents for his work with Splunk, and lives in Marin County, California, with his wife and three children.



Download the Book: ePub | pdf | Kindle

Purchase a Hardcopy:
Amazon | Splunk Store

Benefits:

- Real examples of commands
- Deeper than docs
- Free!

Quick Reference Guide

- Search “splunk quick reference guide”

splunk> QUICK REFERENCE GUIDE

Concepts

Events

An event is a set of values associated with a timestamp. It is a single entry of data and can have one or multiple lines. An event can be a text document, a configuration file, an entire stack trace, and so on. This is an example of an event in a web activity log:

```
10.14.0.172 - - [01/
Mar/2015:12:05:27 -0700] “GET /
trade/app?action=logout HTTP/1.1”
200 2953
```

You can also define transactions to search for and group together events that are conceptually related but span a duration of time. Transactions can represent a multistep business-related activity, such as all events related to a single customer session on a retail website.

At search-time, indexed events that match a specified search string can be categorized into event types.

Indexes

When data is added, Splunk software parses the data into individual events, extracts the timestamp, applies line-breaking rules, and stores the events in an *index*. You can create new indexes for different inputs. By default, data is stored in the "main" index. Events are retrieved from one or more indexes during a search.

Index-Time and Search-Time

During *index-time* processing, data is read from a source on a host and is classified into a source type. Timestamps are extracted, and the data is parsed into individual events. Line-breaking rules are applied to segment the events to display in the search results. Each event is written to an index on disk, where the event is later retrieved with a search request.

Additional Features (Splunk Enterprise only)

Data Model

A *data model* is a hierarchically-organized collection of datasets that Pivot uses to generate reports. Data model objects represent individual datasets, which the data model is composed of.

Pivot

Pivot refers to the table, chart, or other visualization you create using the Pivot Editor. You can map attributes defined by data model objects to data visualizations, without manually writing the searches. Pivots can be saved as reports and used to power dashboards.

Apps

Apps are a collection of configurations, knowledge objects, and customer designed

Search Command Reference

docs.splunk.com -> Splunk Enterprise -> Search and report -> Search Reference -> Commands by category

Hide Contents

Documentation / Splunk® Enterprise / Search Reference / Commands by category

Search Reference

- Introduction
- Quick Reference
  - Splunk Enterprise Quick Reference Guide
  - Command quick reference
  - Commands by category
  - Command types
  - Splunk for SQL users
  - SPL data types and clauses
- Functions
- Time Format Variables and Modifiers
- Search Commands

Commands by category

The following tables list all the search commands, categorized by their usage. Some commands fit into more than one category based on the options that you specify.

Correlation

These commands can be used to build correlation searches.

Command	Description
`append`	Appends subsearch results to current results.
`appendcols`	Appends the fields of the subsearch results to current results, first results to first result, second to second, etc.
`appendpipe`	Appends the result of the subpipeline applied to the current result set to results.
`arules`	Finds association rules between field values.
`associate`	Identifies correlations between fields.

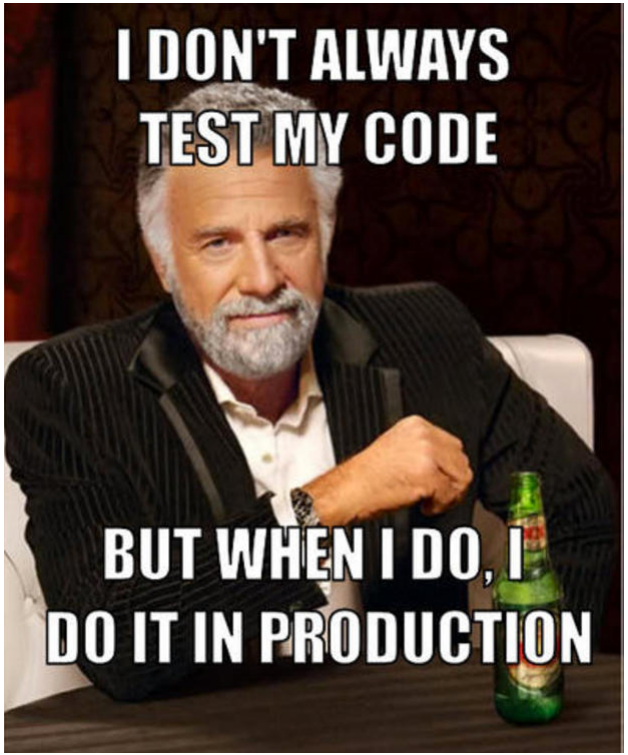
Commands by category

- Correlation
- Data and indexes
- Fields
- Find anomalies
- Geographic and location
- Prediction and trending
- Reports
- Results
- Search
- Subsearch
- Time

