

Splunk Intermediate/Advanced

Course Outline

Module 1: Course Introduction

Module 2: Beyond Search Fundamentals

Module 3: Commands for Visualizations

Module 4: Advanced Visualizations

Module 5: Filtering and Formatting Data

Module 6: Correlating Events

Module 7: Introduction to Knowledge Objects

Module 8: Creating and Managing Fields

Module 9: Creating Field Aliases and Calculated Fields

Module 10: Creating Tags and Event Types

Module 11: Creating and Using Macros

Module 12: Creating and Using Workflow Actions

Module 13: Creating Data Models

Module 14: Using the Common Information Model (CIM) Add-On

Buttercup Games, Inc.

- Multinational company with HQ in San Francisco and offices in Boston and London
- Sells product mainly through its worldwide chain of third party stores, but also sells through its online store



Your Role at Buttercup Games

- You are a Splunk power user with a great understanding of all your company's data
- Your responsibilities are to provide information to users throughout the company and to create and manage Splunk knowledge objects for your stakeholders
- You implement best practices for naming conventions of all knowledge objects
- You gather data and statistics, and report on Security, IT Operations, Operational Intelligence, etc.

Review Search Fundamentals

Basic Search Review

- **Keywords**

For example, search for a single word (e.g., error) or group of words (e.g., error password)

- **Booleans**

NOT, OR, AND; AND is implied; MUST be uppercase; can use ()'s to force precedence

`sourcetype=vendor_sales OR (sourcetype=access_combined
action=purchase)`

- **Phrases**

"web error" (different than web AND error)

- **Field searches**

`status=404, user=admin`

- **Wildcard (*)**

– `status=40*` matches 40, 40a, 404, etc.

– Starting keywords with a wildcard is very inefficient, e.g. `*dmin`

- **Comparisons**

`=, !=, <, <=, >=, > status>399, user!=admin`

Basic Search Review (cont.)

- `table`: returns table containing only specified fields in result set
- `rename`: renames a field in results
- `fields`: includes or excludes specified fields
- `dedup`: removes duplicates from results
- `sort`: sorts results by specified field
- `lookup`: adds field values from external source (e.g., csv files)

Case Sensitivity – Sensitive

Case sensitive	Examples
Boolean operators (uppercase)	AND, OR, NOT (Boolean operators) and, or, not (literal keywords)
Field names	productId vs. productid eval cs_username = "Total Access"
Field values from lookup (default, but configurable)	product_name="Tulip Bouquet" vs. product_name="tulip bouquet"
Regular expressions	\d\d\d vs. \D\D\D
eval and where commands	eval action;if(action=="view",...) where action="Purchase" stats count(eval(action="view")) as...
Tags	tag=DMZ vs. tag=dmz

Case Sensitivity – Insensitive

Case insensitive	Examples
Command names	STATS, stats, sTaTs
Command clauses	AS used by stats, rename, ...; BY used by stats, chart, top, ...; WITH used by replace
Search terms	failed, FAILED, Failed
Statistical functions	avg, AVG, Avg used by stats, chart, ...
Field values	host=www1, host=WWW1 (unless coming from a lookup)

How Splunk Searches – Buckets

- As events come in, Splunk places them into an index's hot bucket (only writable bucket)
- As buckets age, they roll from the hot to warm to cold
- Each bucket has its own raw data, metadata, and index files
- Metadata files track source, sourcetype, and host
- Admins can add more



How Splunk Searches – Searching

- When you search, Splunk uses the time range to choose which buckets to search and then uses the bucket indexes to find qualifying events
- When you search for
index=web password
fail* during the last 24 hours:
 - Splunk identifies the buckets for the last 24 hours
 - And searches the indexes of those buckets for the search terms

Hot: Now to -3h	index	raw events
Hot: -3 to -6h	index	raw events
Hot: -5 to -8h	index	raw events
Warm: -9 to -12h	index	raw events
Warm: -12 to -15h	index	raw events
Warm: -14 to -17h	index	raw events
.....	index	raw events
Warm: -42 to -45h	index	raw events
Warm: -45 to -48h	index	raw events
Cold: -48 to -51h	index	raw events
Cold: -51 to -54h	index	raw events
Cold: -54 to -57h...	index	raw events

General Search Practices

- As events are stored by time, time is the most efficient filter
- After time, most powerful keywords are host, source, sourcetype
- To make searches more efficient, include as many terms as possible
 - e.g., searching for sourcetype=x failure is better than failure
- Use the fields command to extract (discover) only fields you need
- Example: Search last 365 days, scans 566,720 events (in secs):

index=web sourcetype=access_combined	15.16
index=web sourcetype=access_combined fields clientip bytes referrer	4.49

General Search Practices – Wildcards

- Splunk only searches for whole words, but wildcards allowed
- Only *trailing* wildcards make efficient use of index
 - Wildcards at *beginning* of string scan all events within time frame
 - Wildcards in *middle* of string may return inconsistent results
 - So use fail* (not *fail or *fail* or f*il)
- Wildcards tested after all other terms

General Search Practices

- Inclusion is generally better than exclusion
 - Searching for "access denied" is faster than NOT "access granted"
- Filter as early in your search as possible
 - Removing duplicates then sorting is faster than sorting then removing duplicates
- Use the appropriate search mode
 - Fast - performance over completeness
 - Smart [default]
 - Verbose - completeness over performance

Transforming Search Commands

- A transforming command:
 - Massages raw data into a data table
 - 'Transforms' specified cell values for each event into numerical values that you can use for statistical purposes
 - Is required to 'transform' search results into visualizations
- Transforming commands include:
 - top
 - rare
 - chart
 - timechart
 - stats
 - geostats

Reviewing Search Mode – Fast Mode

- Emphasizes performance, returning only essential and required data
- For non-transforming searches:
 - ✓ Events – fields sidebar displays only those fields required for the search
 - ✓ Patterns
- Contents of interesting fields sidebar are lost

Reviewing Search Mode – Smart Mode (Default)

- Designed to give you the best results for your search
- Combination of Fast and Verbose modes
- For non-transforming searches:
 - ✓ Events – fields sidebar displays all fields
 - ✓ Patterns
- For transforming searches:
 - ✓ Statistics or visualizations

Reviewing Search Mode – Verbose Mode

- Emphasizes completeness by returning all possible field and event data
- For non-transforming searches:
 - ✓ Events – fields sidebar displays all fields
 - ✓ Patterns
- For transforming searches:
 - ✓ Events
 - ✓ Patterns
 - ✓ Statistics or visualizations

Search Performance – Modes

- Use the most appropriate search mode:

```
index=web  
sourcetype=access_combined  
| chart count by product_name
```

- Time range: last 365 days

	<u>Returned Results</u>	<u>Events Scanned</u>	<u>Time</u>
	14	566,731	1.82
Smart	14	566,731	1.91
Verbose	14	566,731	15.21

Search Performance – Types of Searches

Search	Description	Indexer throughput
Dense	A large percentage of the data matches the search	Up to 50K matching EPS (Events per second) <i>CPU bound</i>
	Use Cases: computing stats, reporting	
	<code>index=web sourcetype=access_combined timechart count</code>	
Sparse	A small percentage of data matches the search	Up to 5K matching EPS <i>CPU bound</i>
	Use Cases: troubleshooting, error analysis	
	<code>index=web sourcetype=access_combined status=404 timechart count</code>	
Super Sparse	Returns a small number of results from each index bucket matching the search	Up to 2 seconds per index bucket <i>I/O bound</i>
	I/O intensive as the indexer looks through all of an index's buckets	
	With a lot of data, with a lot of buckets, it can take a long time to finish	
	<code>index=web sourcetype=access_combined action=denied src_ip=10.2.13.11</code>	
Rare	The indexer checks all buckets to find results, but bloom filters eliminate those buckets that don't include search results	Up to 10-50 index buckets/second <i>I/O bound</i>
	Use Cases: user behavior tracking	
	<code>index=web sourcetype=access_combined sessionID=1234</code>	

Search Job Inspector

- Tool allows you to examine:
 - Overall stats of search (e.g., records processed and returned, processing time)
 - How search was processed
 - Where Splunk spent its time
- Use to troubleshoot search's performance and understand impact of knowledge objects on processing (e.g., event types, tags, lookups)
- Any existing (i.e., not expired) search job can be inspected

Search Job Inspector – 3 Components

index=web sourcetype=access_combined
| stats count by action

✓ 3,673 events (1/11/18 3:00:00.000 PM to 1/12/18 3:15:47.000 PM) No Event Sampling ▾

Events (3,673) Patterns Statistics (5) Visualization

Format Timeline ▾ List ▾ Format

Job ▾

- Edit Job Settings...
- Send Job to Background
- Inspect Job**
- Delete Job

- Header
- Execution costs
- Search job properties

Search job inspector

This search has completed and has returned 5 results by scanning 3,673 events in 0.778 seconds
(SID: 1515798947.29) [search.log](#)

> Execution costs

> Search job properties

Server Info: Splunk 7.1.0, 34.215.236.159, Fri Jan 12 15:18:14 2018 User: student1

Search Job Inspector – Search Job Properties

Example:



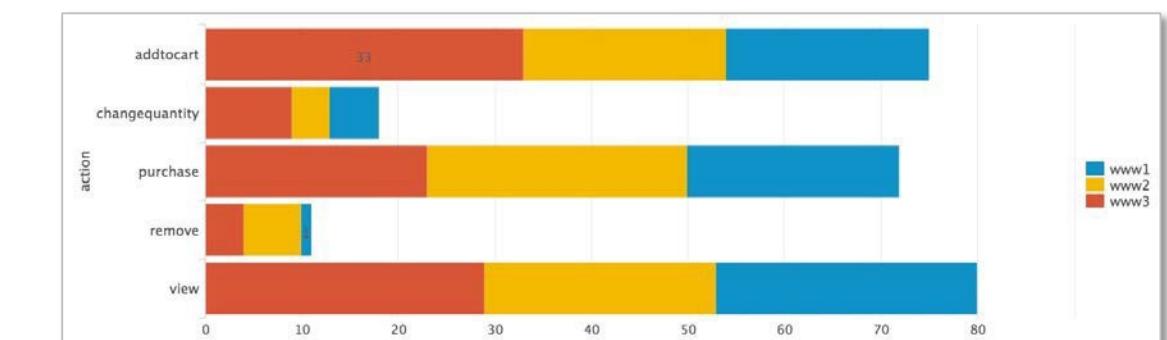
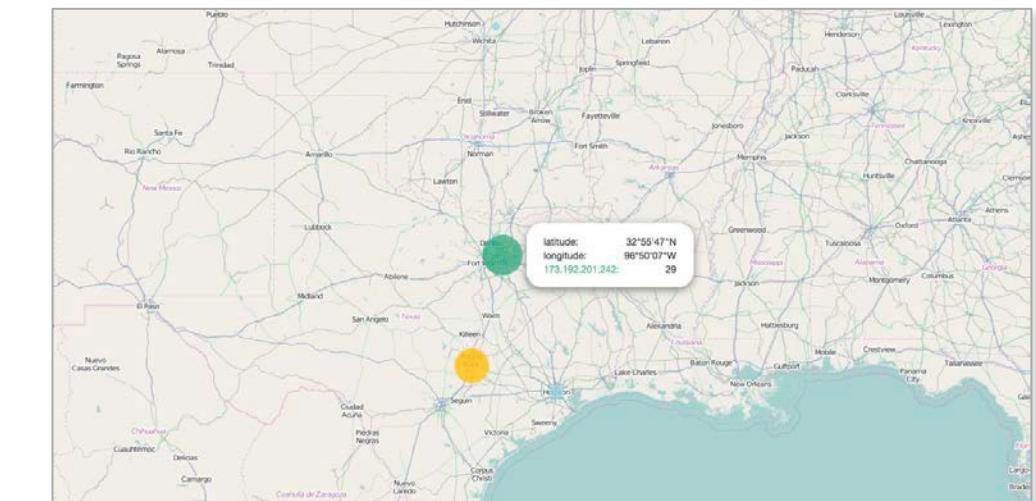
- Produces `scanCount` of **127,201** events
- Returns `resultCount` of **2,144** in 3.01 seconds
- To calculate performance:
 - Do **not** use `resultCount/time` $2,144 / 3.01 = 712$ EPS*
 - Rather, calculate `scanCount/time` $127,201 / 3.01 = 40,892$ EPS

* EPS= events per second

Commands for Visualizations

Visualization Types

- When a search returns statistical values, results can be viewed with a wide variety of visualization types
 - Statistics table
 - Charts: Line, column, pie, etc
 - Single value, gauges
 - Maps
 - Many more



Viewing Results as a Visualization

- Not all searches can be visually represented
- A data series is a sequence of related data points that are plotted in a visualization
- Data series can generate any statistical or visualization results

New Search

index=web sourcetype=access_combined |(404 OR 500 OR 503) OR (error OR fail*) Last 60 minutes

Events (10) Patterns Statistics Visualization

Your search isn't generating any statistic or visualization results. Here are some possible ways to get results.

Pivot
Build tables and visualizations using multiple fields and metrics without writing searches.

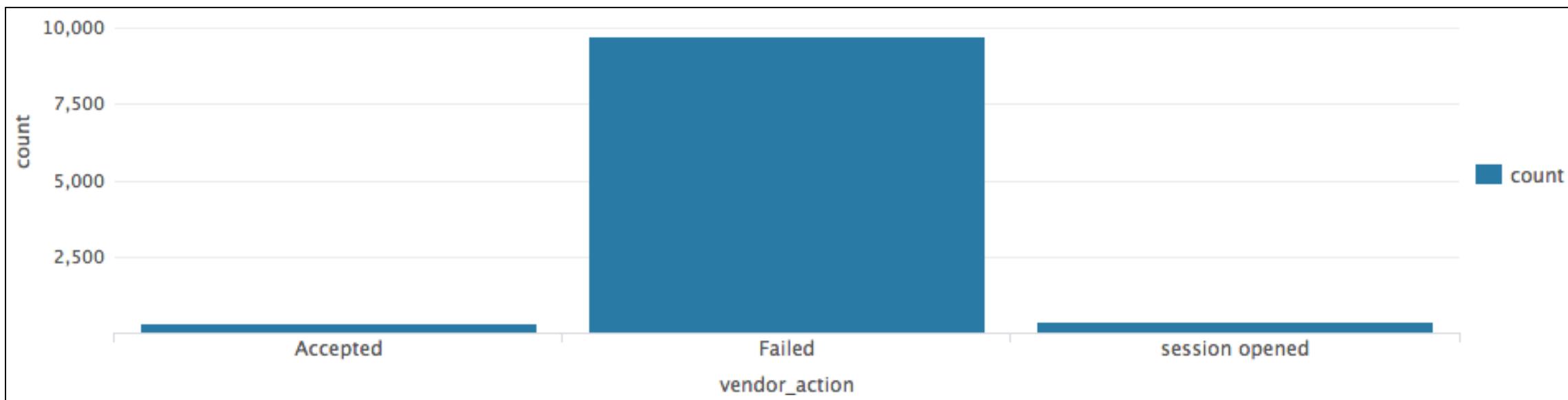
Quick Reports
Click on any field in the events tab for a list of quick reports like 'Top Referrers' and 'Top Referrers by time'.

Search Commands
Use a transforming search command, like timechart or stats, to summarize the data.

Data Structure Requirements – Single Series

- Most visualizations require search results structured as tables, with at least two columns, a **single series**
 - Leftmost column provides x-axis values
 - Subsequent columns provide numeric y-axis values for each series in the chart

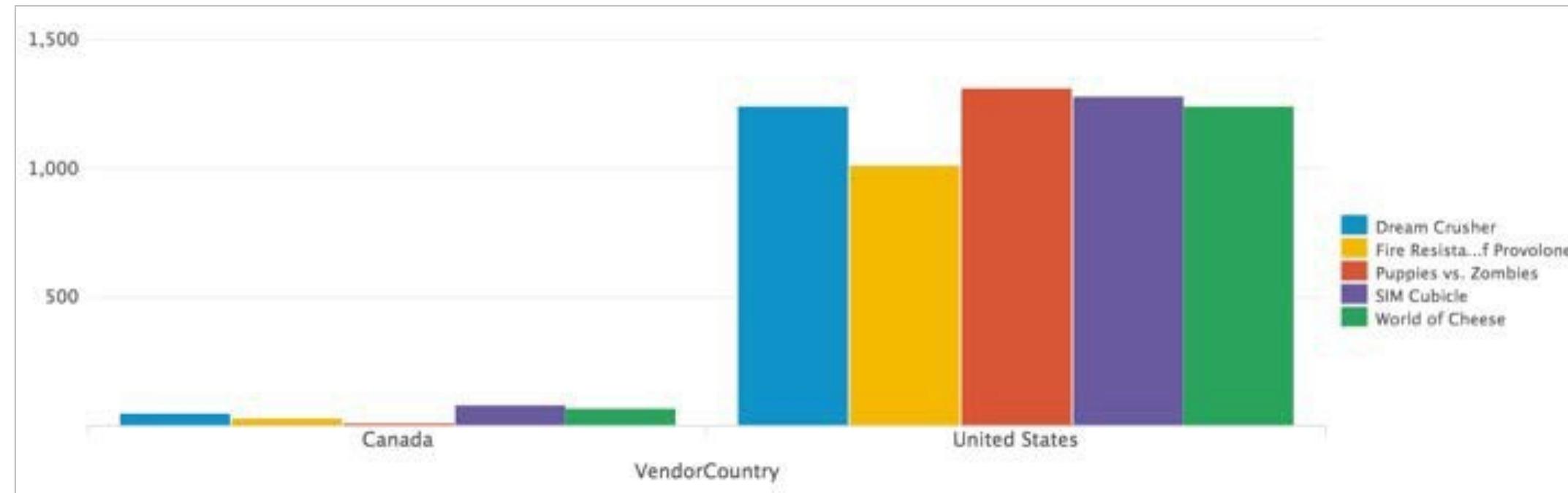
vendor_action	count
Accepted	301
Failed	9740
session opened	400



Data Structure Requirements – Multi-Series

To get **multi-series** tables, you need to set up the underlying search with reporting search commands like `chart` or `timechart`

VendorCountry	Dream Crusher	Fire Resistance Suit of Provolone	Puppies vs. Zombies	SIM Cubicle	World of Cheese
Canada	10	3	2	13	13
United States	230	202	230	228	223



```
index=sales sourcetype=vendor_sales VendorID<4000  
| chart count over VendorCountry by product_name limit=5 useother=f
```

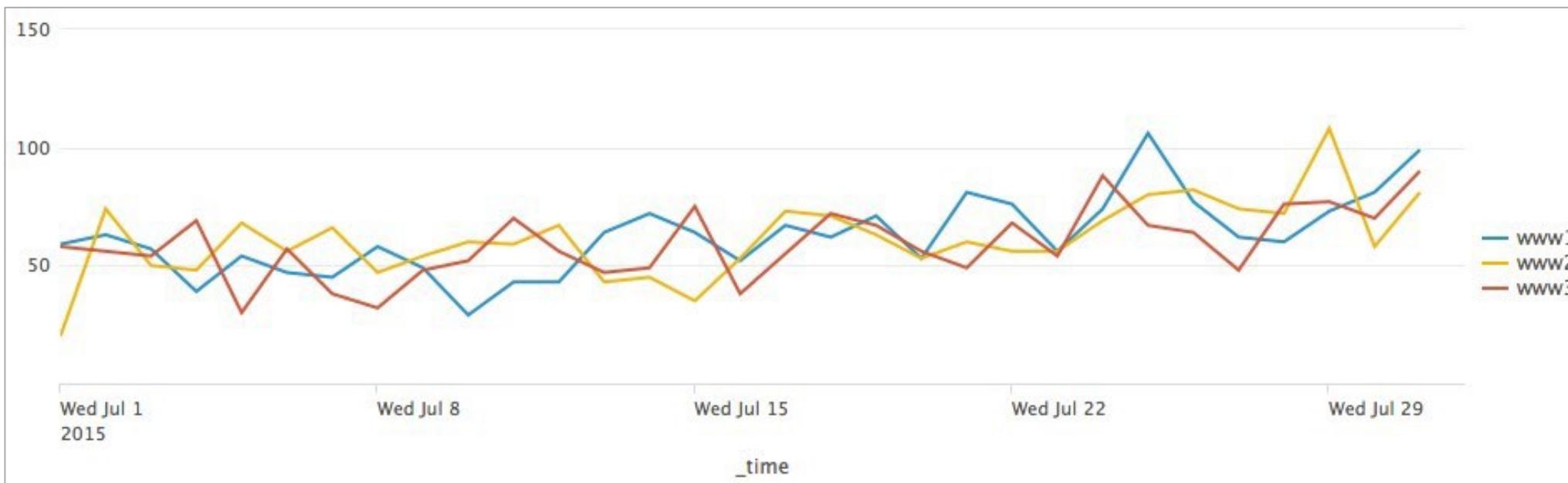
Last 30 days ▾



Data Structure Requirements – Time Series

- **Time series** displays statistical trends over time
- Can be single-series or multi-series

_time	www1	www2	www3
2017-12-01	168	190	158
2017-12-02	156	171	168
2017-12-03	138	150	151
2017-12-04	169	162	195
2017-12-05	135	170	150



```
index=web sourcetype=access_combined action=purchase status=200
| timechart count by host
```

Previous month ▾

Viewing Results as a Chart

- There are seven chart types:
 - Line
 - Area
 - Column
 - Bar
 - Bubble
 - Scatter
 - Pie

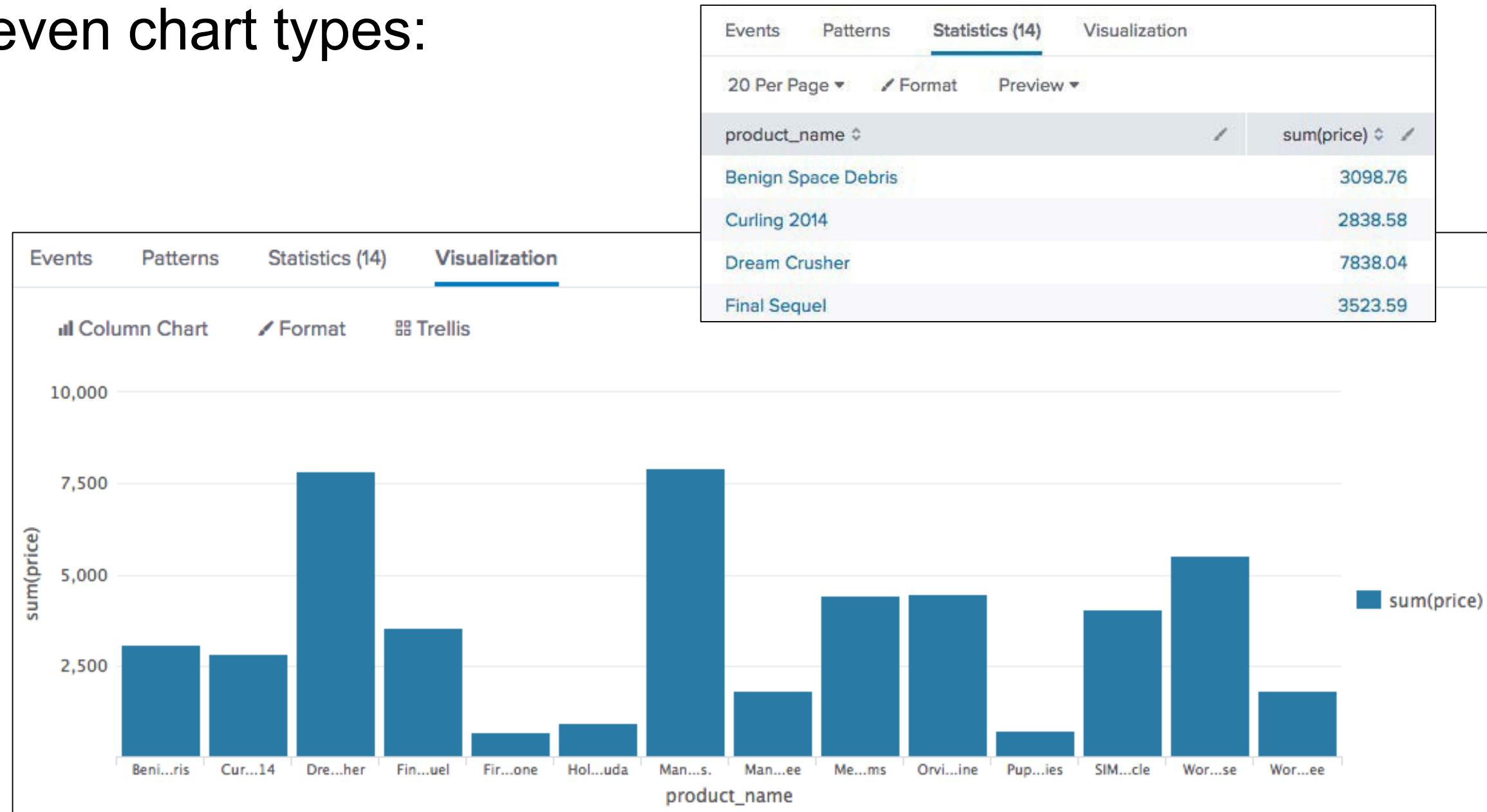


chart Command

- chart command can display any series of data that you want to plot
- You decide which field to plot on the x-axis
 - The function defines the value of the y-axis, therefore it should be numeric
 - The first field after the over clause is the x-axis
 - Using the over and by clauses divides the data into sub-groupings
 - ▶ The values from the by clause display in the legend

chart avg(bytes) over host

- The host values display over the x-axis

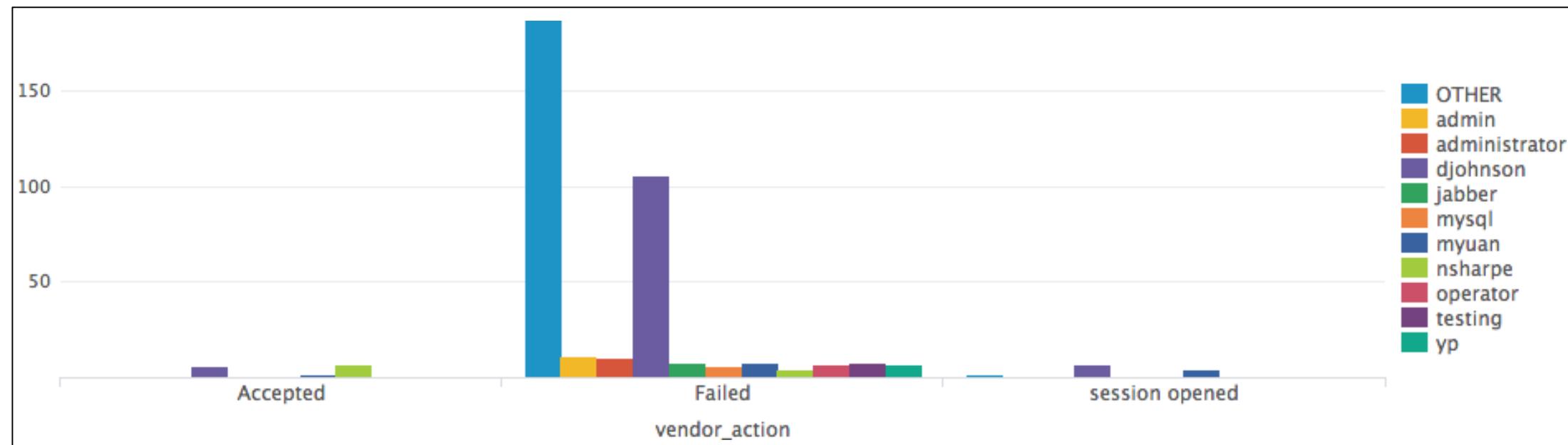
chart avg(bytes) over host by product_name

