Searching & Reporting with Splunk 6.4

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Guidelines

- Hands-on lab exercises reinforce information presented in the lecture modules
- To receive a certificate of completion for the course, you must complete the lab exercises
- The lab exercises must be completed sequentially
  - Later lab exercises often depend on steps completed in previous lab exercises

Course Prerequisites

To be successful in this course, you should have completed:

- Using Splunk

Course Goals

- Create efficient, well-formed searches
- Perform calculations and evaluations on search results
- Generate reports, charts, and visualizations
- Analyze and format results
- Correlate events with transactions

Course Outline

Module 1: Search Fundamentals

Module 2: Transforming Commands, Part 1 Deriving Statistics

Module 3: Transforming Commands, Part 2 Creating Visualizations

Module 4: Transforming Commands, Part 3 Enriching Visualizations

Module 5: Manipulating and Filtering Results

Module 6: Correlating Events

Buttercup Games, Inc.

- Buttercup Games, Inc.
  - Is a multinational company with its 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
- For more information about Buttercup Games, please see Appendix C

Course Scenario

- Use cases in this course are based on Buttercup Games, a gaming company
- Searches and reports are based on:
  - IT operations information from mail and internal network data
  - Security operations information from internal network and badge reader data
  - Business analytics from the web access logs and vendor data

Callouts

Scenarios

- Many of the examples in this course relate to a specific scenario
- For each example, a question is posed from a colleague or manager at Buttercup Games

Scenario

For failed logins into the network during the last 60 minutes, display the `IP` and `user` name.

Notes & Tips

- References for more information on a topic and tips for best practices

Note

Lookups are discussed in the *Creating Splunk Knowledge Objects* course.

Your Role at Buttercup Games

- You are a Splunk power user
- Your responsibility is to provide information to users throughout the company
- You gather data and statistics and report on:
  - Security
  - IT operations
  - Business intelligence
  - Etc.

Useful References

- Search Reference:
  http://docs.splunk.com/Documentation/Splunk/latest/SearchReference
- Search Quick Reference:
  http://www.splunk.com/content/dam/splunk2/pdfs/solution-guides/splunk-quick-reference-guide.pdf

Module 1: Search Fundamentals

Module Objectives

- Become familiar with the source types used during the course
- Review basic search commands and general search practices
- Examine the search pipeline
- Use the following commands to perform searches:
  - `table`
  - `rename`
  - `fields`
  - `dedup`
  - `sort`

Buttercup Games Environment

Data	host	sourcetype
Active Directory data	adldapsv1	WinEventLog:Security
Badge reader data	badgesv1	history_access
BI server data	ecommsv1	sales_entries
Email data	cisco_router1	cisco_esa
Online transactions & Web server	www1	access_combined linux_secure
	www2	
	www3	
Retail sales data	vendorUS1	vendor_sales
Splunk indexer data	splunk1	ps
Web appliance data	cisco_router1	cisco_wsa_squid

Basic Search Review

- **Keywords**

  search for `error`, `password`

- **Booleans**

  `OR`, `AND`, `NOT`; `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
  ```

- **Wildcards**
  - `status=40*` matches 40, 40a, 404, etc., starting keywords with a wildcard is very inefficient, e.g. *dmin
  - Avoid wildcards at the beginning of a string

- **Comparisons**

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

Review: Time Range Abbreviations

- Time ranges specified in the **Advanced** tab of the time range picker
- Time unit abbreviations include:

```
s = seconds   m = minutes   h = hours   d = days   w = week   mon = months   y = year
```

- **@** symbol "snaps" to the time unit you specify
  - Snapping rounds *down* to the nearest specified unit
  - Example: Current time when the search starts is 09:37:12

    -30m@h looks back to 09:00:00

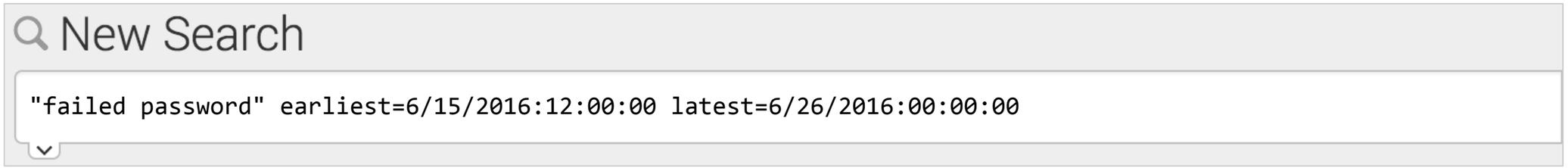
Time Range: `earliest` and `latest`

- You can also specify a time range in the search bar
- To specify a beginning and an ending for a time range, use **`earliest`** and **`latest`**
  - Examples:

`earliest=-h`	looks back one hour
`earliest=-2d@d latest=@d`	looks back from two days ago, up to the beginning of today

Time Range: `earliest` and `latest` (cont.)

- Absolute values can be specified
  - Absolute values take the form: **`%m/%d/%Y:%H:%M:%S`**



General Search Practices

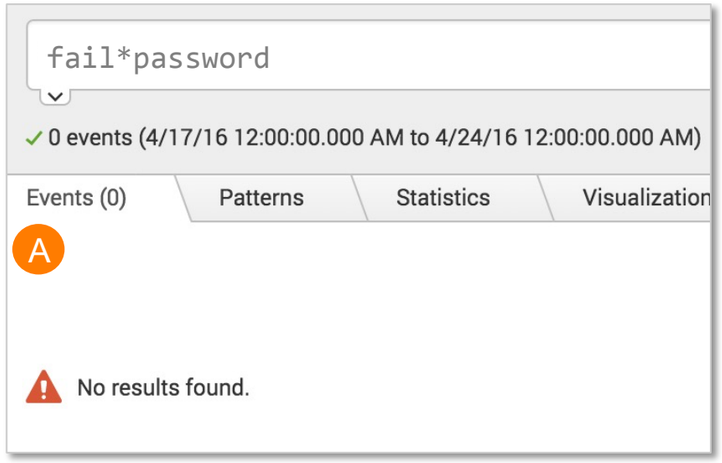
- Time is the most efficient filter
- Be specific
  - Searching for "access denied" is always better than searching for "denied"
  - To make searches more efficient, include as many terms as possible
    - If you want to find events with `"error"` and `"sshd"` and 90% of the events include "error", but only 5% "sshd", include both values in the search
- Inclusion is generally better than exclusion
  - Searching for `"access denied"` is faster than `NOT "access granted"`
- Filter as early as possible
  - For example, remove duplicate events, then sort

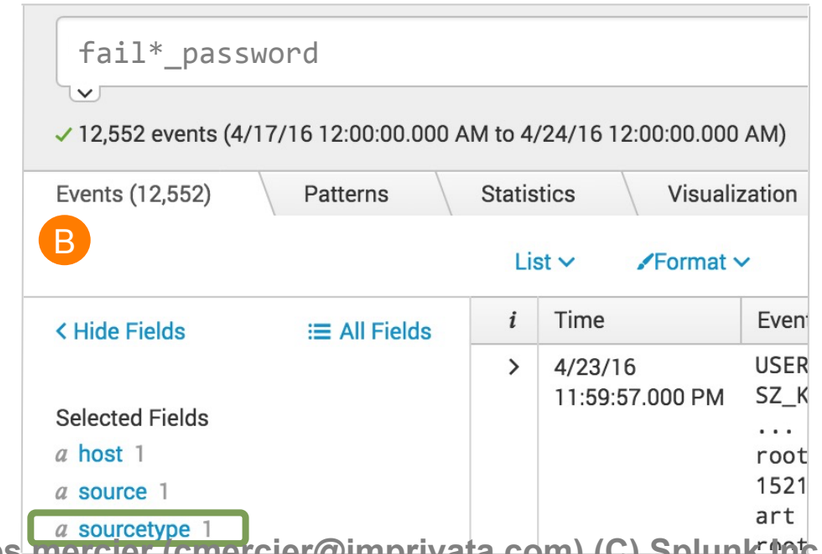
Note

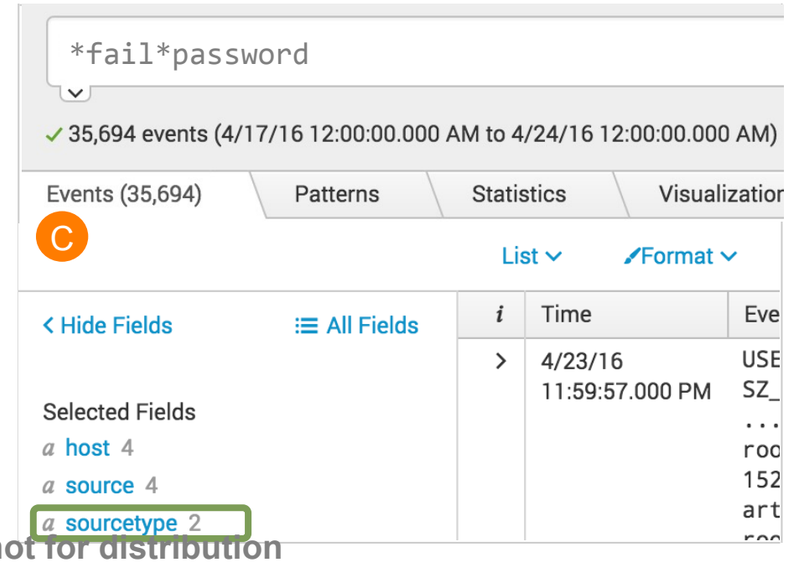
Note that search terms are case-*insensitive* and search fields are case-*sensitive*.

Tips for Using Wildcards

- For fastest performance, try to avoid using wildcards at the beginning of a string
- Inconsistent performance can result from using wildcards in the middle of a string, especially if the string contains punctuation or quotes, for example:
  - A **`fail*password`** (no results)
  - B **`fail*_password`** (results from `sourcetype=ps`)
  - C **`*fail*password`** (results from `sourcetype=ps` and `sourcetype=linux_secure`)







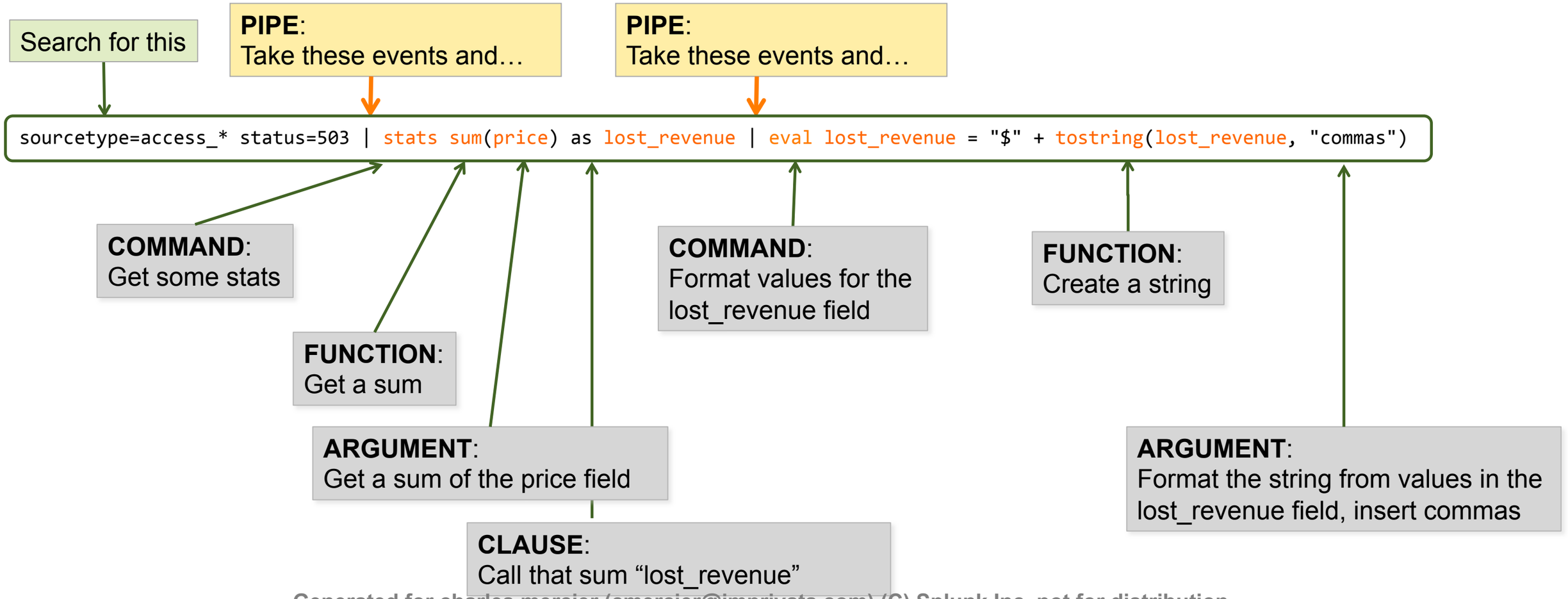
Search Language Syntax Concepts

Searches are made up of 5 basic components

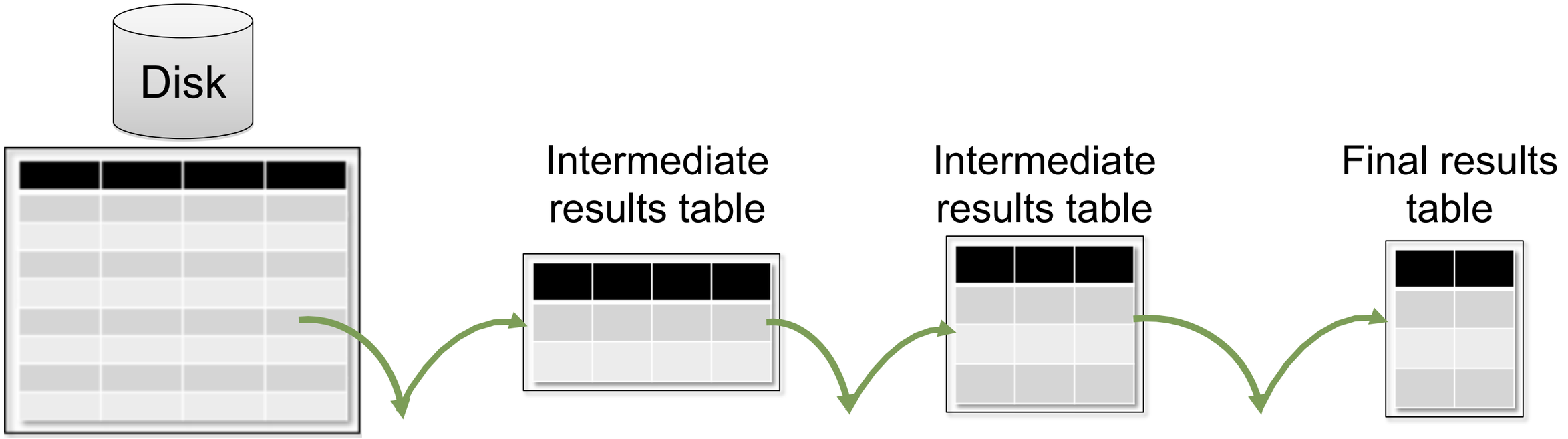
- **Search terms** – what are you looking for?
  - Keywords, phrases, Booleans, etc.
- **Commands** – what do you want to do with the results?
  - Create a chart, compute statistics, evaluate and format, etc.
- **Functions** – how do you want to chart, compute, or evaluate the results?
  - Get a sum, get an average, transform the values, etc.
- **Arguments** – are there variables you want to apply to this function?
  - Calculate average value for a specific field, convert milliseconds to seconds, etc.
- **Clauses –** how do you want to group or rename the fields in the results?
  - Give a field another name or group values by or over

Search Pipeline Example

This diagram represents a search, broken into its syntax components



The Search Pipeline




```
sourcetype=linux_secure failed password | top user | fields – percent
```

Fetch events from disk that match

Summarize into table of top 10 users

Remove column showing percentage

Creating a Table

- `table` command returns a table formed by only fields in the argument list
- Columns are displayed in the order given in the command
  - Column headers are field names
  - Each row is an event
  - Rows are field values

Scenario

Display the `clientip`, `action`, `productId`, and `status` of customer interactions in the online store.

```
sourcetype=access_combined
| table clientip, action, productId, status
```

clientip	action	productId	status
223.205.219.67			200
69.80.0.18	view	WC-SH-A02	200
69.80.0.18		SF-BVS-01	408
91.205.189.15	view	FS-SG-G03	200
91.205.189.15	view	CU-PG-G06	200
91.205.189.15	view	WC-SH-A02	200
91.205.189.15	remove	WC-SH-A01	200
91.205.189.15			200

Note

To make searches more readable, you can break a line in the search bar with shift-enter.

Renaming Fields

- To change the name of a field, use the `rename` command
- Useful for giving fields more meaningful names
- When including spaces or special characters in field names, use double straight quotes:
  - A `rename productId as ProductID`
  - B `rename action as "Customer Action"`
  - C `rename status as "HTTP Status"`

Scenario

Display the `clientip`, `action`, `productId`, and `status` of customer interactions in the online store.

```
sourcetype=access_combined
| table clientip, action, productId, status
| rename productId as ProductID, A
  action as "Customer Action", B
  status as "HTTP Status" C
```

clientip	Customer Action B	ProductID A	HTTP Status C
141.146.8.66		MB-AG-T01	200
141.146.8.66		WC-SH-A01	200
195.80.144.22		DC-SG-G02	200
141.146.8.66		WC-SH-A02	200
195.80.144.22		SC-MG-G10	200
141.146.8.66		PZ-SG-G05	200
195.80.144.22	purchase		200
195.80.144.22	purchase	SC-MG-G10	200

`fields` Command

- Field extraction is one of the most costly parts of a search
- `fields` command allows you to include or exclude specified fields in your search or report
- To include, use `fields +(default)`
  - Occurs before field extraction
  - Improved performance
- To exclude, use `fields -`
  - Occurs after field extraction
  - No performance benefit
  - Exclude fields used in search to make the table/display easier to read

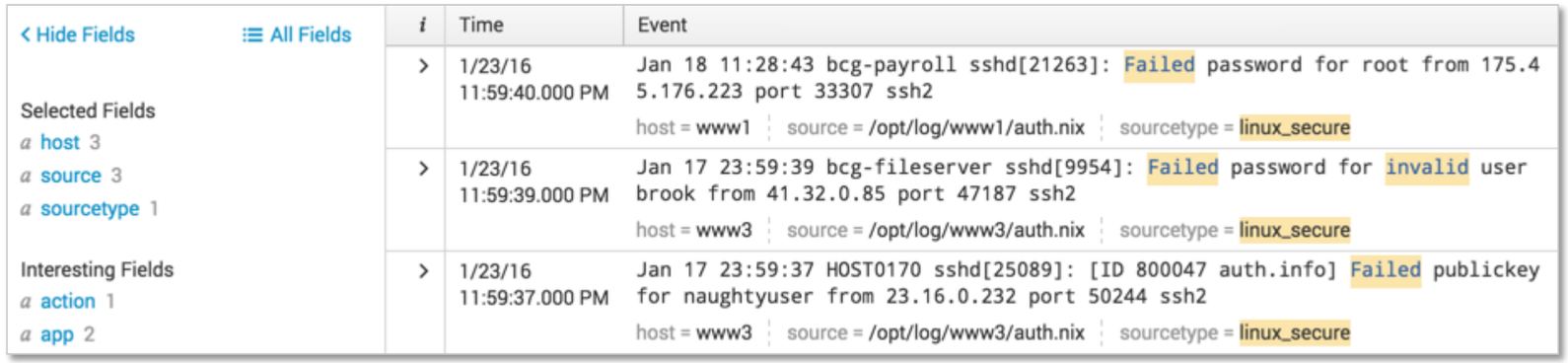
`fields` Command – Examples

Improves performance – only the fields you specify are extracted

Scenario

Display network failures during the previous week.

Returned **6,567** results by scanning **6,567** events in **1.425** seconds:



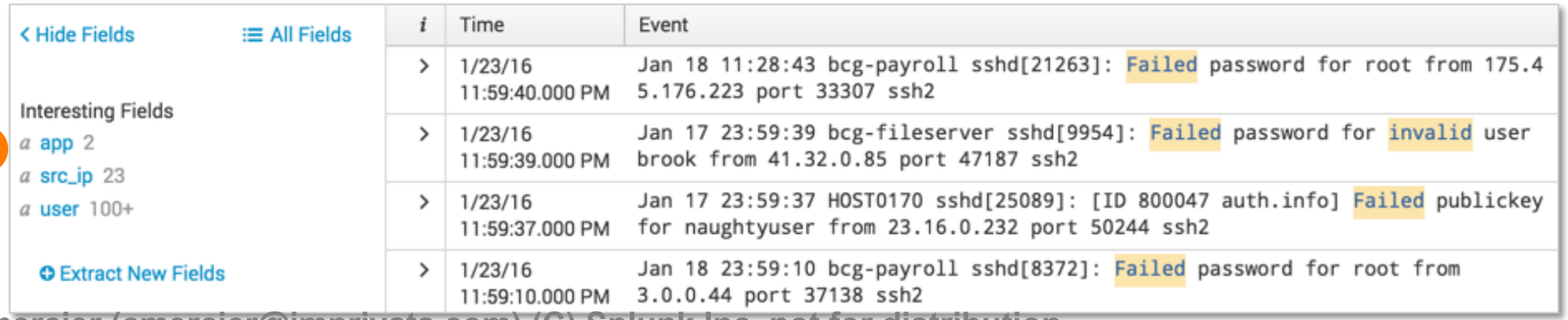
Scenario

Display network failures during the previous week. Retrieve only user, app, and src_ip.