Reference

- Splunk Documentation -> docs.splunk.com
- Smart Answers -> blogs.splunk.com/?s=smart+answers
- Apps -> splunkbase.splunk.com
- User Groups -> usergroups.splunk.com
- FREE Education -> splunk.com/education

Play it safe

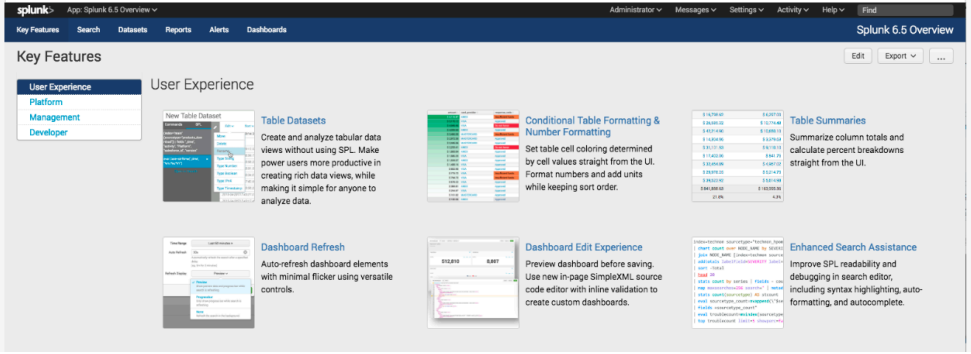


New Stuff

Splunk Enterprise 6.5 Overview

DOWNLOAD

Release 6.5 is the latest version of Splunk Enterprise and Splunk Cloud. We have developed an app to guide you through the powerful new features. This is not an in-depth tutorial, rather a guide to help you understand the new features, and to provide examples as well as sample reports, dashboards and visualizations.