- The host field is the x-axis and the series is further split by product_name

chart Command – over field by field

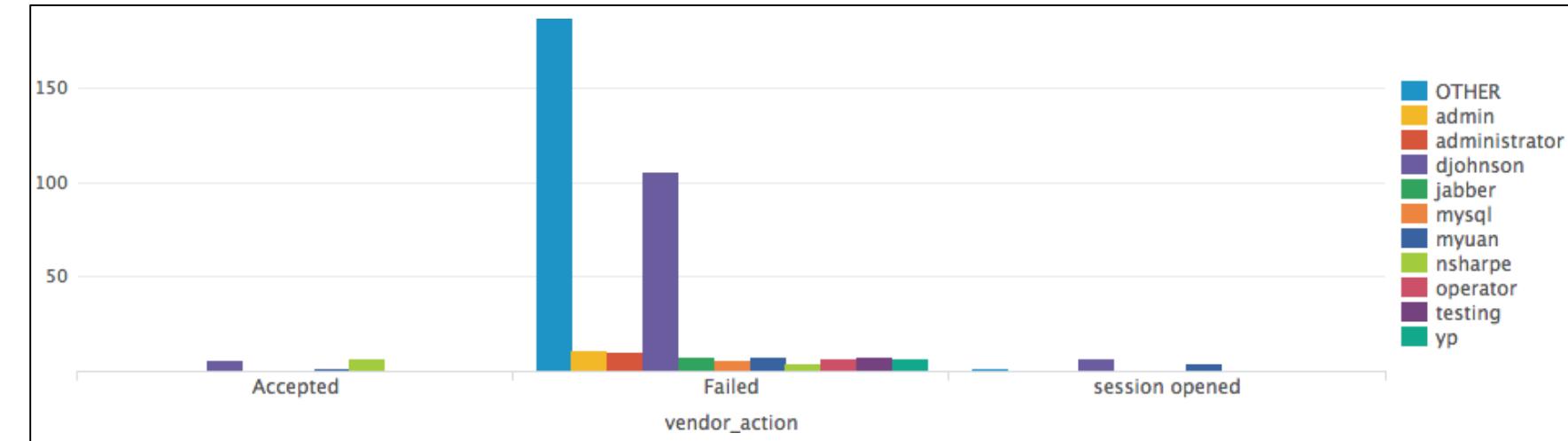
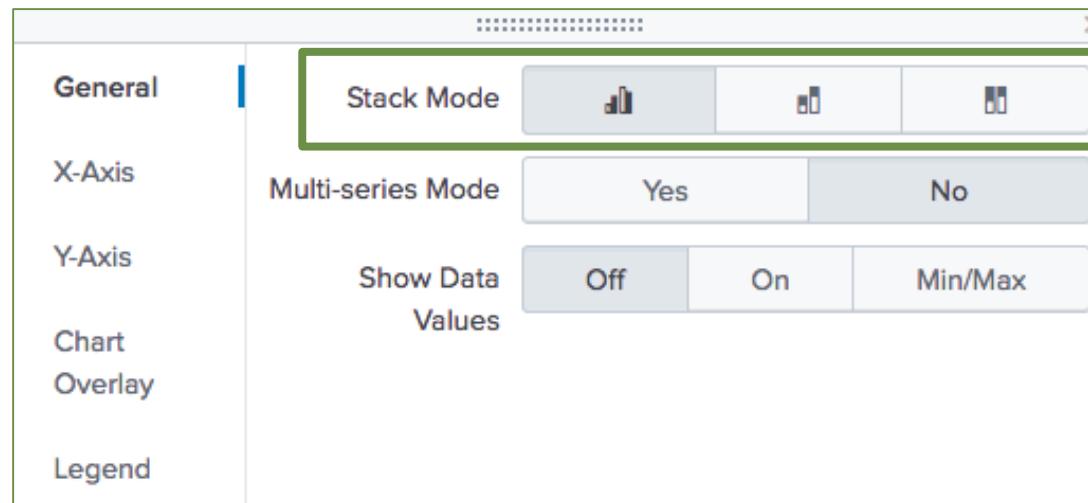
- You can use the `by` clause with the `over` clause to split results (over `vendor_action` by `user`)
- Alternatively, you can just use two `by` clauses (`by vendor_action, user`)
- You can only split chart results over TWO dimensions (unlike stats results)

```
index=security sourcetype=linux secure  
| chart count over vendor_action by user
```

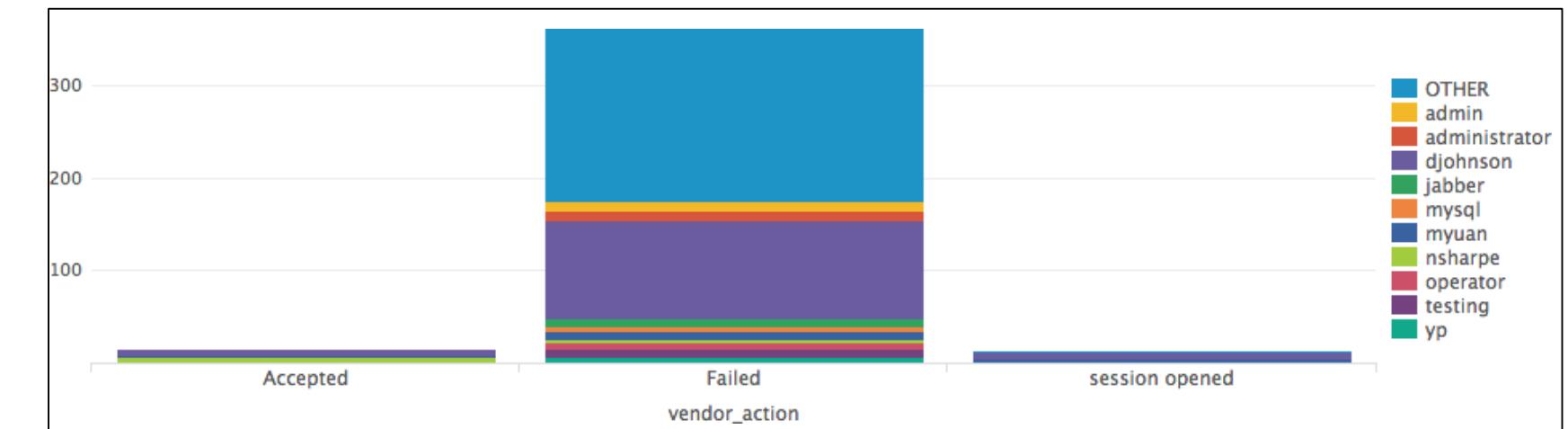


Stack Mode

Stack Mode OFF

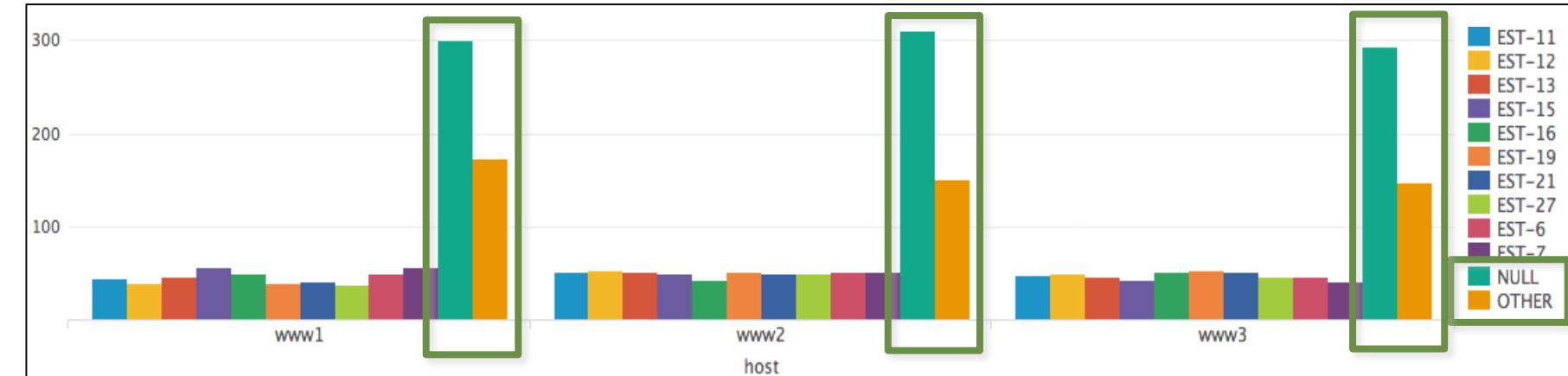


Stack Mode ON

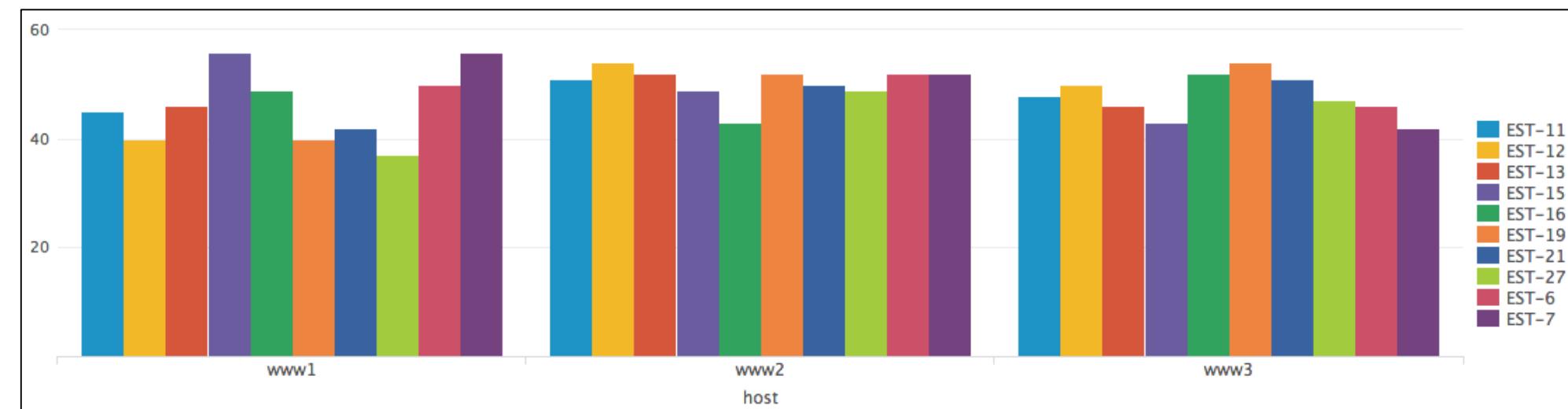


Omitting NULL and OTHER Values

- To remove empty (NULL) and OTHER field values from the display use these options:
 - useother=f
 - usenull=f



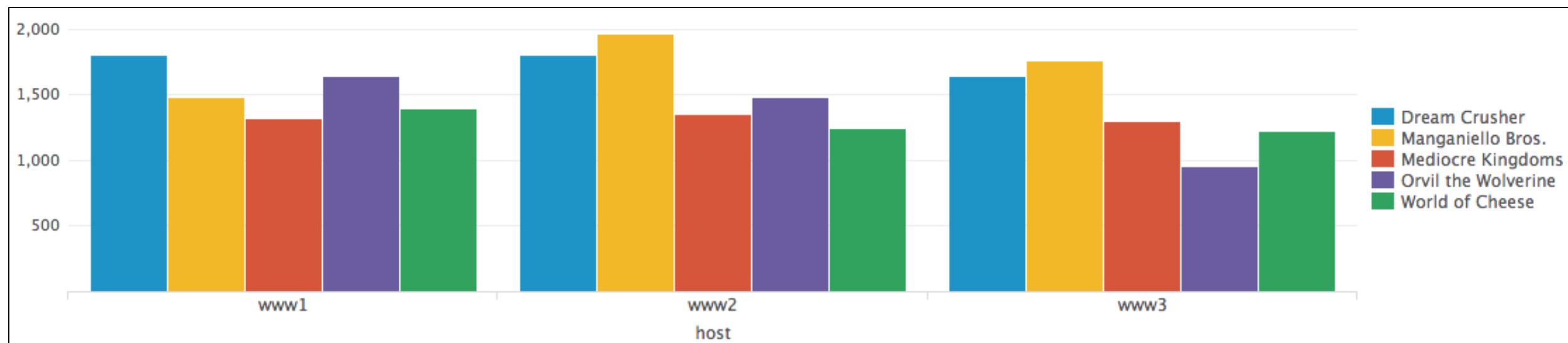
```
index=web sourcetype=access_combined status>399  
| chart count over host by itemId  
useother=f usenull=f
```



Limiting the Number of Values

- To adjust the number of plotted series, use the `limit` argument
- For unlimited values, use `limit=0`

```
index=web sourcetype=access_combined  
action=purchase status=200  
| chart sum(price) over host  
by product_name limit=5 useother=f
```

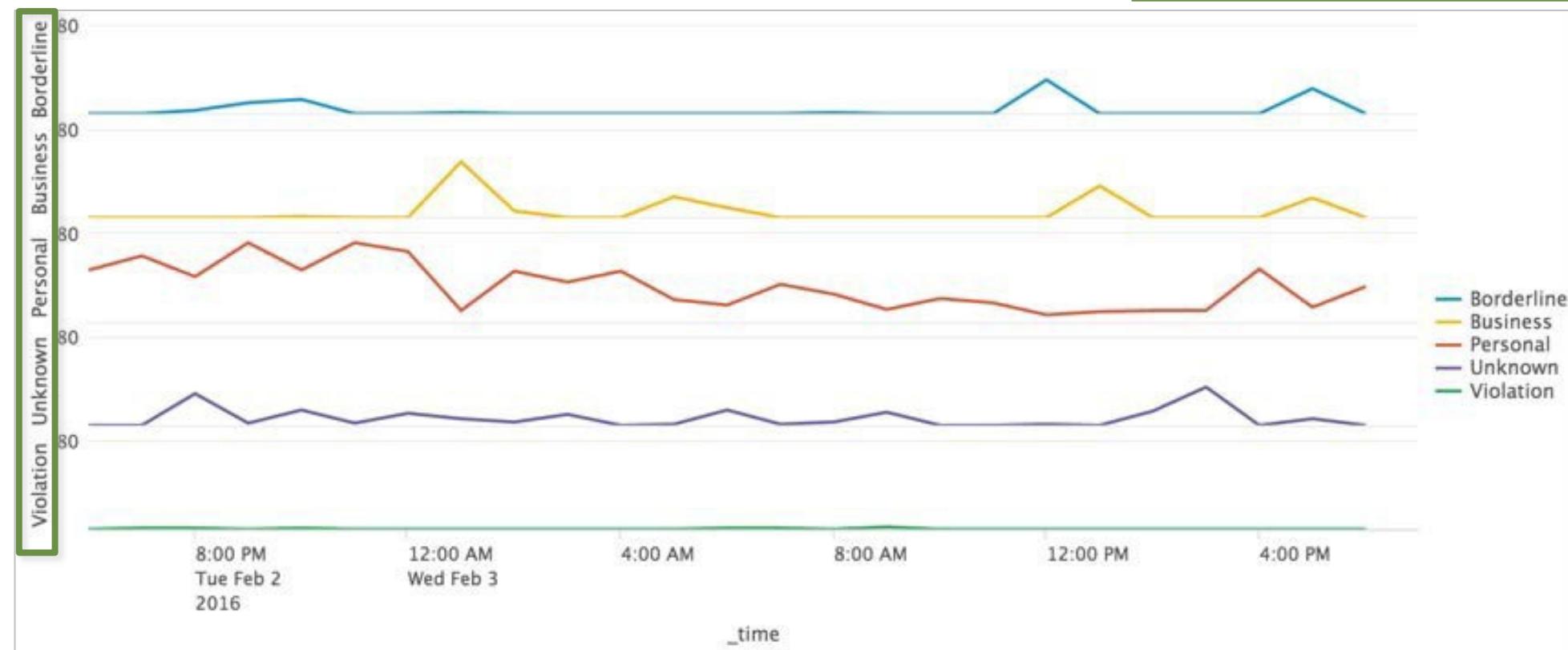
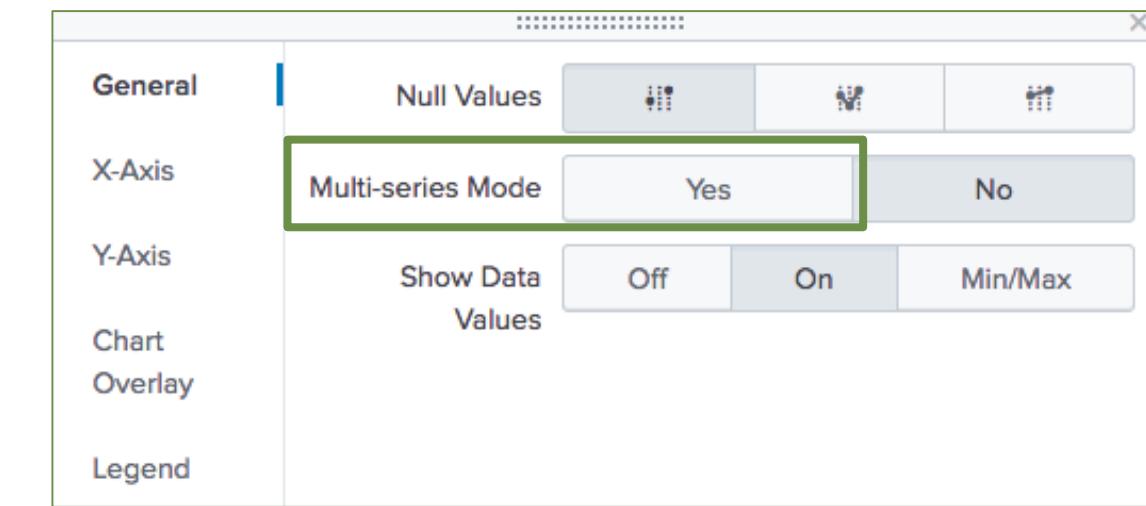


timechart Command – Overview

- timechart command performs statistical aggregations against time
- Plots and trends data over time
- `_time` is always the x-axis
- You can optionally split data using the `by` clause for one other field
 - Each distinct value of the split by field is a separate series in the chart
- Timecharts are best represented as line or area charts

timechart Command – Multi-series: Yes

- Setting multi-series mode to **Yes** causes the y-axis to split for each field value
- y-axis is divided into sections, each spanning the same max and min count



timechart Command – Adjusting the Sampling Interval

- The timechart command "buckets" the values of the _time field
 - This provides dynamic sampling intervals, based upon the time range of the search
- Example defaults:
 - Last 60 minutes uses span=1m
 - Last 24 hours uses span=30m
- Adjust the interval using the span argument, e.g.
span=15m

```
index=security sourcetype=linux_secure  
vendor_action=*
```

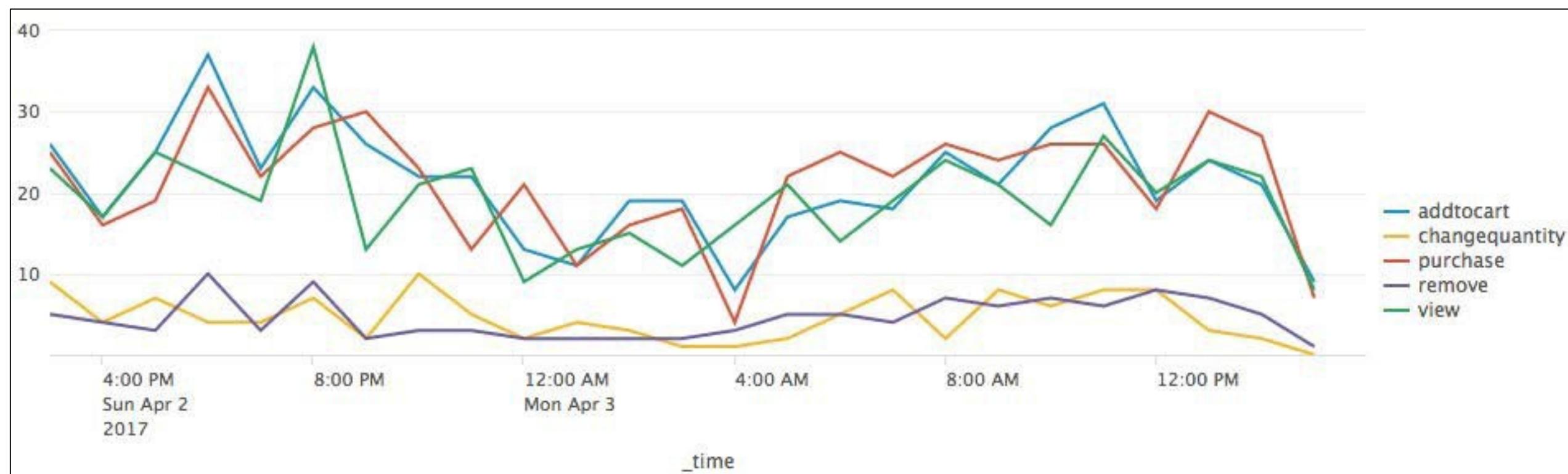
```
| timechart span=15m count by vendor_action
```

_time	Accepted	Failed	session opened
2018-01-16 12:00:00	2	137	3
2018-01-16 12:15:00	2	78	8
2018-01-16 12:30:00	3	61	5
2018-01-16 12:45:00	5	86	4
2018-01-16 13:00:00	3	165	8
2018-01-16 13:15:00	3	65	3
2018-01-16 13:30:00	5	52	7
2018-01-16 13:45:00	2	186	6

timechart Command – Statistical Functions

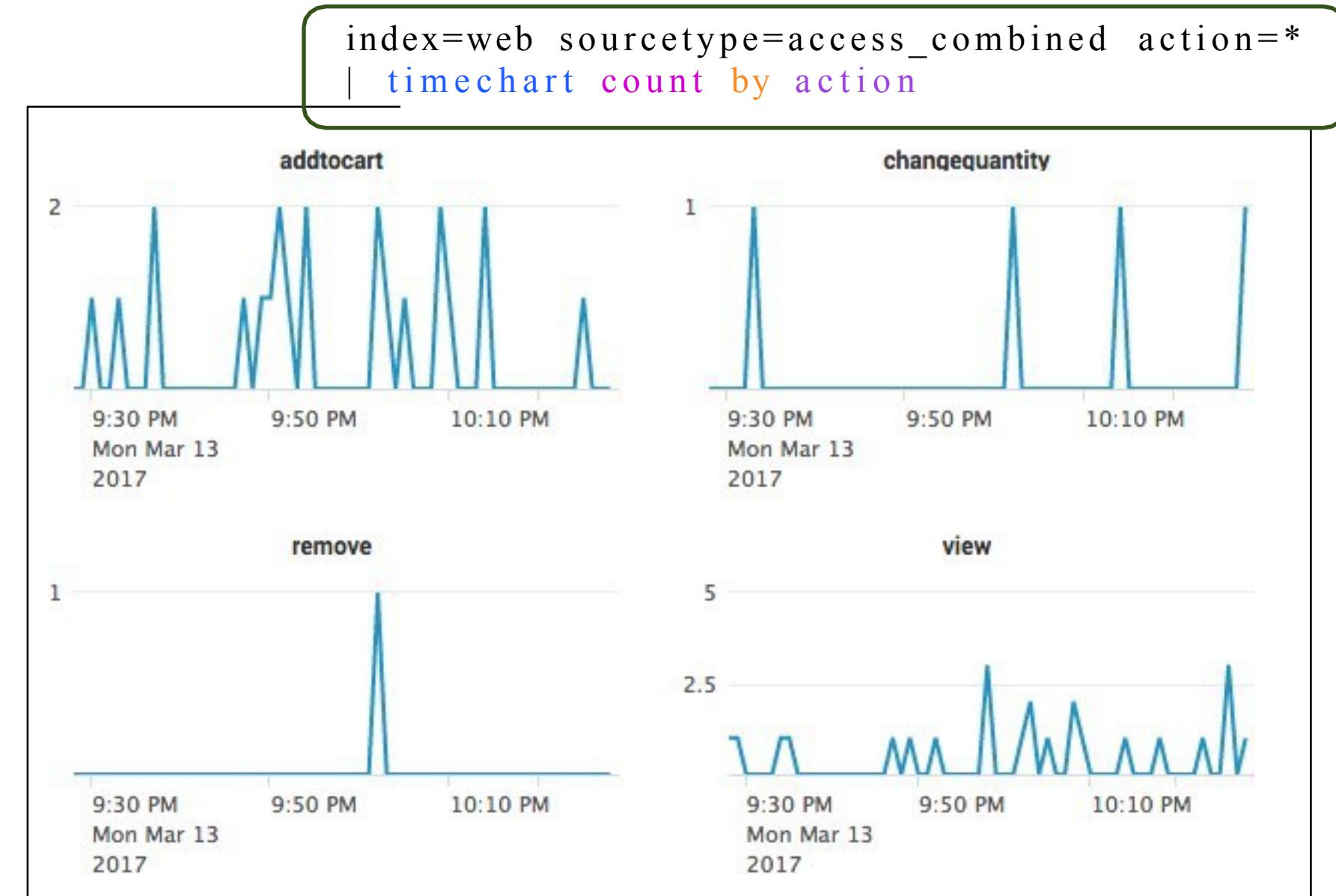
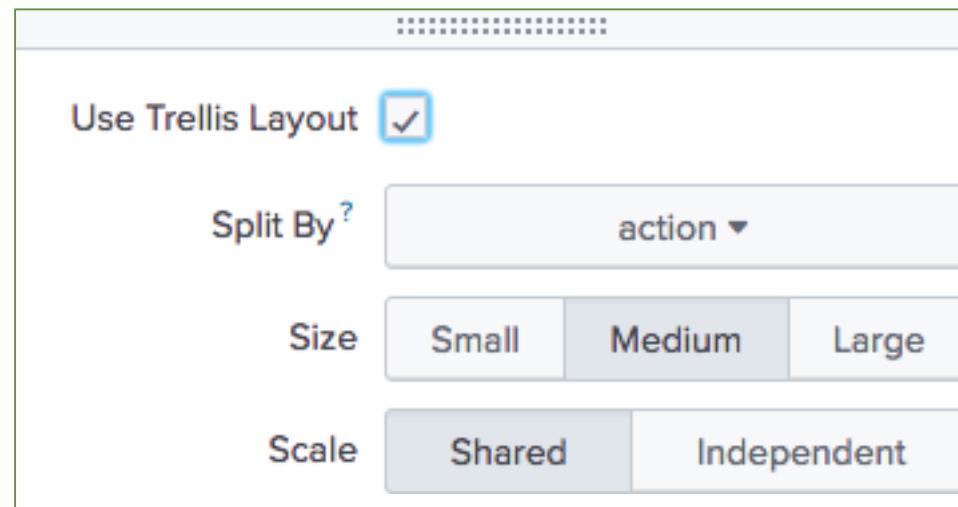
As with the stats and chart commands, you can apply statistical functions to the timechart command

```
index=web sourcetype=access_combined action=*  
| timechart span=1h count by action
```

