



.conf2015

# Smart Splunking

Jeff Champagne, Splunk  
Kate Engel, Morgan Stanley

splunk>

# Who's this dude?

Jeff Champagne

jchampagne@splunk.com

Client Architect

- Splunk user since 2011
- Started with Splunk in Fall 2014
- Former Splunk customer in the Financial Services Industry
- Lived previous lives as a Systems Administrator, Engineer, and Architect



# Who's this gal?

Kate Engel

Analyst at Morgan Stanley

- Started with Morgan Stanley in Summer 2014
- Former Splunk partner Professional Services Consultant
- Philadelphia Sports Fan 😊



# Am I in the right place?

- You should be...
- Familiar with SPL (Splunk Query Language)
- Comfortable creating visualizations and dashboards
- Not afraid of a tiny bit of XML
- A nice person

# Agenda

- Formatting your search results to fit your needs
- Search & Viz. Ticks
- Looking at Splunk's internal data
- Q&A



.conf2015

Make your data look  
as good as it feels

splunk>

# Scenario #1

You run a basic Splunk search to calculate average latency and end up with a value of 4.238765.

How do you turn this into something useful????



# Solution #1 - Round 'em Up



```
| eval latency=round(latency,2)."ms"
```

Store the output of this eval command into the latency field

Round the value of the existing latency field to two decimal places

Concatenate the output of the round function with the string value "ms"

4.238765

4.24ms

[http://docs.splunk.com/Documentation/Splunk/6.2.5/SearchReference/CommonEvalFunctions#Mathematical\\_functions](http://docs.splunk.com/Documentation/Splunk/6.2.5/SearchReference/CommonEvalFunctions#Mathematical_functions)



## Scenario #2

That's great, but I actually need  
to format a big number.

Specifically, Count\_of\_hits=3458826

# Solution #2 – Comma-tose



```
| eval Count_of_hits=tostring(Count_of_hits,"commas")." hits"
```

Store the output of this eval command into the Count\_of\_hits field

Convert the value of the existing Count\_of\_hits field to a string and use commas formatting

Concatenate the output of the tostring function with the string value " hits"

3458826

3,458,826 hits

[http://docs.splunk.com/Documentation/Splunk/6.2.5/SearchReference/CommonEvalFunctions#Conversion\\_functions](http://docs.splunk.com/Documentation/Splunk/6.2.5/SearchReference/CommonEvalFunctions#Conversion_functions)

# Scenario #3

Cool story bro, but now my number columns don't sort properly.

What the HECK!?

# Solution #3 – Just for Looks



```
| fieldformat Count_of_hits=tostring(Count_of_hits,"commas")." hits"
```

Instead of using eval, use the fieldformat command. It will change the appearance of the field without modifying the underlying value.


<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Fieldformat>

# Scenario #4

Your web server logs have these fields:  
UserName, Product, Value, Region, City, IPAddress.

How can you easily report how many products are  
being purchased per region?

# Solution #4 – Stats+Eval to the Rescue!



```
index=my_index sourcetype=my_sourcetype
| fields Region Product
| stats count(eval(Region=="APAC")) AS APAC
count(eval(Region=="EMEA")) AS EMEA
count(eval(Region=="AMER")) AS AMER
by Product
```

} Keep only the fields we need

} Use stats to count the number of events where the Region field equals a specific value. Label those counts as APAC, EMEA, or AMER.

} Group the regional counts by the values of the Product field


<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Stats>

# Scenario #5

You have a new tool to monitor activity on your website. You want to compare it's logs with the current tool's logs but the field names are different...  
How can you easily create an apples to apples search?



# Solution #5 – Coalesce...



```
(sourcetype="old_tool" OR sourcetype="new_tool")
(source_ip=* AND message=*) OR (ip=* AND
messageType=*) =*)
| eval Key1 = (source_ip + "_" + message)
| eval Key2 = (ip + "_" + messageType)
| eval KEY=coalesce(Key1,Key2)
```

} Return the two fields in each sourcetype

} Create key fields for each sourcetype

} Create a master key field to use for reporting.

[http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Eval#9.\\_Coalesce\\_a\\_field\\_from\\_two\\_different\\_source\\_types.2C\\_create\\_a\\_transaction\\_of\\_events](http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Eval#9._Coalesce_a_field_from_two_different_source_types.2C_create_a_transaction_of_events)

# Scenario #6

My month column contains string values  
(Jan, Feb, Mar, etc...)

How do I sort these chronologically  
instead of alphabetically?