★★★★★ 1 ratings

Rate this app

16 downloads

Unsubscribe

Share this app

VERSION 1.1

Utilities

Splunk Enterprise

App

Splunk 6.5

Splunk Software License Agreement

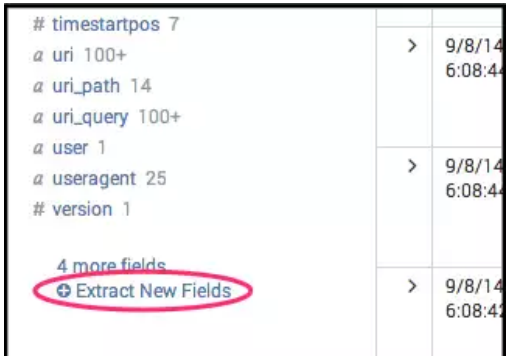
Platform Independent

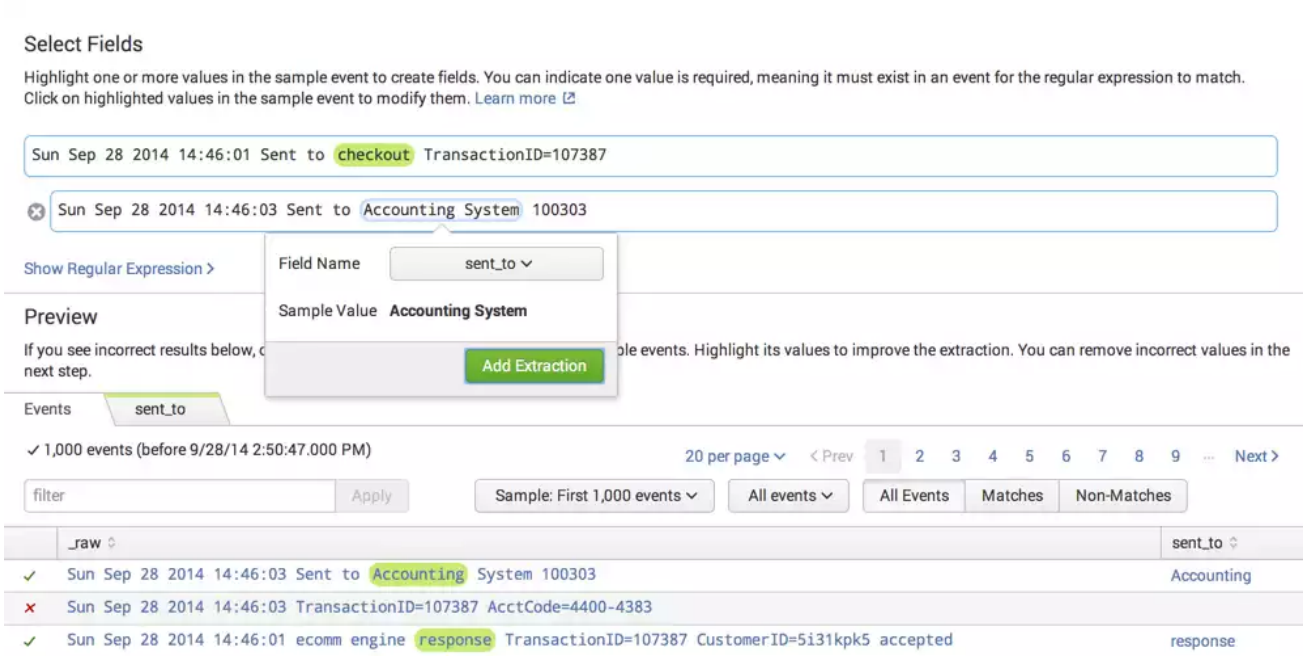
Best Practices

Treat Yo’ Self

Extract Fields

Search: docs.splunk.com field extractor





Don’t Scare Your Admins

Impress them!

- Turn off unused scheduled searches
- You don’t need real time
- Turn off unused acceleration

Dangerous Capabilities

Weak

- Scheduled Search
- Real Time Search
- Acceleration
  - Summary Indexing
  - Report Acceleration
  - Data Models

Strong

- Everyone a ‘user’
- Capabilities only for ‘power’+
- Work with you to implement and learn best practices
- Identify & coach & promote to power
- Don’t be a data butler

9

splunk> listen to your data

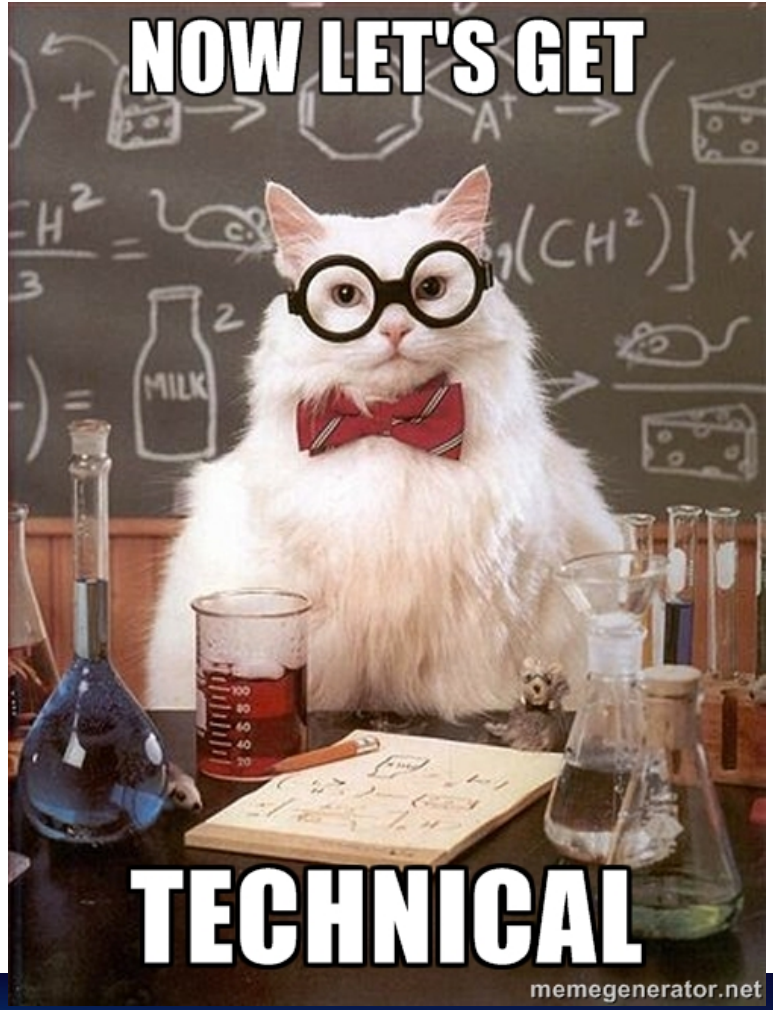
Acceleration Options

	Summary Indexing	Report Acceleration	Data Model Acceleration
Benefits	• Save disk space • Control on impact to system	• Backfill • Simple	• Backfill • Simple • Extensible • Search Agnostic
Limits	• Gaps • Intellectually difficult • Backfill	• Requires transforming • Specific to search	• Massive if misused