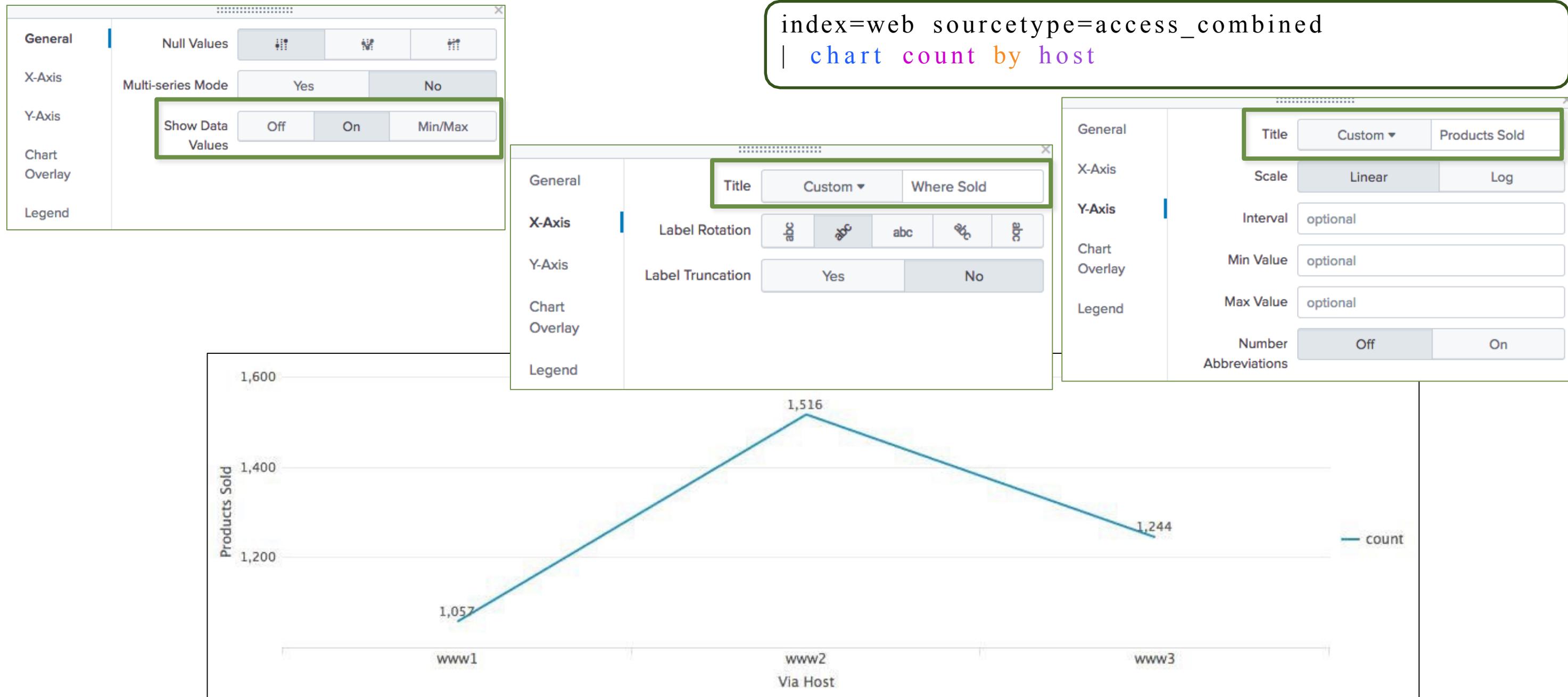


Trellis Layout

- Display multiple charts based on one result set
- Allows visual comparison between different categories
- Data only fetched once

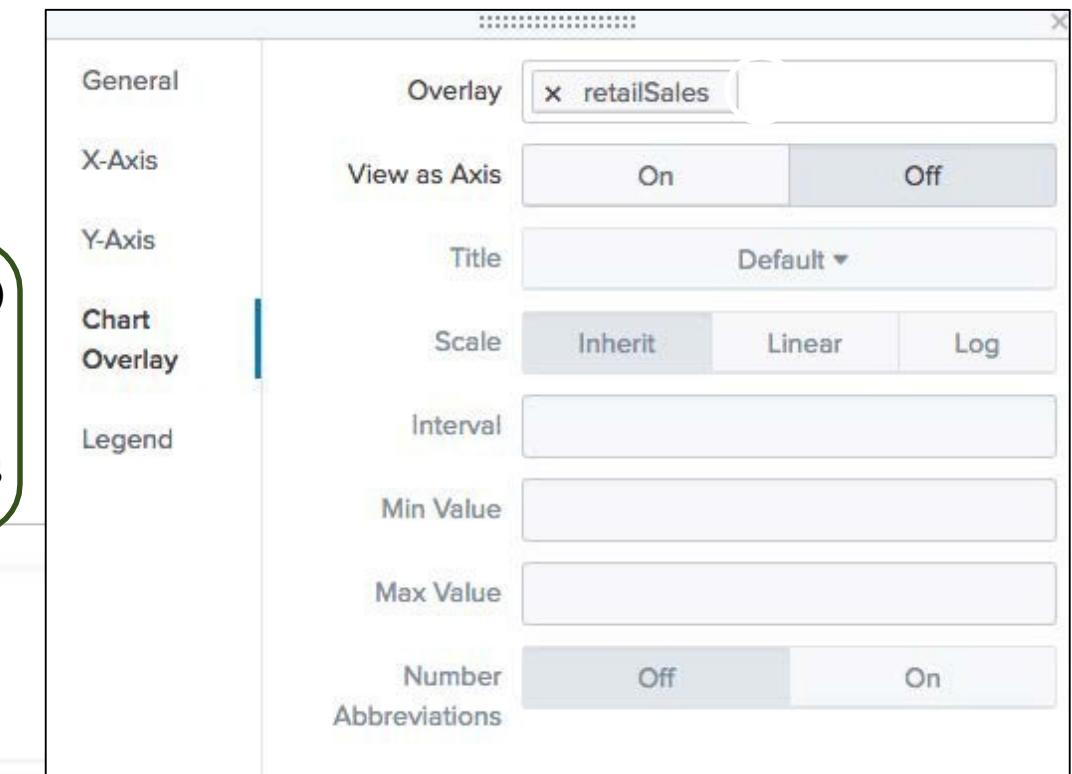


Visualization Formatting



Formatting – Chart Overlay

```
(index=web sourcetype=access_combined action=purchase status<400)
OR (index=sales sourcetype=vendor_sales)
| timechart span=1h sum(price) by sourcetype
| rename access_combined as webSales, vendor_sales as retailSales
```



Transforming Command Summary

Feature	stats	chart	timechart
Multi-level breakdown [by clause]	Many	2	1
Limit # series shown	NA	<code>limit=n</code> <i>Default=10</i>	<code>limit=n</code> <i>Default=10</i>
Filter other series	NA	<code>useother=f</code>	<code>useother=f</code>
Filter null values	NA	<code>usenull=f</code>	<code>usenull=f</code>
Set time value on x axis	NA	NA	<code>span</code>

Transforming Command Summary (cont.)

To count the frequency of a field(s), use top/rare

```
index=security  
sourcetype=linux_secure  
| top src_ip, user, vendor_action
```

src_ip	user	vendor_action	app	count	percent
235.166.61.39	nsharpe	Failed	sshd	122	7.850708
44.248.239.252	myuan	Failed	sshd	113	7.271557
16.201.248.137	nsharpe	Failed	sshd	113	7.271557
43.189.188.26	myuan	Failed	sshd	107	6.885457

Transforming Command Summary (cont.)

Use `stats` to calculate statistics for two or more `by` fields (non time-based)

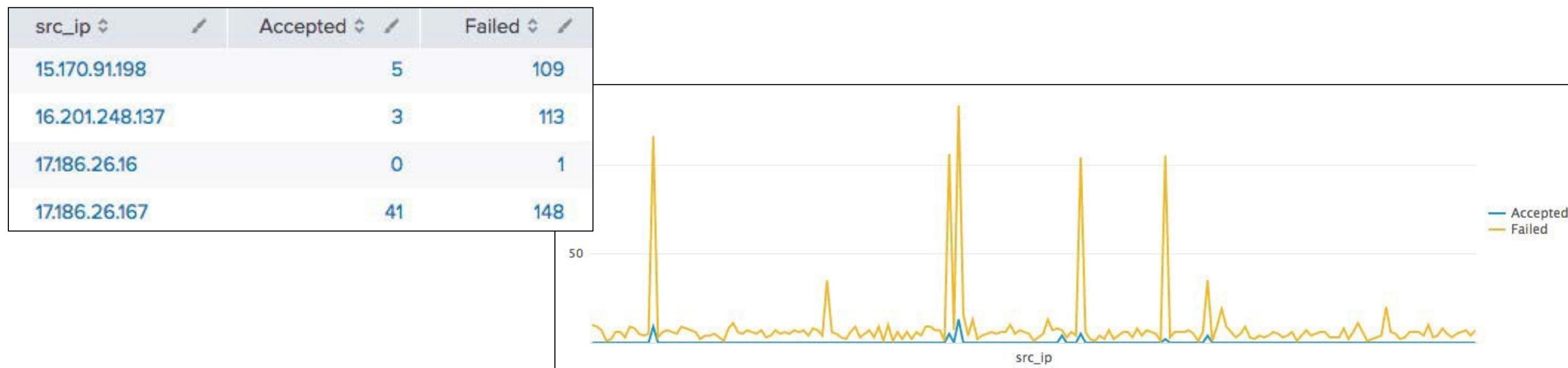
```
index=security sourcetype=linux_secure  
| stats count by src_ip, user, vendor_action, app
```

src_ip	user	vendor_action	app	count
105.131.160.148	djohnson	Failed	sshd	100
107.3.146.207	admin	Failed	sshd	2
107.3.146.207	administrator	Failed	sshd	3
107.3.146.207	angel	Failed	sshd	1
107.3.146.207	bin	Failed	sshd	1
107.3.146.207	couchdb	Failed	sshd	1
107.3.146.207	customer	Failed	sshd	1
107.3.146.207	db	Failed	sshd	1
107.3.146.207	desktop	Failed	sshd	4
107.3.146.207	divine	Failed	sshd	1

Transforming Command Summary (cont.)

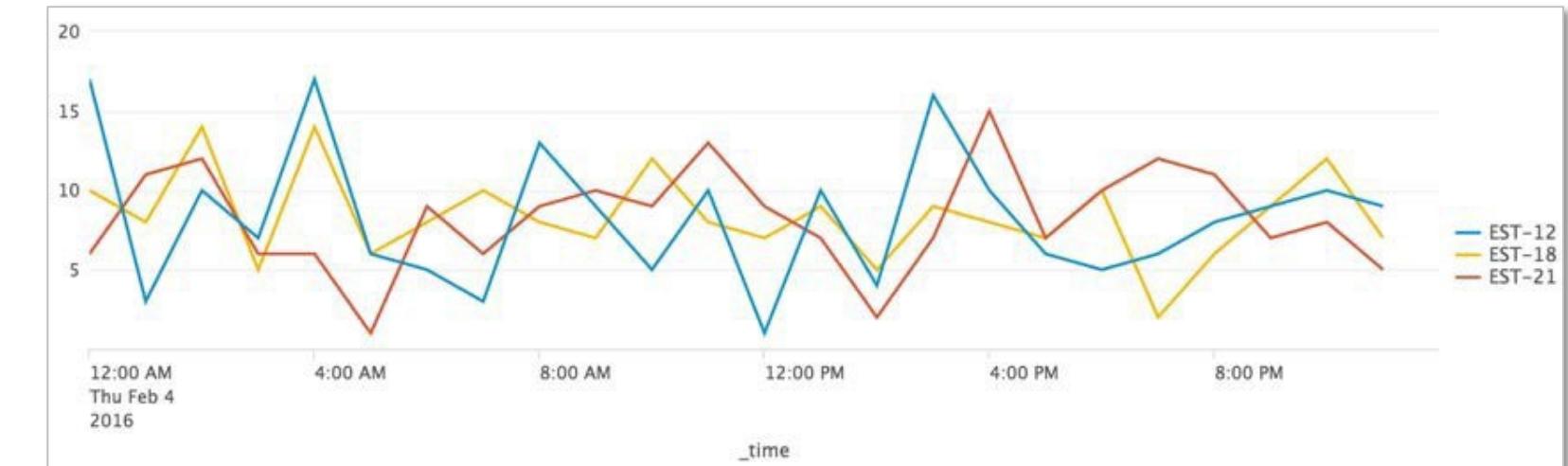
- To calculate statistics with an arbitrary field as the x-axis (not `_time`), use `chart`
 - When you use a `by` field, the output is a table
 - Each column represents a distinct value of the split-by field

```
index=security sourcetype=linux_secure  
| chart count over src_ip  
by vendor_action
```



Transforming Command Summary (cont.)

- Use timechart to calculate statistics with _time as the x-axis
- If a by field is used, the output is a table



- Each column represents a distinct value of the split-by field

```
... | timechart span=1h count by itemId limit=3  
useother=f
```

_time	EST-13	EST-15	EST-7	NULL
2018-01-16 12:00	7	5	4	28
2018-01-16 13:00	15	6	7	53
2018-01-16 14:00	8	20	15	42
2018-01-16 15:00	3	8	9	38
2018-01-16 16:00	5	8	4	34

More Visualizations

trendline Command

- Allows you to overlay a computed moving average on a chart
- `trendline` computes the moving averages of a field

```
trendline <trendtype><period>(field) [AS  
newfield]
```

- *trendtype*:
 - sma - simple moving average
 - ema - exponential moving average
 - wma - weighted moving average

trendline Command (cont.)

- Must define the *period* over which to compute the trend
- *period* must be an integer between 2 and 10000
 - For example, `sma2(sales)` is valid
 - But `sma(sales)` would *fail* as it is missing an integer, the defining period

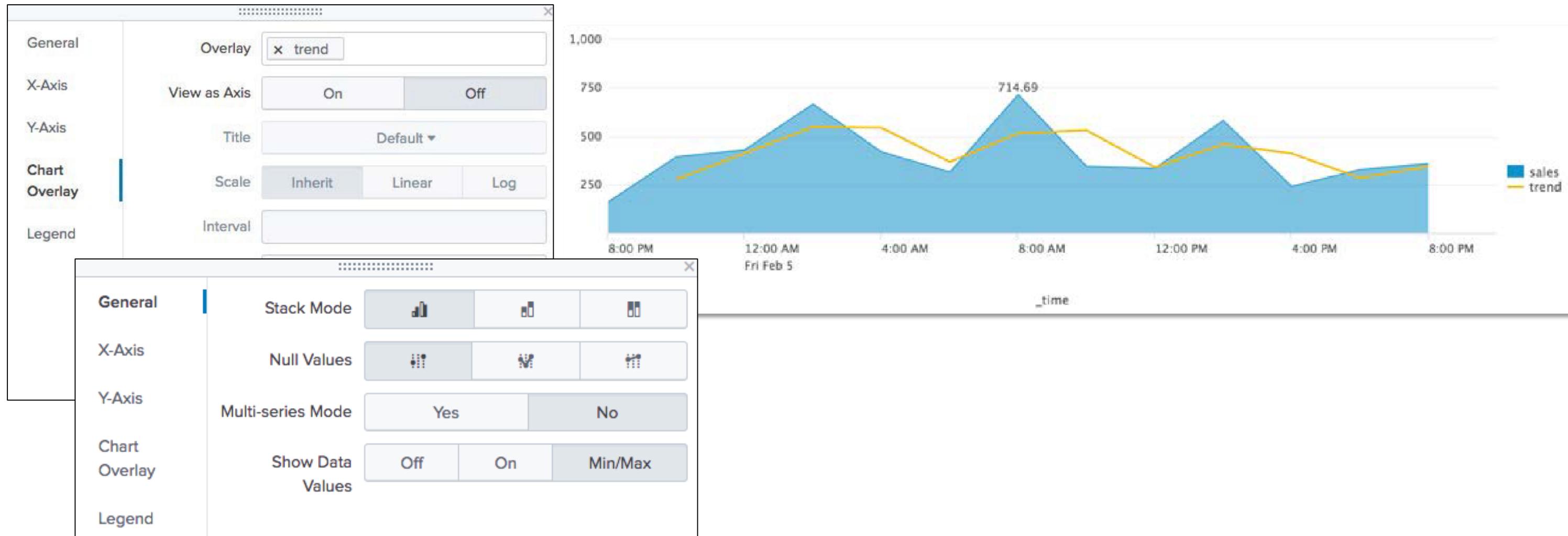
New Search Save As ▾ Close

```
index=web sourcetype=access_combined action=purchase status=200
| timechart span=2h sum(price) as sales
| trendline sma(sales) as trend
```

Last 24 hours ▾ ! Error in 'trendline' command: command="trendline", Invalid trend period for argument 'sma(sales)' Search

trendline Command – Example

```
index=web sourcetype=access_combined action=purchase status=200  
| timechart span=2h sum(price) as sales  
| trendline sma2(sales) as trend
```



Viewing Results as a Map

There are two map types

Cluster Map



Choropleth Map



iplocation Command

```
index=security sourcetype=linux_secure (fail* OR invalid)
| iplocation src_ip
```

- Use iplocation to look up and add location information to an event
 - This information includes city, country, region, latitude and longitude
- Not all of the information is available for all ip address ranges
- Automatically defines the default lat and lon fields required by geostats

INTERESTING FIELDS
a action 1
a app 1
a City 74
a Country 37
date_hour 2
lat 97
linecount 1
lon 98
pid 100+
a process 1
a punct 5
a Region 62

geostats Command

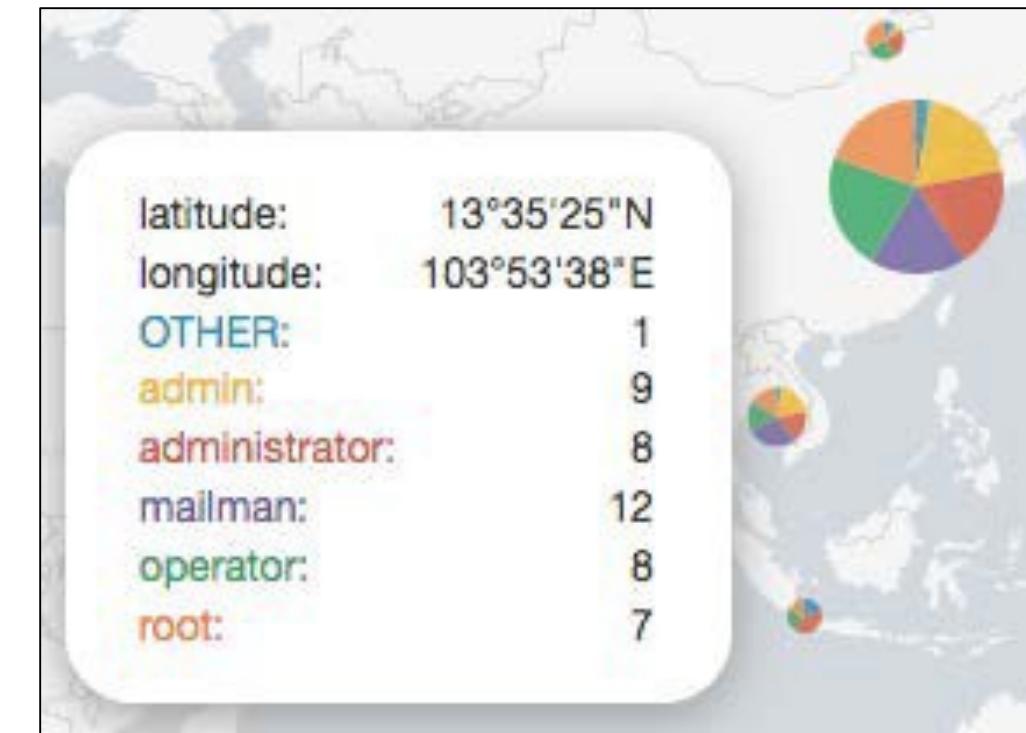
- Use `geostats` to compute statistical functions and render a cluster map

```
geostats [latfield=string] [longfield=string]  
[stats-agg-term]* [by-clause]
```

- Data must include latitude and longitude values
- Define the `latfield` and `longfield` only if they differ from the default `lat` and `lon` fields
- To control the column count:
 - On a global level, use the `globallimit` argument
 - On a local level, depending on where your focus is (i.e., where you've zoomed in), use the `locallimit` argument

geostats Command – Example

```
index=security sourcetype=linux_secure  
(fail* OR invalid)  
| iplocation src_ip  
| geostats globallimit=5 count by user
```



geobin	latitude	longitude	OTHER	admin	administrator	mailman	operator	root
bin_id_zl_0_y_2_x_2	-25.96763	-53.70309	3	4	5	4	2	
bin_id_zl_0_y_2_x_4	-29.00000	24.00000	1		1	3	3	1
bin_id_zl_0_y_2_x_7	-33.22925	151.62210	1	3	2		1	7
bin_id_zl_0_y_3_x_3	-8.05000	-34.90000	110					

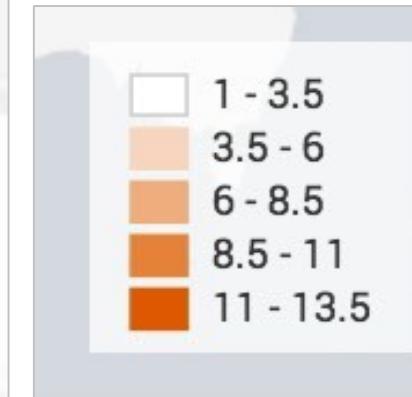
Choropleth Map

- Uses shading to show relative metrics, such as sales, network intruders, etc. for predefined geographic regions
- To define regional boundaries, you must have either a:
 - KML (Keyhole Markup Language) file
 - KMZ (compressed Keyhole Markup Language) file
- Splunk ships with:
 - geo_us_states, United States
 - geo_countries, countries of the world



```
...| geom [featureCollection]  
[featureIdField=string]
```

geom Command



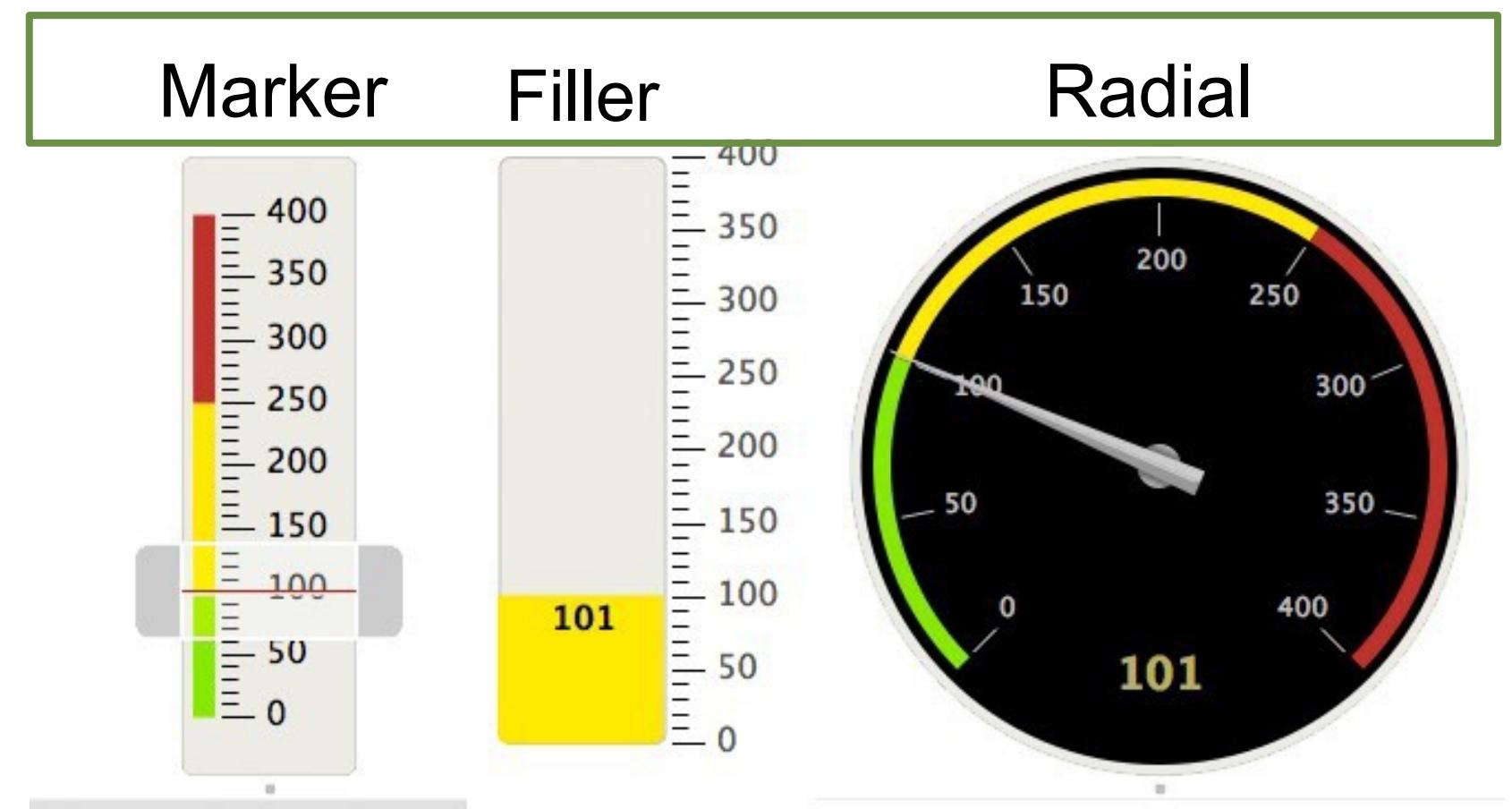
```
index=sales sourcetype=vendor_sales  
VendorID > 4999 AND VendorID < 6000  
| stats count as Sales by VendorCountry  
| geom geo_countries featureIdField=VendorCountry
```

Viewing Results as a Single Value

Single value visualizations provide various formatting options

```
index=security sourcetype=linux_secure vendor_action=failed  
| stats count
```

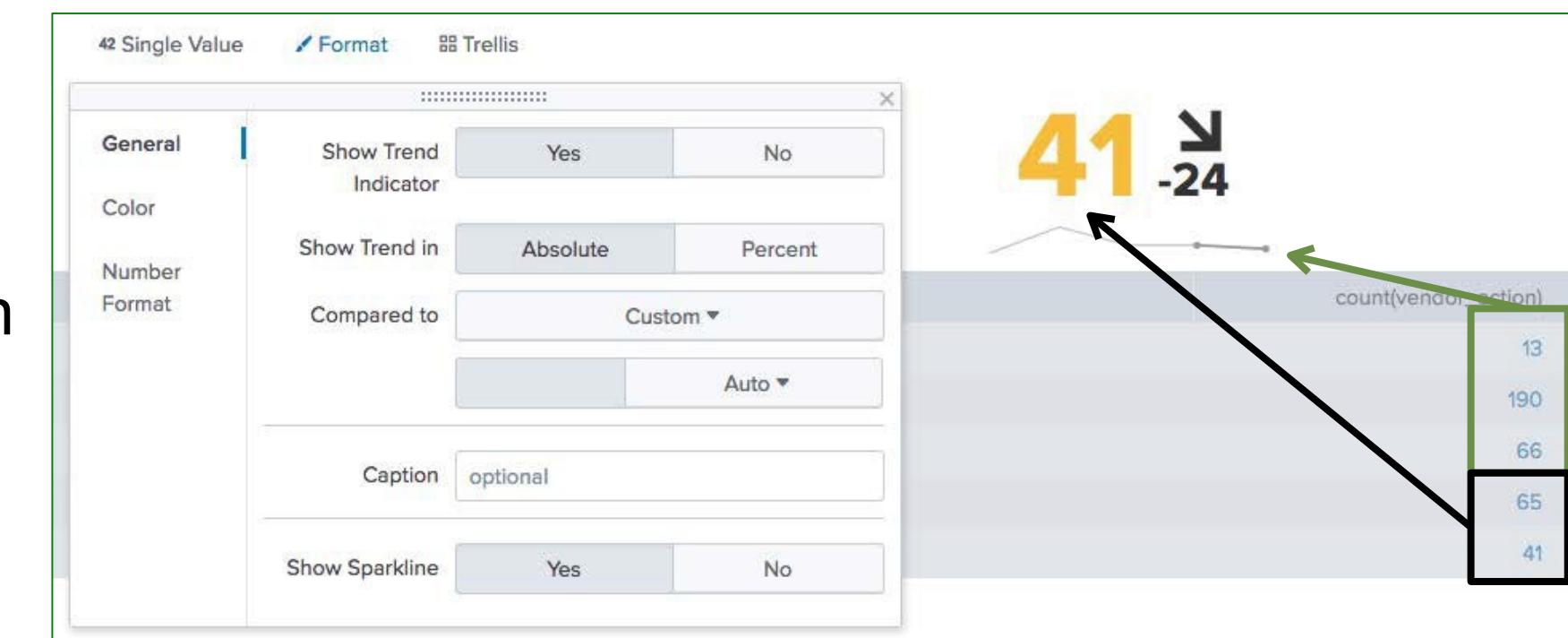
101



Single Value Visualizations: timechart

- With the `timechart` command, you can add a sparkline and a trend
- A **sparkline** is an inline chart
 - It is designed to display time-based trends associated with the primary key
- The **trend** shows the direction in which values are moving
 - It appears to the right of the single value

```
index=security sourcetype=linux_secure  
fail* OR invalid  
| timechart span=15m count(vendor_action)
```