# Solution #6 – Let me count the days



```
index=myData
```

```
| eval month_num=strftime(_time,"%m") | sort month_num
```

Store the output of this eval command into a new field called month\_num

Take the epochtime value in the \_time field and return the Month number

Sort by the numerical value of our new month\_num field

[http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/CommonEvalFunctions#Date\\_and\\_Time\\_functions](http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/CommonEvalFunctions#Date_and_Time_functions)

# Scenario #7

I have a table of items,  
how can I add line numbers?

# Solution #7 – One, Two, Three, Four...



index=myData

```
| eval No=1 | accum No | table No, first_name, last_name, state
```

Create a new field called No and set the value to 1

Use the accum command to increment the value of No for each result

Add the new field to my table

No	first_name	last_name	state
1	Judy	Burton	NC
2	Robin	Henderson	FL
3	Douglas	Henderson	TX

<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Accum>



.conf2015

# Getting' Tricky With It

splunk>

# Scenario #1

I need to join multiple  
sourcetypes by a common field



# Solution #1 – Getting Values



```
index=_internal sourcetype=splunkd OR sourcetype=scheduler  
| stats values(user) AS user values(group) AS group values(run_time) AS run_time by date_hour
```

Values returns  
all of the  
distinct values  
of the field  
specified

Return the values  
of the existing  
user field and call  
the resulting field  
user

Group all of  
the previous  
fields by the  
date\_hour field

- Use `| stats values(<field name>)` -or- `| stats list(<field name>)` instead of `| join`
  - `values()`: returns distinct values in lexicographical order
  - `list()`: returns all values and preserves the order

<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/CommonStatsFunctions>

# Scenario #2

How can I email these sweet results to my coworker super fast?

# Solution #2 – You’ve Got Mail



```
index=myData result_type=sweet  
| sendemail to="godfrey@splunk.com" subject="Sweet Results" sendresults=true
```

Send your  
results out via  
the configured  
email provider

There are several parameters  
available, with the “to” parameter  
being required



<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Sendemail>

# Scenario #3

How can I plot firewall attacks on a map with bubbles the grow and change color based on the # of attacks?

# Solution #3 – Hack the Map



```
index=security sourcetype=firewall
```

```
| iplocation clientip
```

```
| where isnotnull(Country)
```

```
| geostats count as PLOT
```

```
| eval redCount=if(PLOT >= 500,PLOT,0)
```

```
| eval yellowCount=if(PLOT >= 100 AND PLOT < 500,PLOT,0)
```

```
| eval greenCount=if(PLOT < 100,PLOT,0)
```

```
| fields - PLOT
```

Lookup the location info for the IP

Filter out results without a Country

Count the results and format for a map

Create three new range fields for each location on the map.

Get rid of the PLOT field, we don't want it on the map

```
| eval redCount=if(PLOT >= 500,PLOT,0)
```

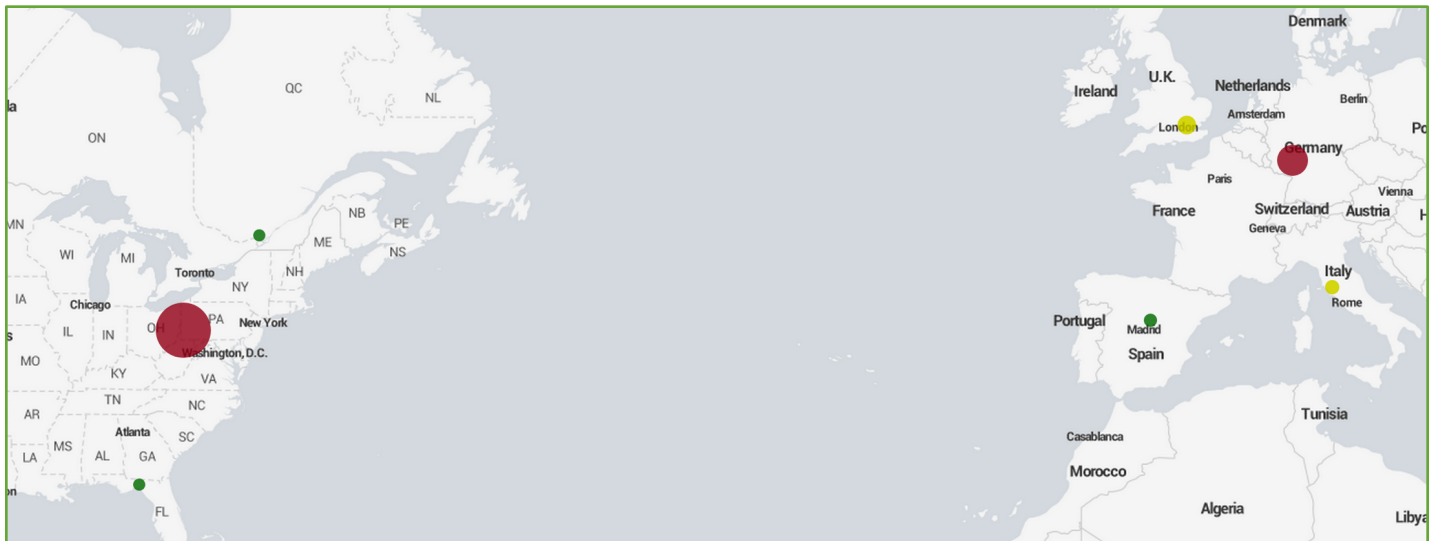
Holds the count for red values at this map location