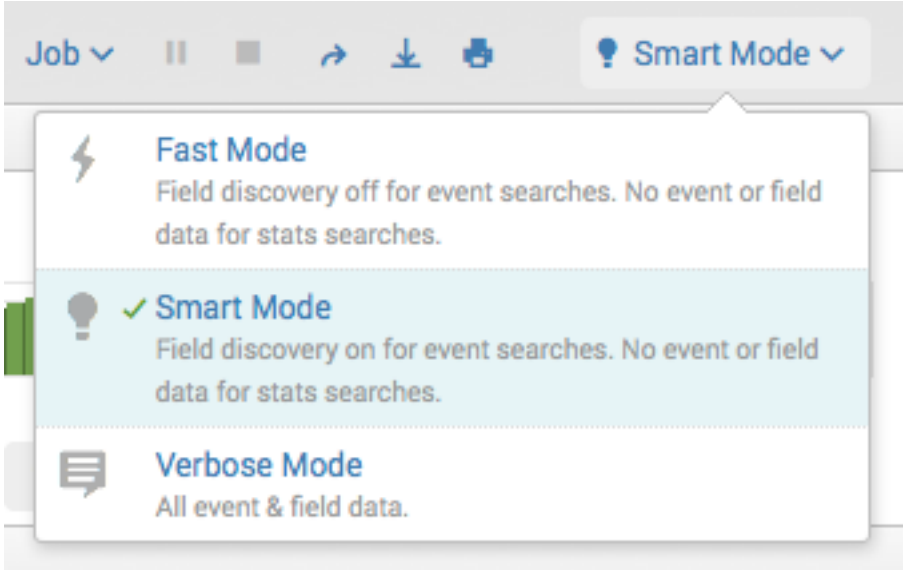
- Great article: Search documentation for “report acceleration”

Best Practices

Searching

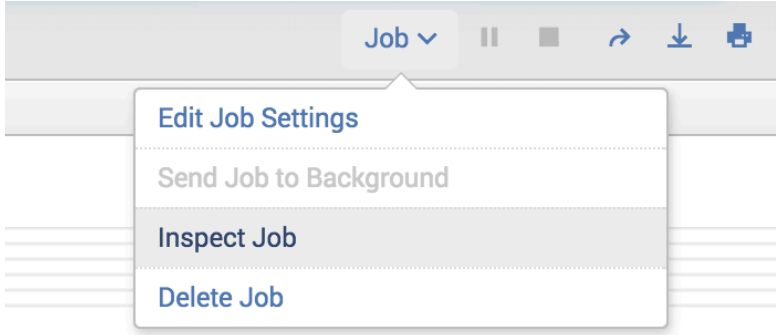

NOW LET'S GET
TECHNICAL
memegenerator.net

Search Speed



Job Inspector

- Job Inspector
  - docs.splunk.com “Search Job Inspector”



This search has completed and has returned **1,000** results by scanning **22,696** events in **1.049** seconds.

- events per second = events / seconds
- results per second = results / seconds

Pretty Searches: Keep it kosher

Weak:

```
… | rename machine as “host for later” | rename net as Subnet |
sort “host for later” | timechart count by “host for later”
span=1h
```

Strong:

```
…| timechart span=1h count by machine
 | sort machine
 | rename machine AS “host for later”,
net AS Subnet
```

- new pipe = new line + space + pipe
- | <command> <params> <processing>
- cosmetics at end
- combine multiple renames and rexs

Faster Searching: Less is more

Weak:

```
iphone
 | stats count by action
 | search action=AppleWebKit
```

Strong:

```
iphone action=AppleWebKit
 | stats count
```

Faster Searching: Require Fields

Weak:

```
iphone
 | stats count by action
```

Wrong Results:

Pulls both phone=iphone and user_agent=*iphone*

Strong:

```
phone=iphone action=*
 | stats count by action
```

Remember:

'iphone' is not the same as 'iphone6s'

Faster Search: Be specific

Weak:

```
iphone
 | stats count by action
```

Strong:

```
index=oidemo host=dmzlog.splunktel.com sourcetype=access_combined
source=/opt/apache/log/access_combined.log iphone
user_agent="*iphone*”
 | stats count by action
```

Time selector!