```
sourcetype=linux_secure
  (fail* OR invalid)
| fields user, app, src_ip
```

A

Returned **6,567** results by scanning **6,567** events in **0.753** seconds:



A

 | 6 July 2016

dedup Command

Use dedup to remove duplicates from your results

```
sourcetype=vendor_sales | table VendorCountry, VendorStateProvince, VendorCity, Vendor
```

VendorCountry	VendorStateProvince	VendorCity	Vendor
United States	Texas	Waco	Wow Games
United States	Utah	Cedar City	Woody's Games
United States	Virginia	Staunton	Woody's Games
United States	Utah	Cedar City	Woody's Games
United States	Utah	Cedar City	Woody's Games
Australia	Western Australia	Perth	Wonderland Hobbies
Australia	Western Australia	Perth	Wonderland Hobbies

```
...| dedup Vendor | table ...
```

VendorCountry	VendorStateProvince	VendorCity	Vendor
United States	Texas	Waco	Wow Games
United States	Utah	Cedar City	Woody's Games
Australia	Western Australia	Perth	Wonderland Hobbies

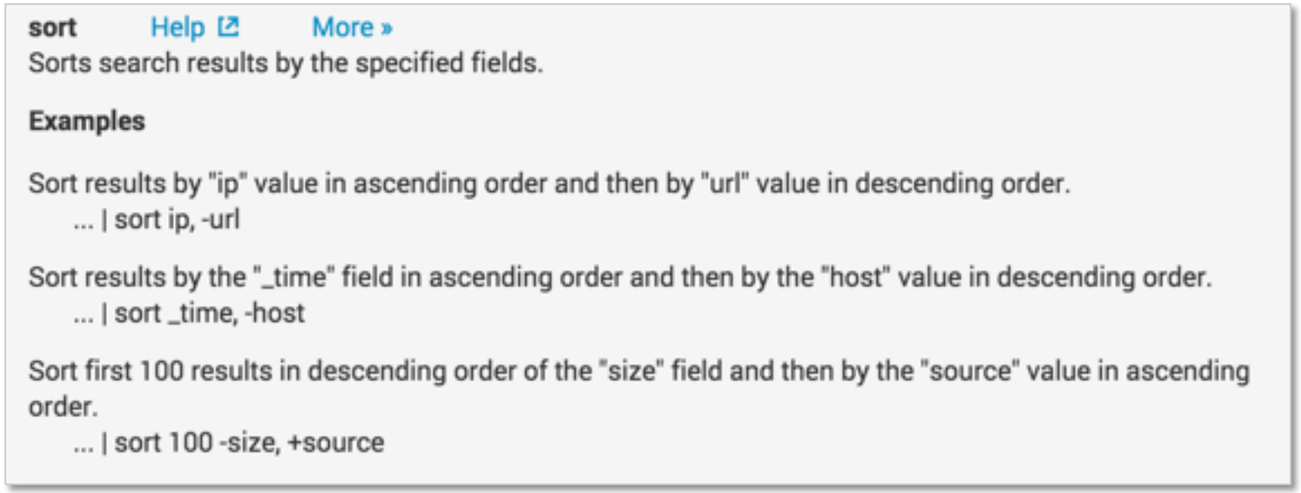
```
...| dedup VendorCity, Vendor | table ...
```

VendorCountry	VendorStateProvince	VendorCity	Vendor
United States	Texas	Waco	Wow Games
United States	Utah	Cedar City	Woody's Games
United States	Virginia	Staunton	Woody's Games
Australia	Western Australia	Perth	Wonderland Hobbies

sort Command

- Use sort to order your results in
  + ascending (default) or
  – descending
- To limit the returned results, use the limit option

```
... | sort limit=20 -categoryId, product_name
... | sort 20 count
```



sort Help More »

Sorts search results by the specified fields.

Examples

Sort results by "ip" value in ascending order and then by "url" value in descending order.

... | sort ip, -url

Sort results by the "_time" field in ascending order and then by the "host" value in descending order.

... | sort _time, -host

Sort first 100 results in descending order of the "size" field and then by the "source" value in ascending order.

... | sort 100 -size, +source

sort Command (cont.)

`sort -/+<fieldname>` sign followed by fieldname sorts results in the sign's order

`sort -/+ <fieldname>` sign followed by space and then fieldname applies sort order to all following fields without a different explicit sort order

```
sourcetype=vendor_sales
| dedup Vendor
| sort - VendorCountry, +VendorStateProvince, VendorCity, Vendor
| table VendorCountry, VendorStateProvince, VendorCity, Vendor
```

VendorCountry	VendorStateProvince	VendorCity	Vendor
United States	Arizona	Yuma	Yumster Games
United States	Arizona	Tucson	Boothill Games
United States	Arizona	Phoenix	Rising Games
United States	Arizona	Phoenix	Phoenix Games
United States	Arizona	Flagstaff	Flaggin Games

Adding Knowledge to Data through Lookups

- There are use cases where you need additional data for the search that is not available in the index
- In this example, `Username` and `Department` are not in the event data

Aug30 2015 23:50:21 Address=1.1.1.R2 Address_Description=San Francisco Device=Proximity Reader Event_Description=Access Granted: Door Used rfid=341402271288

Scenario

Display badge-ins during the last 4 hours to include location, badge ID, user name and department.

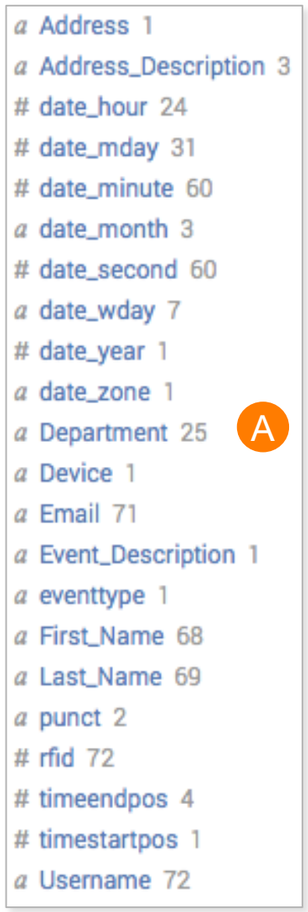
```
sourcetype=history_access
| table Address_Description, rfid,
  Username, Department
```

Address_Description	rfid	Username	Department
London	890313901800	bhussain	ITOps
London	890313901800	bhussain	ITOps
London	890313901800	bhussain	ITOps
London	862417886973	fbryan	Sales
Boston	249772079712	lsagers	SecOps
Boston	398009643042	pbunch	ITOps
Boston	672903009231	dhale	Sales
London	963871339460	rjayaraman	Engineering

Note

Lookups are discussed in the *Creating Splunk Knowledge Objects* course.

Adding Knowledge to Data through Lookups (cont.)



- Lookups allow you to add more fields to your events, such as:
  - Associate RFIDs with user names , IP addresses, and workstation IDs
  - Provide descriptions for http status codes (“file not found”, “service unavailable”)
  - Reveal sale prices and descriptions for products
- Most lookups are automatic and performed in the background
- Lookup fields appear in the Fields sidebar A
- Lookups are Knowledge Objects and are typically created by the Knowledge Manager

Lab Exercise 1

Time: 35 minutes

Scenario:

In this and succeeding lab exercises, you assume the roles of IT operations (IT Ops), security operations (Sec Ops), and business analyst at Buttercup Games

Tasks:

- Log into Splunk on the classroom server
- Make CLASS: Search & Reporting your default app and change your timezone setting
- Explore web server events [`linux_secure`]
- Explore web appliance events [`cisco_wsa_squid`]
- Explore corporate network events [`winauthentication_security`]
- Explore retail and online sales events [`vendor_sales; access_combined`]
- Check for possible attacks on the web servers in the last 15 minutes
- Create a new IT Ops dashboard
  - Check for issues with customer purchases in the online store
  - Check how employees are using corporate resources

Module 2: Transforming Commands, Part 1 Deriving Statistics

Module Objectives

Use the following commands and their functions:

- `top`
- `rare`
- `stats`

Transforming Commands

- Orders search results into a data table that Splunk can use for statistical purposes
- Required to transform search results into visualizations

Getting Top Values

The `top` command finds the most common values of a given field in the result set

- By default, returns top 10 results

src_ip	count	percent
3.0.0.44	1770	49.761035
175.45.176.223	553	15.546809
2.144.0.210	400	11.245432
41.32.0.85	324	9.108800
175.45.176.98	223	6.269328
23.16.0.181	144	4.048355
41.0.0.142	35	0.983975
2.144.0.22	28	0.787180
1.0.32.67	18	0.506044
5.11.128.21	17	0.477931

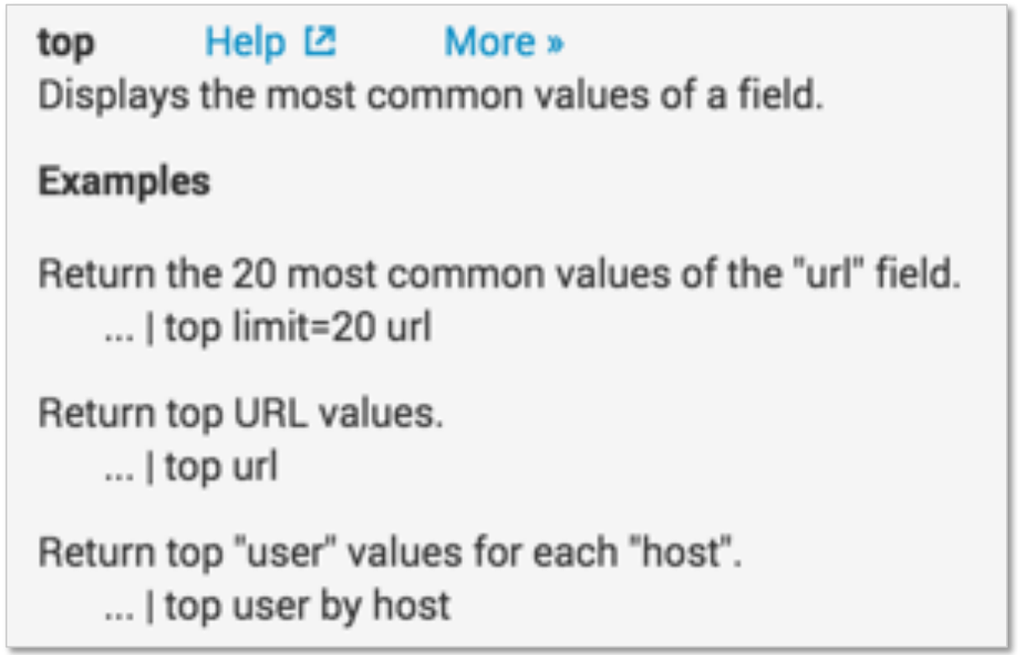
Scenario

During the last 60 minutes, which IPs generated the most attacks?

```
sourcetype=linux_secure (fail* OR invalid)
| top src_ip
```


top Command

- By default, output displays in table format
- Automatically returns **count** and **percent** columns
- `limit=#` returns this number of results
  - By default, 10 results are displayed
  - `limit=0` returns unlimited results
- `countfield=<string>` provides the name of a new field to write the value of count, default is `"count"`
- `showperc=f` specifies whether to create field called `"percent"`, default is true



Note

Refer to the search assistant or Splunk docs for the other available options.

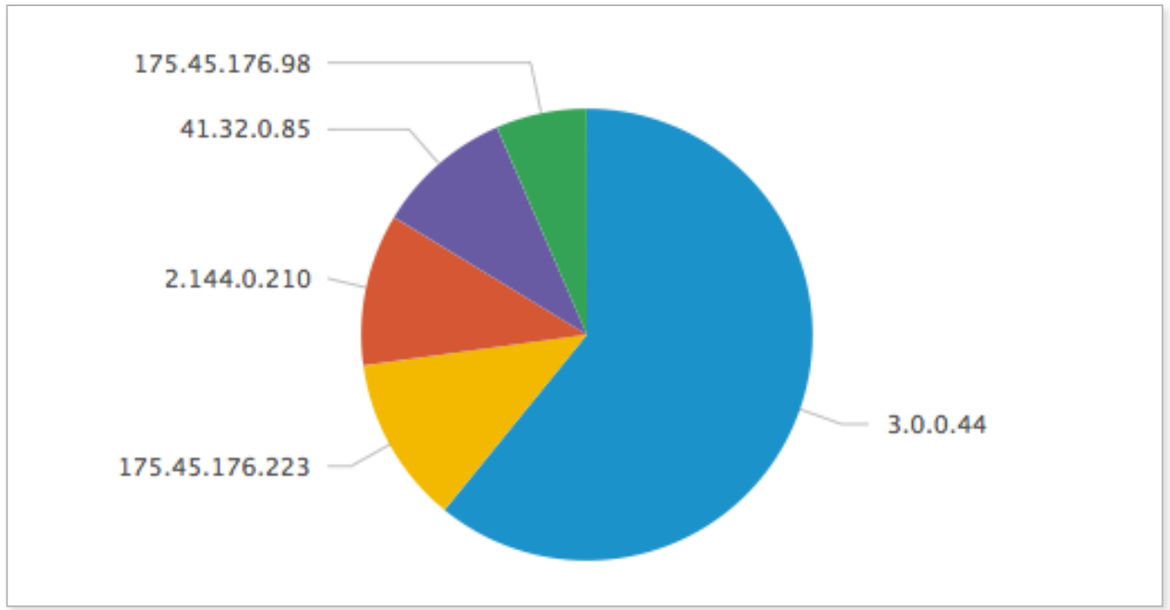
top Command – Single Field Example

Scenario

During the last hour, display the top 5 IPs that generated the most attacks.

```
sourcetype=linux_secure
(fail* OR invalid)
| top limit=5 src_ip
```

src_ip	count	percent
3.0.0.44	56	54.901961
175.45.176.223	11	10.784314
2.144.0.210	10	9.803922
41.32.0.85	9	8.823529
175.45.176.98	6	5.882353



top – Multiple Field Example

Scenario

Display the top 3 common values for users and web categories browsed during the last 24 hours.

```
sourcetype=cisco_wsa_squid
| top user A x_webcat_code_full limit=3 B
```

user	x_webcat_code_full	count	percent
bsimmel@buttercupgames.com	Games	79	6.638655
myuan@buttercupgames.com A	Arts and Entertainment	53	4.453782
acurry@buttercupgames.com	Arts and Entertainment B	47	3.949580

top – Single Field with by Clause Example

Scenario

Display the top 3 web categories browsed for each user during the last 24 hours.

```
sourcetype=cisco_wsa_squid
| top x_webcat_code_full B by user A limit=3
```

user	x_webcat_code_full	count	percent
acurry@buttercupgames.com A	Arts and Entertainment	47	60.256410
acurry@buttercupgames.com	Health and Nutrition B	11	14.102564
acurry@buttercupgames.com	Computers and Internet	6	7.692308
adombrowski@buttercupgames.com	Arts and Entertainment	4	36.363636
adombrowski@buttercupgames.com	Uncategorized URLs	3	27.272727
adombrowski@buttercupgames.com	Business and Industry	2	18.181818

top – Specifying Options

Scenario

Display the top 3 user/web categories browsed combinations during the last 24 hours. Rename the count field and show count, but not the percentage.

```
sourcetype=cisco_wsa_squid
| top user x_webcat_code_full limit=3 A
  countfield="Total Viewed" B showperc=f
```

user	x_webcat_code_full	Total Viewed
svoronoff@buttercupgames.com	Uncategorized URLs	124
cmunson@buttercupgames.com	Society and Culture	99
syoungin@buttercupgames.com	Shopping	87

Note

A Boolean can be t/f, true/false as well as 1/0.

rare Command

- The rare command returns the least common field values of a given field in the results
- Options are identical to the top command

Scenario

Which product is the least sold by Buttercup Games vendors?

```
sourcetype=vendor_sales
  | rare product_name showperc=f limit=1
```

stats Command

- stats allows you to calculate statistics on data that matches your search criteria
- Common functions include:
  - count – returns the number of events that match the search criteria
  - distinct_count, dc – returns a count of unique values for a given field
  - sum – returns a sum of numeric values
  - avg – returns an average of numeric values
  - list – lists all values of a given field
  - values – lists unique values of a given field

stats Help More »

Provides statistics, grouped optionally by field.

Examples

Search the access logs, and return the number of hits from the top 100 values of "referer_domain".

```
sourcetype=access_combined | top limit=100 referer_domain | stats sum(count)
```

Return the average for each hour, of any unique field that ends with the string "lay" (for example, delay, xdelay, relay, etc).

```
... | stats avg(*lay) BY date_hour
```

Remove duplicates of results with the same "host" value and return the total count of the remaining results.

```
... | stats distinct_count(host)
```


stats Command – count

- count returns the number of matching events based on the current search criteria
- Use the as clause to rename the count field

Scenario

Count the invalid or failed login attempts during the last 60 minutes.

```
sourcetype=linux_secure (invalid OR failed)
| stats count
```

Events | Patterns | Statistics (1) | Visualization

10 Per Page | Format | Preview

count
63

```
sourcetype=linux_secure (invalid OR failed)
| stats count as "Potential Issues"
```

Events | Patterns | Statistics (1) | Visualization

10 Per Page | Format | Preview

Potential Issues
63

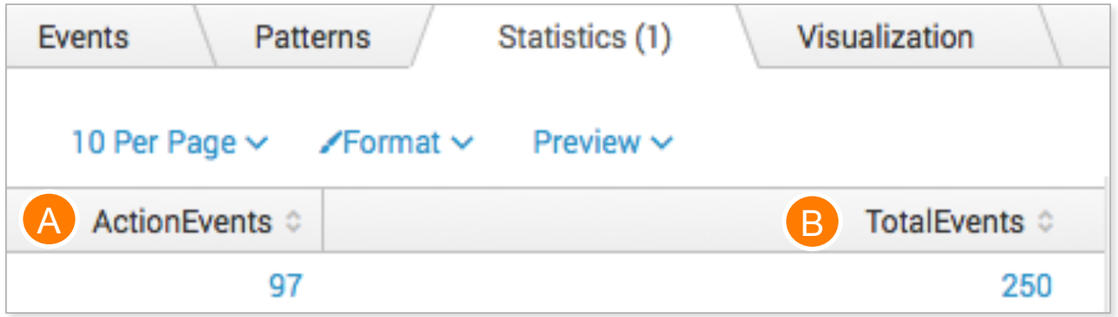
stats Command – count(*field*)

Adding a *field* as an argument to the count function returns the number of events where a value is present for the specified field

Scenario

Count the number of events during the last 15 minutes that contain a vendor action field. Also count the total events.

```
sourcetype=linux_secure
| stats count(vendor_action) as ActionEvents, A
  count as TotalEvents B
```



stats Command – by *fields*

Scenario

Count the number of vendor actions by user and application during the last 15 minutes.

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

- by clause returns a count for each value of a named field or set of fields
- Can use any number of fields in the by *field* list

user	app	vendor_action	count
arangel	sshd	Failed	1
bhussain	cron	session opened	54
eminem	sshd	Failed	1
ftpuser	ftpd	FTP LOGIN	4
jdoe	ftpd	FTP LOGIN	4
lsagers	sshd	Failed	1
madeyemi	sshd	Accepted	2
oracle	sshd	Failed	1
pdabbeville	sshd	Failed	1
root	ftpd	FTP LOGIN	4

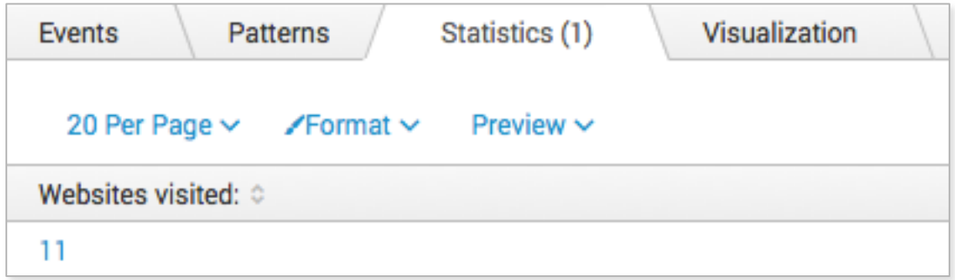
`stats` Command – `distinct_count(`*`field`*`)`

- `distinct_count()` or `dc()` provides a count of how many unique values there are for a given field in the result set
- This example counts how many unique values for `s_hostname`

Scenario

How many unique websites have our employees visited?

```
sourcetype=cisco_wsa_squid
 | stats dc(s_hostname) as "Websites visited:"
```



stats Command – sum(*field*)

Scenario

How much bandwidth did employees spend at each website during the past week?

```
sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as Bandwidth by s_hostname
| sort -Bandwidth
```

A B C

For fields with a numeric value, you can sum the actual values of that field

Events | Patterns | Statistics (54) | Visualization

20 Per Page | Format | Preview

s_hostname	Bandwidth
www.gctsindia.in	4866091
www.heals.co.uk	1233522
www.animationmagazine.net	1203433
www.kare11.com	1202753
www.finedinings.com	747407

stats Command – sum(*field*) – (cont.)

Scenario

Report the number of retail units sold and sales revenue for each product during the past week.

```
sourcetype=vendor_sales
| stats count(price) as "Units Sold" A
  sum(price) as "Total Sales" B by product_name C
| sort -"Total Sales" D
```

- A A single stats command can
- B have multiple functions
- C The by clause is applied to both functions
- D sort in descending order

product_name	A Units Sold	B Total Sales
Dream Crusher	100	3999.00
Manganiello Bros.	83	3319.17
World of Cheese	116	2898.84
C Orvil the Wolverine	60	2399.40 D
SIM Cubicle	111	2218.89
Mediocre Kingdoms	81	2024.19
Final Sequel	65	1624.35
Curling 2014	53	1059.47
World of Cheese Tee	83	829.17
Manganiello Bros. Tee	82	819.18

stats Command – avg(*field*)

- The avg function provides the average numeric value for the given field
- You can only use avg on numeric fields
  - If an event does not have the field or has an invalid value for the field, it is not considered in the calculation

Scenario

What is the average bandwidth used for each website usage type?