Adding Totals Using Format Options

- Automatically total every column using the Format options
- When using this approach, you:
 - Cannot indicate which column to total; all columns are always totaled
 - Cannot add labels

```
index=web sourcetype=access_combined  
| stats sum(bytes) as Bytes,  
avg(bytes) as avgBytes,  
count as totalEvents by host
```

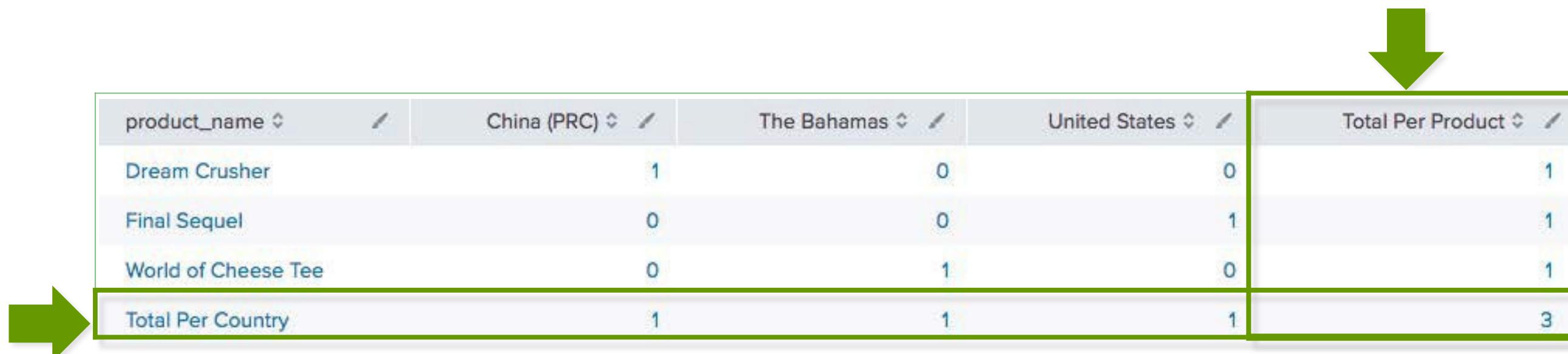
Statistics (3)			
host	Bytes	avgBytes	totalEvents
www1	173626	2066.9761904761904	84
www2	69789	2052.6176470588234	34
www3	92530	2372.5641025641025	39

The screenshot shows the Splunk interface. At the top, there's a navigation bar with 'Events', 'Patterns', 'Statistics (3)', and 'Visualization'. Below it is a search bar with '20 Per Page', 'Format', and 'Preview' dropdowns. The main area displays a table with three rows: www1, www2, and www3. Each row has four columns: host, Bytes, avgBytes, and totalEvents. A yellow arrow points from the 'Format' button in the search bar down to the table, indicating the application of format options. To the right of the table is a 'Format' dialog box with tabs for 'General' and 'Summary'. The 'Summary' tab is selected, showing a 'Totals' section with a 'Yes' button (which is highlighted in yellow) and a 'Percentages' section with a 'No' button. The bottom of the dialog shows the same table structure with the addition of summary rows at the bottom: 335945, 6492.157940099117, and 157.

host	Bytes	avgBytes	totalEvents
www1	173626	2066.9761904761904	84
www2	69789	2052.6176470588234	34
www3	92530	2372.5641025641025	39
		335945	6492.157940099117
			157

Adding Totals Using addtotals Command

- Alternatively, use the addtotals command to:
 - Compute the sum of all **or selected** numeric fields for each column and place the total in the last row
 - Compute the sum of all **or selected** numeric fields for each **row** and place the total in the last column



product_name	China (PRC)	The Bahamas	United States	Total Per Product
Dream Crusher	1	0	0	1
Final Sequel	0	0	1	1
World of Cheese Tee	0	1	0	1
Total Per Country	1	1	1	3

addtotals Command: Syntax

addtotals [row=*bool*] [fieldname=*field*]
[col=*bool*] [labelfield=*field*] [label=*string*]
field-list

		Column Options	
row=true/ false (Default= true)	A column is created that contains numeric totals for each row.	col=true/ false (Default= false)	A row is created that contains numeric totals for each column.
fieldname= <i>field</i> (Default=Total)	Defines a string used to create a field name for the totals column.	label= <i>string</i> (Default=Total)	Defines a string used to name the totals row.
		labelfield= <i>fieldname</i>	Defines where the label string is placed. (Generally, you should make this the first field in the list.)
General Options			
<i>field-list</i> =one or more numeric fields. (Default: all numeric fields)		Defines the numeric fields to be totaled.	

addtotals Command

Do not total rows

Total columns

Add the label totalBytes

Place the label under the host column

Only total the Bytes column

```
index=web sourcetype=access_combined  
| stats sum(bytes) as Bytes, avg(bytes) as avgBytes, count as totalEvents by host  
| addtotals row=f col=t label=totalBytes  
labelfield=host Bytes
```

host	Bytes	avgBytes	totalEvents
www1	141882	2149.7272727272725	66
www2	93781	2084.0222222222224	45
www3	122834	2233.3454545454547	55
totalBytes	358497		

Filtering and Formatting Data

eval Command – Overview

- eval allows you to calculate and manipulate field values in your report

```
eval fieldname1 = expression1 [,  
fieldname2 = expression2...]
```

- Supports a variety of functions
- Results of eval written to either new or existing field you specify
 - If the destination field exists, the values of the field are replaced by the results of eval
 - Indexed data is not modified, and no new data is written into the index
 - Field values are treated in a case-sensitive manner

eval Command

- The eval command allows you to:
 - Calculate expressions
 - Place the results in a field
 - Use that field in searches or other expressions

- Type

Arithmetic

Operators

+ - * / %

Concatenation

+

Boolean

AND OR NOT

Comparison

XOR
< > <= >= != = ==
LIKE

eval

[Learn More ↗](#)

Calculates an expression and puts the resulting value into a field. You can specify to calculate more than one expression.

Example:

... | eval velocity=distance/time

eval Command – Convert Values (cont.)

- Results of eval must be set to a new or existing field
- In this example:

Calculate the number of bytes for each usage type

Create a new field named bandwidth

Convert the values of the Bytes field into MB by dividing Bytes field values by $(1024 * 1024)$

```
index=network sourcetype=cisco_wsa_squid  
| stats sum(bytessent) as Bytes by usage  
| eval bandwidth = Bytes/(1024*1024)
```

usage	Bytes	bandwidth
Borderline	54011022	51.50892448425293
Business	66844576	63.747955322265625
Personal	299584913	285.7064371109009
Unknown	77187989	73.61220264434814
Violation	3203231	3.0548391342163086

eval Command – Round Values

- The results of Bandwidth are hard to read with so many decimal points
- `round(field/number, decimals)` function sets the value of a field to the number of decimals you specify
- In this example:
 - Divide the value of the Bytes field by $(1024 * 1024)$
 - Round the result to two decimal points
- If the number of decimals is unspecified, the result is a whole number

```
index=web      sourcetype=access_combined
| stats sum(sc_bytes) as Bytes by usage
| eval bandwidth =round(Bytes/(1024*1024), 2)
| sort -bandwidth
| rename bandwidth as "Bandwidth (MB)"
```

usage	Bytes	Bandwidth (MB)
Personal	299584913	285.71
Unknown	77187989	73.61
Business	66844576	63.75
Borderline	54011022	51.51
Violation	3203231	3.05

Removing Fields

- The "Bandwidth (MB)" field has the data in the desired format
- The Bytes field is no longer needed
 - The Bytes field can be removed

```
index=web sourcetype=access_combined
| stats sum(sc_bytes) as Bytes by usage
| eval bandwidth = round(Bytes/(1024*1024), 2)
| sort -bandwidth
| rename bandwidth as "Bandwidth (MB)"
| fields - Bytes
```

usage	Bandwidth (MB)
Personal	285.71
Unknown	73.61
Business	63.75
Borderline	51.51
Violation	3.05

eval Command – Calculating Values

You can perform mathematical functions against fields with numeric field values

In this example, stats calculates the total list price and total sale price by product_name

eval calculates the discount percentage and formats the

```
index=web sourcetype=access_combined  
product_name=* action=purchase  
| stats sum(price) as tp, sum(sale_price) as tsp by  
product_name  
| eval Discount = round(((tp - tsp)/ tp)*100)  
| sort -Discount  
| eval Discount = Discount.%  
| rename tp as "Total List Price", tsp as "Total Sale  
Price", product_name as Product
```

Product	Total List Price	Total Sale Price	Discount
Puppies vs. Zombies	578.84	230.84	60%
Fire Resistance Suit of Provolone	594.51	296.51	50%
Holy Blade of Gouda	742.76	370.76	50%
Dream Crusher	6438.39	4023.39	38%
Manganiello Bros.	5758.56	3598.56	38%

eval Command – tostring Function

- `tostring` converts a numeric field value to a string

```
tostring(field, "option")
```

- Options:

- "commas": applies commas
 - If the number includes decimals, it rounds to two decimal places
- "duration": formats the number as "hh:mm:ss"
- "hex": formats the number in hexadecimal

```
index=web sourcetype=access_combined  
action=purchase status=503  
| stats count(price) as NumberOfLostSales,  
  avg(price) as AverageLostSales, sum(price) as  
  TotalLostRevenue  
| eval AverageLostSales  
  = $" + tostring(AverageLostSales, "commas"),  
  TotalLostRevenue = $ + tostring(TotalLostRevenue,  
  "commas")
```

NumberOfLostSales	AverageLostSales	TotalLostRevenue
113	\$21.58	\$2,438.87

toString Function – duration Option

This example shows "duration" option of toString function

stats calculates sessionTime for each session (JSESSIONID)

- Use the range function to return the difference between the max and min values of _time

sort 5 displays the top 5 most frequent values

The duration option formats the time as "hh:mm:ss"

```
index=web sourcetype=access_combined  
| stats range(_time) as sessionTime by JSESSIONID  
| sort 5 -sessionTime  
| eval duration =  
|> tostring(sessionTime,"duration")
```

JSESSIONID	sessionTime	duration
SD6SL8FF3ADFF4951	175	00:02:55
SD6SL1FF10ADFF4966	158	00:02:38
SD6SL3FF1ADFF4959	141	00:02:21
SD0SL1FF2ADFF4963	135	00:02:15
SD4SL10FF2ADFF4961	132	00:02:12

`eval` Commands with Multiple Expressions

- Multiple expressions can be combined into one `eval` command
- Each subsequent expression references the results of previous expressions
- Expressions must be separated by commas

```
eval fieldname1 =  
    expression1, fieldname2 =  
    expression2,  
    fieldname3 = expression3...
```

eval Command – if Function Syntax

`if(X , Y , Z)`

- The `if` function takes three arguments
- The first argument, X , is a Boolean expression
 - If it evaluates to TRUE, the result evaluates to the second argument, Y
 - If it evaluates to FALSE, the result evaluates to the third argument, Z
- Non-numeric values must be enclosed in "double quotes"
- Field values are treated in a case-sensitive manner

eval Command – if Function Example

- Create a new field, SalesTerritory
- Evaluate VendorID

- If ≥ 7000 AND < 8000 is TRUE, set result to "Asia"
 - ▶ Remember, arguments must be enclosed in quotes
- If it evaluates to FALSE, set result to "Rest of the World"

```
index=sales sourcetype=vendor_sales
| eval SalesTerritory
= if((VendorID >= 7000 AND VendorID < 8000), "Asia", "Rest of the
  World")
| stats sum(price) as TotalRevenue by SalesTerritory
| eval TotalRevenue = "$" + tostring(TotalRevenue, "commas")
```

SalesTerritory	TotalRevenue
Asia	\$1,371.26
Rest of the World	\$14,495.25

eval Command – case Function

case(X1,Y1,X2,Y2...)

Note

- The first argument, $X1$, is a Boolean expression
- If it evaluates to TRUE, the result evaluates to $Y1$
- If it evaluates to FALSE, the next Boolean expression, $X2$, is evaluated, etc.
- If you want an “otherwise” clause, just test for a condition you know is true at the end (e.g., $0=0$)

```
index=web sourcetype=access_combined  
| eval rating = case(productId LIKE "WC%", "Teen", productId LIKE "FS%", Mature,  
0=0, "Unrated")
```

eval function

- To count the number of events that contain a specific field value, use the `count` and `eval` functions

- Used within a transforming command, such as `stats`
- Requires an `as` clause
- Double quotes are required for character field values
- Field values are case-sensitive

```
index=security sourcetype=linux_secure  
vendor_action=*  
| stats count(eval.vendor_action="Accepted") as Accepted,  
count(eval.vendor_action="Failed") as Failed,  
count(eval.vendor_action="session opened") as SessionOpened
```

Accepted	Failed	SessionOpened
322	9676	407

Filtering Results – search and where

- The `search` and `where` commands both filter results

- `search`

- ▶ May be easier if you're familiar with basic search syntax
 - ▶ Treats field values in a case-insensitive manner
 - ▶ Allows searching on keyword
 - ▶ Can be used at any point in the search pipeline

- `where`

- ▶ Can compare values from two different fields
 - ▶ Functions are available, such as `is not null()`
 - ▶ Treats field values in a case-sensitive manner
 - ▶ Can't appear before first pipe in search pipeline

search Command

- To filter results, use search at any point in the search pipeline
- Behaves exactly like search strings before the first pipe
 - Uses the "*" wildcard
 - Treats field values in a case-insensitive manner

```
index=web sourcetype=access_combined  
action=purchase status=200  
| stats sum(price) as sales by product_name  
| search sales>500  
| sort -sales  
| eval sales="$"+sales  
| rename sales as "Popular Products",  
product_name as "Product Name"
```

Product Name	Popular Products
Manganiello Bros.	\$839.79
Dream Crusher	\$799.80
SIM Cubicle	\$679.66

where Command

where eval-expression

- Uses same expression syntax as eval command
- Uses boolean expressions to filter search results and only keeps results that are True
- Double quoted strings are interpreted as field values
 - Treats field values in a case-sensitive manner
- Unquoted or single-quoted strings are treated as fields

where Command – Example

- Filters search results using eval expressions
- Used to compare two different fields

```
index=web sourcetype=access_combined  
| timechart count(eval(action="changequantity"))  
    as changes, count(eval(action="remove")) as  
    removals  
| where removals > changes
```

_time	changes	removals
2018-01-07	108	130
2018-01-09	126	135

where Command With like Operator

- Can do wildcard searches with where command
- Use (_) for one character and (%) for multiple characters
- Must use the like operator with wildcards

```
index=security sourcetype=linux_secure  
| stats count by src_ip  
| where src_ip like "10_.%"
```

src_ip	count
102.2.83.137	102
107.3.146.207	55
108.65.113.83	34
109.169.32.135	72

fillnull Command

- Use `fillnull` to replace null values in fields
- Use `value=string` to specify a string you want displayed instead
 - Example: `fillnull value=NULL`
- If no `value=` clause, default replacement value is 0
- Optionally, restrict which fields `fillnull` apply to by listing them at end of command
 - Example: `fillnull VALUE="N/A" discount refund`

fillnull Command – Examples

```
index=sales sourcetype= vendor_sales  
| chart sum(price) over product_name by  
VendorCountry  
| fillnull
```

product_name	Morocco	United States
Dream Crusher	39.99	39.99
Puppies vs. Zombies	0	4.99

```
index=sales sourcetype= vendor_sales  
| chart sum(price) over product_name by  
VendorCountry  
| fillnull value="No Value"
```

product_name	Morocco	United States
Dream Crusher	39.99	39.99
Puppies vs. Zombies	No Value	4.99

Correlating Events

What is a Transaction?

- A transaction is any group of related events that span time
- Events can come from multiple applications or hosts
 - Events related to a single purchase from an online store can span across an application server, database, and e-commerce engine
 - One email message can create multiple events as it travels through various queues
 - Each event in the network traffic logs represents a single user generating a single http request
 - Visiting a single website normally generates multiple http requests
 - ▶ HTML, JavaScript, CSS files
 - ▶ Flash, images, etc.

transaction Command

- `transaction field-list`
 - `field-list` can be one field name or a list of field names
 - Events are grouped into transactions based on the values of these fields
 - If multiple fields are specified and a relationship exists between those fields, events with related field values are grouped into a single transaction
- Common constraints:
`maxspan maxpause startswith endswith`

Events That Have the Same JSESSIONID

- The log shows a number of events that share the same JSESSIONID value (SD0SL10FF3ADFF4950)
- However, it is difficult to:
 - View the events as a group
 - Gain insight as to what is happening with these events
 - Know if there are other events scattered in the results set

> 1/17/18	175.44.1.122 - - [18/Jan/2018:00:01:08]	"GET /oldlink?itemId=EST-6&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1"	200	4:01:08.000 PM	3516 "http://www.buttercupgames.com/oldlink?itemId=EST-6" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 614	JSESSIONID = SD0SL10FF3ADFF4950	host = www3	source = /opt/log/www3/access.log	sourcetype = access_combined
> 1/17/18	175.44.1.122 - - [18/Jan/2018:00:01:05]	"GET /category.screen?categoryId=STRATEGY&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1"	200	4:01:05.000 PM	2453 "http://www.buttercupgames.com/oldlink?itemId=EST-18" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 989	JSESSIONID = SD0SL10FF3ADFF4950	host = www3	source = /opt/log/www3/access.log	sourcetype = access_combined
> 1/17/18	175.44.1.122 - - [18/Jan/2018:00:00:53]	"POST /category.screen?categoryId=SIMULATION&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1"	200	4:00:53.000 PM	3526 "http://www.buttercupgames.com" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 222	JSESSIONID = SD0SL10FF3ADFF4950	host = www3	source = /opt/log/www3/access.log	sourcetype = access_combined

transaction Command – Example 1

- The transaction command creates a single event from a group of events
 - The events must share the same value in a specified field
- Transactions can cross multiple tiers such as web servers or application servers
- For example, you can easily view the events for JSESSIONID SD0SL10FF3ADFF4950

```
index=web sourcetype=access_combined  
| transaction JSESSIONID
```

> 1/17/18 4:00:53.000 PM	175.44.1.122 -- [18/Jan/2018:00:00:53] "POST /category.screen?categoryId=SIMULATION&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1" 200 3526 "http://www.buttercupgames.com" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1 .4322; InfoPath.1; MS-RTC LM 8)" 222 175.44.1.122 -- [18/Jan/2018:00:01:05] "GET /category.screen?categoryId=STRATEGY&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1" 200 2453 "http://www.buttercupgames.com/oldlink?itemId=EST-18" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 989 175.44.1.122 -- [18/Jan/2018:00:01:08] "GET /oldlink?itemId=EST-6&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1" 200 3516 "http://www.buttercupgames.com/oldlink?itemId=EST-6" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1 .1.4322; InfoPath.1; MS-RTC LM 8)" 614 175.44.1.122 -- [18/Jan/2018:00:01:17] "GET /category.screen?categoryId=ACCESSORIES&JSESSIONID=SD0SL10FF3ADFF4950 HTTP 1.1" 200 652 "http://www.buttercupgames.com/oldlink?itemId=EST-16" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 118
-----------------------------	---

transaction Command – Specific Fields

The transaction command produces additional fields, such as:

- duration – the difference between the timestamps for the first and last event in the transaction
- eventcount – the number of events in the transaction

transaction Command – maxspan/maxpause

- You can also define a max overall time span and max gap between events

- maxspan=10m

- ▶ Maximum total time between the *earliest* and *latest* events

- ▶ If not specified, default is -1 (or no limit)

- maxpause=1m

- ▶ Maximum total time *between* events

- ▶ If not specified, default is -1 (or no limit)

```
index=web sourcetype=access_combined  
| transaction clientip maxspan=10m maxpause=1m  
| eval duration = tostring(duration,"duration")  
| sort -duration  
| table clientip duration action  
| rename clientip as "Client IP", action as  
"Client Actions"
```

Client IP	duration	Client Actions
91.199.80.24	00:01:27	addtocart changequantity purchase view
12.130.60.5	00:01:27	addtocart purchase view
60.220.218.88	00:01:25	view

transaction Command – starts with/end with

- To form transactions based on terms, field values, or evaluations, use starts with and ends with options

- In this example:

- The first event in the

- transaction includes addtocart

- The last event includes purchase

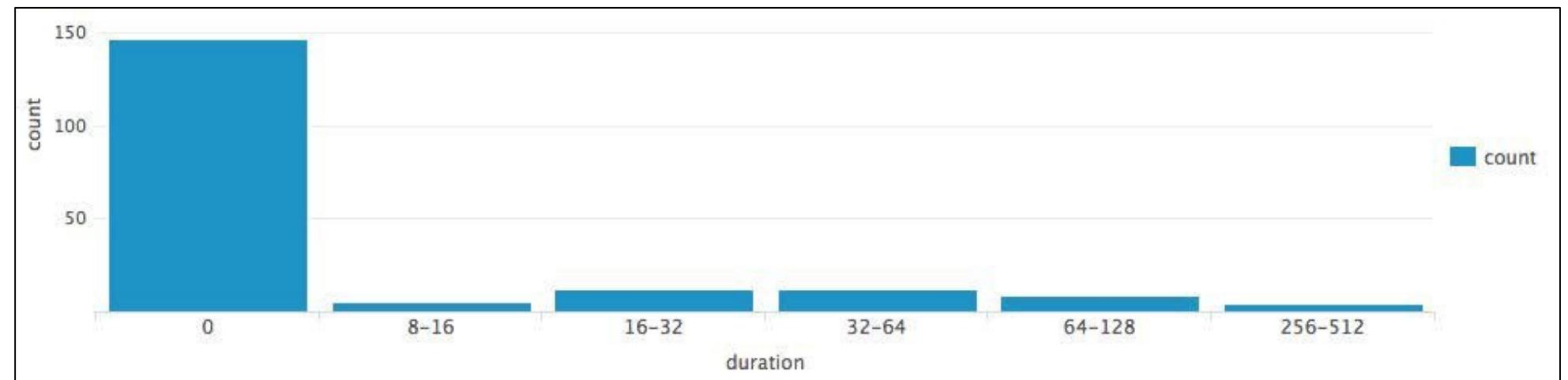
```
index=web sourcetype=access_combined
| transaction clientip JSESSIONID
| startswith=eval(action="addtocart")
| endswith=eval(action="purchase")
| table clientip, JSESSIONID, duration, eventcount
```

clientip	JSESSIONID	duration	eventcount
223.205.219.198	SD1SL9FF5ADFF4953	3	2
201.3.120.132	SD10SL5FF2ADFF4956	4	2
201.3.120.132	SD10SL5FF2ADFF4956	1	2
203.45.206.135	SD10SL7FF4ADFF4964	2	2

Reporting on Transactions

- You can use statistics and reporting commands with transactions
- `count()` function counts number of transactions and separates the count by the duration of each

```
index=web sourcetype=access_combined  
status=200 action=purchase  
| transaction clientip maxspan=10m  
| chart count BY duration span=log2
```



transaction vs. stats

- When you have a choice, use stats — it's faster and more efficient, especially in large Splunk environments
- Only use transaction when you:
 - Need to see events correlated together
 - Must define event grouping based on start/end values or segment on time
- Use stats when you:
 - Want to see the results of a calculation
 - Can group events based on a field value (e.g., by src_ip)
- By default, there's a limit of 1,000 events per transaction
 - No such limit applies to stats
 - Admins can change limit by configuring max_events_per_bucket in limits.conf

transaction vs. stats: Example 2

```
index=security  
sourcetype=linux_secure failed  
| transaction src_ip  
| table src_ip, eventcount  
| sort - eventcount
```

src_ip	eventcount
87.194.216.51	1000
211.166.11.101	840
128.241.220.82	662
194.215.205.19	617
87.194.216.51	134
236.45.28.248	133
229.156.63.239	133

- Transaction took 6.163 seconds
- Stats took 4.643 seconds
- transaction has a limit of 1,000
- Count of transactions vs. count of IPs

```
index=security  
sourcetype=linux_secure failed  
| stats count as eventcount by src_ip  
| sort - eventcount
```

src_ip	eventcount
87.194.216.51	1134
211.166.11.101	840
128.241.220.82	662
194.215.205.19	617
109.169.32.135	443
216.221.226.11	432
188.138.40.166	429

Knowledge Objects

Module Objectives

- Identify the categories of knowledge objects
- Define the role of a knowledge manager
- Identify naming conventions
- Review permissions
- Manage knowledge objects
- Describe the Splunk Common Information Model (CIM)

What are Knowledge Objects?

- Knowledge objects are tools you use to discover and analyze various aspects of your data
 - Data interpretation – Fields and field extractions
 - Data classification – Event types
 - Data enrichment – Lookups and workflow actions
 - Normalization – Tags and field aliases
 - Datasets – Data models



What are Knowledge Objects? (cont.)