If the value of PLOT at this map location is greater than or equal to 500, then set the value of redCount at this map location to the value of PLOT. If it is less than 500, set redCount to 0, as the count at this location is not a value that should be red.

# Solution #3 – Hack the Map



```
index=security sourcetype=firewall | iplocation clientip | where isnotnull(Country) | geostats count as PLOT | eval  
redCount=if(PLOT >= 500,PLOT,0) | eval yellowCount=if(PLOT >= 100 AND PLOT < 500,PLOT,0) | eval  
greenCount=if(PLOT < 100,PLOT,0) | fields - PLOT
```



To make the dots specific colors, you'll need to add the following line to your dashboard panel XML:

```
<option name="mapping.fieldColors">{"redCount":0x8f0017,"yellowCount":0xCCCC00, "greenCount":0x006700}</option>
```



.conf2015

# Finding Beauty on the Inside

splunk>



# Scenario #1

How can I look at the volume of data my  
Universal Forwarders are sending?

# Solution #1 – Turn up the Volume



```
index=_internal source=*metrics.log group=tcpin_connections  
| eval Forwarder=coalesce(sourceHost,hostname)  
| timechart limit=500 span=1d sum(kb) as DailyKB by Forwarder
```


- Look at Splunk's internal metrics log
- Set Forwarder value to sourceHost or hostname
- Timechart the average daily KB by forwarder per day

## Scenario #2

That looks good, but how can I compare today's volume to last week?

# Solution #2 – If I could turn back time

Search Time Range: 1 Week



```
index=_internal source=*metrics.log group=tcpin_connections
| eval Forwarder=coalesce(sourceHost,hostname)
| bucket span=1d _time
| stats sum(kb) AS DailyKB by Forwarder _time
| stats earliest(DailyKB) AS EarliestVolume latest(DailyKB) AS
LatestVolume by Forwarder
| eval PercChange=round((((LatestVolume-
EarliestVolume)/EarliestVolume)*100,2)
```

- Look at Splunk's internal metrics log
- Set Forwarder value to sourceHost or hostname
- Bucket the results by day
- Sum the kb field grouped by Forwarder
- Get the DailyKB value from last week and today for each Forwarder
- Calculate the % change, round to two decimal places, and put the value in the PercChange field

# Scenario #3

How can I quickly see details about the hosts,  
sourcetypes, and sources  
in an index?

# Solution #3 – That's so META



```
| metadata type=sourcetypes index=MyIndex
```

The metadata command searches summary data instead of the raw events...so its really fast!

Specify the index and type of data you'd like to return. Splunk keeps summary information on the sources, sourcetypes, and hosts in every index.

firstTime ↕	lastTime ↕	recentTime ↕	sourcetype ↕	totalCount ↕	type ↕
1438127979	1440781315	1440780386	Kepware	66196	sourcetypes
1438127978	1440780456	1440780456	Perfmon:MSExchange Throttling	7692	sourcetypes
1438127681	14407771072	1440770446	WinEventLog:Security	1020	sourcetypes
1438127974	1440781143	1440780451	access_combined	108200	sourcetypes
1274457218	1440781131	1440780356	cisco:asa	21619	sourcetypes
1438128130	1440780999	1440780357	cisco:fwsm	441300	sourcetypes
1392026571	1440780926	1440780207	cisco:pix	19830	sourcetypes
1438127876	1440781180	1440780414	git-json	1313	sourcetypes


<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Metadata>

# Scenario #4

What if I want to see detail about all of the sourcetypes in an index?