```
sourcetype=cisco_wsa_squid
| stats avg(sc_bytes) as "Average Bytes" A
  by usage B
```

usage	Average Bytes
Borderline	13709.812627
Business	13166.405498
Personal	17105.070920
Unknown	13528.095825
Violation	7139.151515

B

A

stats Command – list(*field*)

- list function lists all field values for a given field
- This example lists the websites visited by each employee
  - Since the security logs generate an event for each network request, the same hostname appears multiple times
  - If you want a list of "unique" field values, use the values function

Scenario

Which websites have our employees accessed during the last 60 minutes?

```
sourcetype=cisco_wsa_squid
| stats list(s_hostname) as "Websites visited:"
  by cs_username
```

cs_username	Websites visited:
basselin@buttercupgames.com	- -
blu@buttercupgames.com	www.lowermybills.com
cquinn@buttercupgames.com	- static.pochta.ru
dhale@buttercupgames.com	-
dpiazza@buttercupgames.com	www.ayles.com www.ayles.com www.ayles.com www.ayles.com www.ayles.com www.ayles.com

stats Command – values(*field*)

Scenario

Display the user names of failed attempts by IP in the last 60 minutes.

```
sourcetype=linux_secure fail*
| stats values(user) as "User Names",
  count(user) as Attempts by src_ip
```

values function lists “unique” values for the specified field

src_ip	User Names	Attempts
1.0.32.67	root	2
10.232.44.142	twilliam	1
10.232.44.71	jsimon1	1
175.45.176.223	gbottazzi oracle root scanner user	37
175.45.176.98	abc andrew cvs enquiries logs michael test test3	8

Lab Exercise 2

Time: 30 minutes

Scenarios:

- Find out from where visitors to our website are coming
  - Add this search to your IT Ops dashboard
- Display the top status codes for each of our web servers
- Identify the types of content employees are viewing
- Count the number of distinct employee badge swipes, by location, during the last 24 hours
- List the actions (without duplicates) on our Active Directory server during the last hour

****CHALLENGE Exercise**

- Calculate the number of events, the average price, and the total price for each action in the online store during the previous week

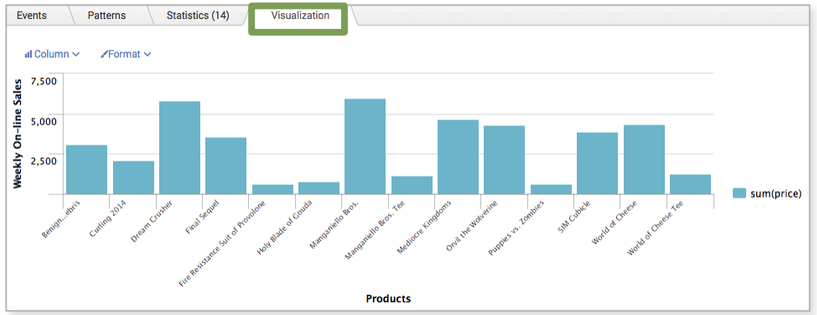
Module 3: Transforming Commands, Part 2 Creating Visualizations

Module Objectives

- Explore data structure requirements
- Explore visualization types
- Create and format charts
- Create and format `timecharts`
- Explain when to use each type of reporting command

Visualization Types

When a search returns statistical values, you can view results as a visualization

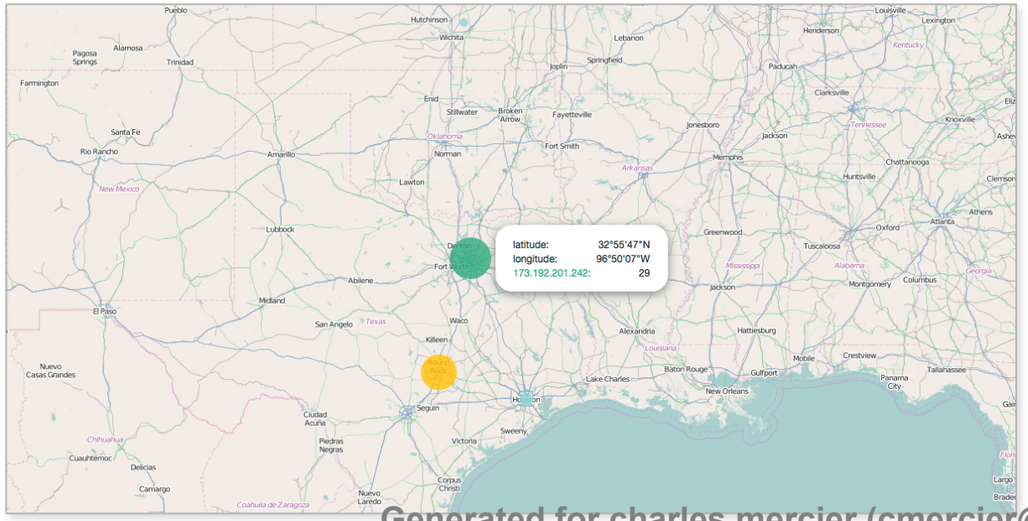


Chart

Events | Patterns | Statistics (14) | Visualization

10 Per Page | Format | Preview

product_name	sum(price)
Benign Space Debris	3073.77
Curling 2014	2118.94
Dream Crusher	5838.54
Final Sequel	3573.57
Fire Resistance Suit of Provolone	626.43
Holy Blade of Gouda	766.72
Manganiello Bros.	5998.50
Manganiello Bros. Tee	1168.83
Mediocre Kingdoms	4648.14
Orvil the Wolverine	4278.93

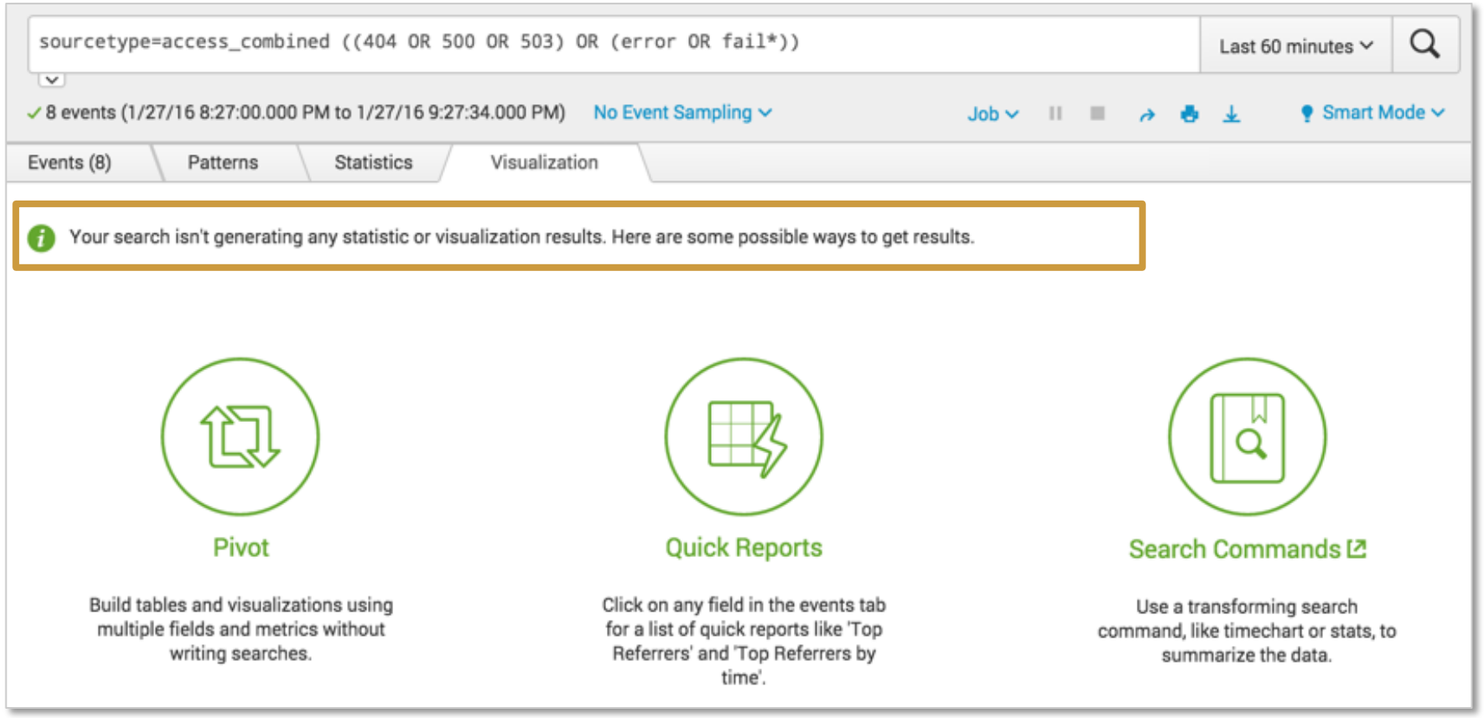


Map

Single value

Viewing Results as a Visualization

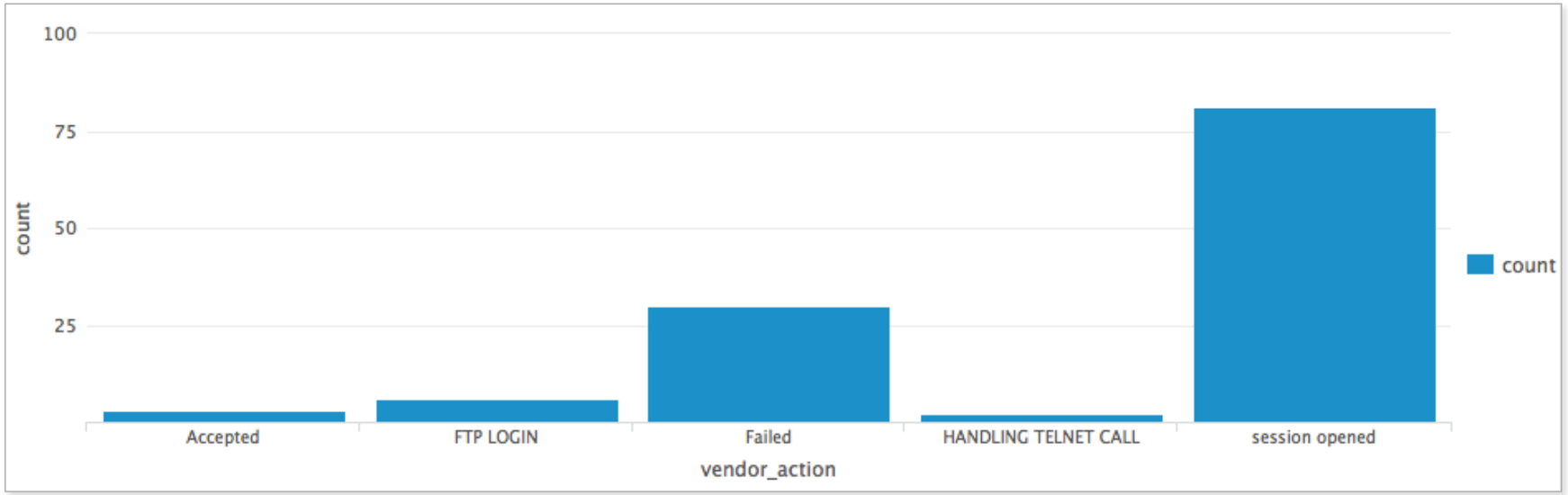
- Not all searches can be visually represented
- A data series is a sequence of related data points that are plotted in a visualization
- Data series can generate any statistical or visualization results



Data Structure Requirements – Single Series

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

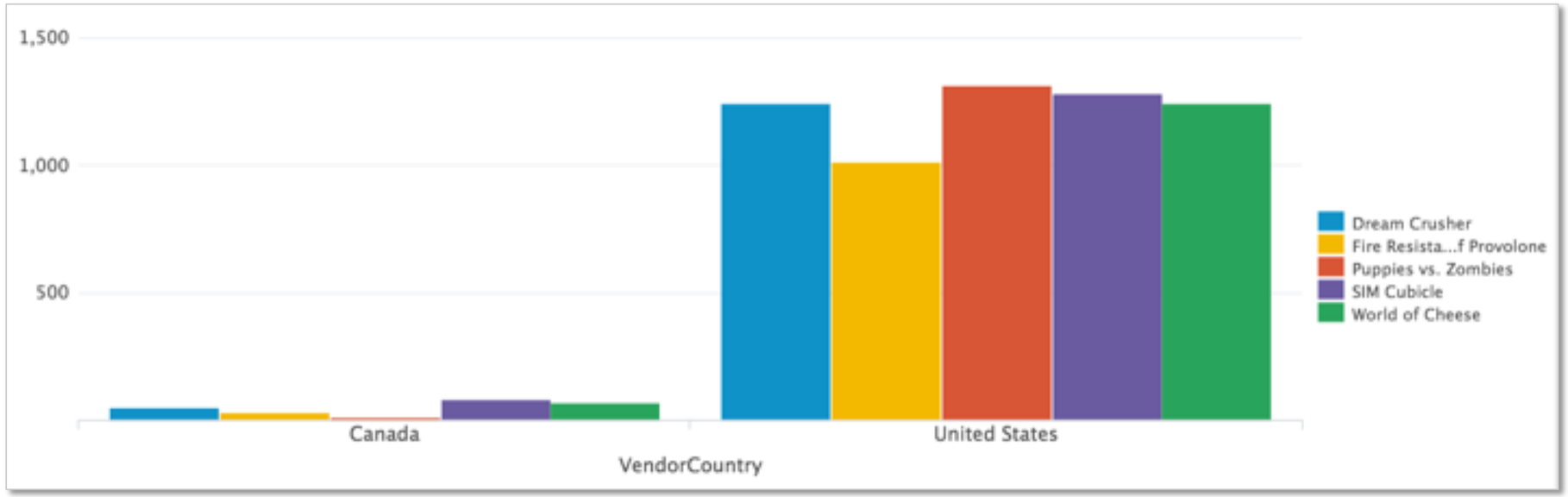
vendor_action	count
Accepted	2
FTP LOGIN	6
Failed	29
HANDLING TELNET CALL	2
Invalid user	3
session opened	59



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	48	31	14	83	69
United States	1245	1012	1317	1283	1246

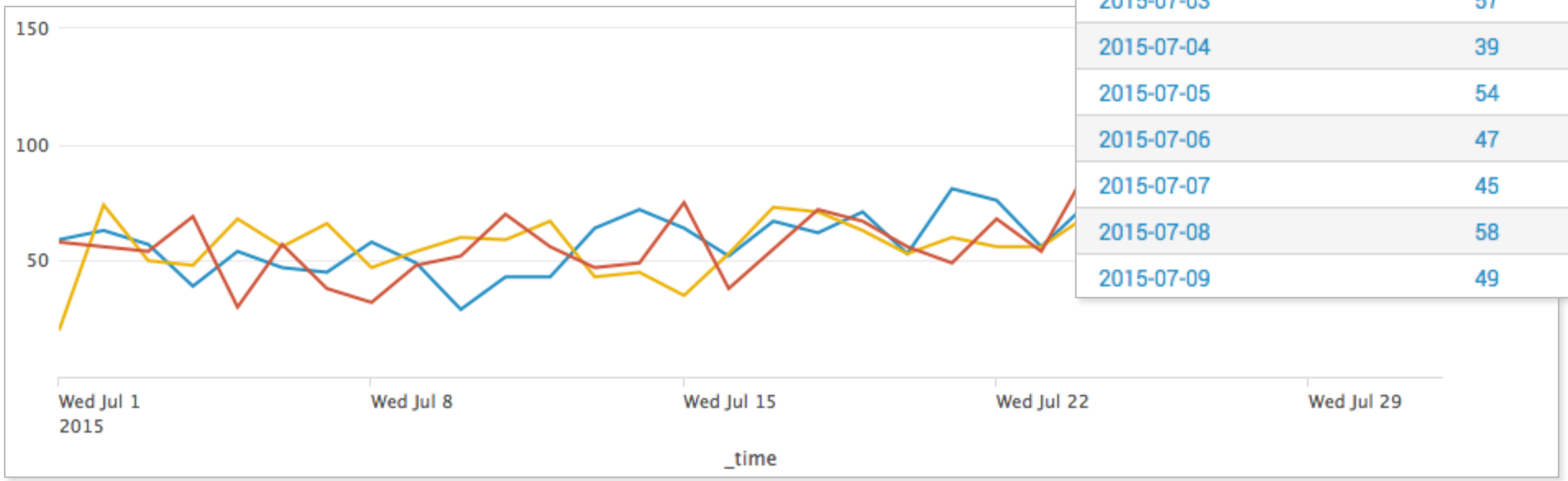



```
sourcetype=vendor_sales VendorID<4000 | chart count over VendorCountry by product_name limit=5 useother=f
```

Last 30 days

Data Structure Requirements – Time Series

Time series display statistical trends over time



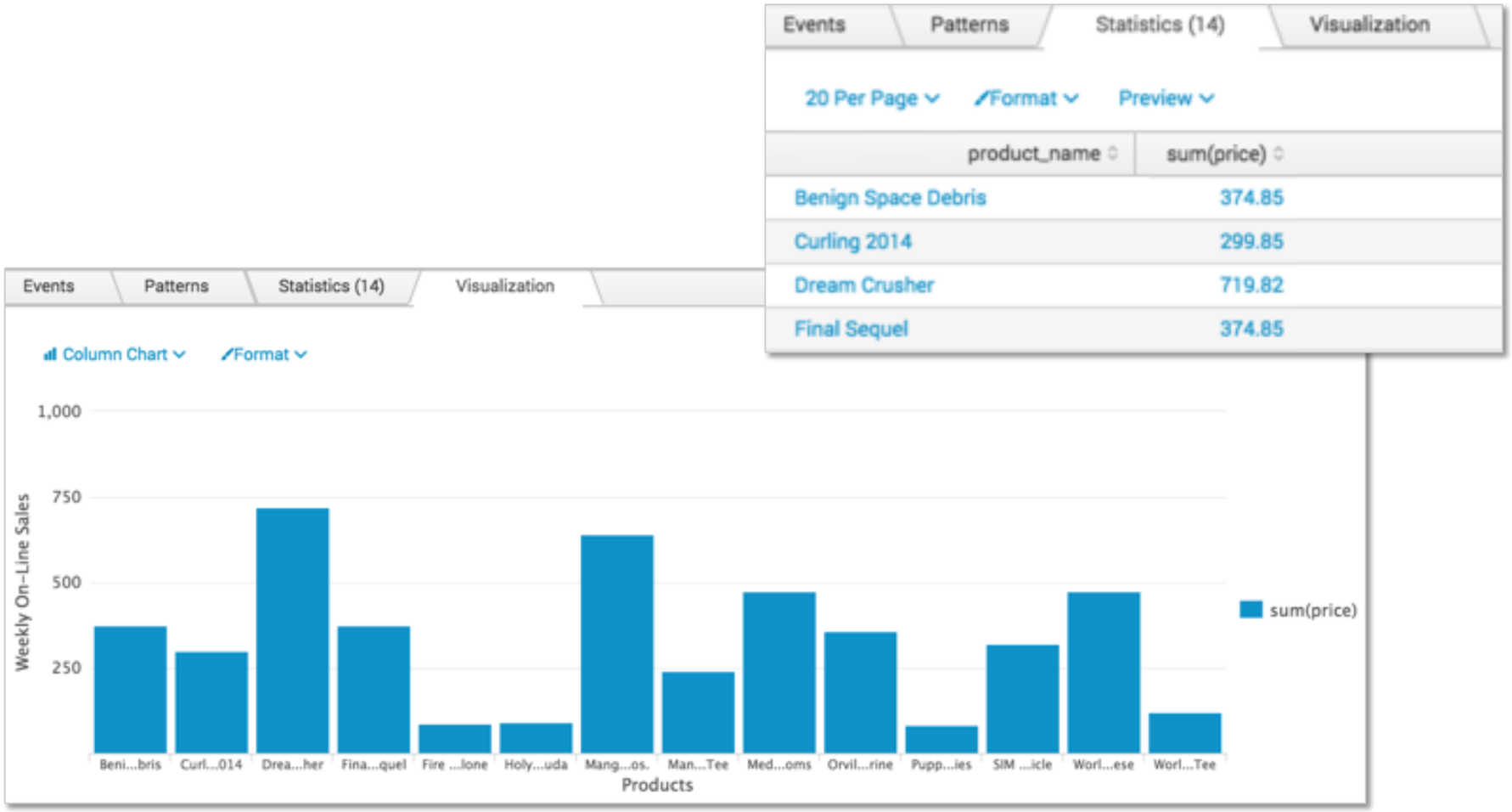
_time	www1	www2	www3
2015-07-01	59	20	58
2015-07-02	63	74	56
2015-07-03	57	50	54
2015-07-04	39	48	69
2015-07-05	54	68	30
2015-07-06	47	56	57
2015-07-07	45	66	38
2015-07-08	58	47	32
2015-07-09	49	54	48

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

Previous month

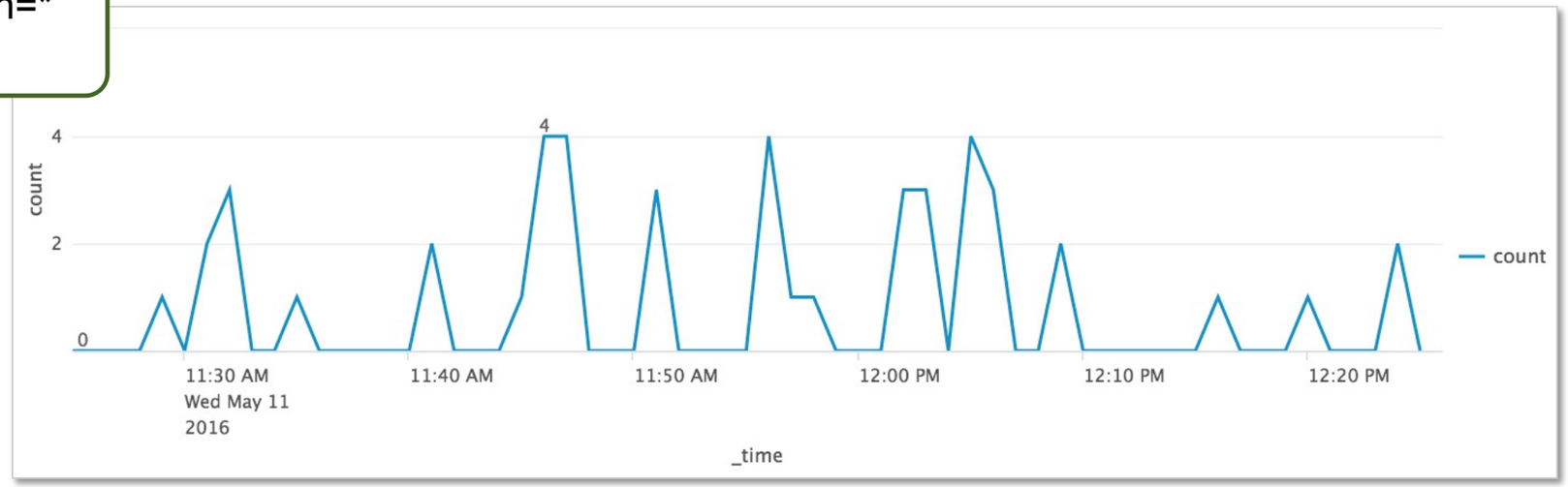
Viewing Results as a Chart

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

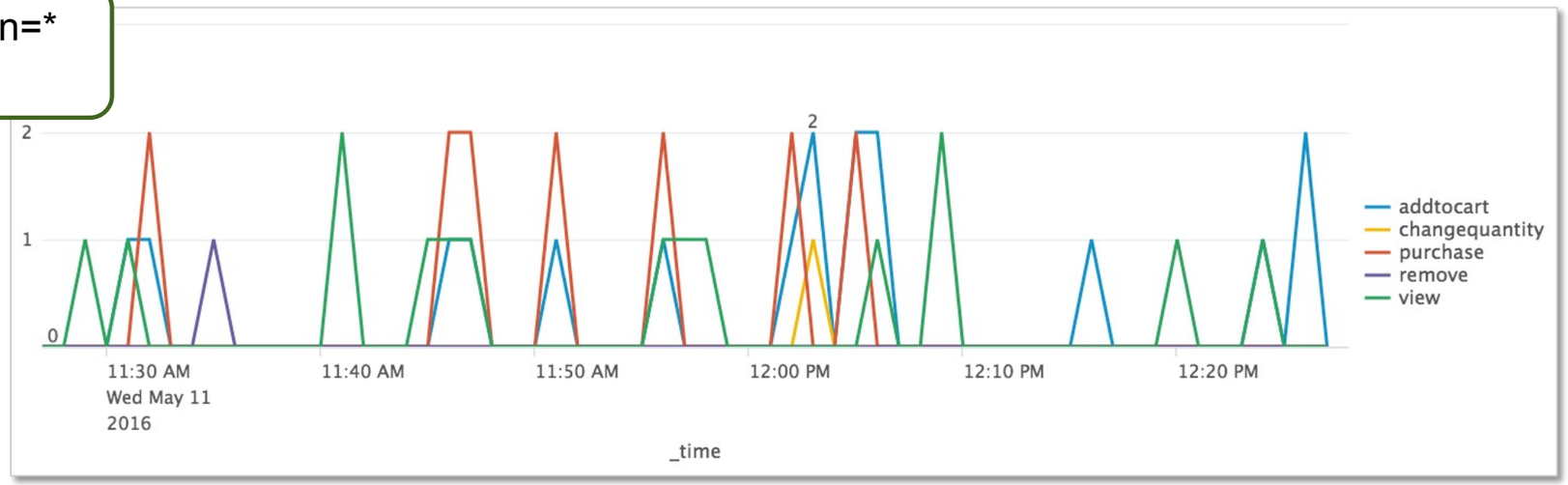


Charts – Line