Search Tangent: Event Types & Tags

Weak:

```
index=oidemo host=dmzlog.splunktel.com sourcetype=access_combined
source=/opt/apache/log/access_combined.log iphone
user_agent="*iphone*”
 | stats count by action
```

Strong:

```
tag=iphone_event
```

or

```
eventtype=web_logs
```

Faster Searching: stats vs dedup/transaction

Weak:

```
… phone=*
  | dedup phone
  | table phone
  | sort phone
```

```
… phone=*
  | transaction host
  | table host, phone
```

Strong:

```
… phone=*
  | stats count by phone, host
  | fields - count
```

Pro Tip:

- Table is cosmetic
- Fields is reducing

Faster & Pretty Searching: multi-eval

Weak:

```
… | eval this=”is”
  | eval a=“verbose”
  | eval example=“of eval”
```

Strong:

```
… | eval this=”is”, a=“verbose”, example=“of eval”
```

http://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels#Enable_multi-eval_to_improve_datamodel_acceleration

Pretty Searches: foreach is clean

Weak:

```
…| timechart span=1h limit=0 sum(eval(b/pow(1024,3))) as size by
st
```

Strong:

```
…| timechart span=1h limit=0 sum(b) by st
 | foreach * [ eval <<FIELD>> = '<<FIELD>>' / pow( 1024 , 3 ) ]
```

Pretty Searches: Dereference Finesse

```
index=_internal | eval {log_level} = message
```

Selected Fields

a ERROR 100+

a INFO 100+

a WARN 100+

a WARNING 92

Pretty Searches: coalesce's cooler than if

Weak:

```
…| eval size = if( isnull(bytes) , if( isnull(b) , "N/A" , b ) ,
bytes )
```

Strong:

```
…| eval size = coalesce( bytes , b , "N/A" )
```

Faster Searching: Avoid Subsearches

Weak:

```
index=burch | eval blah=yay
 | append [ search index=simon | eval blah=duh ]
```

Strong:

```
( index=burch … ) OR ( index=simon …)
 | eval blah=case( index==”burch" , "yay" , index==”simon" ,
"duh" )
```

(`format` and `return` commands for returning results)

Faster Searching: NOT NOTs

Weak:

```
index=burch NOT blah=yay blah=cool
```

Strong:

```
index=burch blah=duh
```

```
index=burch blah!=yay
```

Implies `( blah!=yay blah=* )`

Search Commands: Transaction

Weak:

```
…| transaction host
```

Mo data, Mo problems!

Strong:

```
…| transaction maxspan=10m maxevents=100 …
```

Search Commands: Time and Units

Weak:

```
…| eval new_time = <ridiculous string edits>
```

Strong:

```
…| convert ctime(duration)
```

```
…| bin span=1h _time
```

```
…| eval pause = tostring( pause , "duration" )
```

```
…| rename new_time as _time
```

Search Commands: metadata

Weak:

```
index=*
 | stats count by host
```

Strong:

```
| metadata index=* type=hosts
```

Search Commands: eventcount

Weak:

```
index=*
 | stats count by index
```

Strong:

```
| eventcount summarize=false index=*
```

Accurate Results: Snap-To Times

Weak

Time range

Start time: -60min

Finish time:

Time specifiers: y, mon, d, h, m, s

Learn more

Acceleration

☐ Accelerate this search

Schedule and alert

☑ Schedule this search

Schedule type *: Basic

Run every *: hour

Strong

Time range

Start time: @hour-1hour

Finish time: @hour

Time specifiers: y, mon, d, h, m, s

Learn more

Acceleration

☐ Accelerate this search

Schedule and alert

☑ Schedule this search

Schedule type *: Basic

Run every *: hour

Accurate Results: Time Fields

Weak

Search

```
earliest=-24hours latest=now
...
```

Strong

Time range

Start time: @hour-1hour

Finish time: @hour

Time specifiers: y, mon, d, h, m, s

Learn more

Acceleration

☐ Accelerate this search

Schedule and alert

☑ Schedule this search

Schedule type *

Basic

Run every *

hour

Accurate Results: Realistic Alerts

Weak

- Static conditions
  - | where count>10
- Spam
  - Avg

Strong

- Actionable:
- stddev
- percXX

- Find anomalies when outside “normal”

- Writing Actionable Alerts Blog

Dashboard Performance

Weak

- Many similar searches
- Viewed by Many
- Viewed by Few

Strong

- Post Process
- Cache Results (scheduled search)
- Inline Searches

Best Practices

Next Steps

What Now?

Related breakout sessions and activities…

- Rate this! (be honest)
- More talks:
  - conf.splunk.com/speakers.html
  - Search for
    - Burch
    - Champagne
    - Optimization
    - Practices
    - tips
    - Worst

Free Discussion

Questions, ideas, experiences

...have you?

THANK YOU

splunk>