- Shareable
 - Can be shared between users
- Reusable
 - Persistent objects that can be used by multiple people or apps, such as macros and reports
- Searchable
 - Since the objects are persistent, they can be used in a search

Defining Naming Conventions

- This course uses simple names for lab exercises, but using a naming convention in your production environment is recommended. For example:
 - Group:** Corresponds to the working group(s) of the user saving the object (examples: SEG. NEG. OPS. NOC)
 - Object Type:** Indicates the type of object (alert, report, summary-index-populating) (examples: Alert, Report, Summary)
 - Description:** A meaningful description of the context and intent of the search, limited to one or two words if possible; ensures the search name is unique
- So, for example:
SEG_Alert_WinEventlogFailures

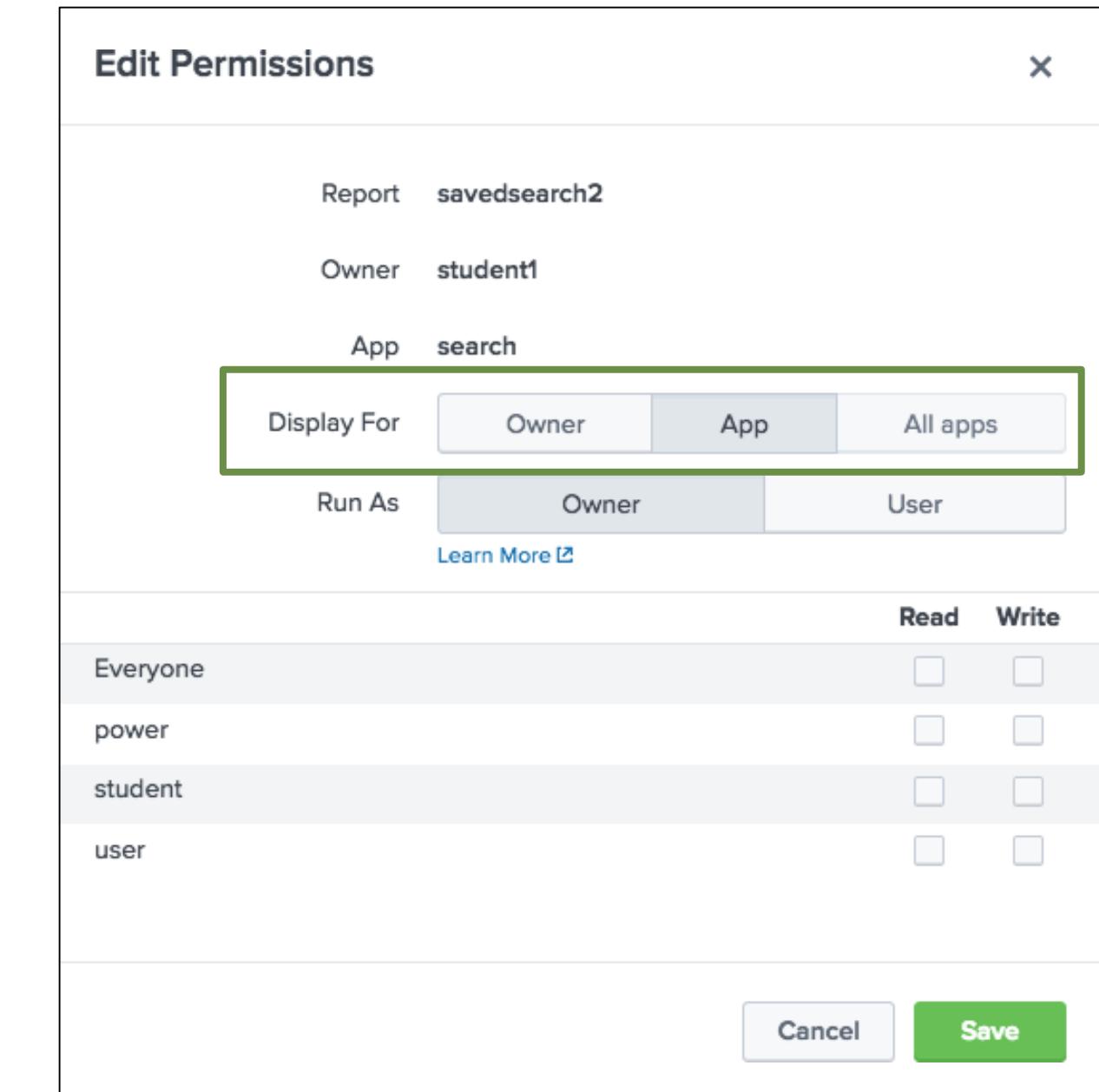
Reviewing Permissions

Description	Create	Read	Edit (write)
Private	Only the person who created the object can use it and edit it	User Power Admin	Person who created it Admin
This app only	Object persists in the context of a specific app	Power Admin	User* Power* Admin
All apps	Object persists globally across all apps	Admin	User* Power* Admin

* Permission to read and/or write if creator gives permission to that role

Reviewing Permissions (cont.)

- When an object is created, Display For is set to **Owner** by default
- When object's permissions are set to **App** or **All apps**, all roles are given read permission
 - Write permission is reserved for admin and the object creator unless the creator edits permissions
- Only the admin role can promote an object to **All apps**
 - Other roles have **All Apps** button grayed out



Managing Knowledge Objects

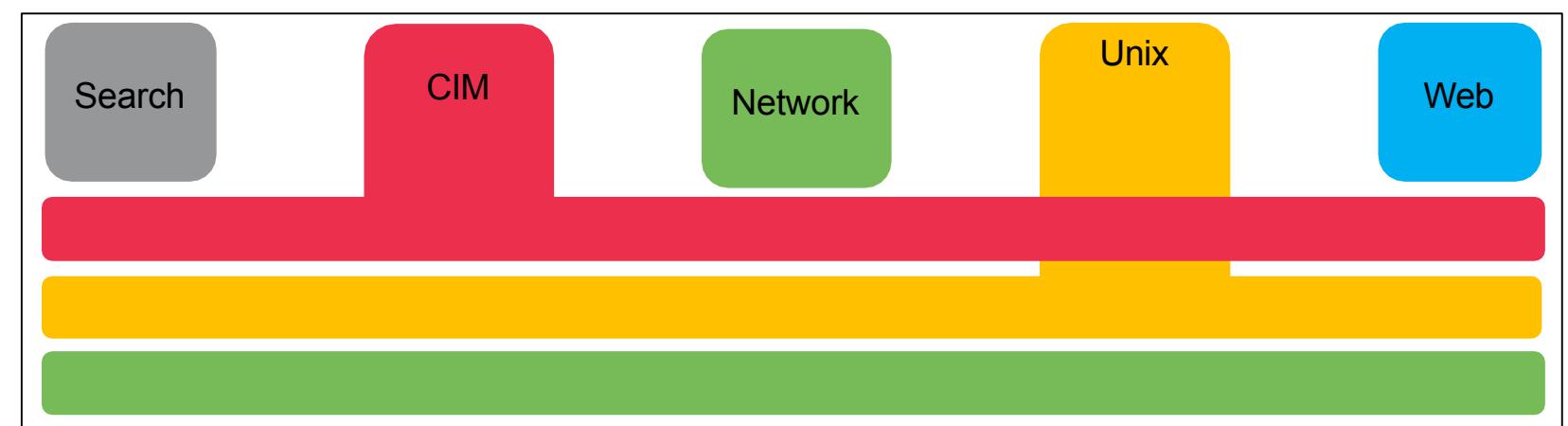
- Knowledge objects are centrally managed from **Settings > Knowledge**
- Your role and permissions determine your ability to modify an object's settings

The screenshot shows the 'All configurations' page under the 'KNOWLEDGE' section of the Settings menu. The left sidebar lists various configuration categories: Searches, reports, and alerts; Data models; Event types; Tags; Fields; Lookups; User interface; Advanced search; All configurations (which is selected); and DATA. The main content area displays a table titled 'All configurations' showing 1-25 of 70 items. The table has columns for Name, Config type, Owner, App, Sharing, and Status. A green box highlights the 'Sharing' and 'Status' columns. A green arrow points from the 'Sharing' column to the 'Status' column. The table data is as follows:

Name	Config type	Owner	App	Sharing	Status
Errors in the last 24 hours	savedsearch	No owner	search	App Permissions	Enabled
Errors in the last hour	savedsearch	No owner	search	App Permissions	Enabled
License Usage Data Cube	savedsearch	No owner	search	App Permissions	Enabled Disable
Orphaned scheduled searches	savedsearch	No owner	search	App Permissions	Enabled Disable
alert	views	No owner	search	Global Permissions	Enabled
alerts	views	No owner	search	Global Permissions	Enabled

Using the Splunk Common Information Model (CIM)

- Methodology for normalizing data
- Easily correlate data from different sources and source types
- Leverage to create various objects discussed in this course—field extractions, field aliases, event types, tags
- More details discussed in Module 13



Creating and Managing Fields

Field Auto-Extraction

- Splunk automatically discovers many fields based on source type and key/value pairs found in the data
- Prior to search time, some fields are already stored with the event in the index
 - Meta fields, such as `host`, `source`, and `sourcetype`
 - Internal fields such as `_time` and `_raw`
- At search time, *field discovery* discovers fields directly related to the search's results
- Splunk may also extract other fields from raw event data that aren't directly related to the search

Performing Field Extractions

- In addition to the many fields Splunk auto-extracts, you can also extract your own fields with the Field Extractor (FX)
- Use FX to extract fields that are static and that you use often in searches
 - Graphical UI
 - Extract fields from events using regex or delimiter
 - Extracted fields persist as knowledge objects
 - Can be shared and re-used in multiple searches
- Access FX via Settings, Fields Sidebar, or Event Actions menu

Field Extraction Methods

- **Regex**
 - Use this option when your event contains unstructured data like a system log file
 - FX attempts to extract fields using a Regular Expression that matches similar events
- **Delimiter**
 - Use this option when your event contains structured data like a .csv file
 - The data doesn't have headers and the fields must be separated by delimiters (spaces, commas, pipes, tabs, or other characters)

Field Extraction Workflows – RegEx

Settings

The screenshot shows the Splunk Extract Fields workflow interface. It consists of four main stages:

- Select Sample:** Choose a source or source type, Data Type (sourcetype), and Source Type (linux_secure). A preview window shows event details: "Thu Jan 18 2018 19:30:28 www3 sshd[3014]: Failed password for myuan from 187.60.191.199 port 2961 ssh2".
- Select Method:** Indicate the method to extract fields. Options include "I prefer to write the regular expression myself" (selected) and "I prefer to write the regular expression myself". Below this, the source type is listed as "linux_secure". A preview window shows the same event log entry.
- Select Fields:** This stage is currently active, indicated by a green circle on the progress bar.
- Save:** Final step of the workflow.

On the right side of the interface, there are three sections:

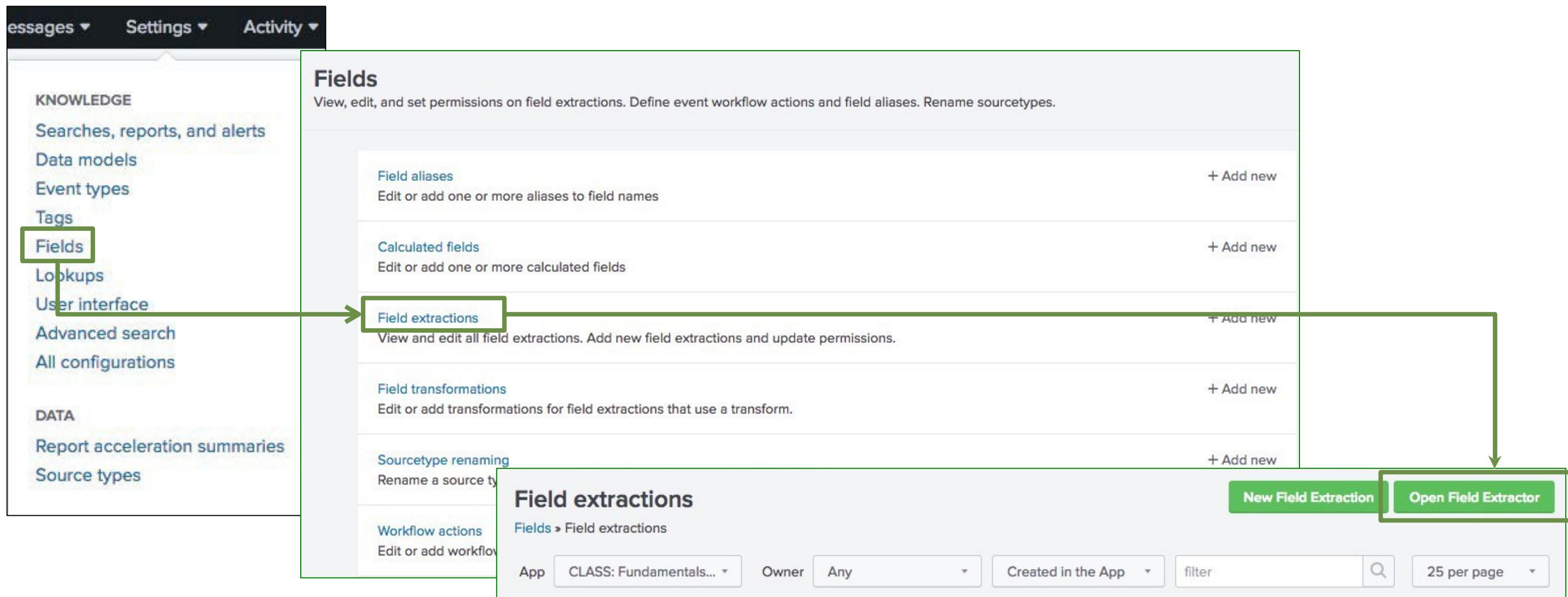
- Regular Expression:** A box containing the regular expression pattern `(.*?)`.
- Delimiters:** A box containing the delimiter pattern `x|y|z`.
- Notes:** Text explaining that Splunk Enterprise will extract fields using a Regular Expression.

Fields Sidebar

Event Actions

Regex Field Extractions from Settings

Settings > Fields > Field extractions > Open Field Extractor



Delimited Field Extractions

- Use delimited field extractions when the event log does not have a header and fields are separated by spaces, commas, or characters
- In this example, the fields are separated by commas

i	Time	Event
>	1/30/18 10:43:39.000 PM	"2018-01-30T22:43:39000-0400",29,1>Error,HOST0167,System,772103058 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit
>	1/30/18 10:43:37.000 PM	"2018-01-30T22:43:37000-0400",35,4,Information,HOST0201,System,507701378 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit
>	1/30/18 10:43:36.000 PM	"2018-01-30T22:43:36000-0400",35,4,Information,HOST0201,System,753380719 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit

Field Extraction Workflows – Delimiters

Settings

The screenshot shows the Splunk Extract Fields workflow interface. The process consists of four steps: Select Sample, Select Method, Select Fields, and Save. The first step, "Select Sample", is currently active, indicated by a green dot on the progress bar. The "Next >" button is visible at the top right of this step. Below the progress bar, there is a "Existing fields >" link.

Select Sample Event

Choose a source or source type, select a sample event, and click Next to go to the next step. The field extractor will use the event to extract fields. [Learn more](#)

I prefer to write the regular expression myself

Data Type: sourcetype ▾

Source Type: Select Source Type

Events

✓ 1,000 events (before 1/31/18 10:05)

filter

_raw

"2018-01-31T20:58:15000-0400", 7036, 4, Information, HOST0167, System, 296651410

Select Method

Indicate the method you want to use to extract your field(s). [Learn more](#)

I prefer to write the regular expression myself

Source type: win_audit

Regular Expression

Splunk Enterprise will extract fields using a Regular Expression.

Delimiters

Splunk Enterprise will extract fields using a delimiter (such as commas, spaces, or characters). Use this method for delimited data like comma separated values (CSV files).

(.*?)

x|y|z

Fields Sidebar

Event Actions

Delimited Field Extractions (Settings) – Rename Field

Click the icon next to the default field name

Enter a new field name

Extract Fields Select Sample Select Method Rename Fields Save < Back Next >

Rename Fields

Select a delimiter. In the table that appears, rename fields by clicking on field names or values. [Learn more](#)

Delimiter

Space Comma Tab Pipe Other

Field Name	time	field2	field3	field4	field5	field6	field7
		7036	4	Information	HOST0167	System	296651410

Rename Field

Events field1 field2 field3 field4 field5 field6 field7

✓ 1,000 events (before 1/31/18 10:08:33.000 PM) 20 per page < Prev 1 2 3 4 5 6 7 8 9 ... Next >

filter Apply Sample: 1,000 events All events All Events Matches Non-Matches

_raw	field1	field2	field3	field4	field5
"2018-01-31T20:58:15000-0400",7036,4,Information,HOST0167,System,296651410	"2018-01-31T20:58:15000-0400"	7036	4	Information	HOST0167
"2018-01-31T20:03:14000-0400",29,1>Error,HOST0167,System,772103058	"2018-01-31T20:03:14000-0400"	29	1	Error	HOST0167
"2018-01-31T20:03:12000-0400",35,4,Information,HOST0201,System,507701378	"2018-01-31T20:03:12000-0400"	35	4	Information	HOST0201

Creating Field Aliases and Calculated Fields

Field Aliases

- A way to normalize data over any default field (host, source or sourcetype)
- Multiple aliases can be applied to one field
- Applied after field extractions, before lookups
- Can apply field aliases to lookups

Field Alias Example

- Several source types contain some type of a username field
- To make data correlation and searching easier, normalize the username field

The screenshot shows a Splunk search interface with three main sections:

- Selected:** Shows the selected host as "cisco_router1" and the selected source as "/opt/log/cisco_router1/cisco_firewall.log".
- Event:** A table of event fields and their values. The "Username" field is highlighted with a green box and has an arrow pointing up to a yellow box labeled "sourcetype=cisco_firewall".
- Results:** A list of events from the "cisco_firewall" source. The "username" field is highlighted with a green box and has an arrow pointing up to a yellow box labeled "sourcetype=access_combine".
- Results:** A list of events from the "winauthentication_security" source. The "User" field is highlighted with a green box and has an arrow pointing up to a yellow box labeled "sourcetype=winauthentication_security".

Event Fields (highlighted in the Event section):

- Severity: 5-None
- splunk_role: power
- src: 123.196.113.11
- src_ip: 123.196.113.11
- status: 200
- url: http://www.uncw.edu /www/screenNewMast.css
- usage: Personal
- username: dhale
- x_acltag: DEFAULT_CASE-DefaultGroup-Demo_Clients-NONE-NONE-DefaultRouting

Result Fields (highlighted in the Results sections):

- ComputerName: BG03-dhale
- EventCode: 4634
- EventType: 8
- LogName: Security
- Message: Successful
- RecordNumber: 9787
- Sid: S-1-5-21-57989841-920026266-725345543-6444
- SidType: 1
- SourceName: Security
- Type: Success
- User: dhale

Creating a Field Alias

**Settings > Fields > Field Aliases >
New Field Alias**

1. Select the app associated with the field alias
2. Enter a Name for the field alias
3. Apply the field alias to a default field:
 - Host
 - Source
 - Sourcetype
4. Enter the name for the existing field and the new alias

Add new

Fields > Field aliases > Add new

Destination app: class_Fund2

Name*: cisco_firewall_aliases

Apply to: sourcetype | named* | cisco_firewall

Field aliases: Username = user

+ Add another field

existing field name

new field alias

Cancel Save

Creating a Field Alias (cont.)

In this example, one field alias used for new ‘user’ fields in multiple source types

New field alias required for each sourcetype

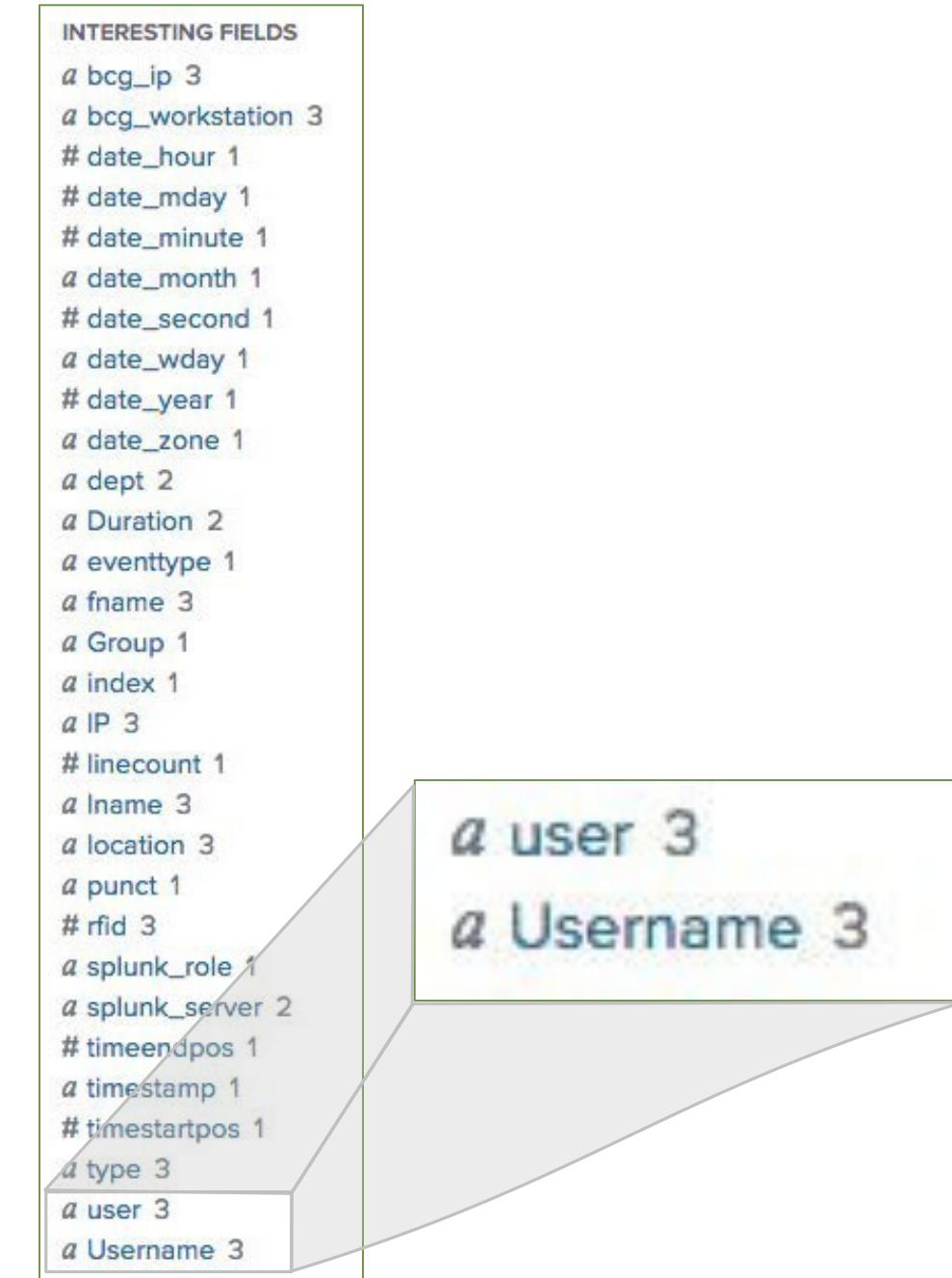
The screenshot shows the Splunk interface for creating field aliases across three separate windows:

- Top Window:** Shows a configuration for a destination app "class_Fund2" with a name "cisco_firewall_aliases" and an "Apply to" setting of "sourcetype".
- Middle Window:** An "Add new" dialog for "Fields > Field aliases > Add new". It shows a destination app "class_Fund2", a name "cisco_wsa_squid_aliases", and an "Apply to" setting of "sourcetype" with a named value "cisco_wsa_squid".
- Bottom Window:** Another "Add new" dialog for "Fields > Field aliases > Add new". It shows a destination app "class_Fund2", a name "winauthentication_security_aliases", and an "Apply to" setting of "sourcetype" with a named value "winauthentication_security".

Each window includes a "Field aliases" section where a field like "username" or "User" is mapped to a new field like "user". Buttons for "Save" and "Cancel" are visible at the bottom right of the bottom window.

Field Alias and Original Fields

- When you create a field alias, the original field is not affected
- Both fields appear in the All Fields and Interesting Fields lists, if they appear in at least 20% of events



Field Aliases and Lookups

After you have defined your field aliases, you can reference them in a lookup table

The screenshot shows a CSV file titled "employees.csv" and a configuration interface for field aliases.

CSV File Content:

```
rfid, fname, lname, user, email, dept, location, ip
108423575302, Allen, Pucci, apucci, apucci@buttercupgames.com, Sales, Boston, 10.3.10.53
672903009231, Dwight, Hale, dhale, dhale@buttercupgames.com, Sales, Boston, 10.3.10.241
398009643042, Phyllis, Bunch, pbunch, pbunch@buttercupgames.com, ITOps, Boston, 10.3.10.227
374765319282, Enrique, Maxwell, emaxwell, emaxwell@buttercupgames.com, ITOps, Boston, 10.3.10.46
227128834140, David, Johnson, djohnson, djohnson@buttercupgames.com, Engineering, Boston, 10.3.10.180
371211812887, Galina, Zuyeva, gzuyeva, gzuyeva@buttercupgames.com, Engineering, Boston, 10.3.10.67
249772079712, Louis, Sagers, lsagers, lsagers@buttercupgames.com, SecOps, Boston, 10.3.10.21
.....
417852300683, Amanda, Curry, acurry, acurry@buttercupgames.com, SecOps, San Francisco, 10.1.10.252
542830538161, Alan, Dombrowski, adombrowski, adombrowski@buttercupgames.com, SecOps, San Francisco, 10.1.10.129
768166372290, Cerys, Farrell, cfarrell, cfarrell@buttercupgames.com, Sales, San Francisco, 10.1.10.107
153218951159, Placido, Toscani, ptoscani, ptoscani@buttercupgames.com, Sales, San Francisco, 10.1.10.38
994499284304, Ian, King, iking, iking@buttercupgames.com, Sales, San Francisco, 10.1.10.201
531253083348, Gabriel, Voronoff, gvoronoff, gvoronoff@buttercupgames.com, Marketing, San Francisco, 10.1.10.163
520156890727, Bao, Lu, blu, blu@buttercupgames.com, Marketing, San Francisco, 10.1.10.100
727896988001, Lien, Teng, lteng, lteng@buttercupgames.com, ITOps, San Francisco, 10.1.10.15
936901629743, Gabriel, Voronoff, gvoronoff, gvoronoff@buttercupgames.com, ITOps, San Francisco, 10.1.10.163
230876363319, Meng, Yuan, myuan, myuan@buttercupgames.com, Engineering, San Francisco, 10.1.10.172
271108583080, Patrick, Callahan, pcallahan, pcallahan@buttercupgames.com, Engineering, San Francisco, 10.1.10.98
569361105570, Kathleen, Percy, kpercy, kpercy@buttercupgames.com, Compliance Officer, San Francisco, 10.1.10.216
.....
145297537706, Nigella, Pearce, npearce, npearce@buttercupgames.com, SecOps, London, 10.2.10.70
632071692298, Yanto, Owen, yowen, yowen@buttercupgames.com, Sales, London, 10.2.10.170
862417886973, Finlay, Bryan, fbryan, fbryan@buttercupgames.com, Sales, London, 10.2.10.166
890313901800, Bradley, Hussain, bhussain, bhussain@buttercupgames.com, ITOps, London, 10.2.10.22
425932411002, Naomi, Sharpe, nsharpe, nsharpe@buttercupgames.com, ITOps, London, 10.2.10.163
....
```

Field aliases Configuration:

Field aliases	Username	=	user
		=	

+ Add another field

A green arrow points from the "user" alias in the configuration interface to the "user" field in the CSV file.

What is a Calculated Field?