```
sourcetype=access_combined action=*
| timechart count
```

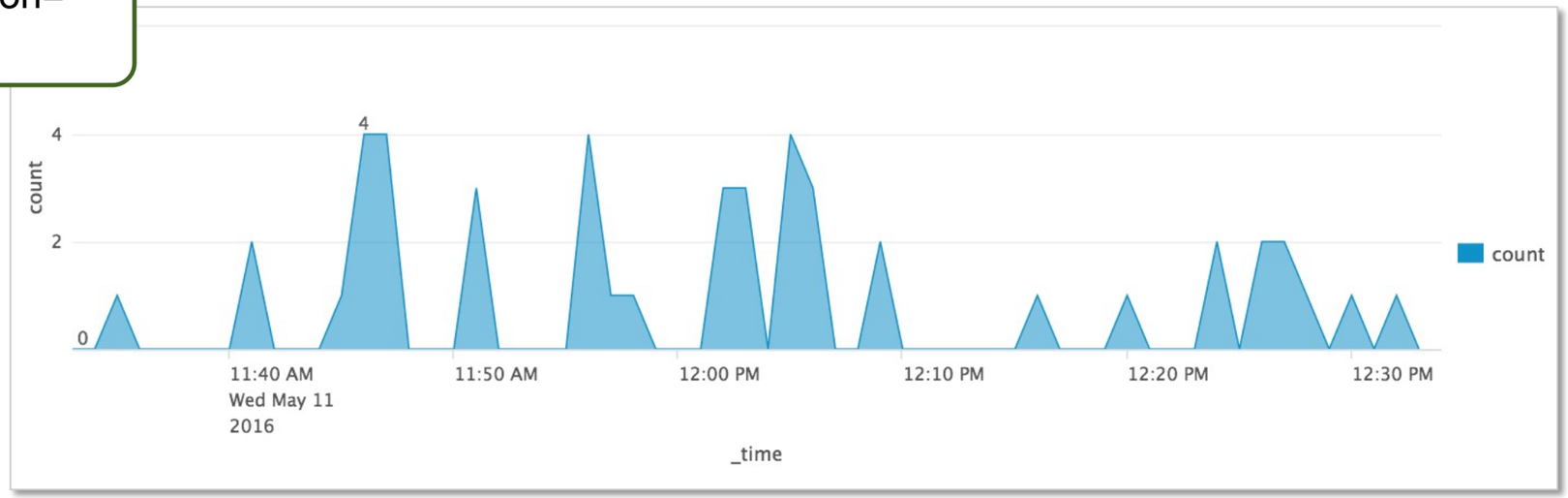



```
sourcetype=access_combined action=*
| timechart count by action
```

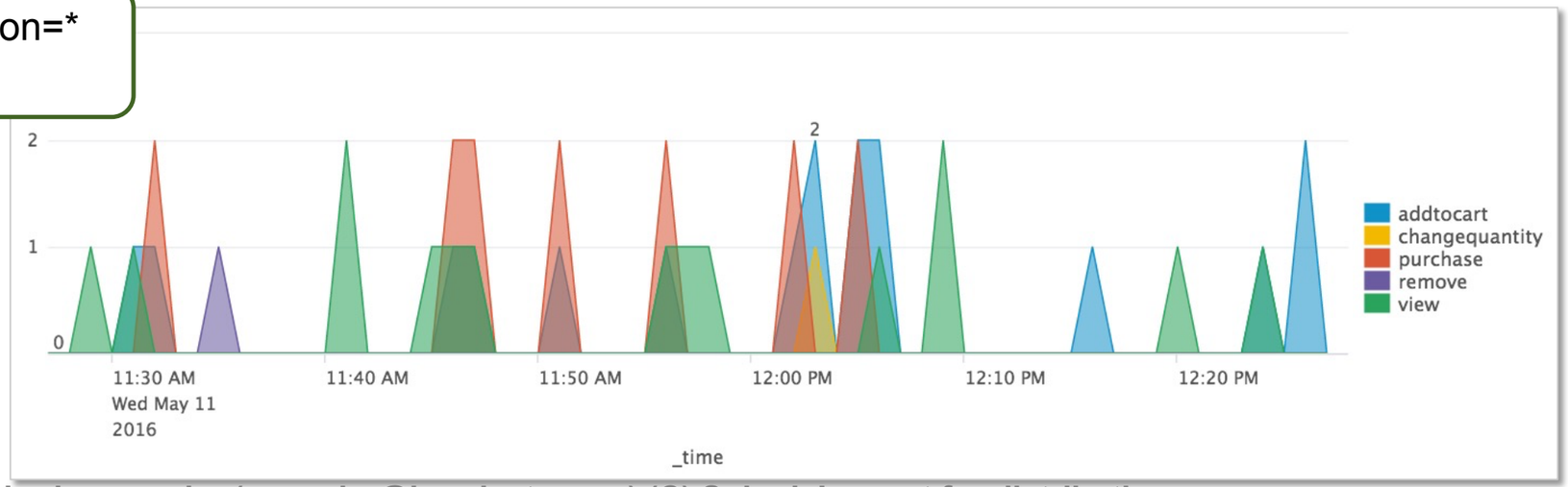


Charts – Area

```
sourcetype=access_combined action=*
| timechart count
```

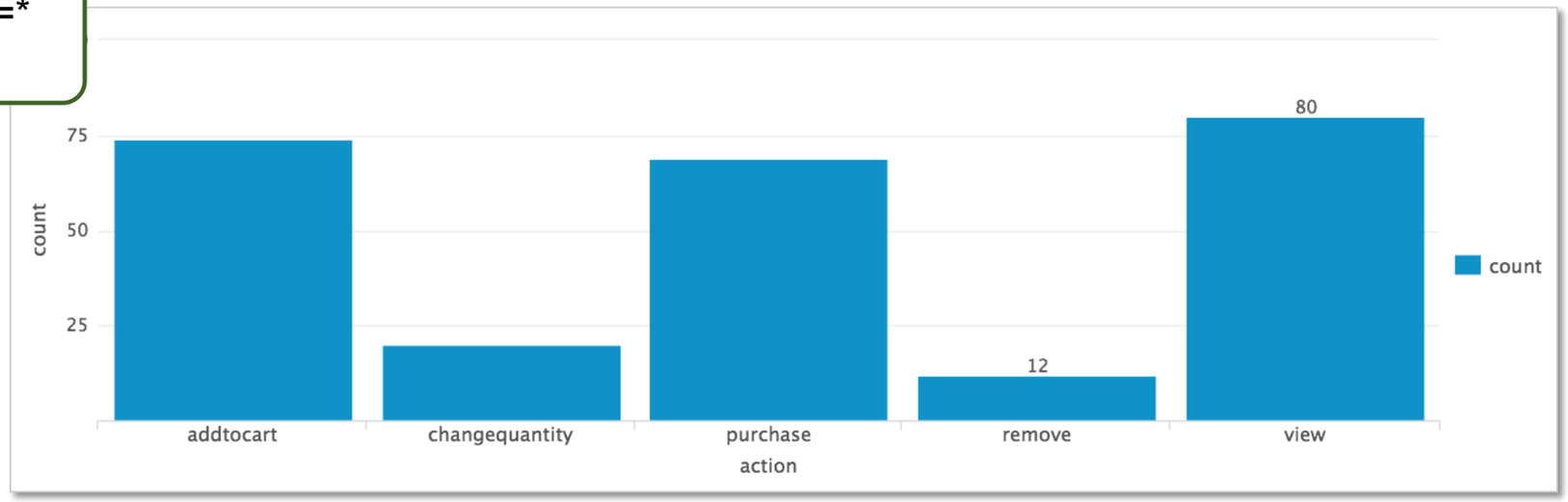



```
sourcetype=access_combined action=*
| timechart count by action
```

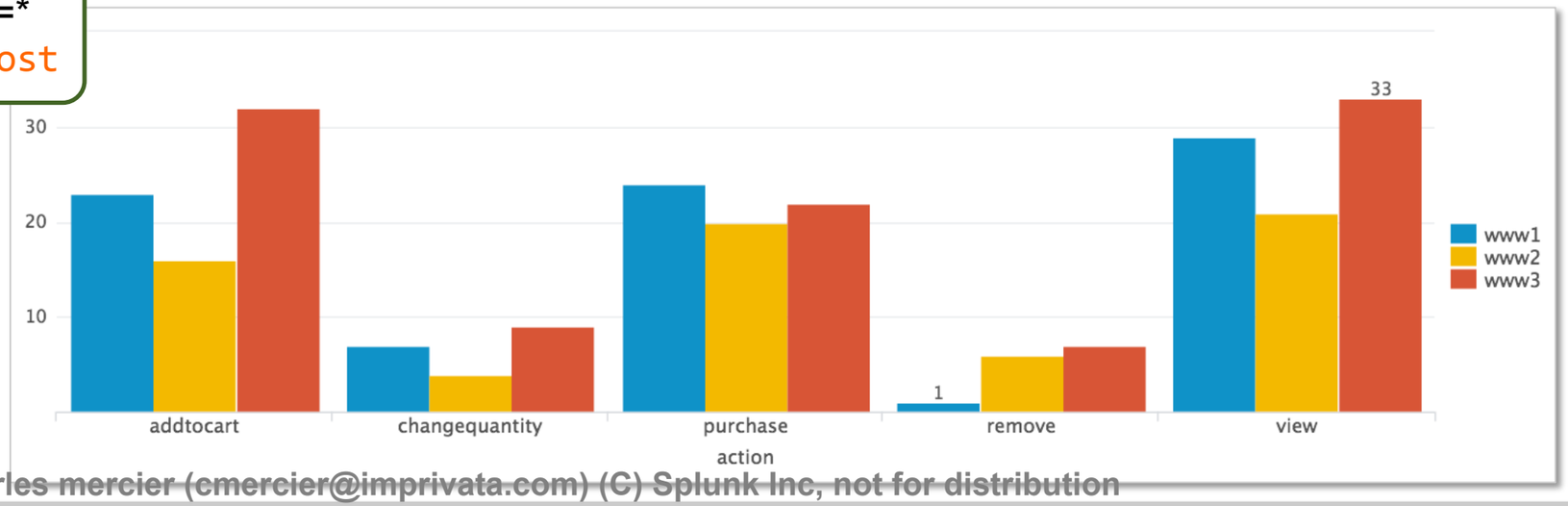


Charts – Column

```
sourcetype=access_combined action=*
| chart count over action
```

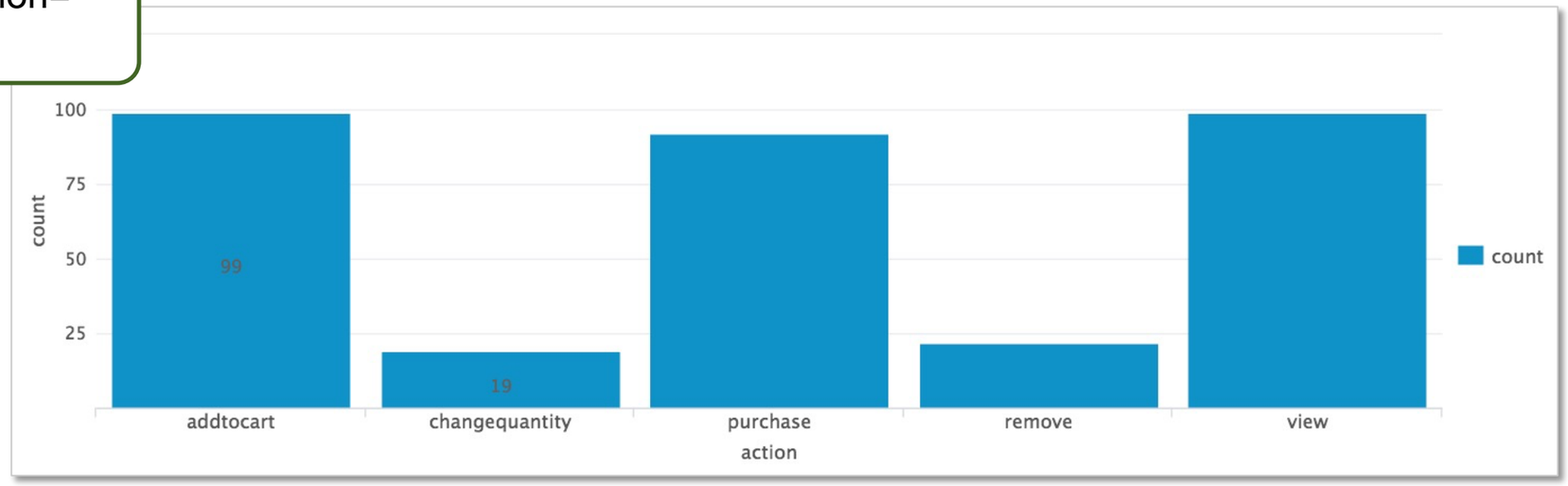



```
sourcetype=access_combined action=*
| chart count over action by host
```

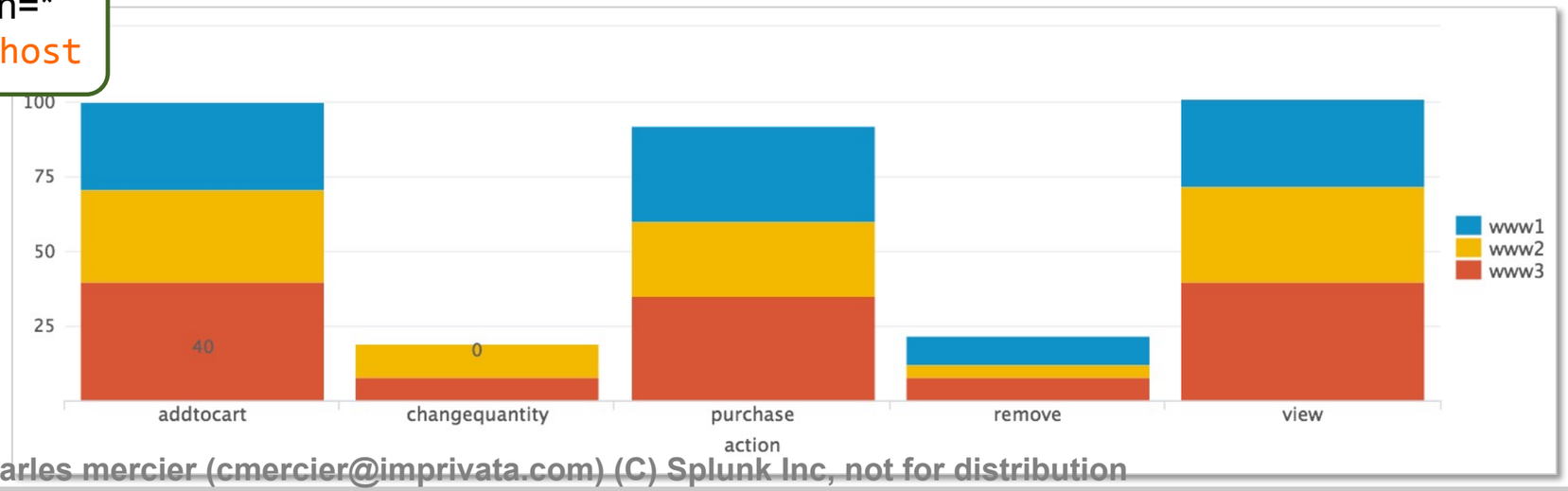


Charts – Column (Formatted as Stacked)

```
sourcetype=access_combined action=*
| chart count over action
```

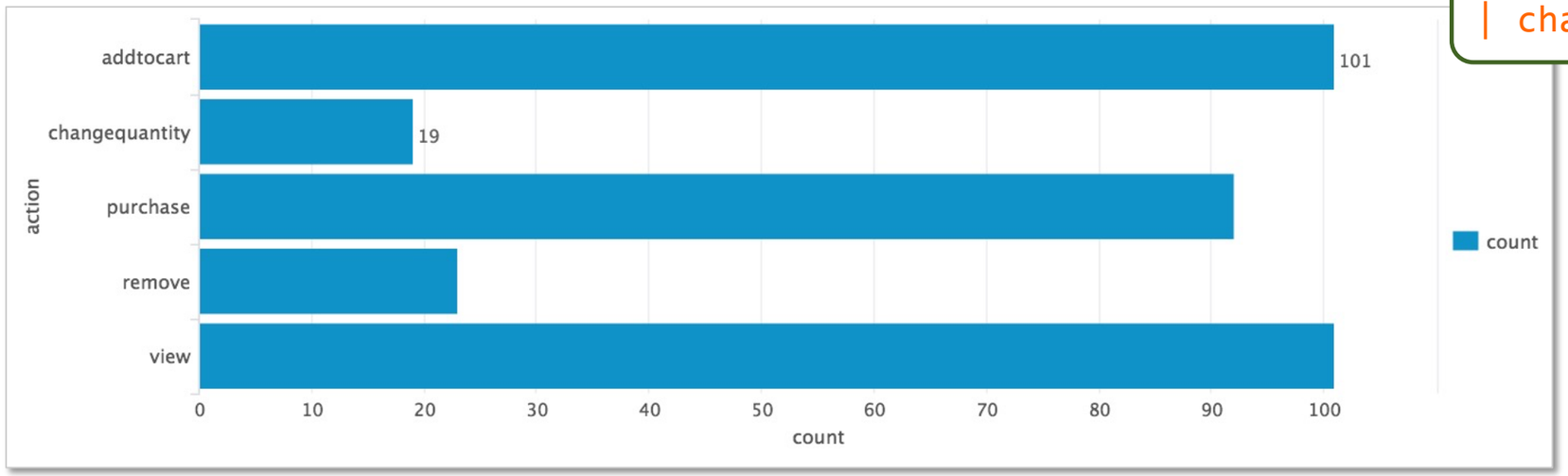