# Solution #4– Sourcetype Detail



```
| metadata type=sourcetypes index=myIndex  
| eval current_time = now()  
| eval seconds_since_last_event=(current_time - lastTime)  
| convert ctime(lastTime) as LastTime  
| convert ctime(firstTime) as FirstTime  
| convert ctime(recentTime) as RecentTime  
| convert ctime(current_time)  
| eval hours_since_last_event= round(seconds_since_last_event/(60*60),3)  
| table sourcetype seconds_since_last_event hours_since_last_event  
RecentTime FirstTime LastTime totalCount
```

- Get the metadata for sourcetypes
- Set current\_time to the current time
- Calculate the seconds since the last event
- Format the time so we can read it
- Convert seconds to hours
- Create a table with the fields we want

sourcetype	seconds_since_last_event	hours_since_last_event	RecentTime	FirstTime	LastTime	totalCount
Kepware	3794218	1053.949	07/29/2015 01:01:15	07/28/2015 23:59:39	07/29/2015 01:13:13	6373
Perfmon:MSExchange Throttling	3794968	1054.158	07/29/2015 01:00:43	07/28/2015 23:59:38	07/29/2015 01:00:43	744
WinEventLog:Security	3791435	1053.176	07/29/2015 01:00:09	07/28/2015 23:54:41	07/29/2015 01:59:36	255
access_combined	3794155	1053.932	07/29/2015 01:01:33	07/28/2015 23:59:34	07/29/2015 01:14:16	10210


<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Metadata>

# Scenario #5

Nice!

What if I wanted to determine if there is a delay indexing events from my hosts?

# Solution #5– Keep up the pace



```
| metadata type=hosts
| eval seconds_lag=(recentTime - lastTime)
| convert ctime(lastTime) as LastTime
| convert ctime(firstTime) as FirstTime
| convert ctime(recentTime) as IndexTime
| rangemap field=seconds_lag low=0-1800 elevated=1801-3600
| fields host,FirstTime,LastTime,IndexTime,seconds_lag,range
```

- Get the metadata for hosts
- Calculate the difference between the timestamp on the latest event and the time it was indexed
- Format the time so we can read it
- Assign a severity based on the lag
- Filter down to the fields we want


host	FirstTime	LastTime	IndexTime	seconds_lag	range
127.0.0.1	05/21/2010 09:38:12	07/29/2015 01:02:42	09/10/2015 23:03:51	3794469	elevated
network_span-01	12/29/2014 08:01:19	07/29/2015 01:10:04	09/10/2015 20:36:19	3785175	low
websphere-01	09/03/2010 19:17:18	07/29/2015 01:09:46	09/10/2015 23:04:12	3794066	low
websphere-02	09/03/2010 19:17:18	07/29/2015 01:09:45	09/10/2015 23:05:06	3794121	elevated
websphere-03	09/03/2010 19:17:18	07/29/2015 01:09:31	09/10/2015 23:04:57	3794126	elevated

<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Metadata>

# Scenario #6

How can I get a list of all indexes?

# Solution #6 – Take a REST



```
| rest /services/data/indexes  
| search eai:acl.app=my_app title=*  
| stats sum(totalEventCount) AS totalEventCount by title
```

- Query the indexes REST endpoint
- Limit to the app context my\_app and return all indexes
- Calculate the total event count per index

title	totalEventCount
_audit	21938
_blocksignature	0
_internal	321475
_introspection	42002
_thefishbucket	0
history	0


<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/Rest>

# Scenario #6

Awesome!  
Can I see anything else cool?

# Solution #6 – Take a REST


## Lookup Tables:



```
| rest /services/data/lookup-table-files  
| search eai:acl.app=fantastic_lookups*  
| dedup title | rename title as LookupTable | table LookupTable
```

- } Query the lookups REST endpoint
- } Limit to my app context
- } Dedup the list and create a table

## Dashboards:



```
| rest /servicesNS/-/-/data/ui/views  
| search eai:acl.app=my_awesomeApp  
| table title
```

- } Query the views REST endpoint
- } Limit to my app context
- } Create a table

<http://docs.splunk.com/Documentation/Splunk/6.2.5/SearchReference/Rest>

# Resources

- Search Command Reference  
<http://docs.splunk.com/Documentation/Splunk/latest/SearchReference>
- Splunk Answers  
<http://answers.splunk.com/>
- Smart Answers Blog – Patrick Pablo, Community Content Manager  
<http://blogs.splunk.com/2014/11/24/smart-answers>
- Splunk Book – Exploring Splunk  
<http://www.splunk.com/goto/book>



# What Now?

Related breakout sessions and activities...

## Today

- 4:15 - Beyond the Lookup Glass – Room 320
- 5:15 – Search Efficiency Optimization – Room 320

## Tomorrow

- 10:00 & 5:15 – Getting Started with Maps – Room 318
- 11:15 – Building Powerful Analytics with Ease – Room 113
- 3:15 – Search Efficiency Optimization – Room 318
- 4:15 – SPLing Bee – Community Theatre Marquee Ballroom



.conf2015

THANK YOU

splunk>