- Shortcut for performing repetitive, long, or complex transformations using the `eval` command
- Must be based on an extracted field

New Search

```
index=network sourcetype=cisco_wsa_squid
| eval megabytes = sc_bytes/(1024*1024)
| stats sum(megabytes) as Megabytes by usage
| sort Megabytes
```

Last 24 hours ▾

✓ 1,211 events (1/24/18 12:00:00.000 PM to 1/25/18 12:21:26.000 PM) No Event Sampling ▾ Job ▾ II ▾ Smart Mode ▾

Events Patterns Statistics (5) Visualization

100 Per Page ▾ Format Preview ▾

usage	Megabytes
Violation	0.0172176361083984380
Business	0.29117774963378906000
Borderline	2.04081058502197270000
Unknown	2.59564876556396500000
Personal	7.97706794738769500000

Creating a Calculated Field

Settings > Fields > Calculated Fields > New Calculated Field

1. Select the app that will use the calculated field
2. Select host, source, or sourcetype to apply to the calculated field and specify the related name
3. Name the calculated field
4. Define the eval expression

Add new

Fields > Calculated fields > Add new

Destination app	class_Fund2
Apply to	sourcetype named * cisco_wsa_squid
Name *	megabytes
Name of the field whose value will be calculated	
Ev	megabytes sc_bytes/(1024*1024)
A valid eval expression, e.g. x + 3	
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

Working with Tags and Event Types

Describing Tags

- Tags are like nicknames that you create for related field/value pairs
- Tags make your data more understandable and less ambiguous
- You can create one or more tags for any field/value combination
- Tags are case sensitive

Creating Tags

To create a tag:

1. Click on the arrow for event details
2. Under Actions, click the down arrow
3. Select Edit Tags
4. Name the tags, separated by commas

1/25/18 Thu Jan 25 2018 21:09:26 www2 sshd[43407]: Failed password for user nsharpe from 26.171.161.77 port 3073 ssh2
1:09:26.000 PM

Event Actions ▾

Type	Field	Value	Actions
Selected	host	www3	▼
	source	/opt/log/www3/secure.log	▼
	sourcetype	linux_secure	▼
Event	action	failure(failure)	▼
	eventtype	errOr(error) failed_login nix-all-logs nix_errors(error) nix_security(os unix) sshd_authentication(authentication remote)	▼ ▼ ▼ ▼ ▼ ▼
	user	nsharpe	▼
	vendor_action	Failed	▼
Time +	_time	2018-01-25T13:09:26.000-06:00	▼
Default	index	security	▼
	linecount	1	▼

Edit Tags

> 1/25/18 Thu Jan 25 2018 21:09:26 www2 sshd[43407]: Failed password for user nsharpe
1:09:26.000 PM .77 port 3073 ssh2

host = www3 | source = /opt/log/www3/secure.log | sourcetype = linux_secure

Create Tags

Field Value	user=root
Tag(s)	privileged

Comma or space separated list of tags.

Cancel Save

Viewing Tags

- When tagged field/value pairs are selected, the tags appear:
 - In the results as tags
 - In parentheses next to the associated field/value pairs

The screenshot shows the Splunk user interface with a search results table and a detailed event view.

Event Details:

Field	Value
action	failure(failure)
app	sshd
dest	mailsv1
eventtype	errOr(error)
	failed_login
	nix-all-logs

Event Log:

```
> 1/25/18    Thu Jan 25 2018 21:36:43 mailsv1 sshd[2411]: Failed password for root from 190.113.128.150 port  
1:36:43.000 PM 4655 ssh2  
host = mailsv1 | source = /opt/log/mailsv1/secure.log | sourcetype = linux_secure |  
tag = authentication tag = error tag = failure tag = os tag = privileged tag = remote tag = unix
```

Selected Tags:

Tag	Value
authentication	
error	
failure	
os	
privileged	
remote	
unix	

Selected Fields:

Field	Value
process	sshd
src_ip	190.113.128.150
src_port	4655
sshd_protocol	ssh2
user	root(privileged)

Using Tags

To use tags in a search, use the syntax: `tag=<tag name>`

New Search

Save As ▾ Close

```
index=security sourcetype=linux_secure tag=priv* src_ip!=NULL  
| stats count by src_ip, host
```

Last 24 hours ▾

✓ 228 events (1/24/18 2:00:00.000 PM to 1/25/18 2:02:52.000 PM) No Event Sampling ▾ Job ▾ II III ⌂ ⌃ ⌄ ⌅ Smart Mode ▾

Events Patterns Statistics (192) Visualization

100 Per Page ▾ Format Preview ▾ < Prev 1 2 Next >

src_ip	host	count
107.3.146.207	mailsv1	1
108.65.113.83	www2	1
109.169.32.135	www2	2
110.159.208.78	www2	1
111.161.27.20	www1	1
111.161.27.20	www2	1
111.161.27.20	www3	1
112.111.162.4	www1	1
117.21.246.164	www2	2

Searching for Tags

- To search for a tag associated with a value:
 - `tag=<tagname>`

```
tag=privileged
```

- To search for a tag associated with a value on a specific field:
 - `tag::<field>=<tagname>`

```
tag::user=privileged
```

- To search for a tag using a partial field value:
 - Use (*) wildcard

```
tag=p*
```

Managing Tags – List by Field Value Pair

- **Settings > Tags > List by field value pair**
 - Edit permissions
 - Disable all tags for pair – disables the tag in searches and prevents it from being listed under List by Tag Name and All unique tag objects

List by field value pair						New Tag
Tags » List by field value pair						
Showing 1-2 of 2 items						
App	CLASS: Fundamentals...	Owner	Any	Created in the App	filter	25 per page
Field value pair	Tag name	App	Sharing	Status	Actions	
user=administrator	privileged	class_Fund2	Private	Enabled Disable all tags for pair	Clone Move Delete	
user=root	privileged	class_Fund2	Private	Enabled Disable all tags for pair	Clone Move Delete	

Adding/Changing the Tag Name

Click **List by field value pair** to add another tag or change the name of the tag

List by field value pair

Tags > List by field value pair

Showing 1-2 of 2 items

App CLASS: Fundamentals... Owner Any

Field value pair	Tag name	App
user=administrator	privileged	class_Fund2
user=root	privileged	class_Fund2

user=administrator

Tags > List by field value pair > user=administrator

Tag name Enter one tag per textfield

privileged

+ Add another field

Cancel Save

Adding/Changing the Field Value Pair

Click **List by tag name** to add or edit the field value pair for the tag

The screenshot shows the Splunk interface for managing tags. On the left, the main page is titled "List by tag name" with a "New Tag" button. It includes filters for "App" (CLASS: Fundamentals...), "Owner" (Any), "Created in the App", and a search bar. The results table shows one item: "privileged" with field value pairs "user=administrator, user=root". A green arrow points from this row to a modal window on the right.

Main Page: List by tag name

- Tags > List by tag name
- Showing 1-1 of 1 item
- App: CLASS: Fundamentals...
- Owner: Any
- Created in the App
- filter: Search icon
- 25 per page

Tag name	Field value pair
privileged	user=administrator, user=root

Modal Window: privileged

Tags > List by tag name > privileged

Field value pair: 'example: host=splunk.com'

user=administrator	Delete
user=root	Delete
(empty)	Delete

+ Add another field

Cancel Save

Describing Event Types

- A method of categorizing events based on a search
- A useful method for institutional knowledge capturing and sharing
- Can be tagged to group similar types of events

Creating an Event Type from the Search Page

1. Run a search and verify that all results meet your event type criteria
2. From the Save As menu, select **Event Type**
3. Provide a Name for your event type (name should not contain spaces)

The screenshot illustrates the process of creating an event type. On the left, the 'New Search' interface shows a search query 'index==* status>499' and a results table with three log entries. A green box highlights the search bar. On the right, a modal window titled 'Save As Event Type' is open, showing fields for 'Name' (set to 'web_error'), 'Tags' (set to 'Optional'), 'Color' (set to 'yellow'), and 'Priority' (set to '1 (Highest)'). A green arrow points from the 'Event Type' option in the 'Save As' dropdown on the search page to the 'Save As Event Type' dialog.

New Search

index==* status>499

195 events (1/24/18 3:00:00.000 PM to 1/25/18 3:06:20.000 PM) No Event Sampling

Events (195) Patterns Statistics Visualization

Format Timeline - Zoom Out + Zoom to Selection Deselect

1 hour per column

List Format 20 Per Page < Prev 1 2 3 4 5 6 7 8 9 10 Next >

< Hide Fields All Fields i Time Event

SELECTED FIELDS
host 4 source 4 sourcetype 2 tag 1

INTERESTING FIELDS
action 5 # bytes 100+ categoryId 7 clientip 100+ date_hour 24 date_mday 2 date_minute 59

1/25/18 3:06:04.000 PM 202.91.242.117 -- [25/Jan/2018:23:06:04] "POST /cart.do?action=view&itemId=EST-19&JSESSIONID=SD0SL3FF2ADFF4962 HTTP/1.1" 500 2017 "http://www.buttercupgames.com" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.1; .NET4.0C; .NET4.0E; MS-RTC LM 8)" 609 host = www2 | source = /opt/log/www2/access.log | sourcetype = access_combined | tag = web

1/25/18 3:01:52.000 PM 211.25.254.234 -- [25/Jan/2018:23:01:52] "GET /cart.do?action=addtocart&itemId=EST-11&JSESSIONID=SD4SL4FF1ADFF4952 HTTP/1.1" 500 3569 "http://www.buttercupgames.com/category.screen?categoryId=NULL" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)" 957 host = www3 | source = /opt/log/www3/access.log | sourcetype = access_combined | tag = web

1/25/18 2:56:30.000 PM 195.69.252.22 -- [25/Jan/2018:22:56:30] "GET /category.screen?categoryId=NULL&JSESSIONID=SD8SL10FF6ADFF4960 HTTP/1.1" 500 1189 "http://www.buttercupgames.com/product.screen?productId=SF-BVS-01" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 258 host = www3 | source = /opt/log/www3/access.log | sourcetype = access_combined

Save As Event Type

Name: web_error

Tags: Optional

Color: yellow

Priority: 1 (Highest)

Determines which style wins, when an event has more than one event type.

Cancel Save

Using the Event Type Builder

1. From the event details, select **Event Actions > Build Event Type**

The screenshot shows the Splunk interface for a search titled "New Search". The search bar contains the query "index=*" status>499 and the time range "Last 24 hours". The results section shows 195 events from 1/24/18 to 1/25/18. The "Events (195)" tab is selected. Below the timeline, a single event is selected, showing its details in the table below. The "Event Actions" dropdown menu is open, and the "Build Event Type" option is highlighted with a green border.

Time	Event
1/25/18 3:06:04.000 PM	202.91.242.117 - - [25/Jan/2018:23:06:04] "POST /cart.do?action=view&itemId=EST-19&JSESSIONID=SD0SL3FF2ADFF4962 HT TP 1.1" 500 2017 "http://www.buttercupgames.com" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0. 50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.1; .NET4.0C; .NET4.0E; MS-RTC LM 8)" 609

Event Actions

Action	Value	Actions
Build Event Type		
Extract Fields	www2	
Show Source	/opt/log/www2/access.log	

Using the EventType Builder (cont.)

2. Refine the criteria for your event type such as:
 - Search string
 - Field values
 - Tags

3. Verify your selections and click Save

The screenshot shows the Splunk Event Type Builder interface. At the top, it displays a "Generated event type" search bar with the query "(index==* status>499)". Below the search bar are three buttons: "Edit", "Test", and "Save". To the right of the search bar, there is a "Suggested field values" section and a "Sample events" section.

Suggested field values:

- ident:** -
- linecount:** 1
- tag:** web
- tag:eventtype:** web
- version:** 1.1
- eventtype:** nix-all-logs
 bvg_online_sales
- index:** web
 network
- method:** POST
 GET
- sourcetype:** access_combined
 cisco_wsa_squid

Sample events:

Note: Sample events match the current event type search.

```
202.91.242.117 -- [25/Jan/2018:23:06:04] "POST /cart.do?action=view
211.25.254.234 -- [25/Jan/2018:23:01:52] "GET /cart.do?action=addto
195.69.252.22 -- [25/Jan/2018:22:56:30] "GET /category.screen?cate
223.5.16.102 -- [25/Jan/2018:22:47:52] "GET /cart.do?action=purchas
188.143.232.202 -- [25/Jan/2018:22:38:50] "POST /cart.do?action=purc
188.143.232.202 -- [25/Jan/2018:22:36:32] "GET /oldlink?itemId=EST-
188.143.232.202 -- [25/Jan/2018:22:36:14] "GET /oldlink?itemId=EST-
87.194.216.51 -- [25/Jan/2018:22:30:52] "GET /product.screen?produ
1516919448.147 48958 94.229.0.21 NONE/503 1860 GET http://gen6iz.co
94.229.0.20 -- [25/Jan/2018:22:20:50] "GET /cart.do?action=addtocar
88.12.32.208 -- [25/Jan/2018:22:16:47] "GET /oldlink?itemId=EST-128
210.192.123.204 -- [25/Jan/2018:22:13:18] "GET /product.screen?prod
1516917581.325 48948 209.160.24.63 NONE/503 1863 GET http://trucov.
128.241.220.82 -- [25/Jan/2018:21:59:38] "GET /oldlink?itemId=EST-1
1516917509.028 120 92.1.170.135 NONE/503 1872 GET http://hitpresent.
209.114.36.109 -- [25/Jan/2018:21:52:10] "GET /cart.do?action=addto
221.207.229.6 -- [25/Jan/2018:21:19:17] "GET /oldlink?itemId=EST-13
188.173.152.100 -- [25/Jan/2018:21:11:37] "GET /cart.do?action=addt
212.58.253.71 -- [25/Jan/2018:21:04:10] "GET /oldlink?itemId=EST-21
```

Generated event type:

(index==* status>499)

Buttons: Edit, Test, Save

Suggested field values:

Check the below field values, to build up your new event type. Blue field values are from your selected event; other values are from other events.

Ident: -

Linecount: 1

Tag: web

Tag:eventtype: web

Version: 1.1

Eventtype: nix-all-logs
 bvg_online_sales

Index: web
 network

Method: POST
 GET

Sourcetype: access_combined
 cisco_wsa_squid

Generated event type:

(index==* status>499)

Buttons: Edit, Test, Save

Suggested field values:

Check the below field values, to build up your new event type. Blue field values are from your selected event; other values are from other events.

Ident: -

Linecount: 1

Tag: web

Tag:eventtype: web

Version: 1.1

Eventtype: nix-all-logs
 bvg_online_sales

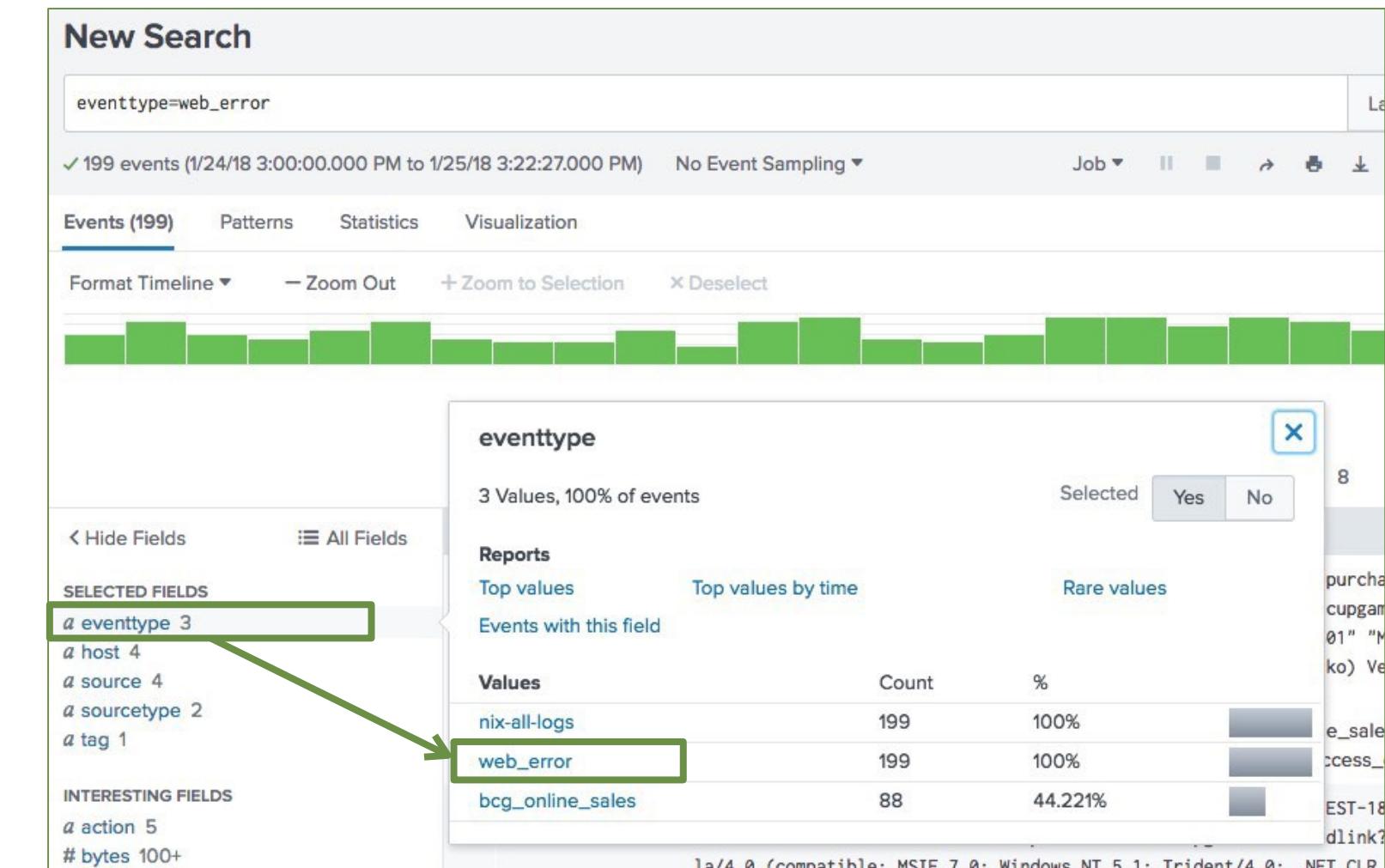
Index: web
 network

Method: POST
 GET

Sourcetype: access_combined
 cisco_wsa_squid

Using Event Types

- To verify the event type, search for `eventtype=web_error`
- ‘eventtype’ displays in the Fields sidebar and can be added as a selected field
- Splunk evaluates the events and applies the appropriate event types at search time
- Using the Fields sidebar, you can easily view the individual event types, the number of events, and percentage



Tagging Event Types

You can tag event types two ways:

1. **Settings > Event Types**
2. **Event details > Actions**

Event Actions ▾

Type	Field	Value	Actions
Selected	<input checked="" type="checkbox"/> eventtype	nix-all-logs	▼
		web_error(error web)	▼
		bog_online_sales(wet)	▼
<input checked="" type="checkbox"/> host		www1	Edit Tags

Create Tags

Field Value: eventtype=web_error

Tag(s): error, web

Comma or space separated list of tags.

Cancel Save

web_error

Event types > web_error

Search string * index==* status>499

Tag(s) error web

Enter a comma-separated list of tags.

Color yellow

Priority 1 (Highest)

Highest priority shows up first in a result.

Cancel Save

Priority determines which event type color displays for an event

Event Types vs. Saved Reports

- Event Types
 - Categorize events based on a search string
 - Tag event types to organize data into categories
 - The `eventtype` field can be included in a search string
 - Does not include a time range
- Saved Reports
 - Search criteria will not change
 - Includes a time range and formatting of the results
 - Can be shared with Splunk users and added to dashboards

Creating and Using Macros

Macros Overview

- Useful when you frequently run searches or reports with similar search syntax
- The time range is selected at search time
- Macros can be a full search string or a portion of a search that can be reused in multiple places
- Allows you to define one or more arguments within search segment
 - Pass parameter values to macro at execution time
 - Macro uses values to resolve search string

Creating a Basic Macro

Settings > Advanced search > Search Macros

1. Click New Search Macro
2. Select the destination app
3. Enter a name
4. Type the search string
5. Save



Add new

Advanced search > Search macros > Add new

Destination app class_Fund2

Destination app class_Fund2

Name * Enter the name of the macro. If the search macro takes an argument, enter the argument name followed by a colon and the argument value. For example: mymacro(2)

US_sales

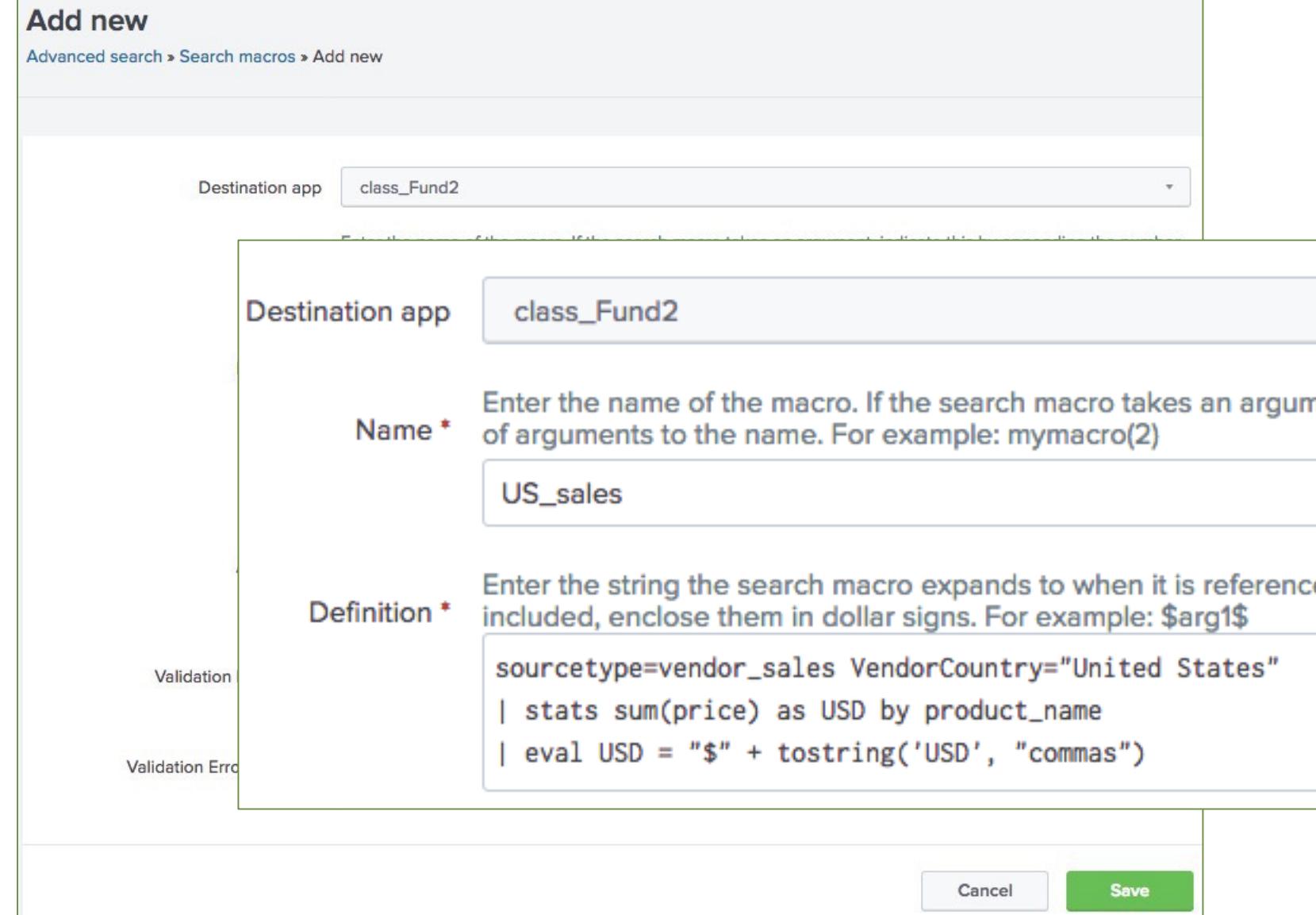
Definition * Enter the string the search macro expands to when it is referenced. You can include variables, enclose them in dollar signs. For example: \$arg1\$

sourcetype=vendor_sales VendorCountry="United States"
| stats sum(price) as USD by product_name
| eval USD = "\$" + tostring('USD', "commas")

Validation

Validation Err

Cancel Save



Using a Basic Macro

- Type the macro name into the search bar
- Surround the macro name with the **backtick** (or grave accent) character
 - **`macroname`** != 'macroname'
 - Do not confuse with single-quote character (')
- Pipe to more commands, or precede with search string

The screenshot shows a search interface with a search bar containing the macro name `US_sales`. Below the search bar is a table of sales data.

Macro Definition:

```
sourcetype=vendor_sales  
VendorCountry="United States"  
| stats sum(price) as USD by product_name  
| eval USD = "$" + tostring('USD', "commas")
```

Table Data:

product_name	USD
Benign Space Debris	\$349.86
Curling 2014	\$779.61
Dream Crusher	\$2,959.26
Final Sequel	\$824.67
Fire Resistance Suit of Provolone	\$207.48
Holy Blade of Gouda	\$173.71
Manganiello Bros.	\$1,879.53
Manganiello Bros. Tee	\$599.40
Mediocre Kingdoms	\$799.68

Adding Arguments

- Include the number of arguments in parentheses after the macro name
 - `monthly_sales(3)`
- Within the search definition, use `arg`
 - `currency=$currency$`
 - `symbol=$symbol$`
 - `rate=$rate$`
- In the **Arguments** field, enter the name of the argument(s)
- Provide one or more variables of the macro at search time

Add new

Advanced search > Search macros > Add new

Destination app class_Fund2

Destination app class_Fund2

Name * Enter the name of the macro. If the search macro takes an argument, indicate the number of arguments to the name. For example: mymacro(2)
monthly_sales(3)

Definition * Enter the string the search macro expands to when it is referenced in another search macro. You can include variables included, enclose them in dollar signs. For example: \$arg1\$
`stats sum(price) as USD by product_name
| eval $currency$ = "$symbol$" + tostring(USD*$rate$, "commas")
| USD = "$" + tostring(USD, "commas")`

Use eval-based definition?

Arguments Enter a comma-delimited string of argument names. Argument names may only contain letters, numbers, and '-' characters.
currency,symbol,rate

Validating Macros

You can validate argument values in your macro

- Validation Expression: you can enter an expression for each argument
 - Argument must be enclosed in dollar signs
- Validation Error Message: message that appears when you run the macro

Arguments Enter a comma-delimited string of argument names. Argument names may only contain alphanumeric, '_' and '-' characters.
currency,symbol,rate

Validation Expression Enter an eval or boolean expression that runs over macro arguments.
isnum(\$rate\$)

Validation Error Message Enter a message to display when the validation expression returns 'false'.
The 'rate' value must start with a number

New Search

index= sales sourcetype=vendor_sales VendorCountry=Germany OR VendorCountry=France OR VendorCountry=Italy | 'monthly_sales(euro,€,\$.79)' Last 7 days ▾

! Error in 'SearchParser': Encountered the following error while validating macro 'monthly_sales(euro,€,\$.79)': The 'rate' value must start with a number.

Creating and Using WorkflowActions

What are Workflow Actions?

- Execute workflow actions from an event in your search results to interact with external resources or run another search
 - **GET** - retrieve information from an external resource
 - **POST** - send field values to an external resource
 - **Search** - use field values to perform a secondary search

The screenshot shows a Splunk event details page. At the top, there is a timestamp and source information: "1/26/18 1:38:17.210 PM 1517002697.210 68 69.80.0.18 TCP_REFRESH_HIT/200 977 GET http://www.manta.com/manta /images/header_and_nav/btn_search.gif lsagers@buttercupgames.com DIRECT/www.manta.c om - ALLOW_WBRS-DefaultGroup-Demo_Clients-NONE-NONE-DefaultRouting <IW_busi,6.5,-,-,-,-,-,-,-,-,-,-,-,-,-> - http://www.manta.com/". Below this is a "Event Actions" button. A dropdown menu is open, showing the following options:

- Build Event Type
- Get info for IP:69.80.0.18** (This option is highlighted with a green border.)
- Extract Fields
- Show Source

To the right of the dropdown, there is a table with two columns: "Value" and "Actions". The table contains the following data:

Value	Actions
nix-all-logs	▼
cisco_router1	▼
/opt/log/cisco_router1	▼
/cisco_ironport_web.log	▼

Creating a GET Workflow Action

Settings > Fields > Workflow actions > New Workflow Action

1. Select the app
2. Name the workflow action with no spaces or special characters
3. Define the label, which will appear in the Event Action menu
4. Determine if your workflow action applies to a field or event type

Add new

Fields » Workflow actions » Add new

Destination app	class_Fund2
Name *	get_whois_info
Enter a unique name without spaces or special characters later on within Splunk Settings.	
Label *	Get info for IP:\$src\$
Enter the label that appears for this action. Only use dollar signs, e.g. 'Search for ticket number'	
Apply only to the following fields	src
Specify a comma-separated list of fields that it. When fields are specified, the workflow action appears in all field menus.	
Apply only to the following event types	
Specify a comma-separated list of event types to apply to it.	