```
sourcetype=access_combined action=*
| chart count over action by host
```

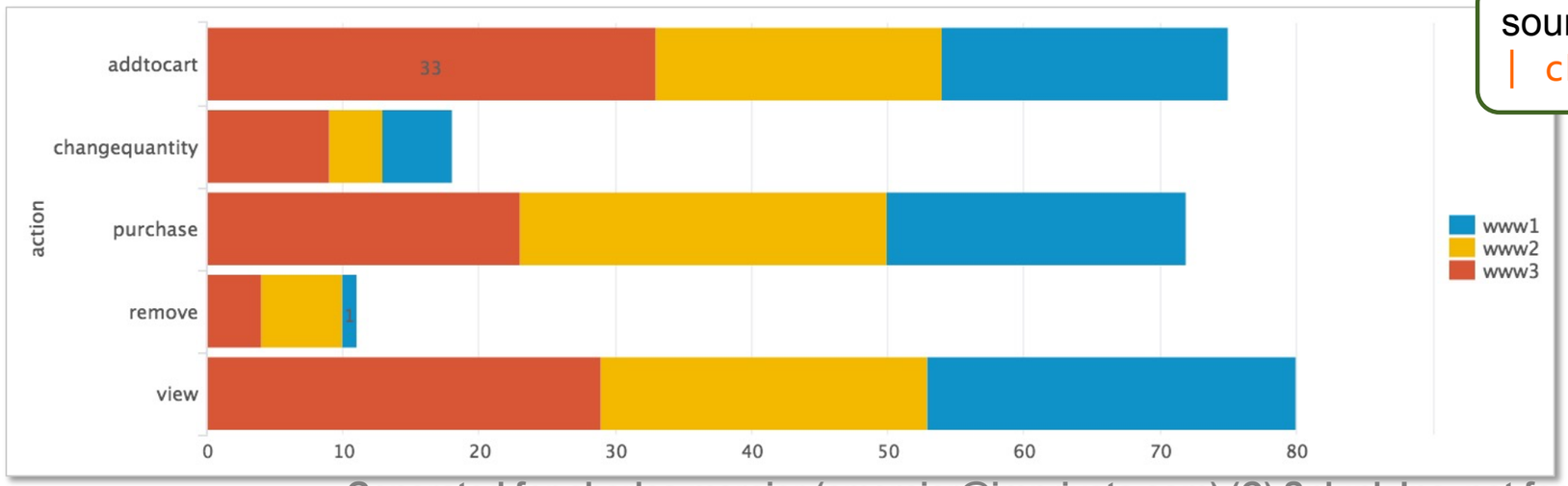


Charts – Bar

```
sourcetype=access_combined action=*
| chart count over action
```

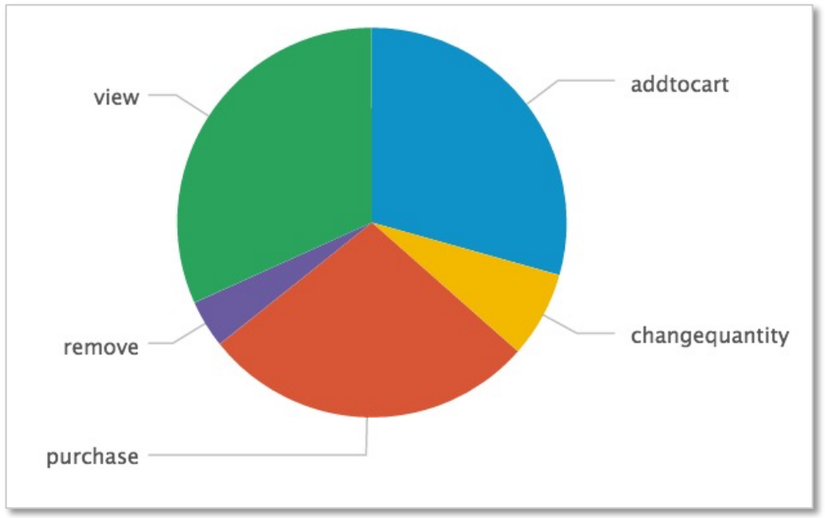



```
sourcetype=access_combined action=*
| chart count over action by host
```



Charts – Pie

```
sourcetype=access_combined action=*
| chart count over action
```



Charts – Scatter

- Scatter chart shows trends in the relationships between discrete data values

```
sourcetype=vendor_sales
VendorID >=7000 AND VendorID <=8999
| stats sum(price) as "Total Sales",
  values(price) as "Price", count as Count
  by VendorCountry, product_name
```

- Generally, it shows discrete values that do not occur at regular intervals or belong to a series

Charts – Bubble

- Bubble chart provides a visual way to view a three dimensional series
- Each bubble plots against two dimensions on the X and Y axes
- The size of the bubble represents the value for the third dimension

```
sourcetype=vendor_sales
VendorID >=7000 AND VendorID <=8999
| stats sum(price) as "Total Sales",
  values(price) as "Price", count as Count
  by VendorCountry, product_name
```

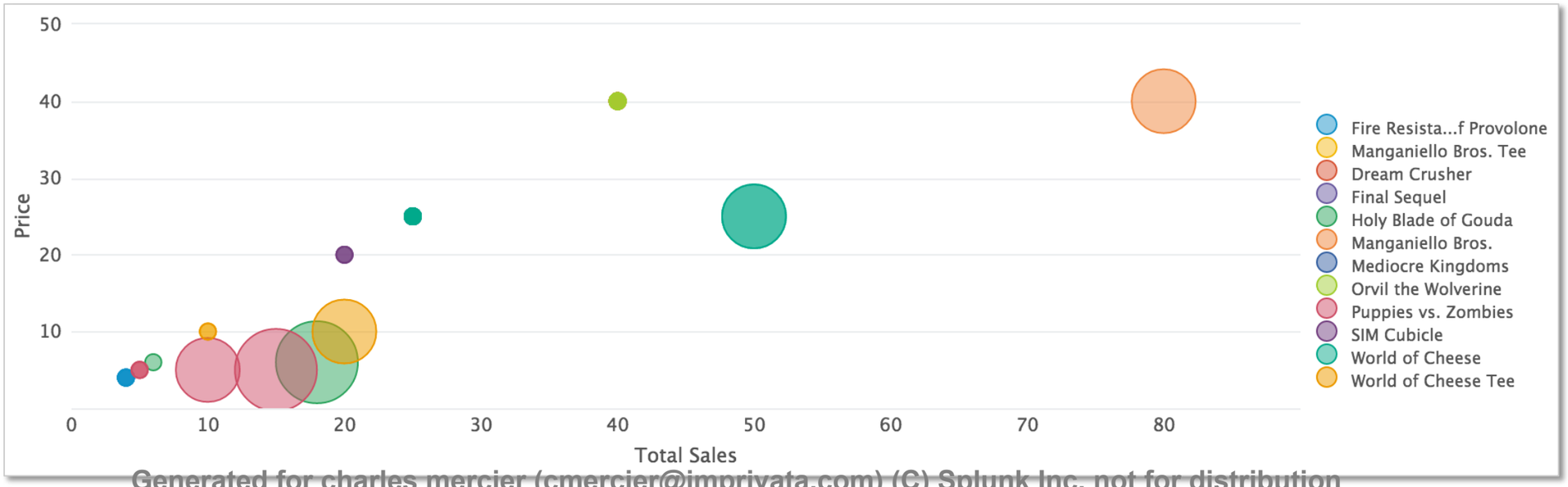


chart Command

- 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 over is the x-axis
  - Using the over and by clauses divides the data into sub-groupings
    - the values from by will display in the legend

```
chart avg(bytes) over host
```

  - The host values will 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

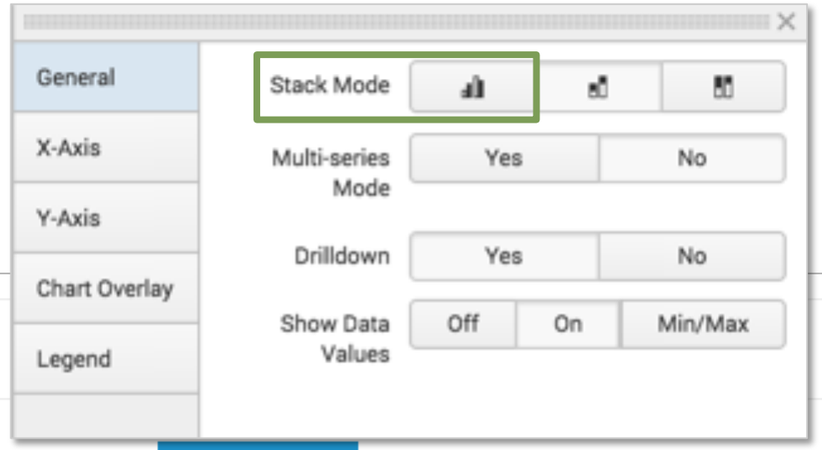
chart Command – over <field> Example

- This example shows a basic chart
- count function tallies the number of events for each value in the result set

Scenario

Display a count of vendor actions over the last 24 hours.

```
sourcetype=linux_secure
| chart count over vendor_action
```



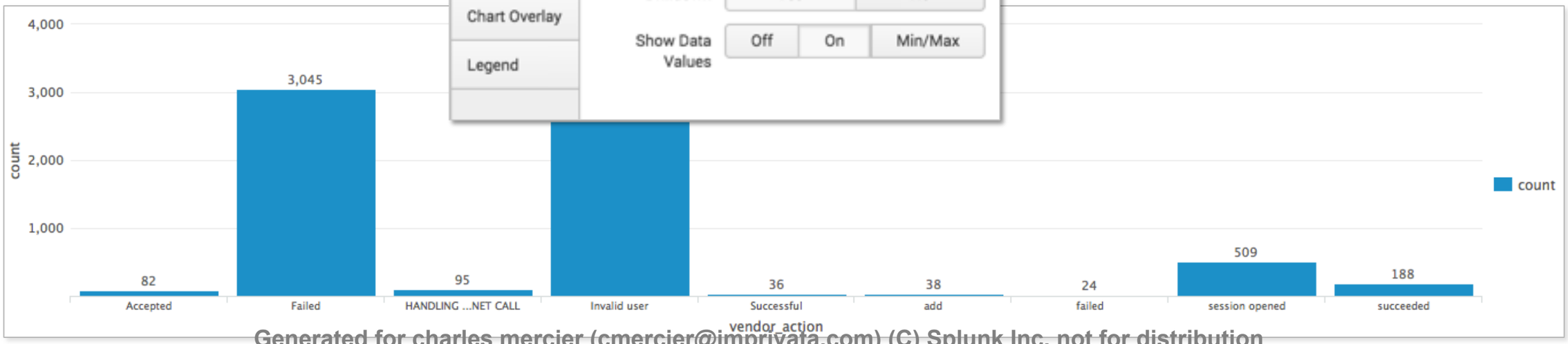


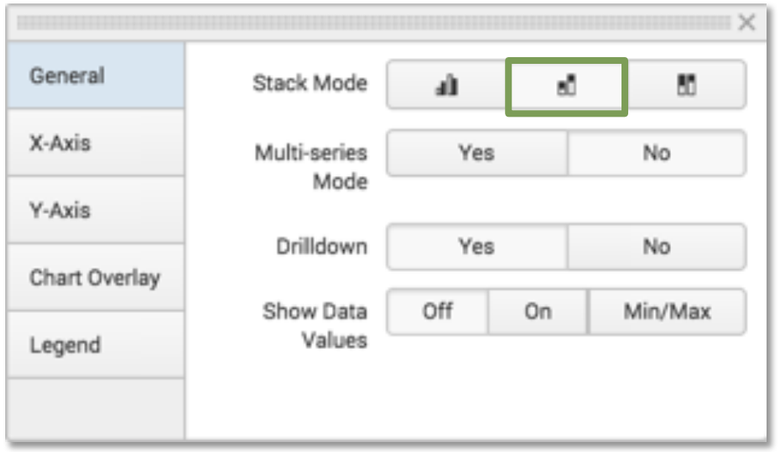
chart Command – over <field> by <field>

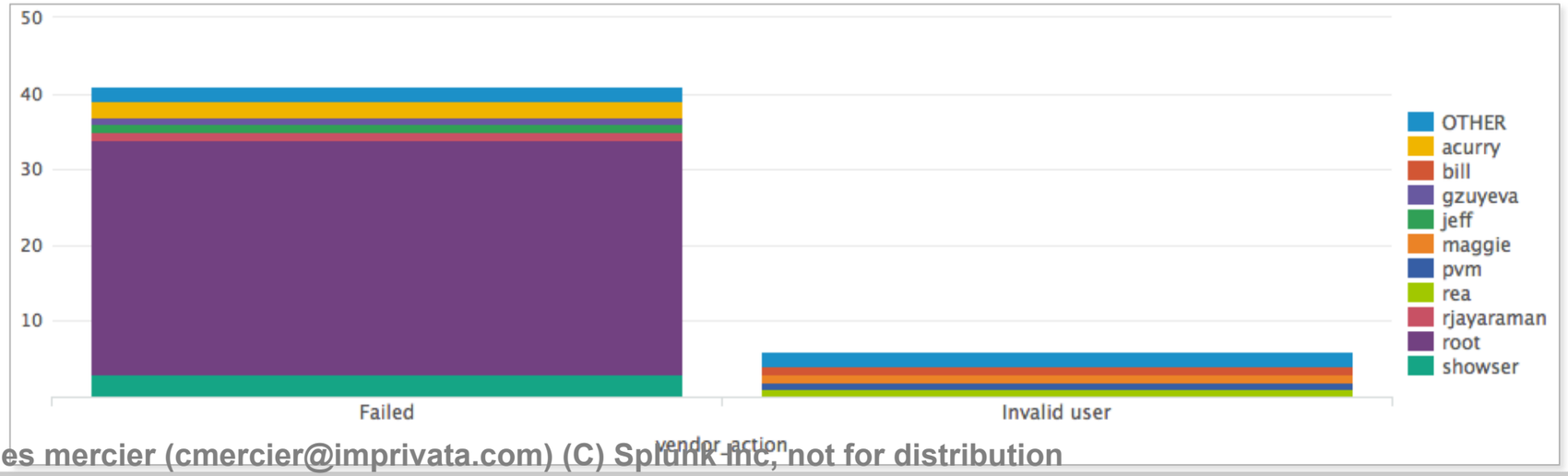
- Use **by**, followed by a field name, to split results
  - Unlike `stats`, only ONE value can be specified after by
- In this example, results are grouped by `vendor_action`, then split by user
- This example displays stacked columns

Scenario

Display a count of vendor actions by user over the last 60 minutes.

```
sourcetype=linux_secure (invalid OR fail*)
| chart count over vendor_action by user
```





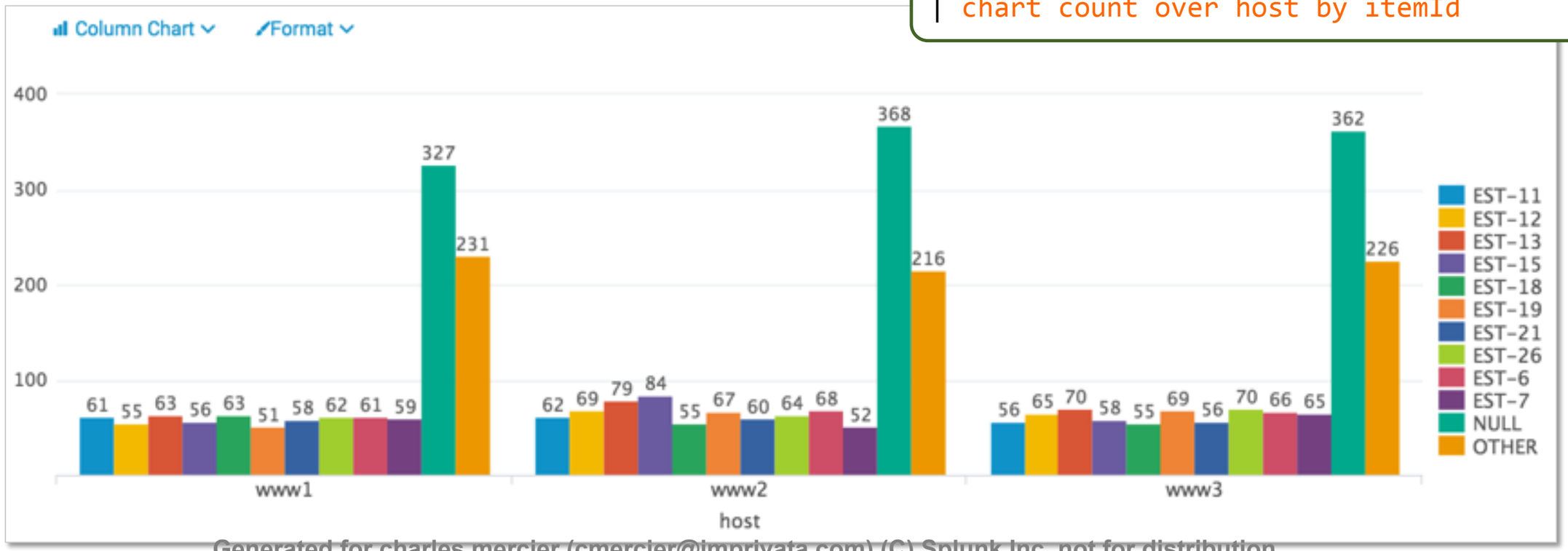
Including Null and Other Values

Notice the results are skewed by NULL and OTHER values, shown by default, that we may not want to show

Scenario

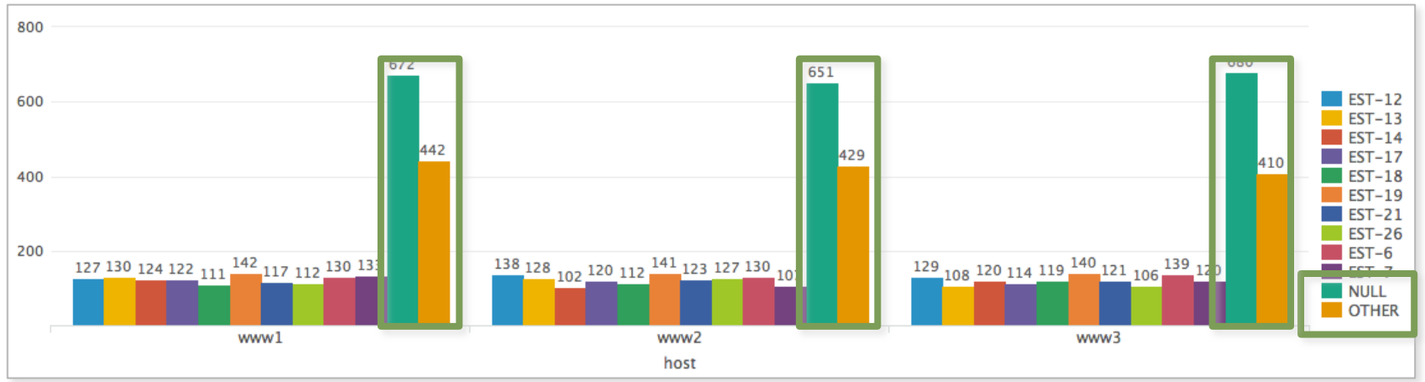
Display a count of unsuccessful web transactions by host for each item over the last 7 days.

```
sourcetype=access_combined status>399
| chart count over host by itemId
```

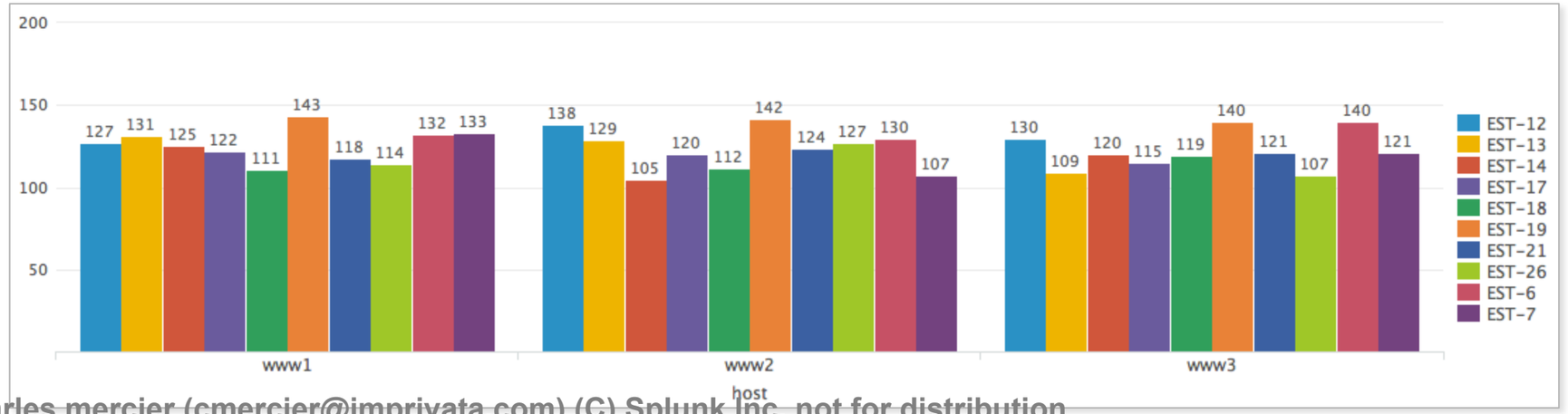


Omitting Null and Other Values

- `chart` and `timechart` commands automatically filter the series they plot to the 10 highest values
  - Surplus values are grouped into OTHER
- To remove empty (`null`) and other field values from the display, use these options:
  - `useother=f`
    `usenull=f`




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



Best Practice

To remove null values, add `itemId=*` to base search.

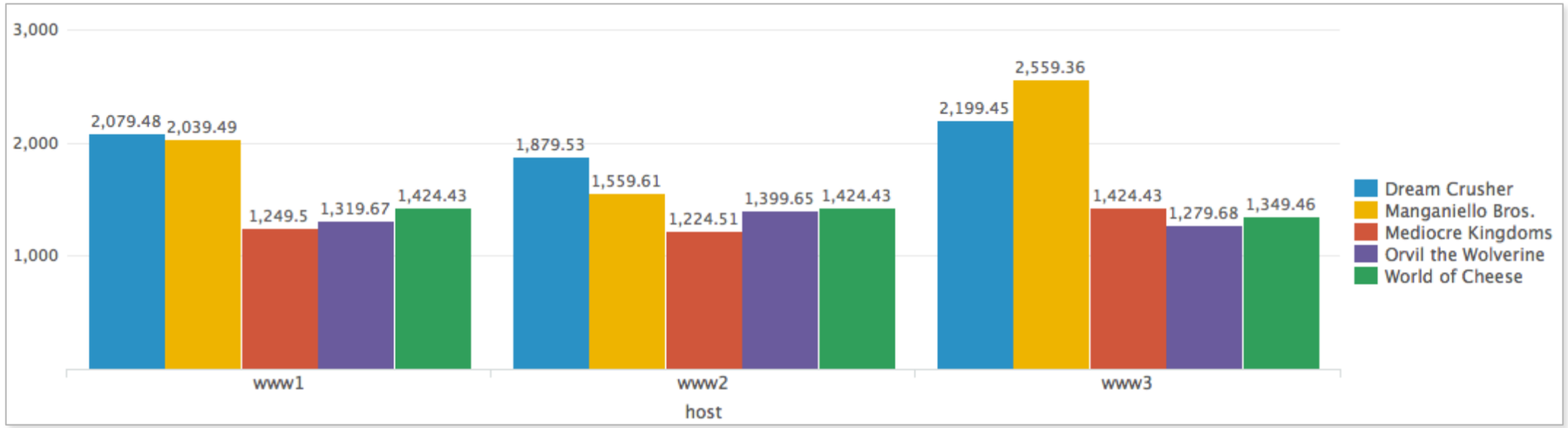
Limiting the Number of Values

To adjust the number of plotted series, use the `limit` argument, `limit=0` for unlimited

Scenario

Display sales per host for the top 5 products over the last 7 days.

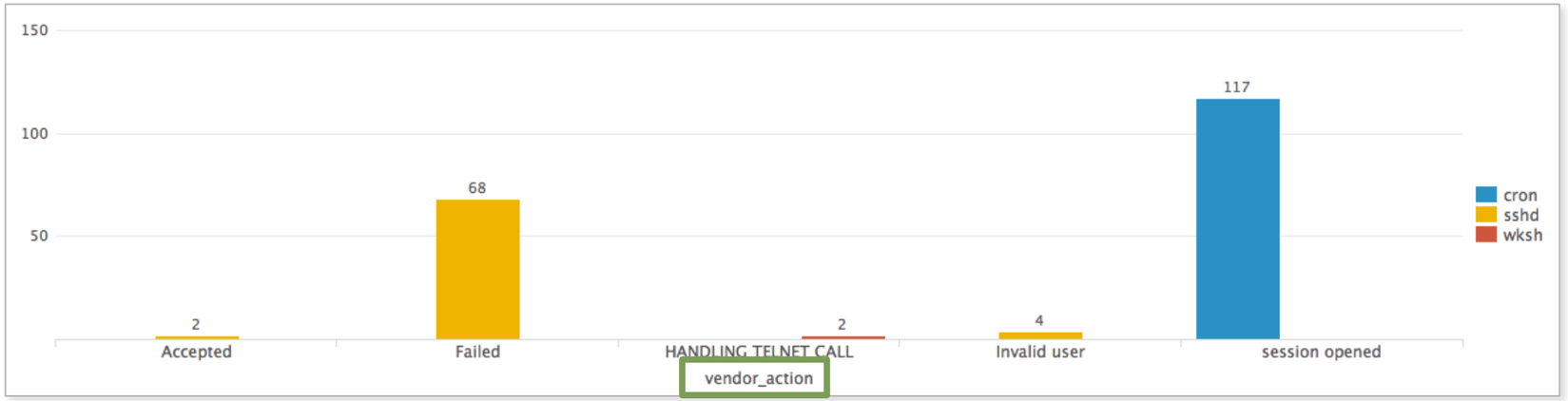
```
sourcetype=access_combined
  action=purchase status=200
| chart sum(price) over host
  by product_name limit=5 useother=f
```



Visualizations – x and y Axes

- For line, area, and column charts, the x axis is horizontal

```
sourcetype=linux_secure
| chart count over vendor_action by app
```



- For bar chart, the x axis is vertical

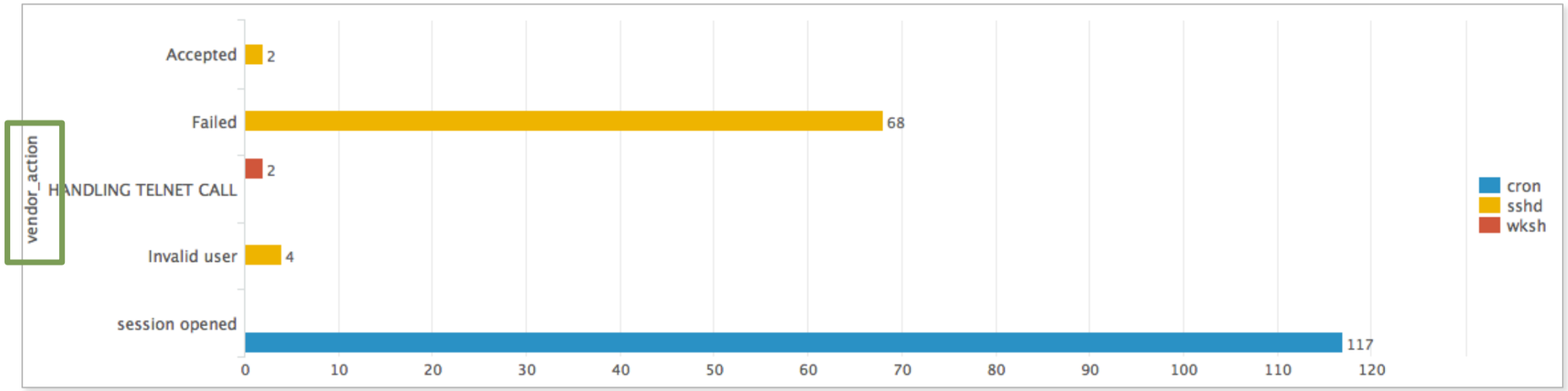
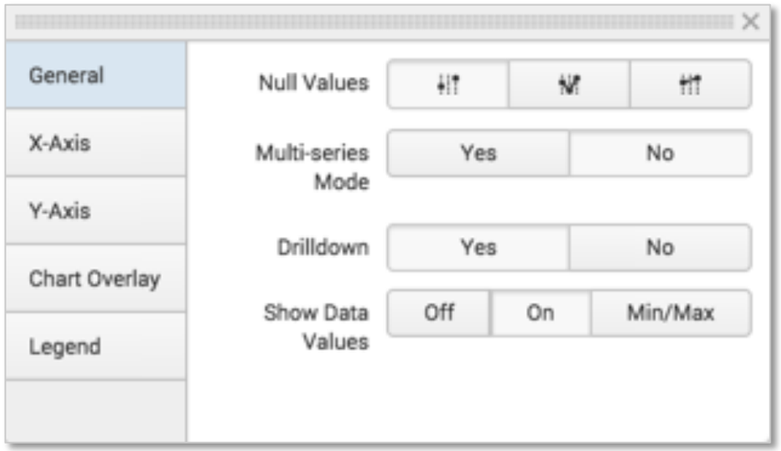
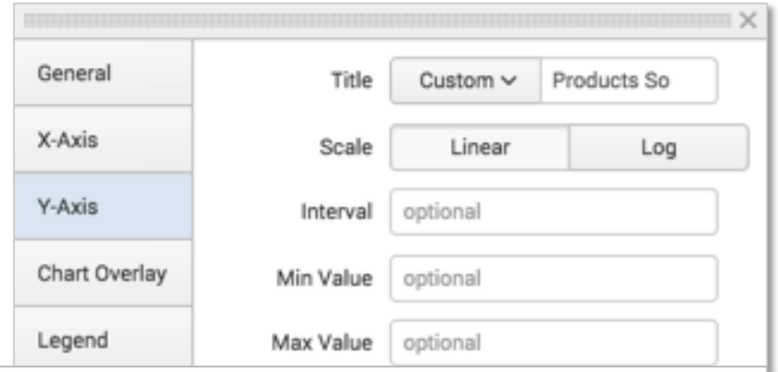
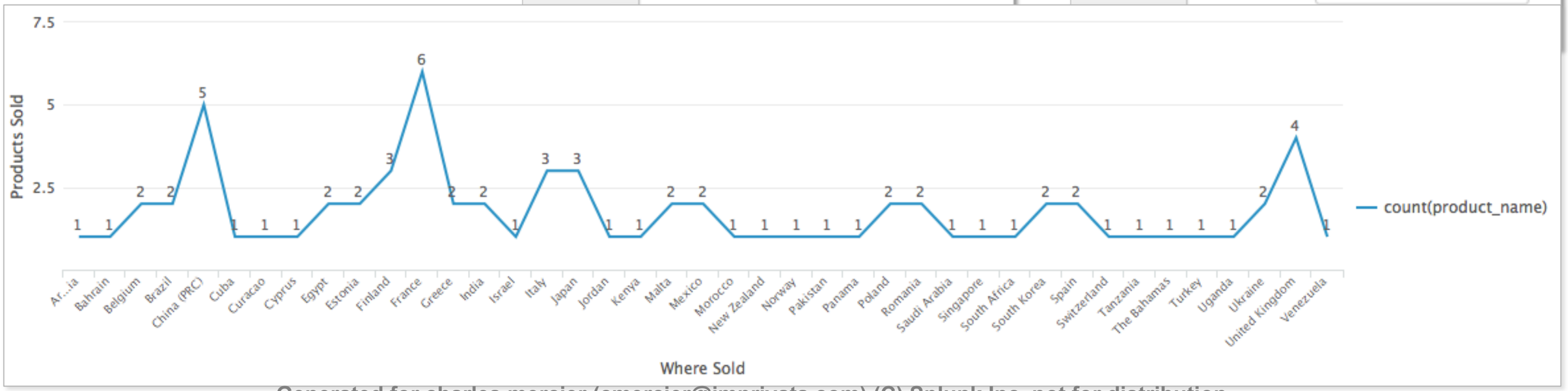


chart Command – Formatting

```
sourcetype=vendor_sales VendorID > 4000
| chart count(product_name) by VendorCountry
```







`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

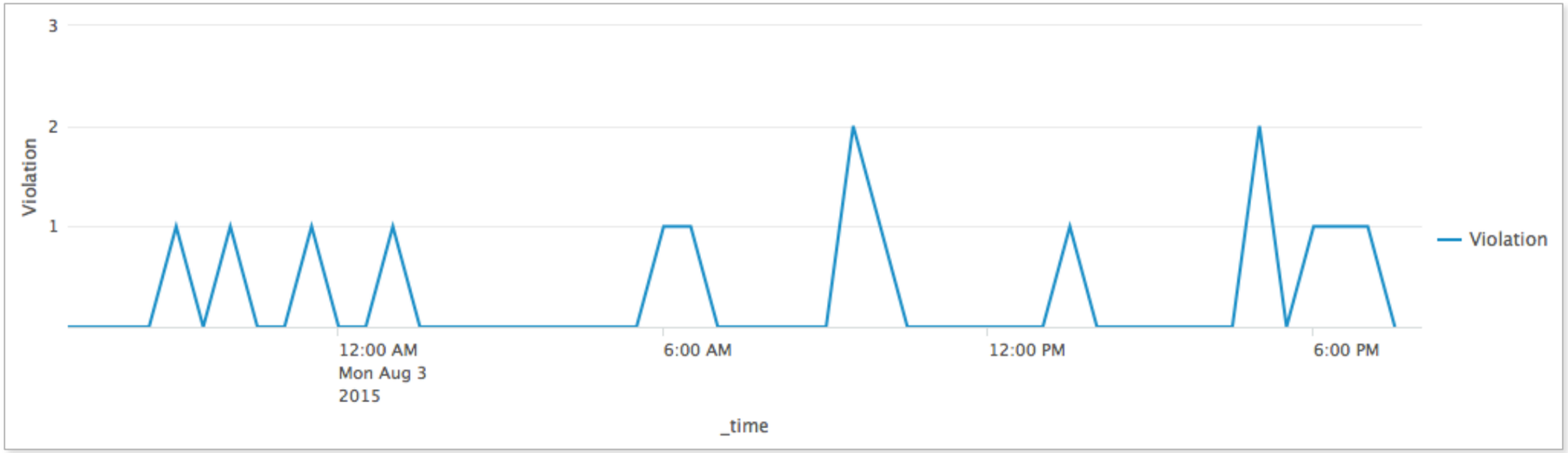
`timechart` Command – Example

This basic timechart displays the number of usage violations

Scenario

How many usage violations have occurred in the last 24 hours?

```
sourcetype=cisco_wsa_squid usage=Violation
| timechart count
```



Note

Functions and arguments used with stats and chart can also be used with timechart.

timechart Command – Multiple Values

- Splitting by the usage field, each line represents a unique field value
  - Unlike stats, only ONE value can be specified after by
- y-axis represents the count for each field value

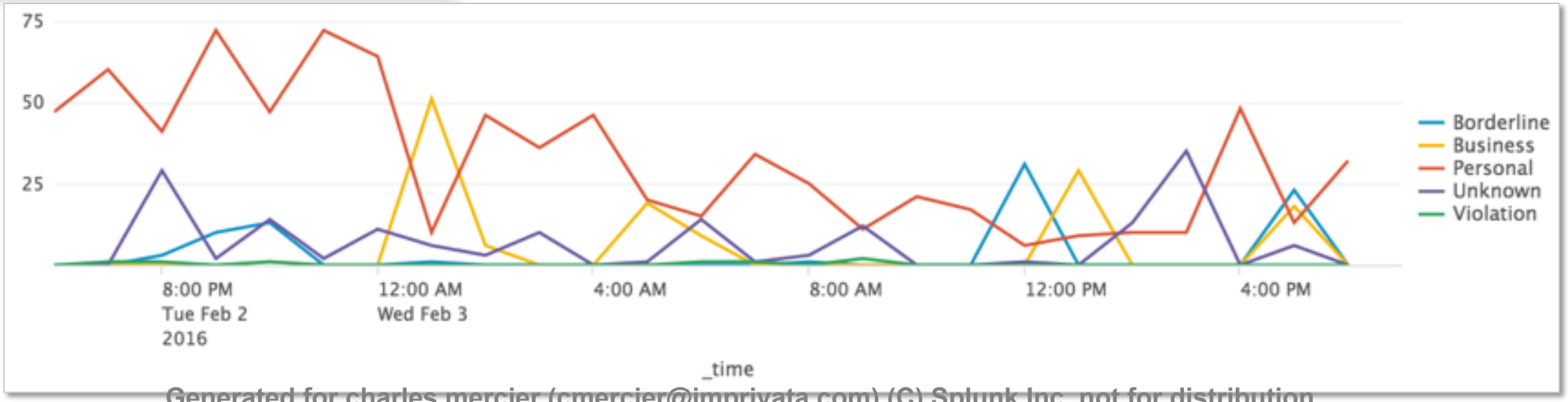
Scenario

What is the overall usage trend for the last 24 hours?