Creating a GET Workflow Action (cont.)

5. From the **Show action in** dropdown list, select **Event menu**
6. From Action type dropdown list, select link
7. Enter the URI of where the user will be directed
8. Specify if the link should open in a New window or Current window
9. Select the Link method of get
10. Save

The screenshot shows a configuration interface for a workflow action. The fields are as follows:

- Show action in:** Event menu
- Action type ***: link
- URI ***: http://who.is/whois-ip/ip-address/\$src\$
A tooltip below the field says: "Enter the location to link to. Optionally, specify fields by enclosing the field name in dollar signs, e.g. http://www.google.com/search?q=\$host\$."
- Open link in:** New window
- Link method:** get

At the bottom right are two buttons: **Cancel** and **Save**.

Testing the GET Workflow Action

1/26/18 1517002697.210 68 69.80.0.18 TCP_REFRESH_HIT/200 977 GET http://www.manta.com/manta/images/header_and_nav/btn_search.gif lsagers@buttercupgames.com DIRECT/www.manta.com - ALLOW_WBRS-DefaultGroup-Demo_Clients-NONE-NONE,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,IW_busi,-> - http://www.manta.com

Event Actions ▾

Build Event Type	Value
Get info for IP:69.80.0.18	nx-all-logs
Extract Fields	cisco_router1
Show Source	/opt/log/cisco/cisco_ironport

IP Information for 69.80.0.18

Quick Stats

IP Location	Barbados Bridgetown Cable & Wireless (barbados) Limited
ASN	AS14813 BB-COLUMBUS - Columbus Telecommunications (Barbados) Limited, BB (registered Mar 15, 2007)
Whois Server	whois.arin.net
IP Address	69.80.0.18

NetRange: 69.80.0.0 - 69.80.63.255
CIDR: 69.80.0.0/18
NetName: BDS-NET9
NetHandle: NET-69-80-0-0-1
Parent: NET69 (NET-69-0-0-0-0)
NetType: Direct Allocation
OriginAS: AS11139
Organization: Cable & Wireless (Barbados) Limited (CWBL-1)
RegDate: 2010-03-23
Updated: 2012-03-02
Ref: <https://whois.arin.net/rest/net/NET-69-80-0-0-1>

Creating a POST Workflow Action

**Settings > Fields >
Workflow actions > New
Workflow Action**

Complete steps 1 – 6 as described in the previous example, Creating a GET Workflow Action

Add new

Fields > Workflow actions > Add new

Destination app	class_Fund2
Name *	multiple_attempts_to_open_ports
Enter a unique name without spaces or special characters later on within Splunk Settings.	
Label *	Create ticket - multiple attempts to port:\$port\$
Enter the label that appears for this action. Optionally, in dollar signs, e.g. 'Search for ticket number : \$ticketn	
port	Specify a comma-separated list of fields that must be present. When fields are specified, the workflow action only appears in all field menus.
Specify a comma-separated list of event types that an event must have to apply to it.	
Show action in	Event menu
Action type *	link

Creating a POST Workflow Action (cont.)

7. Enter the URI of where the user will be directed
8. Open the link in a **New window** or Current window
9. Select the Link method of **post**
10. Provide post argument parameters
11. Save

Link configuration

URI * Enter the location to link to. Optionally, specify fields by enclosing the field name in dollar signs, e.g. http://www.google.com/search?q=\$host\$.

Open link in

Link method

Post arguments

<input type="text" value="title"/>	= <input type="text" value="Multiple attempts to open\$port\$"/>	<input type="button" value="Delete"/>
<input type="text" value="description"/>	= <input type="text" value="\$_raw\$"/>	<input type="button" value="Delete"/>

Creating a Search Workflow Action

Settings > Fields > Workflow actions> New

Complete steps 1 – 5 as described in the previous example, Creating a GET Workflow Action

6. From the Action type dropdown list, select **search**

Add new

Fields » Workflow actions » Add new

Destination app	class_Fund2
Name *	search_login-by_IP
Enter a unique name without spaces or punctuation. It can be changed later on within Splunk Settings.	
Label *	Search failed login from IP:\$src\$
Enter the label that appears for this action in the UI. Use dollar signs, e.g. 'Search for ticket number \$id\$'.	
Apply only to the following fields	src
Specify a comma-separated list of fields to apply to it. When fields are specified, the workflow appears in all field menus.	
Apply only to the following event types	
Specify a comma-separated list of event types to apply to it.	
Show action in	Event menu
Action type *	search

Creating a Search Workflow Action (cont.)

7. Enter the Search string
8. Select the app if it is different from the current app
9. Enter the view name where the search will execute
10. Indicate if the search should run in a New window or the Current window
11. Enter the time range for the search or choose to use the same time range as the search
12. Save

Search configuration

Search string *

Enter the search for this action. Optionally, specify fields as \$fieldname\$, e.g. sourcetype=rails controller=\$controller\$ error=^.

Run in app

Choose an app for the search to run in. Defaults to the current app.

Open in view

Enter the name of a view for the search to open in. Defaults to the current view.

Run search in

Time range

Earliest time

Latest time

Use the same time range as the search that created the field listing

Testing the Search Workflow Action

The screenshot shows the Splunk interface with a search results page and a sidebar.

Search Results:

- Event Actions:** A dropdown menu containing:
 - Build Event Type
 - Extract Fields
 - Search failed login from IP:188.173.152.100** (highlighted with a green box)
 - Show Source
- New Search:** A search bar containing the query `index=security sourcetype=linux_secure failed src_ip=188.173.152.100`. Below it is a timeline visualization showing event counts over time.
- Events (22):** A table listing 22 events. The first two events are shown in detail:

Time	Event
Fri Jan 26 2018 22:09:38	Failed password for invalid user administrator from 188.173.152.100 port 3454 ssh2 eventtype = errOr error eventtype = failed_login eventtype = nix-all-logs eventtype = nix_errors error eventtype = nix_security os unix eventtype = sshd_authentication authentication remote host = www2 source = /opt/log/www2/secure.log sourcetype = linux_secure tag = authentication tag = error tag = failure tag = os tag = privileged tag = remote tag = unix
Fri Jan 26 2018 21:59:13	Failed password for invalid user rdb from 188.173.152.100 port 1623 ssh2 eventtype = errOr error eventtype = failed_login eventtype = nix-all-logs eventtype = nix_errors error eventtype = nix_security os unix eventtype = sshd_authentication authentication remote host = www2 source = /opt/log/www2/secure.log sourcetype = linux_secure tag = authentication tag = error tag = failure tag = os tag = remote tag = unix

Sidebar:

- Selected Fields:** eventtype 6, host 4, source 4, sourcetype 1, tag 7
- Interesting Fields:** action 1, app 1, date_hour 9, date_mday 2, date_minute 20

Creating Data Models

Reviewing Pivot

- Used for creating reports and dashboards (discussed in Splunk Fundamentals 1)
- As a knowledge manager, you're responsible for building the data model that provides the datasets for Pivot

The screenshot shows two main Splunk interfaces. On the left is the 'Datasets' listing page, which displays a table of 9 datasets. One dataset, 'Buttercup Games Online Sales > Web Requests', has its 'Actions' menu open, showing options like 'Explore', 'Visualize with Pivot' (which is highlighted with a green box and an arrow pointing to the right), and 'Investigate in Search'. On the right is the 'New Pivot' interface, which is a powerful reporting tool. It includes sections for 'Filters' (set to 'Last 30 days'), 'Split Columns' (with a field for 'product name'), 'Split Rows' (with a field for 'action'), and 'Column Values' (including a 'Count of Web...'). The main area is a pivot table with columns for various actions and products. A sample of the data is shown below:

action	Benign Space Debris	Curling 2014	Dream Crusher	Final Sequel	Fire Resistance Suit of Provolone	Holy Blade of Gouda	Manganiello Bros.	Manganiello Bros. Tee	Mediocre Kingdoms	NULL	Orvil the Wolverine	Puppies vs. Zombies	SIM Cubicle	World of Cheese
addtocart	807	767	1183	1041	1123	959	1131	1038	1272	281	835	895	1291	1317
changequantity	179	165	305	239	250	225	277	223	292	129	209	227	293	278
purchase	460	398	661	569	616	577	611	575	685	118	482	512	707	730
remove	170	171	274	231	267	188	247	267	301	120	188	190	300	277
view	689	697	1136	939	1066	912	975	996	1141	503	753	823	1136	1227

Overview of Data Models

- Hierarchically structured datasets that generate searches and drive Pivot
- Pivot reports are created based on datasets
- Each event, search or transaction is saved as a separate dataset

The screenshot shows the Splunk Enterprise interface for managing data models. The title bar indicates 'splunk>enterprise' and 'App: Search & Reporting'. The main page is titled 'Buttercup Games Site Activity' under the 'Buttercup_Games_Site_Activity' dataset. On the left, there's a sidebar with 'Datasets' and an 'Add Dataset' button. Below it, under 'EVENTS', is a section for 'Web Requests' which is currently selected. This section contains two main categories: 'Successful Requests' (with sub-options like 'purchases', 'addtocart', 'remove') and 'Failed Requests' (with sub-options like 'failed purchases', 'failed addtocart', 'failed remove'). To the right, the 'Web_Requests' data model is detailed. It includes a 'CONSTRAINTS' section with the query 'index=web sourcetype=access_combined' and a 'Constraint' button. Below this are sections for 'INHERITED' fields (_time, host, source, sourcetype) and 'EXTRACTED' fields (action, bytes, categoryId, change_type, clientip, cookie, status). Each field has a checkbox, a type indicator (Time, String, Number, IPv4), and an 'Override' or 'Edit' button.

Data Model Dataset Types

A data model can consist of 3 types of datasets

- Events
- Searches
- Transactions

Events

Searches

Transactions

Buttercup Games Site Activity

Buttercup_Games_Site_Activity

< All Data Models

Datasets Add Dataset ▾

EVENTS

Web Requests

Successful Requests

- purchases
- addtocart
- remove

Failed Requests

- failed purchases
- failed addtocart
- failed remove

SEARCHES

User

TRANSACTIONS

visit duration

Web Requests Web_Requests

CONSTRAINTS

index=web sourcetype=access_combined Constraint Edit

Bulk Edit ▾

INHERITED

_time	Time	
<input type="checkbox"/> host	String	Override
<input type="checkbox"/> source	String	Override
<input type="checkbox"/> sourcetype	String	Override

EXTRACTED

<input type="checkbox"/> action	String	Edit
<input type="checkbox"/> bytes	Number	Edit
<input type="checkbox"/> categoryId	String	Edit
<input type="checkbox"/> change_type	String	Edit
<input type="checkbox"/> clientip	IPv4	Edit
<input type="checkbox"/> cookie	String	Edit

Rename Delete Add Field ▾

Data Model Events

- **Event datasets** contain constraints and fields
- **Constraints** are essentially the search broken down into a hierarchy
- **Fields** are properties associated with the events

The screenshot shows a user interface for managing event datasets. On the left, there's a sidebar titled 'EVENTS' containing a tree view of 'Web Requests' events, which are further divided into 'Successful Requests' (with sub-items like 'purchases', 'addtocart', 'remove') and 'Failed Requests' (with sub-items like 'failed purchases', 'failed addtocart', 'failed remove'). To the right, the main panel is titled 'Web Requests' and shows its 'CONSTRAINTS' section with the query: `index=web sourcetype=access_combined`. A yellow box highlights this section and is labeled 'base search'. Below the constraints, there's an 'INHERITED' section with the field `_time` (Time type). The main area then splits into two columns: 'fields' (left) and 'EXTRACTED' (right). The 'fields' column lists `host` (String), `source` (String), and `sourcetype` (String), each with an 'Override' link. The 'EXTRACTED' column lists several fields with their types and edit links: `action` (String), `bytes` (Number), `categoryid` (String), `change_type` (String), `clientip` (IPv4), `cookie` (String), and `date_hour` (Number). A yellow box highlights the 'fields' section and is labeled 'fields'. At the top right of the main panel, there are 'Rename' and 'Delete' buttons.

Event Object Hierarchy and Constraints

Datasets

Web Requests

EVENTS

CONRAINTS

base search – all access_combined events

Each constraint inherits the parent search string

Successful Requests

purchases

addtocart

remove

Failed Requests

failed purchase

failed addtocart

failed remove

Datasets

Events

Web Requests

Successful Requests

purchases

addtocart

remove

Failed Requests

failed purchases

failed addtocart

failed remove

Constraint

Edit

Rename

Delete

index=web sourcetype=access_combined

status<400

successful requests

Inherited

Successful Requests

purchases

addtocart

remove

Failed Requests

failed purchases

failed addtocart

failed remove

Datasets

Events

Web Requests

Successful Requests

purchases

addtocart

remove

Failed Requests

failed purchases

failed addtocart

failed remove

INHERITED

_time

Time

action

String

bytes

Number

Bulk Edit

Add Field

index=web sourcetype=access_combined

status<400

action=purchase productId=*

successfull requests for purchases

Inherited

Dataset Fields

- Select the fields you want to include in the dataset
- Like constraints, fields are inherited from parent objects

The screenshot shows a dataset configuration page for "Web Requests". At the top right are "Rename" and "Delete" buttons. Below the title "Web Requests" and subtitle "Web_Requests" is a "CONSTRAINTS" section containing the constraint "index=web sourcetype=access_combined" with "Constraint" and "Edit" buttons. A "Bulk Edit" dropdown is also present. The main area is divided into "INHERITED" and "EXTRACTED" sections. The "INHERITED" section lists fields: "_time" (Time), "host" (String, Override), "source" (String, Override), and "sourcetype" (String, Override). The "EXTRACTED" section lists fields: "action" (String, Edit), "bytes" (Number, Edit), "categoryid" (String, Edit), "change_type" (String, Edit), "clientip" (IPv4, Edit), "cookie" (String, Edit), "date_hour" (Number, Edit), and "status" (Number, Edit).

Web Requests		
Web_Requests		
CONSTRAINTS		
index=web sourcetype=access_combined	Constraint	Edit
Bulk Edit ▾		
INHERITED		
_time	Time	
<input type="checkbox"/> host	String	Override
<input type="checkbox"/> source	String	Override
<input type="checkbox"/> sourcetype	String	Override
EXTRACTED		
<input type="checkbox"/> action	String	Edit
<input type="checkbox"/> bytes	Number	Edit
<input type="checkbox"/> categoryid	String	Edit
<input type="checkbox"/> change_type	String	Edit
<input type="checkbox"/> clientip	IPv4	Edit
<input type="checkbox"/> cookie	String	Edit
<input type="checkbox"/> date_hour	Number	Edit
<input type="checkbox"/> status	Number	Edit

Creating a Data Model

Settings > Data Models

The screenshot shows the Splunk Enterprise interface with the following details:

- Header:** splunk>enterprise, Apps ▾, student1 ▾, Messages ▾, Settings ▾, Activity ▾, Help ▾, Find, Search icon.
- Data Models Page:** Displays 24 Data Models. A green arrow points from the "New Data Model" button in the top right to the "New Data Model" dialog box.
- Table Headers:** i, Title ▾, Type ▾, Actions, App ▾.
- Table Rows:** Alerts, Application State, Authentication, Buttercup Games Online Sales, all listed as data models.
- New Data Model Dialog:** A modal window titled "New Data Model".
 - Title:** Buttercup Games Site Activity
 - ID:** Buttercup_Games_Site_Activity
Note: Can only contain letters, numbers and underscores.
 - App:** Search & Reporting ▾
 - Description:** optional
- Buttons:** Cancel, Create.

Adding a Root Event

The diagram illustrates the process of adding a root event dataset. It starts with a main interface titled "Buttercup Games Site Activity" showing a "Root Event" option in the "Add Dataset" menu. A green arrow points from this menu to a detailed "Add Event Dataset" configuration screen.

Add Event Dataset
Data Model: Buttercup Games Site Activity

Dataset Name: Web Requests
Dataset ID ? Web_Requests
Can only contain letters, numbers and underscores.

Constraints: index=web sourcetype=access_combined

Examples:
uri="*.php*" OR uri="*.py*"
NOT (referer=null OR referer="-")

Buttons: Cancel, Preview, Save

A yellow callout box on the right side of the "Constraints" field states: "Constraints are essentially search terms – add child events (discussed later in this section) to further "narrow" your search".

Below the main interface, a preview window shows the results of the applied constraint. It displays a sample of 1,000 events with the following log entries:

```
90.205.111.169 - - [29/Jan/2018:18:56:34] "GET /oldlink?itemId=EST-15&JSESSIONID=SD3SL8FF8ADFF4965 HTTP 1.1" 200 2732 "http://www.buttercupgames.com/oldlink?itemId=EST-15" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; MS-RTC LM 8; InfoPath.2)" 987
90.205.111.169 - - [29/Jan/2018:18:56:19] "POST /cart/success.do?JSESSIONID=SD3SL8FF8ADFF4965 HTTP 1.1" 200 867 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-6" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; MS-RTC LM 8; InfoPath.2)" 735
90.205.111.169 - - [29/Jan/2018:18:56:19] "POST /cart.do?action=purchase&itemId=EST-6&JSESSIONID=SD3SL8FF8ADFF4965 HTTP 1.1" 200 3202 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-6&categoryId=ARCADE&productId=BS-AG-C09" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; MS-RTC LM 8; InfoPath.2)" 815
```

A yellow callout box in the preview window says: "Click Preview to view the events that the constraint returns".

Adding a Root Event (cont.)

- In this example, the root event of this data model represents all web requests
- The Inherited attributes are default fields
- Use **Add Field > Auto-Extracted** to add more fields

Buttercup Games Site Activity
Buttercup_Games_Site_Activity

[Edit](#) [Download](#) [Pivot](#) [Documentation](#)

[All Data Models](#)

Datasets [Add Dataset](#)

EVENTS

Web Requests

Web_Requests

[Rename](#) [Delete](#)

CONSTRAINTS

index=web sourcetype=access_combined

Constraint [Edit](#)

Bulk Edit

[Add Field](#)

INHERITED

<input type="checkbox"/>	_time	Time
<input type="checkbox"/>	host	String
<input type="checkbox"/>	source	String
<input type="checkbox"/>	sourcetype	String

Calculated fields are processed in the order above, so ensure any dependent fields are defined first. Drag to rearrange.

Auto-Extracted

- Eval Expression
- Lookup
- Regular Expression
- Geo IP

Adding Fields

- **Auto-Extracted** – can be default fields or manually extracted fields
- **Eval Expression** – a new field based on an expression that you define
- **Lookup** – leverage an existing lookup table
- **Regular Expression** – extract a new field based on regex
- **Geo IP** – add geographical fields such as latitude/longitude, country, etc.

Web Requests
Web_Requests

CONSTRAINTS

index=web sourcetype=access_combined

Bulk Edit ▾

INHERITED

	Type	Action
_time	Time	
host	String	Edit
source	String	Edit
sourcetype	String	Edit

EXTRACTED

	Type	Action
action	String	Edit
bytes	Number	Edit
categoryid	String	Edit
change_type	String	Edit
clientip	IPv4	Edit
cookie	String	Edit
date_hour	Number	Edit
status	Number	Edit

Rename Delete

Add Field ▾

Auto-Extracted
Eval Expression
Lookup
Regular Expression
Geo IP

Adding Fields – Auto-Extracted

Fields that already exist for the constraint can be added as attributes to the data model

Add Auto-Extracted Field

Sample: 1,000 events ✓ 1,000 events (before 1/29/18 11:10:13.000 AM) Missing field? Add by Name

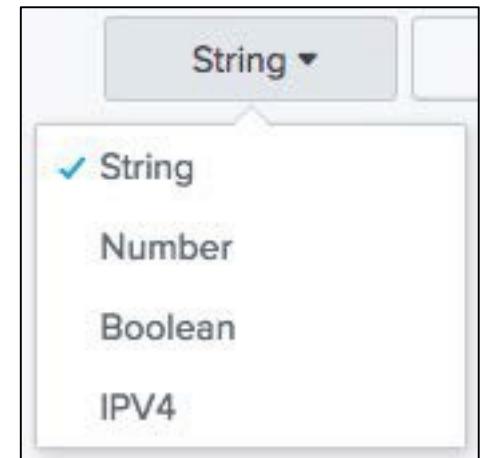
<input type="checkbox"/> Field Name	Display Name	Type and Flags
> <input type="checkbox"/> JSESSIONID		
✓ <input checked="" type="checkbox"/> action Example values: purchase addtocart view remove changequantity	action	String ▾ Optional ▾
> <input type="checkbox"/> app		
> ✓ <input checked="" type="checkbox"/> bytes	size	Number ▾ Optional ▾
> ✓ <input checked="" type="checkbox"/> categoryId	category	String ▾ Optional ▾

View a field's example values

Give the field a friendly name for use in Pivot

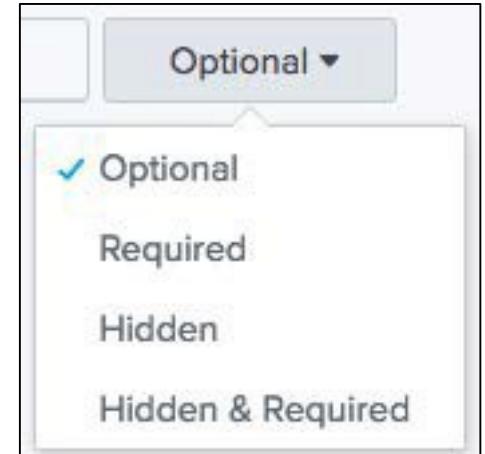
Field Types

- String: Field values are recognized as alpha-numeric
- Number: Field values are recognized as numeric
- Boolean: Field values are recognized as true/false or 1/0
- IPV4: Field values are recognized as IP addresses
 - This is an important field type, as at least one IPV4 attribute type must be present in the data model in order to add a Geo IP attribute



Field Flags

- Optional: This field doesn't have to appear in every event
- Required: Only events that contain this field are returned in Pivot
- Hidden: This field is not displayed to Pivot users when they select the dataset in Pivot
 - Use for fields that are only being used to define another field, such as an eval expression
- Hidden & Required: Only events that contain this field are returned, and the fields are hidden from use in Pivot



Adding Fields – Eval Expressions

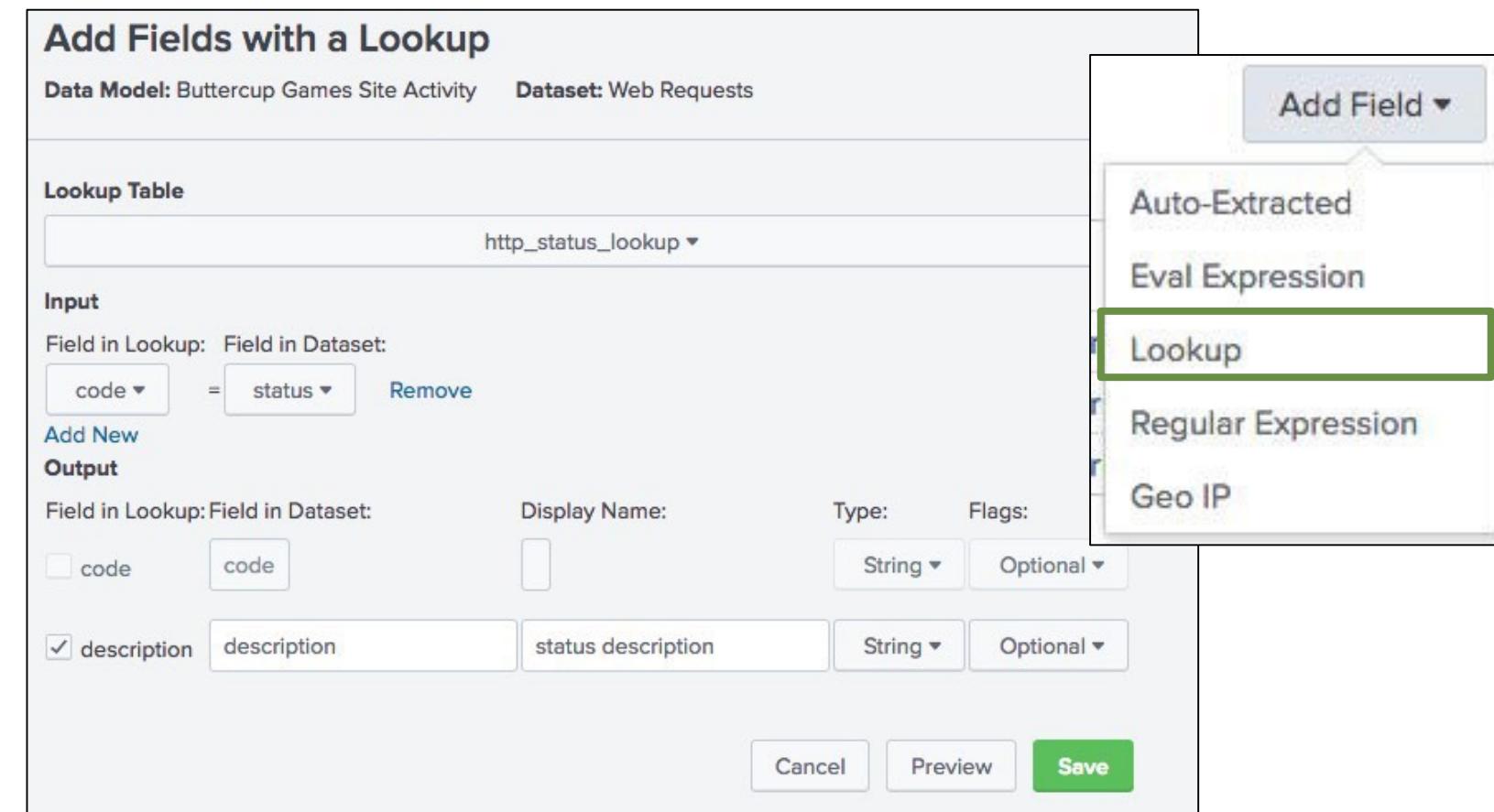
- You can define a new field using an eval expression
 - In this example, you create a field named Error Reason that evaluates the value of the status field

The screenshot shows the 'Add Fields with an Eval Expression' interface in Splunk. On the left, a sidebar has 'Eval Expression' selected. The main area shows an eval expression: `if(status>399,"Web error","OK")`. To the right, a 'Field' section defines a new field named `errorReason` with a `Display Name: Error Reason`, `Type: String`, and `Flags: Optional`. Below this is a preview table showing event samples. A yellow callout bubble with the text 'Click Preview to verify your eval expression returns events' points to the 'Preview' button at the bottom right of the preview table.

_time	errorReason	host	source	sourcetype	JSESSIONID	action	app	bytes	categoryId	change_type
2018-01-29 11:14:59	OK	www3	/opt/log/www3/access.log	access_combined	SD2SL2FF3ADFF4952			1327	STRATEGY	
2018-01-29 11:14:51	OK	www3	/opt/log/www3/access.log	access_combined	SD2SL2FF3ADFF4952	changequantity		1639		
2018-01-29 11:14:37	OK	www3	/opt/log/www3/access.log	access_combined	SD2SL2FF3ADFF4952	view		800		
2018-01-29 11:14:22	OK	www3	/opt/log/www3/access.log	access_combined	SD2SL2FF3ADFF4952			1978	ARCADE	

Adding Fields – Lookups

- Leverage an existing lookup definition to add fields to your event object
- Configure the lookup attribute in the same way as an automatic lookup



Adding Fields – Lookups (cont.)