```
sourcetype=cisco_wsa_squid
| timechart span=1h count by usage
```

Note

Using timechart, you can split by a maximum of one field because _time is the implied first by field.



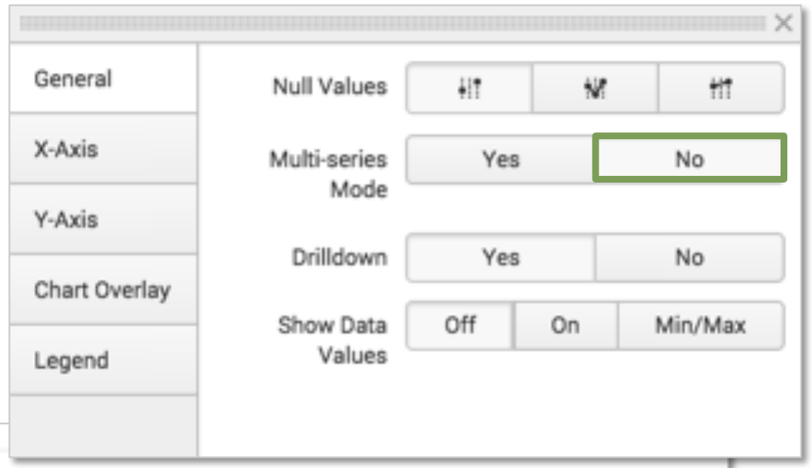
timechart Command – Multi-series: No

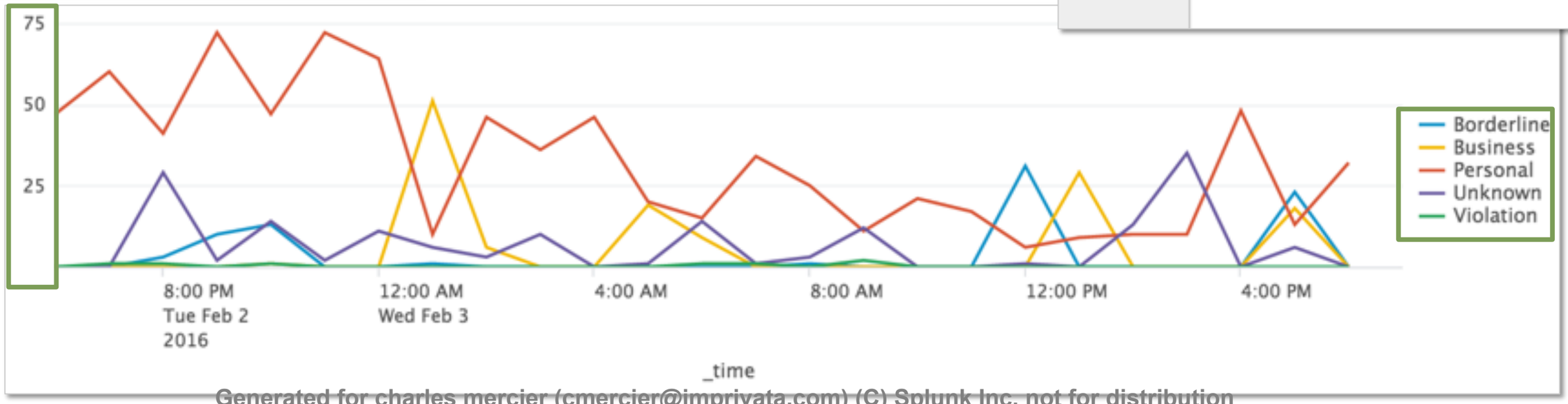
When the Multi-series mode is set to **No**, all fields share the y-axis

Scenario

What is the overall usage trend for the last 24 hours?

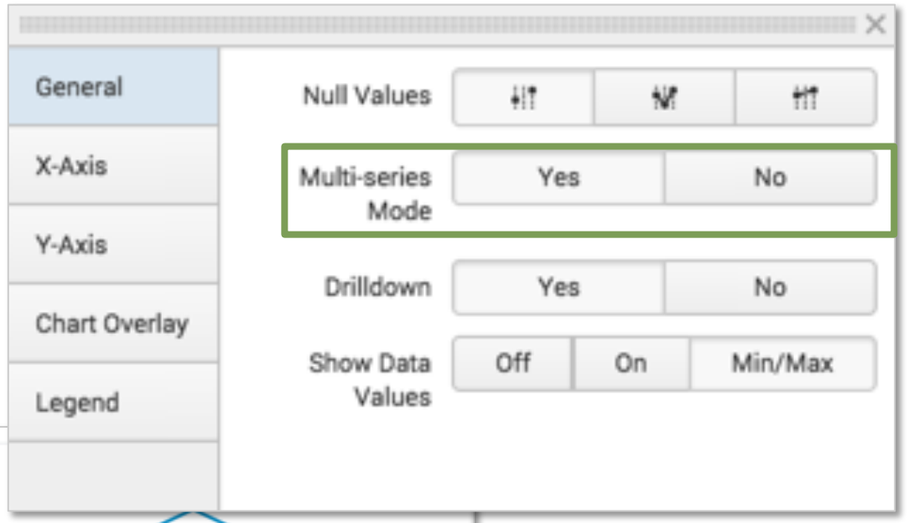
```
sourcetype=cisco_wsa_squid
| timechart span=1h count by usage
```

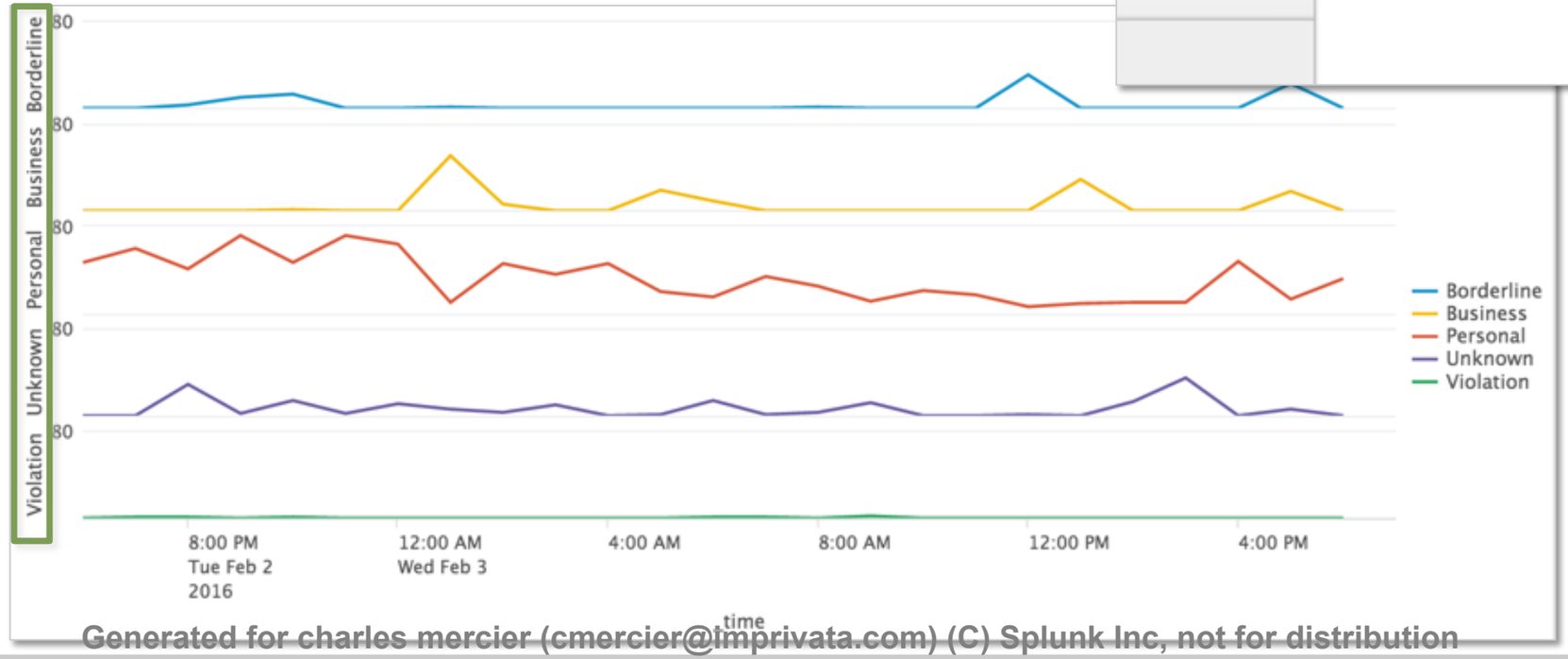




`timechart` Command – Multi-series: Yes

- 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 to provide 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`

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

✓ 462 events (2/3/16 10:53:00.000 AM to 2/3/16 2:53:04.000 PM) | No Event Sampling

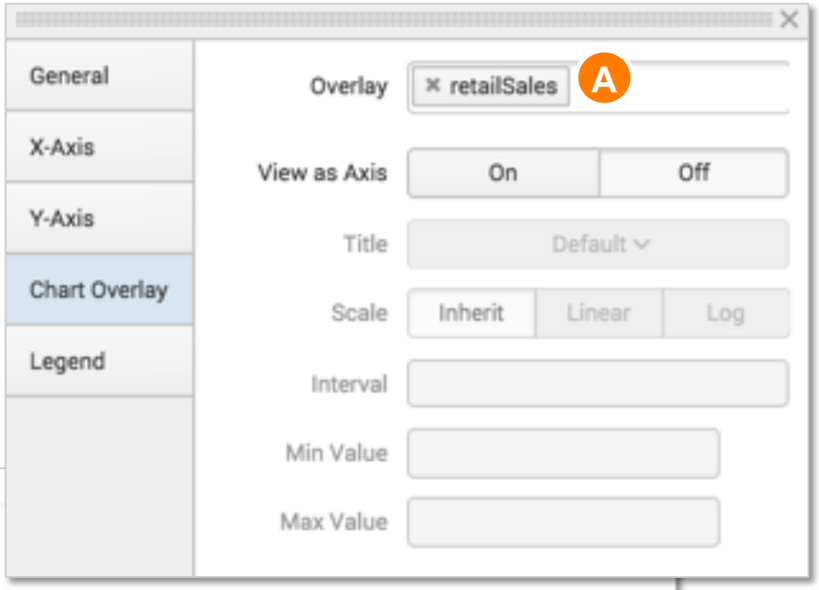
Events | Patterns | Statistics (17) | Visualization

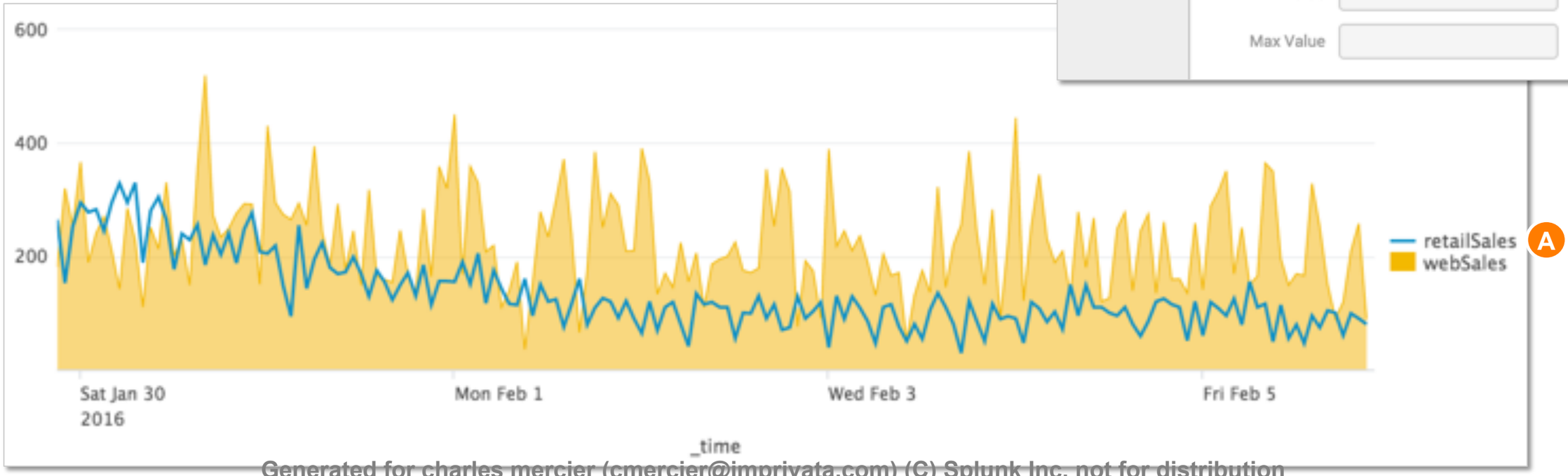
20 Per Page | Format | Preview

_time	Failed	Invalid user
2016-02-03 10:45:00	13	3
2016-02-03 11:00:00	31	1
2016-02-03 11:15:00	25	4
2016-02-03 11:30:00	20	6
2016-02-03 11:45:00	33	5
2016-02-03 12:00:00	37	3
2016-02-03 12:15:00	27	1
2016-02-03 12:30:00	34	2
2016-02-03 12:45:00	22	5
2016-02-03 13:00:00	28	4

Formatting – Chart Overlay

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





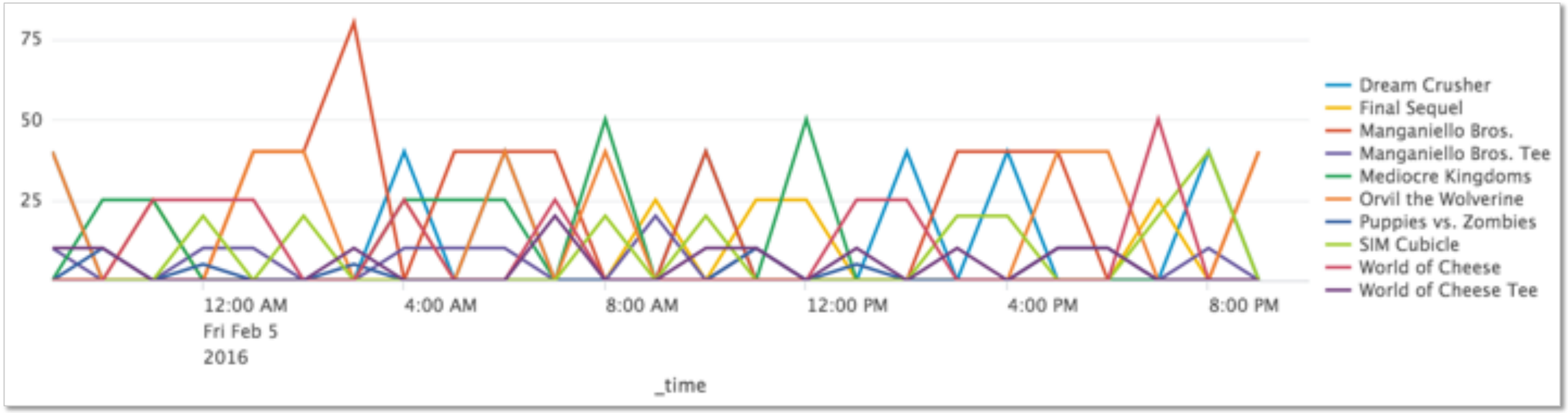
timechart Command – Statistical Functions

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

Scenario

How much retail revenue did we receive from each product during the last 24 hours?

```
sourcetype=vendor_sales
| timechart span=1h sum(price) by product_name
  useother=f usenull=f
```



Transforming Command Summary

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

Transforming Command Summary (cont.)

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

```
sourcetype=linux_secure
 | top src_ip, user, vendor_action, app
```

src_ip	user	vendor_action	app	count	percent
3.0.0.44	root	Failed	sshd	156	32.032854
175.45.176.223	root	Failed	sshd	66	13.552361
2.144.0.210	root	Failed	sshd	39	8.008214
152.206.0.63	madeyemi	Accepted	sshd	20	4.106776