- Use Preview to test your lookup settings
- Use the Events and Values tab to verify your results

Add Fields with a Lookup

Data Model: Buttercup Games Site Activity Dataset: Web Requests Documentation

Lookup Table http_status_lookup

Input
Field in Lookup: Field in Dataset:
 = Remove

Add New Output
Field in Lookup: Field in Dataset:
 code
 description status

Events **Values**

✓ 1,000 events (before 1/29/18 11:59:20.000 AM)

Sample: 1,000 events

20 per page

Values	Count	%
OK.	881	88.100
Bad Request.	20	2.000
Internal Server Error.	20	2.000
Service Unavailable.	19	1.900
Request Timeout.	16	1.600
Not Found.	14	1.400
HTTP Version Not Supported.	13	1.300
Not Acceptable.	12	1.200
Forbidden.	5	0.500

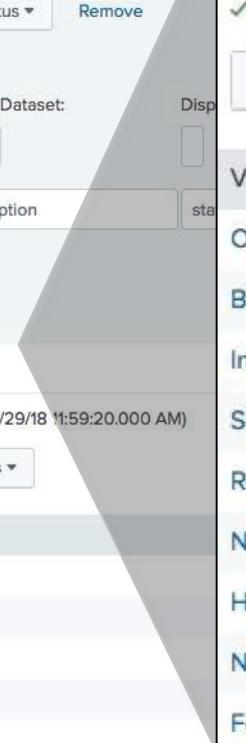
Events **Values**

✓ 1,000 events (before 1/29/18 11:59:20.000 AM)

Sample: 1,000 events

Values

OK.
Bad Request.
Internal Server Error.
Service Unavailable.
Request Timeout.
Not Found.
HTTP Version Not Supported.
Not Acceptable.
Forbidden.



Adding Fields – Regular Expression

You can define a new field using a regular expression

The screenshot shows the 'Add Fields with a Regular Expression' interface. In the top right corner, a vertical menu has 'Regular Expression' highlighted with a green border. A green arrow points from this highlighted item down to the 'Matches' tab in the event list below.

Extract From: _raw

Regular Expression: userAgent=(?<browser>[*]{0,1})

Field(s):

- Field Name: browser
- Display Name: browser
- Type: String
- Flags: Optional

Events: browser

✓ 1,000 events (before 1/29/18 12:12:34.000 PM)

20 per page ▾ < Prev 1 2 3 4 5 6 7 8 9 ... Next >

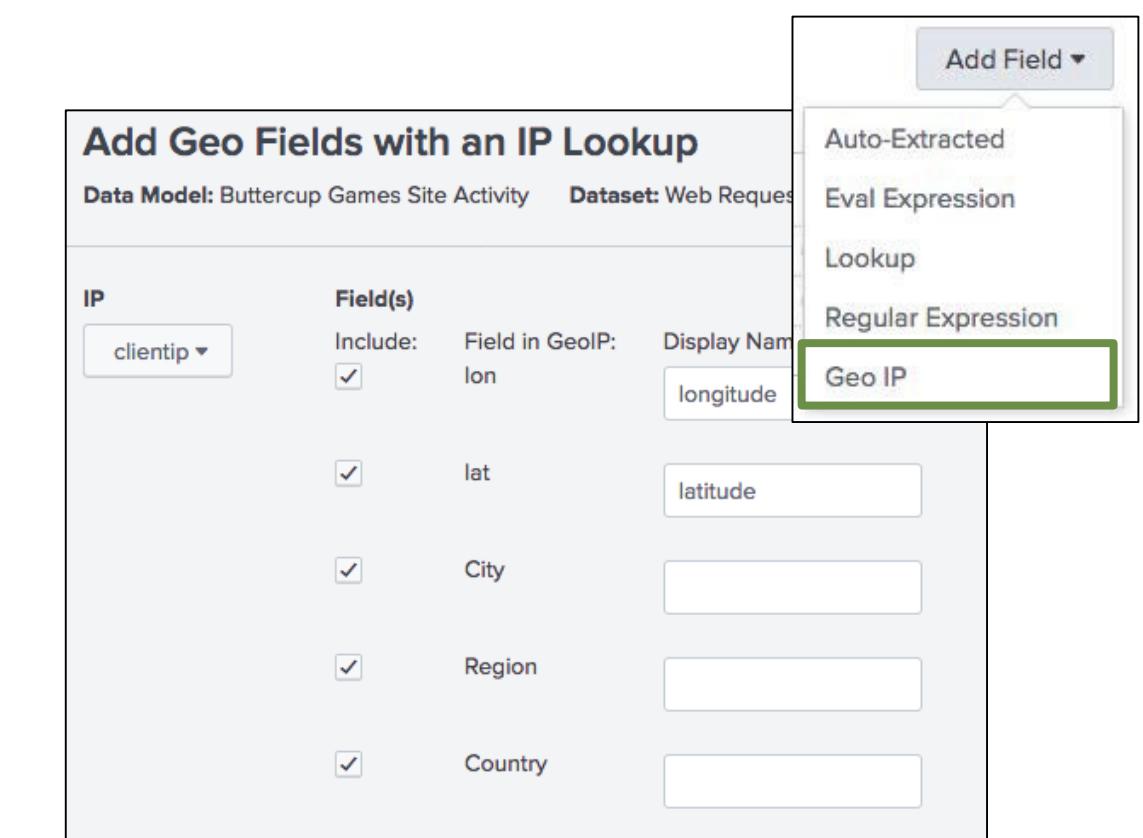
filter Apply Sample: 1,000 events ▾ All events ▾ All Events Matches Non-Matches

_raw

```
173.192.201.242 - - [29/Jan/2018:20:12:30] "GET /category.screen?categoryId=SHOOTER&JSESSIONID=SD10SL5FF8ADFF4952 HTTP/1.1" 200 1814 "http://www.buttercupgames.com/product.screen?"  
173.192.201.242 - - [29/Jan/2018:20:12:17] "GET /category.screen?categoryId=STRATEGY&JSESSIONID=SD10SL5FF8ADFF4952 HTTP/1.1" 200 1032 "http://www.buttercupgames.com/cart.do?action=...  
173.192.201.242 - - [29/Jan/2018:20:12:11] "GET /cart.do?action=addtocart&itemId=EST-13&productId=FS-SG-G03&JSESSIONID=SD10SL5FF8ADFF4952 HTTP/1.1" 200 3700 "http://www.buttercupg...  
173.192.201.242 - - [29/Jan/2018:20:11:55] "GET /oldlink?itemId=EST-6&JSESSIONID=SD10SL5FF8ADFF4952 HTTP/1.1" 200 2096 "http://www.buttercupgames.com/product.screen?productId=MB-AG...  
173.192.201.242 - - [29/Jan/2018:20:11:49] "POST /cart.do?action=remove&itemId=EST-15&productId=DC-SG-G02&JSESSIONID=SD10SL5FF8ADFF4952 HTTP/1.1" 200 2259 "http://www.yahoo.com"  
233.77.49.94 - - [29/Jan/2018:20:09:13] "GET /product.screen?productId=BS-AG-G09&JSESSIONID=SD2SL5FF5ADFF4952 HTTP/1.1" 200 3473 "http://www.buttercupgames.com/oldlink?itemId=EST-..."
```

Adding Fields - GeolP

- Map visualizations require latitude/longitude fields
- To use Geo IP Lookup, at least one IP field must be configured as an IPv4 type
- While the map function isn't available in Pivot, the data model can be called using the `ipivot` command and `<map>` element in a dashboard population search
 - Select the field that contains the mapping to lat/lon
 - Identify the lat/lon and geo fields in the data



Adding Child Datasets

When you create a new child dataset, you give it one or more additional constraints

Buttercup Games Site Activity

Buttercup_Games_Site_Activity

[Edit](#) [Download](#) [Pivot](#) [Documentation](#)

[All Data Models](#)

Datasets

EVENTS

Web Requests

Root Event
Root Transaction
Root Search
Child

Add Dataset ▾

Web Requests
Web_Requests

CONSTRAINTS

index=web sour

Bulk Edit ▾

Add Child Dataset

Data Model: Buttercup Games Site Activity

Dataset Name
Successful Requests

Additional Constraints
status<400

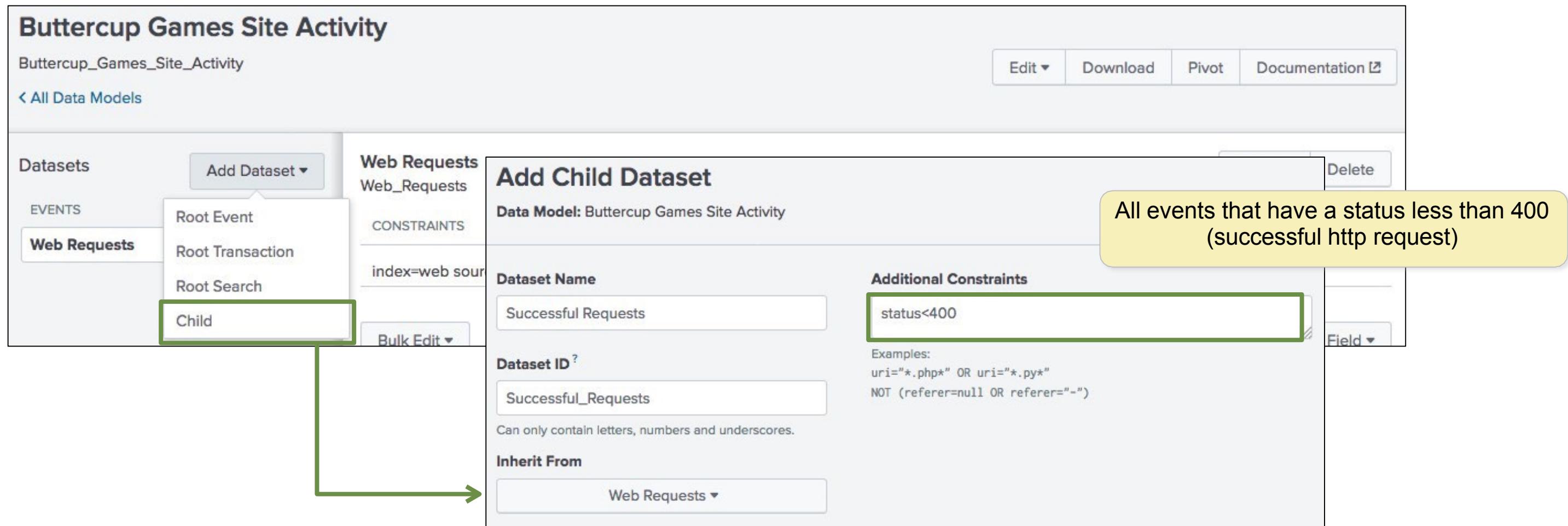
All events that have a status less than 400
(successful http request)

Examples:
uri="*.php*" OR uri="*.py*"
NOT (referer=null OR referer="-")

Dataset ID ?
Successful_Requests

Can only contain letters, numbers and underscores.

Inherit From
Web Requests ▾



Adding Child Datasets (cont.)

- Child datasets inherit all fields from the parent events
- You can add more fields to child dataset

The screenshot shows the Splunk Data Model Editor interface for a child dataset named "Successful Requests".

CONSTRAINTS:

- index=web sourcetype=access_combined
- status<400

INHERITED Fields:

- _time
- clientip
- host
- source
- sourcetype
- status

Fields (Auto-Extracted):

- Eval Expression
- Lookup
- Regular Expression
- Geo IP

Buttons:

- Rename
- Delete
- Edit
- Add Field
- Override

Testing the Data Model

- Click Pivot to access the Select a Dataset window
- Choose an object from the selected data model to begin building the report

Buttercup Games Site Activity

Buttercup_Games_Site_Activity

< All Data Models

Datasets Add Dataset ▾

EVENTS

Web Requests

- Successful Requests
 - purchases
 - addtocart
 - remove
- Failed Requests
 - failed purchases
 - failed addtocart
 - failed remove

Web Requests

Web_Requests

CONSTRAINTS

index=web sourcetype=access_combined

Bulk Edit ▾

INHERITED

_time Time

host String

source String

sourcetype String

EXTRACTED

Constraint

Edit

Documentation ▾

Pivot

Select a Dataset

11 Objects in Buttercup Games Site Activity

Web Requests

Successful Requests

purchases

addtocart

remove

Failed Requests

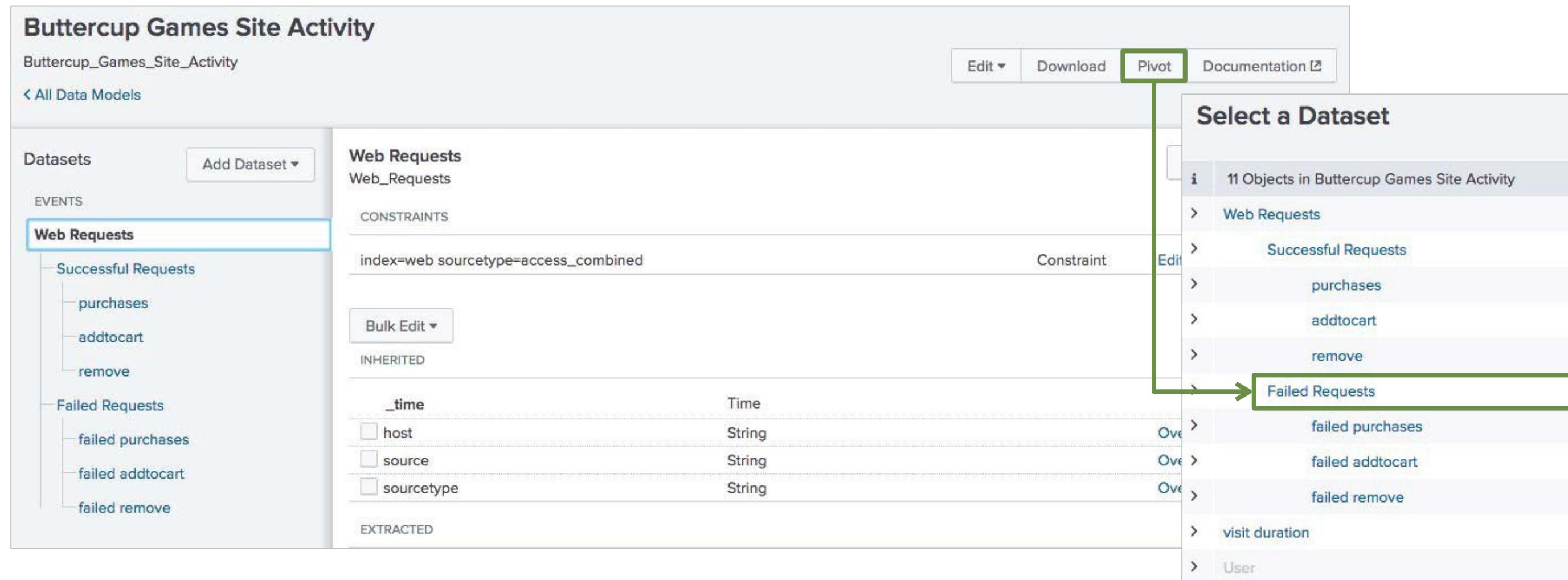
failed purchases

failed addtocart

failed remove

visit duration

User



Data Model Transaction Datasets

- Enable the creation of datasets that represent transactions
- Use fields that have already been added to the model using event or search datasets

Buttercup Games Site Activity

Buttercup_Games_Site_Activity

< All Data Models

Datasets Add Dataset ▾

visit duration visit_duration

CONSTRAINTS

Group Datasets	Web_Requests	Transaction	Edit
Group By	clientip		
Max Pause	10s		
Max Span			

Bulk Edit ▾

Add Field ▾

INHERITED

_time	Time	Required
duration	Number	Required
eventcount	Number	Required
host	String	Override
source	String	Override
sourcetype	String	Override

TRANSACTIONS

visit duration

EXTRACTED

action	String	Edit
bytes	Number	Edit
categoryid	String	Edit
change_type	String	Edit

Search and Transaction Dataset Considerations

- There must be at least one event or search dataset before adding a transaction dataset
- Search and transaction datasets cannot benefit from persistent data model acceleration
- As you learn to create data models, consider the types of reports your users will run
 - Can the same report be achieved with event datasets?
 - Will users need raw events or transactional data?

Set Permissions

- When a data model is created, the owner can determine access based on the following permissions:
 - Who can see the data models
 - ▶ Owner, App, or All Apps
 - Which users can perform which actions (Read/Write)
 - ▶ Everyone
 - ▶ Power
 - ▶ User
 - ▶ Admin-defined roles, if applicable

Edit Permissions

Data Model Buttercup Games Site Activity

Owner student1

App search

Display For

	Read	Write
Everyone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
power	<input type="checkbox"/>	<input type="checkbox"/>
student	<input type="checkbox"/>	<input type="checkbox"/>
user	<input type="checkbox"/>	<input type="checkbox"/>

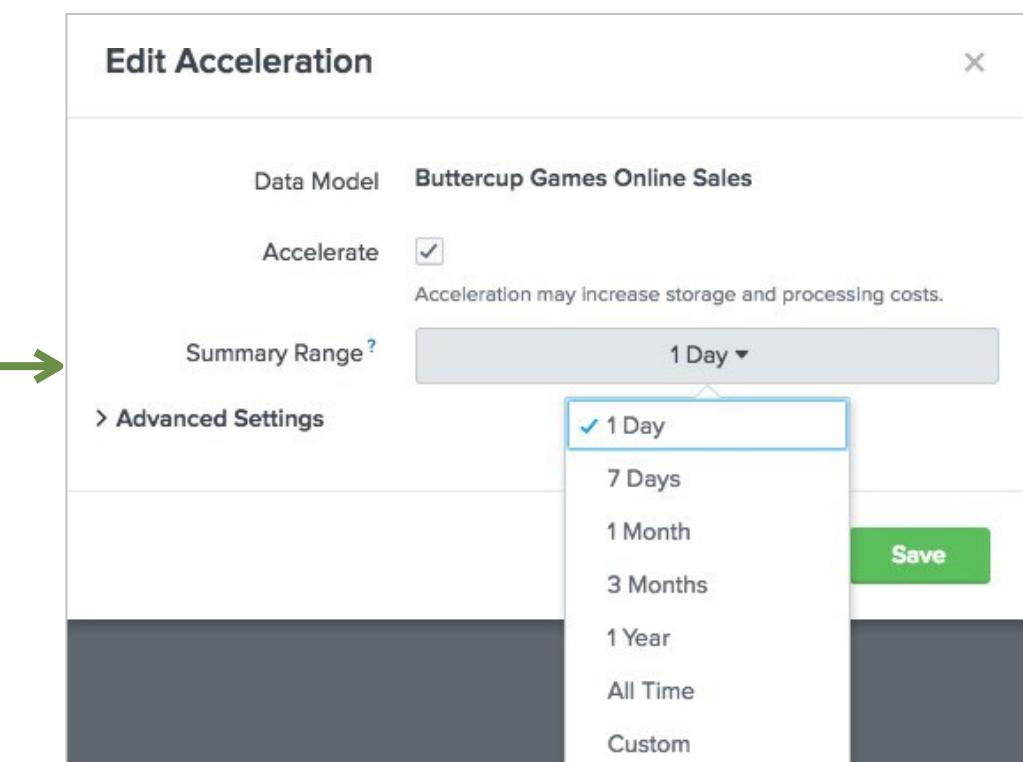
Data Model Acceleration

- Uses automatically created summaries to speed completion times for pivots
- Takes the form of inverted time-series index files (`tsidx`) that have been optimized for speed
- Discussed in more detail in *Advanced Searching and Reporting*

Accelerating a Data Model

- With persistent data model acceleration, all fields in the model become "indexed" fields
- You must have administrative permissions or the `accelerate_datamodel` capability to accelerate a data model
- Private data models cannot be accelerated
- Accelerated data models cannot be edited

The screenshot shows the Data Models page with one data model listed: Buttercup Games Online Sales. The model is a data model owned by ao-bcg and nobody, with Global sharing. A context menu is open for this model, with the 'Edit Acceleration' option highlighted by a green box.

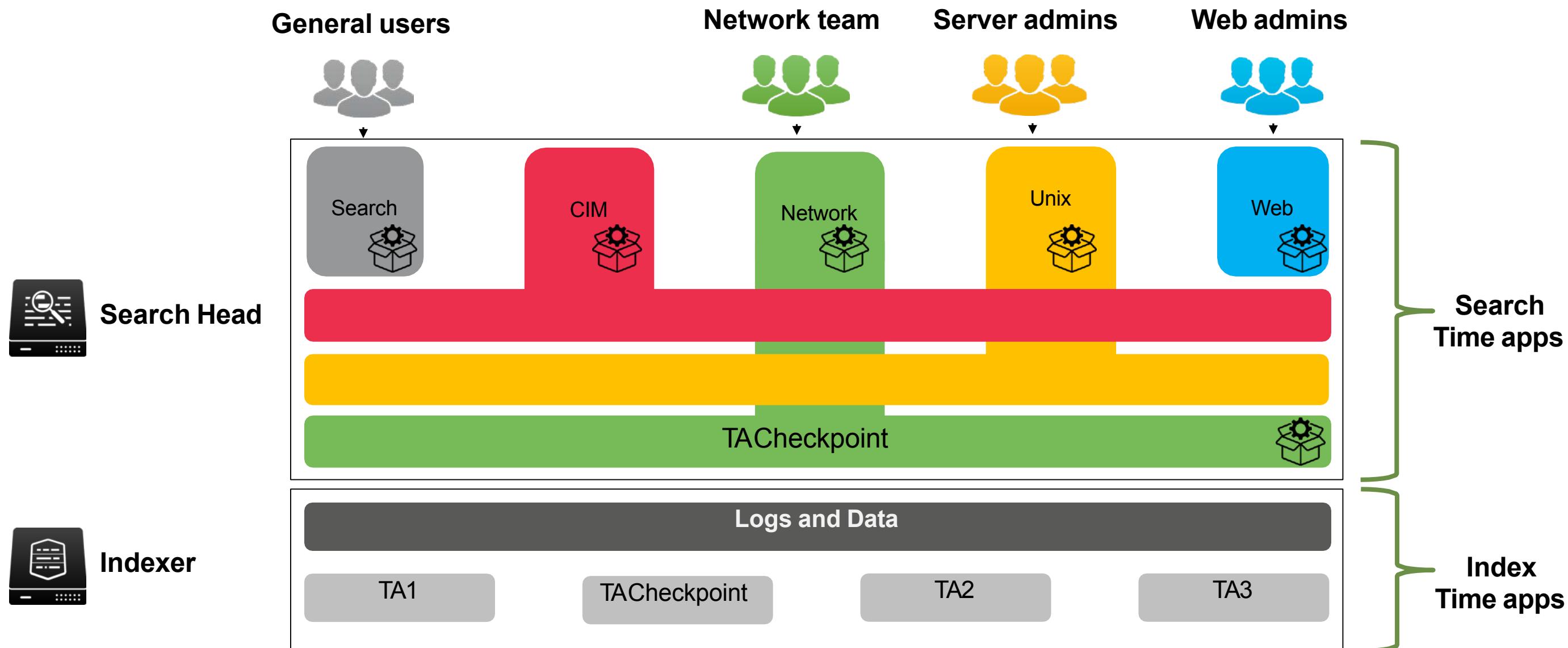


Using the Common Information Model (CIM) Add-On

What is the Common Information Model (CIM)?

- The Splunk Common Information Model provides a methodology to normalize data
- Leverage the CIM when creating field extractions, field aliases, event types, and tags to ensure:
 - Multiple apps can co-exist on a single Splunk deployment
 - Object permissions can be set to global for the use of multiple apps
 - Easier and more efficient correlation of data from different sources and source types

How the Splunk CIM Works



Normalized Field Names – Email Data

Field name	Data type	Description	Possible values
action	string	Action taken by the reporting device.	delivered, blocked, quarantined, deleted, unknown
duration	number	The amount of time for the completion of the messaging event, in seconds.	Email
src	string	The system that sent the message. May be aliased from more specific fields such as src_host, src_ip, or src_name.	

Normalized Field Names – Network Traffic

Field name	Data type	Description	Possible values
action	string	The action taken by the network device.	allowed, blocked, dropped, unknown
bytes	number	Total count of bytes handled by this device/interface (bytes_in + bytes_out).	
bytes_in	number	How many bytes this device/interface received.	
bytes_out	number	How many bytes this device/interface transmitted.	
src	string	The source of the network traffic (the client requesting the connection). May be aliased from more specific fields such as src_host, src_ip, or src_name.	

Normalized Field Names – Web Data

Field name	Data type	Description	Possible values
action	string	The action taken by the server or proxy.	
duration	number	The time taken by the proxy event, in milliseconds.	
http_method	string	The HTTP method used in the request.	GET, PUT, POST, DELETE, etc.
src	string	The source of the network traffic (the client requesting the connection).	
status	string	The HTTP response code indicating the status of the proxy request.	404, 302, 500, and so on.

Splunk CIM Add-on

- Set of 22 pre-configured data models
 - Fields and event category tags
 - Least common denominator of a domain of interest
- Leverage the CIM so that knowledge objects in multiple apps can co-exist on a single Splunk deployment
- Available on splunkbase:
 - <https://splunkbase.splunk.com/app/1621/>
- Use the CIM Reference Tables
 - <https://docs.splunk.com/Documentation/CIM/4.9.0/User/Howtousetheserreferencetables>

Splunk CIM Add-On Data Models	
Alerts	Java Virtual Machines (JVM)
Application State	Malware
Authentication	Network Resolution (DNS)
Certificates	Network Sessions
Change Analysis	Network Traffic
CIM Validation (S.o.S)	Performance
Databases	Splunk Audit Logs
Email	Ticket Management
Interprocess Messaging	Updates
Intrusion Detection	Vulnerabilities
Inventory	Web

datamodel Command

- Search against a specified data model object
- Return a description of all or a specified data model and its objects
- Is a generating command and should be the first command in the pipeline

datamodel

[Learn More ↗](#)

Allows user to examine data models and run the search for a datamodel object.

Example:

```
I datamodel
```

datamodel Command – Example

```
datamodel Web Web search | fields Web*
```

Dataset name
prepended to field
names in your data

- Command
- Data model name
- Data model dataset name
- Command
- Find field names with Web prefix

INTERESTING FIELDS

```
a Web.action 7
# Web.bytes 100+
a Web.dest 10
a Web.http_content_type 1
a Web.http_method 2
a Web.http_referrer 1
a Web.http_user_agent 1
# Web.http_user_agent_length 1
# Web.is_not_Proxy 1
# Web.is_Proxy 1
a Web.src 10
# Web.status 10
a Web.uri_path 11
a Web.uri_query 100+
a Web.url 10
# Web.url_length 7
a Web.user 9
a Web.vendor_product 2
```

from Command

- Retrieves data from a data model or named dataset
- Must be the first command in a search
- Different than just using datamodel
 - datamodel returns **all** fields prepended with data model name
 - from datamodel returns specified fields only
- from can also retrieve data from saved searches, reports, or lookup files

```
| from datamodel:"internal_server.splunkdaccess"
```

```
| from savedsearch:mysecurityquery
```

Additional CIM Resources

- Understand and use the CIM Add-on

<http://docs.splunk.com/Documentation/Splunk/latest/Knowledge/UnderstandandusetheCommonInformationModel>

- Overview of the Splunk CIM

<http://docs.splunk.com/Documentation/CIM/latest/User/Overview>

- Use the CIM to normalize data at search time

<http://docs.splunk.com/Documentation/CIM/latest/User/UsetheCIMtonormalizedataatsearchtime>