```
sourcetype=linux_secure
 | rare src_ip, user, vendor_action, app
```

src_ip	user	vendor_action	app	count	percent
152.206.0.61	user	Failed	sshd	1	0.206186
175.45.176.223	ftp1	Failed	sshd	1	0.206186
175.45.176.223	man	Failed	sshd	1	0.206186
175.45.176.223	mark	Invalid user	sshd	1	0.206186

Transforming Command Summary (cont.)

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

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

src_ip	user	vendor_action	app	count
10.1.10.172	admin	Failed	sshd	2
10.1.10.172	administrator	Failed	sshd	1
10.1.10.172	apache	Failed	sshd	3
10.1.10.172	art	Failed	sshd	1
10.1.10.172	db	Failed	sshd	3
10.1.10.172	db2inst1	Failed	sshd	1
10.1.10.172	db4	Failed	sshd	1
10.1.10.172	desktop	Failed	sshd	1
10.1.10.172	email	Failed	sshd	1
10.1.10.172	games	Failed	sshd	1

Transforming Command Summary (cont.)

To calculate statistics with an arbitrary field as the x-axis (not `_time`), use `chart`

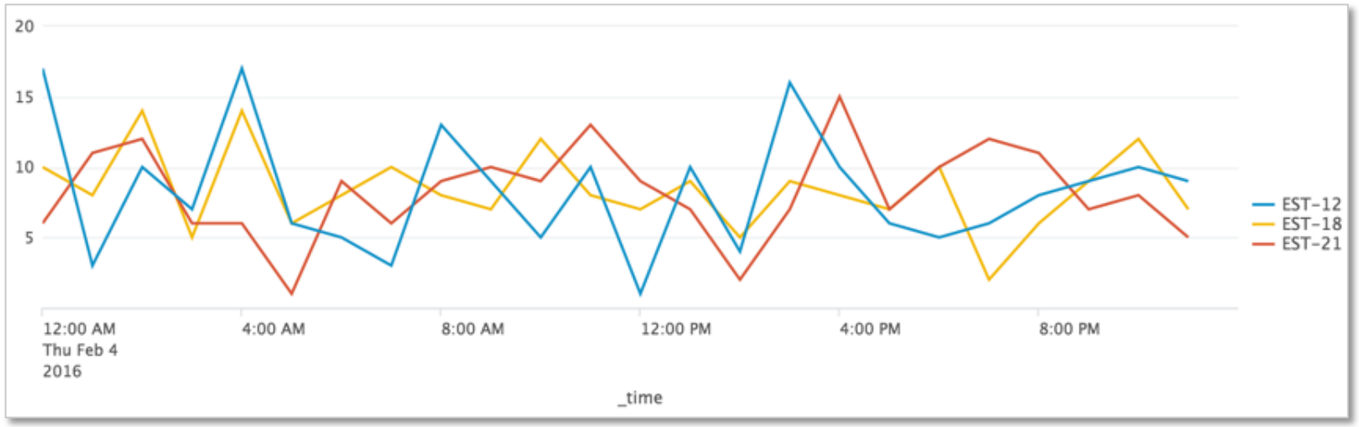
- When you use a by field, the output is a table where each column represents a distinct value of the split-by field

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

src_ip	Accepted	Failed	Invalid user	NULL
1.0.32.67	0	3	0	0
2.144.0.22	0	3	1	0
2.144.0.210	0	46	1	0

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 where each column represents a distinct value of the split-by field




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

_time	EST-12	EST-18	EST-21
2016-02-04 00:00	17	10	6
2016-02-04 01:00	3	8	11
2016-02-04 02:00	10	14	12
2016-02-04 03:00	7	5	6

Lab Exercise 3

Time: 30 minutes

Scenarios:

- Report the failures on the network during the last 60 minutes and add it to a new security dashboard as a column chart
- Chart the five best selling products in North America by country during the last 7 days
- Display Internet usage in a timechart during the last 24 hours

****Challenge Exercises:**

- Display and compare online and vendor sales during the previous week

Module 4: Transforming Commands, Part 3 Enriching Visualizations

Module Objectives

- Use the `trendline` command
- Create `maps`
  - `iplocation`
  - `geostats`
  - `geom`
- Create and format `single values`
- Use the `addtotals` command

trendline Command

- trendline computes the moving averages of a field

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

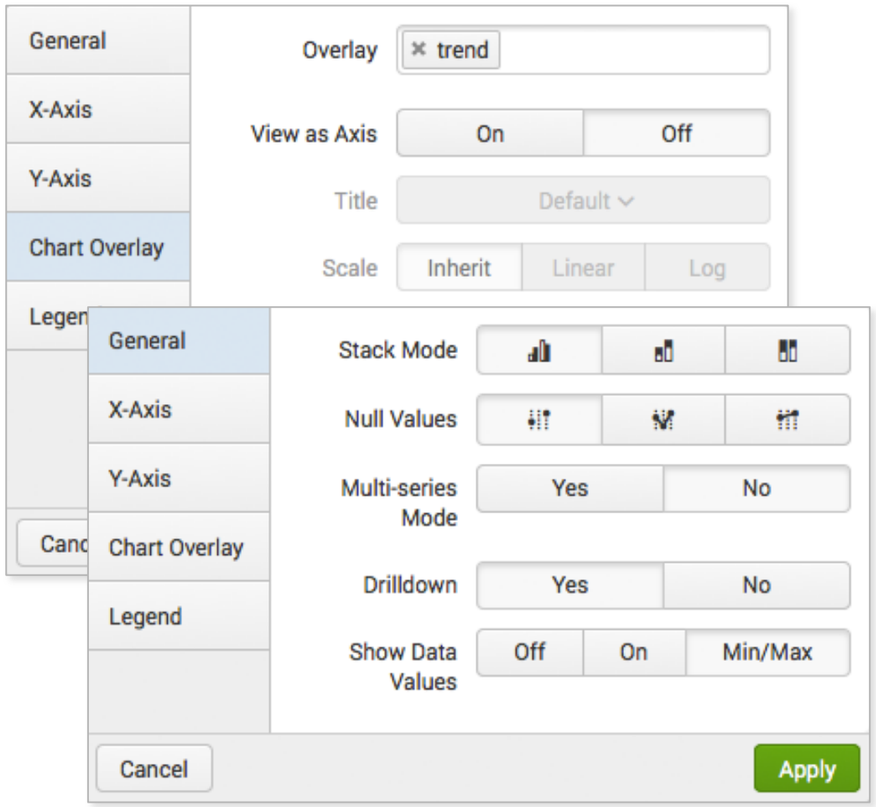
- trendtype:
  - sma - simple moving average
  - ema - exponential moving average
  - wma - weighted moving average
- Define the period over which to compute the trend; an integer between 2 and 10000, e.g., sma2
- trendtype requires the period parameter; for example, sma(sales) would fail as it is missing an integer, the defining period

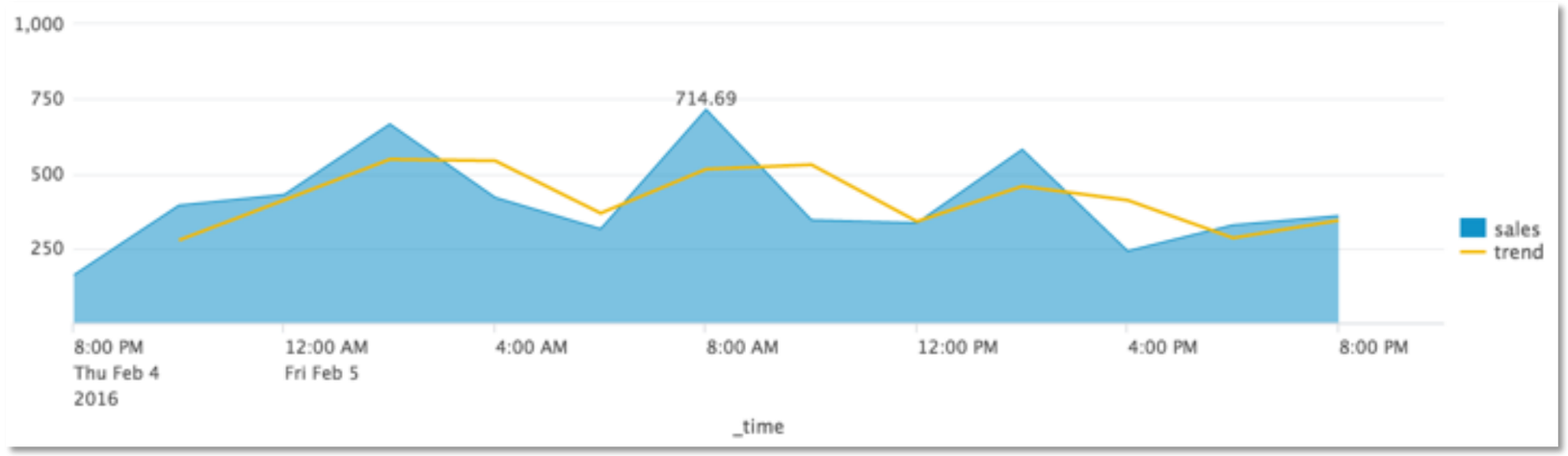
trendline Command – Example

Scenario

Display total sales and sales trend over the 24 hours.

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

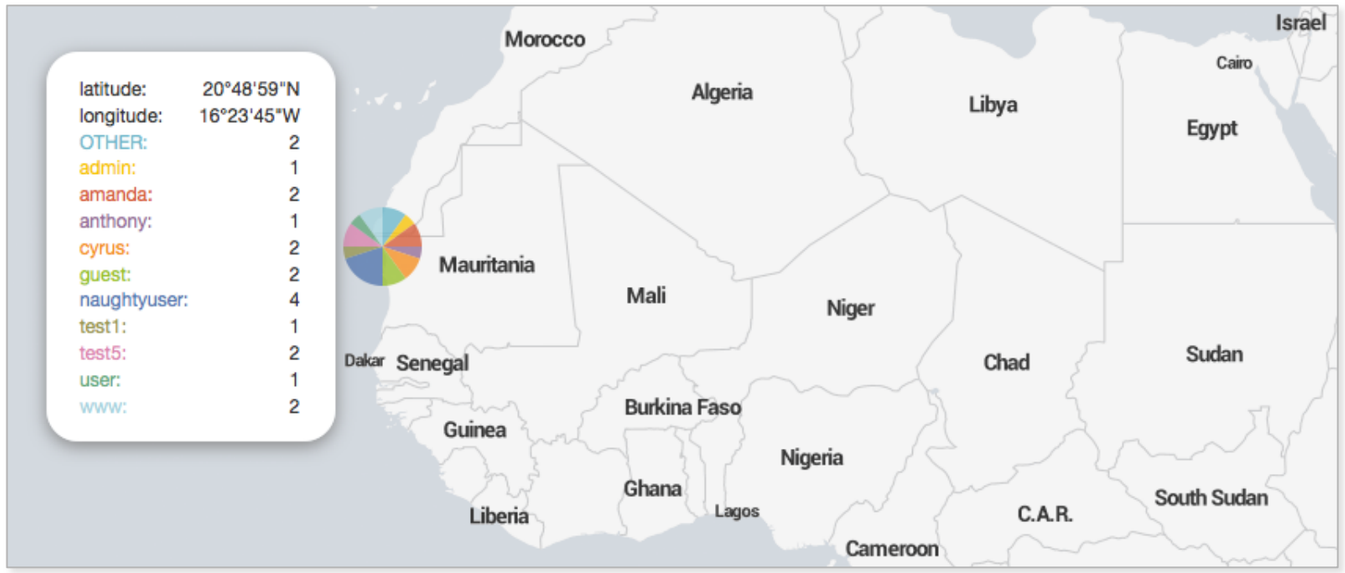




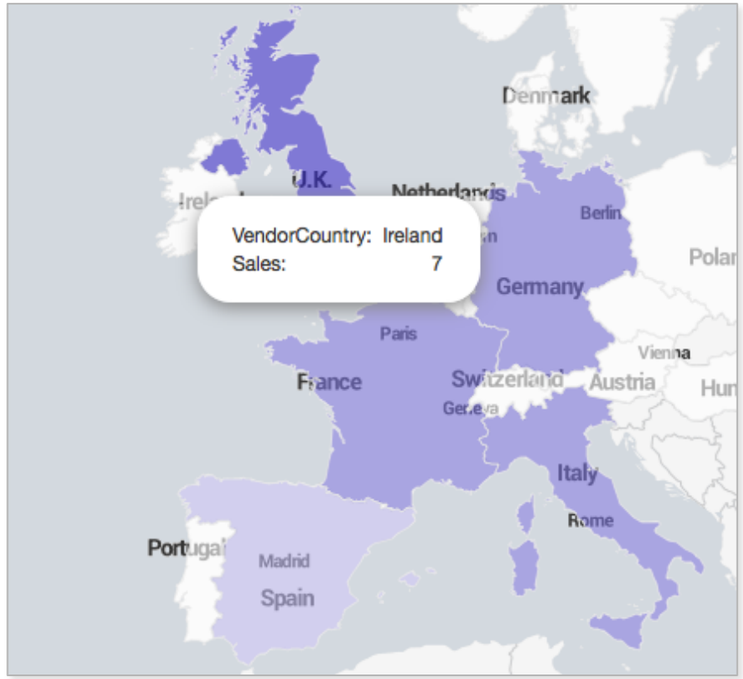
Viewing Results as a Map

There are two map types:

Cluster Map



Choropleth Map



iplocation Command

Scenario

Discover longitude and latitude data for src_ip.

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

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

geostats Command

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

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

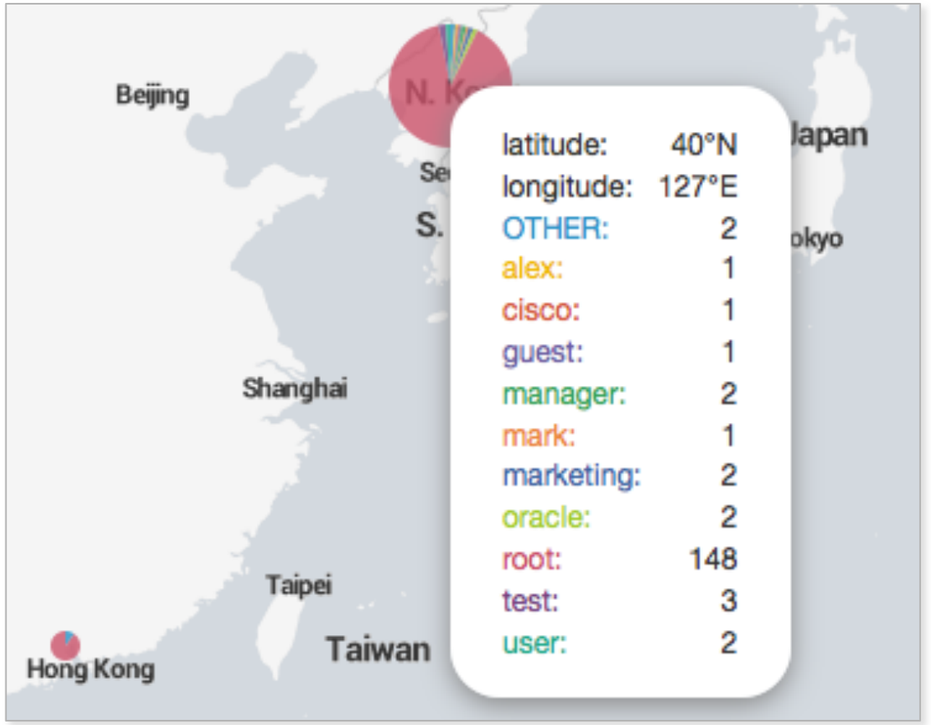
- Data must include latitude and longitude values
- Define the latfield and longfield only if they differ from the default lat and lon fields
- Use with the iplocation command to define lat and lon fields
- To control the column count, use the globallimit argument

geostats Command – Example

Scenario

Map the users of failed actions on the network worldwide during the last 24 hours.

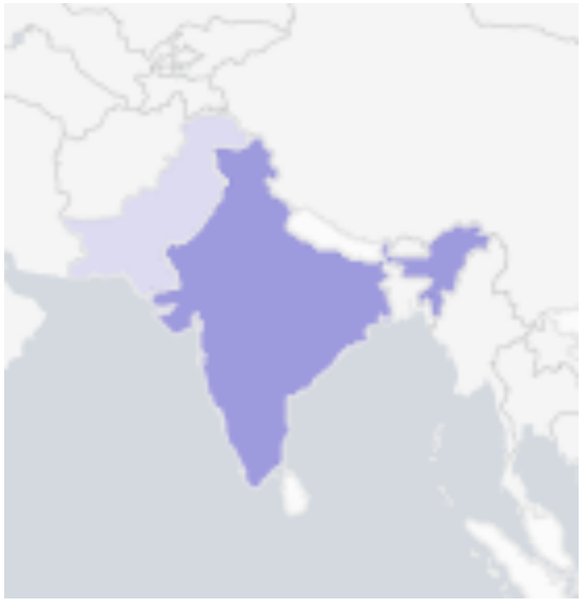
```
sourcetype=linux_secure (fail* OR invalid)
| iplocation src_ip
| geostats globallimit=12 count by user
```



geobin	latitude	longitude	OTHER	alex	cisco	guest	manager	mark	marketing	oracle	root	test	user
bin_id_zl_0_y_2_x_4	-29.00000	24.00000	2							2		1	4
bin_id_zl_0_y_5_x_2	41.14120	-73.26370	14								348		
bin_id_zl_0_y_5_x_4	30.26289	31.31977	3	2		5	2			2		2	2
bin_id_zl_0_y_5_x_5	35.69610	51.42310	3		1			1	1	5	104	1	2
bin_id_zl_0_y_5_x_6	39.29653	126.42708	2	1	1	1	2	1	2	2	157	3	2

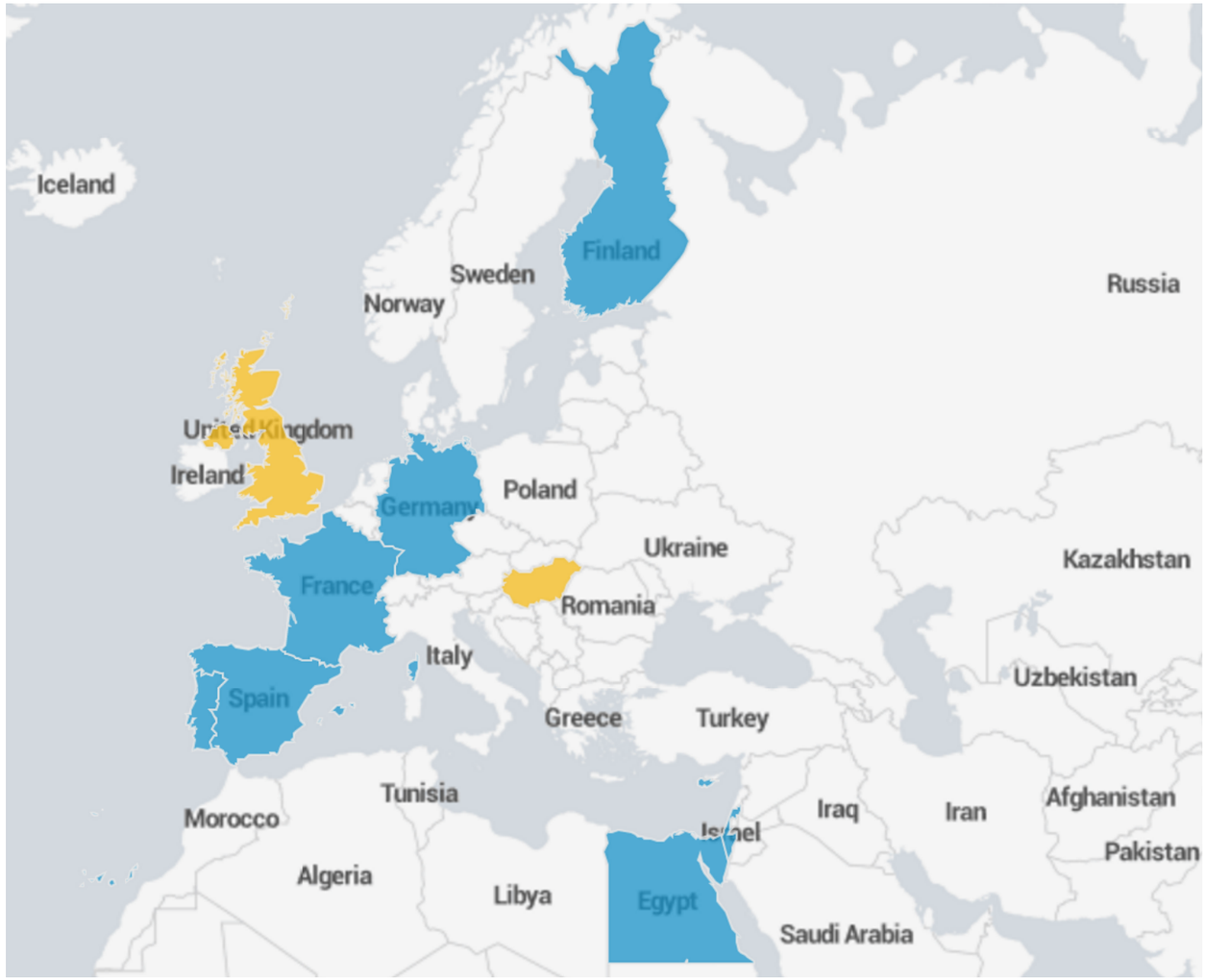
Choropleth Map

- Uses shading to show relative metrics, such as sales, network intruders, etc, for predefined geographic regions
- Must have a KMZ, or compressed Keyhole Markup Language, file that defines region boundaries
- Splunk ships with:
  - `geo_us_states`, United States
  - `geo_countries`, countries of the world



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

geom Command



Scenario

Display previous week's retail sales in EMEA.

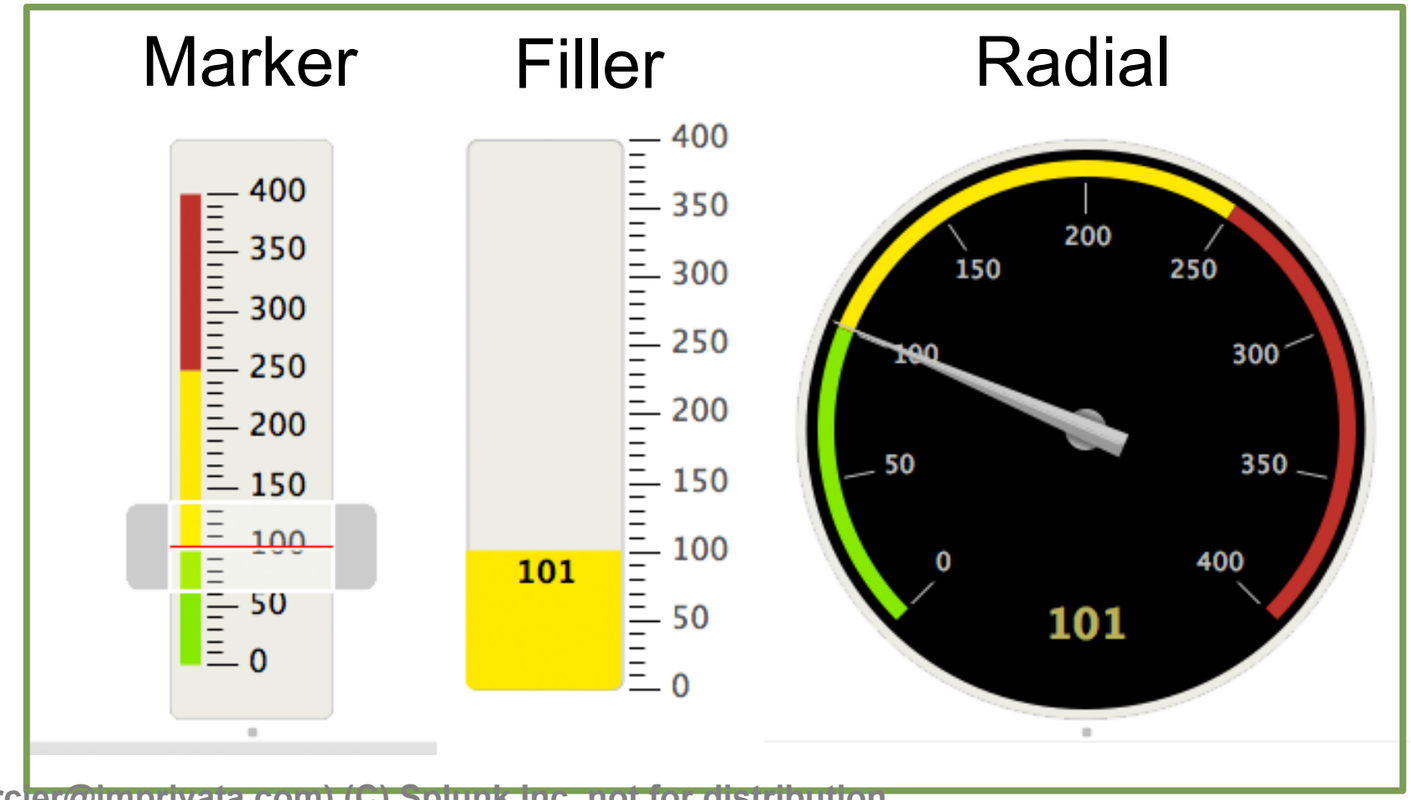
```
sourcetype=vendor_sales
VendorID > 4999 AND VendorID < 6000
| stats count as Sales by VendorCountry
| geom geo_countries featureIdField=VendorCountry
```

Viewing Results as a Single Value

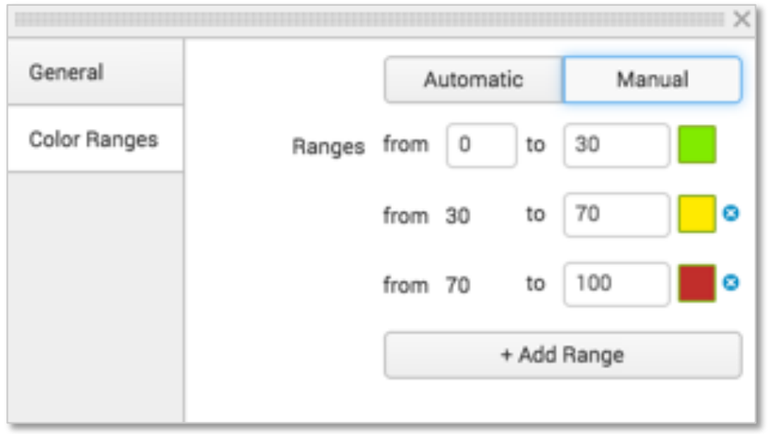
- Single value visualizations provide various formatting options

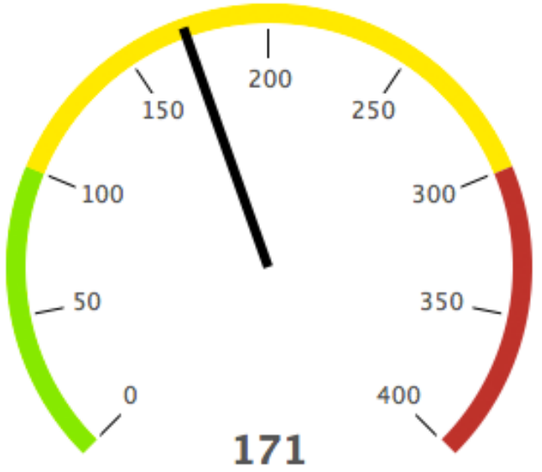
```
sourcetype=linux_secure vendor_action=failed
| stats count as count
```

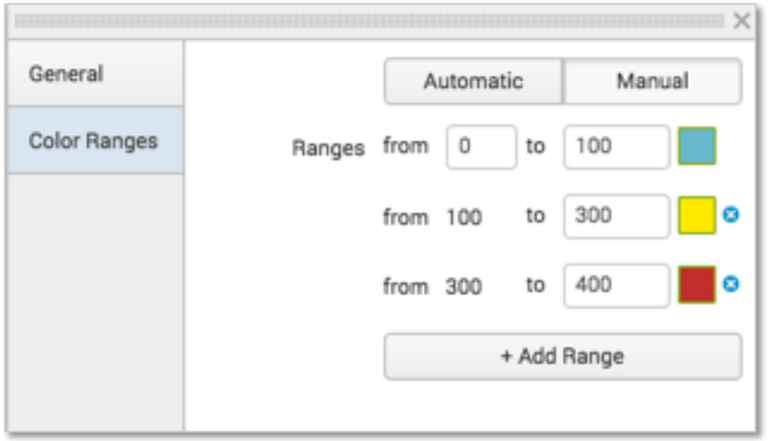
101

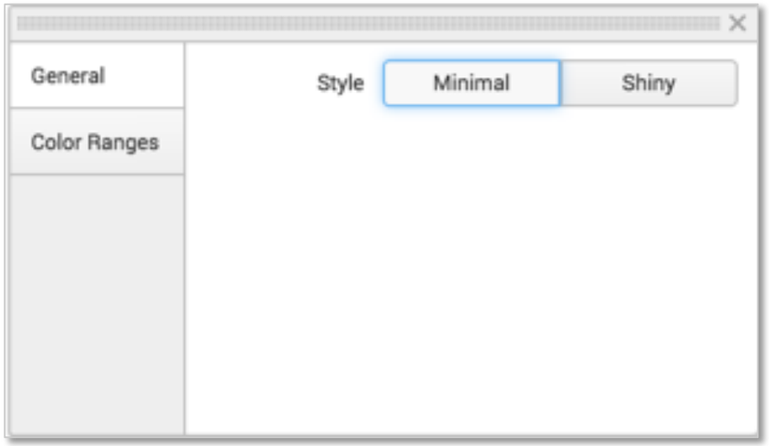


Single Value Visualizations: Format









Single Value Visualizations: Single Value

116

Failed/Invalid last 60 minutes

```
sourcetype=linux_secure (fail* OR invalid)
| stats count(vendor_action)
```

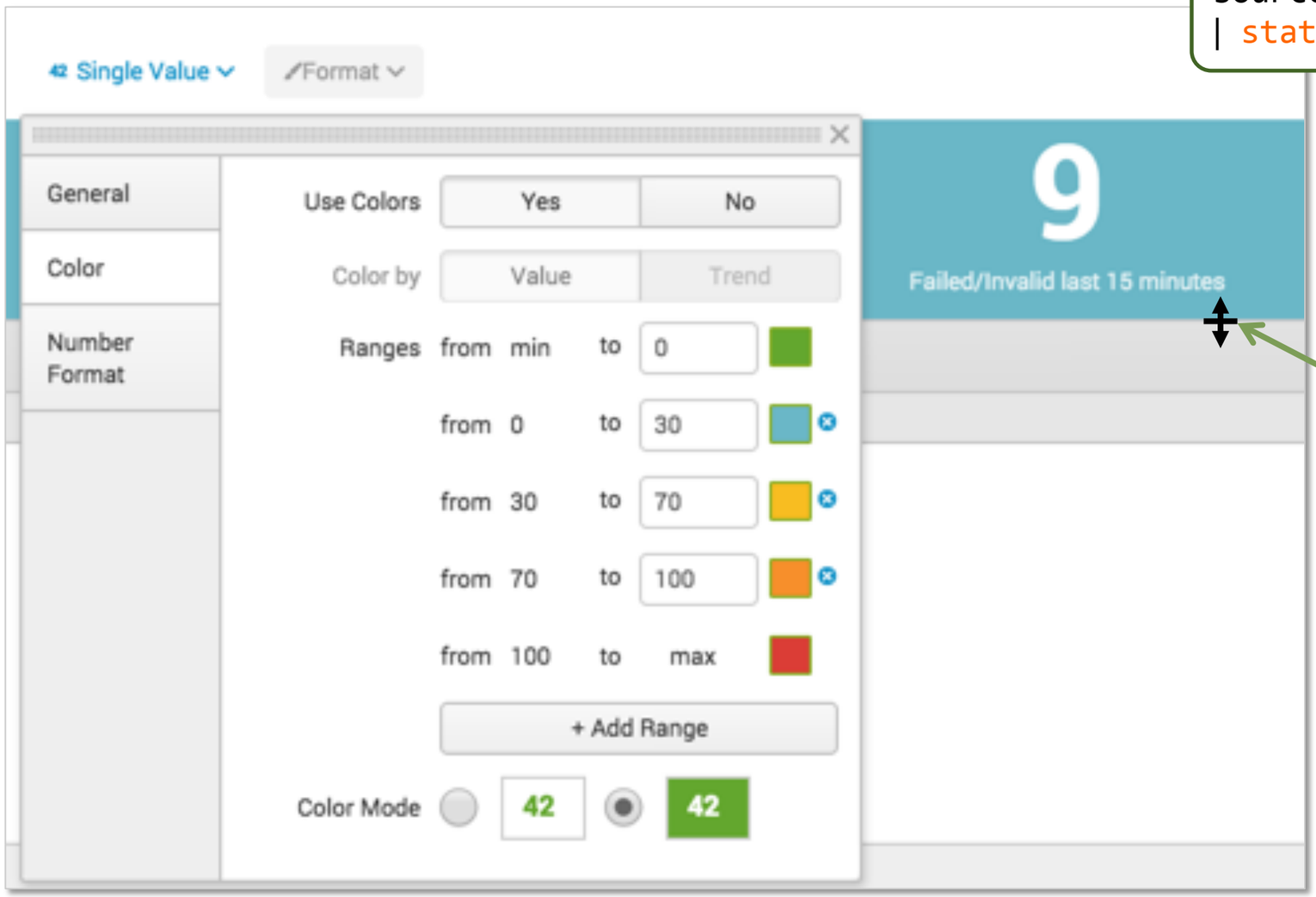
```
sourcetype=linux_secure (fail* OR invalid)
| chart count by src_ip
| sort -count
```

175.45.176.98

Most failures (IP), last 15m

Single Value Visualizations: Formatting

```
sourcetype=linux_secure (fail* OR invalid)
| stats count
```

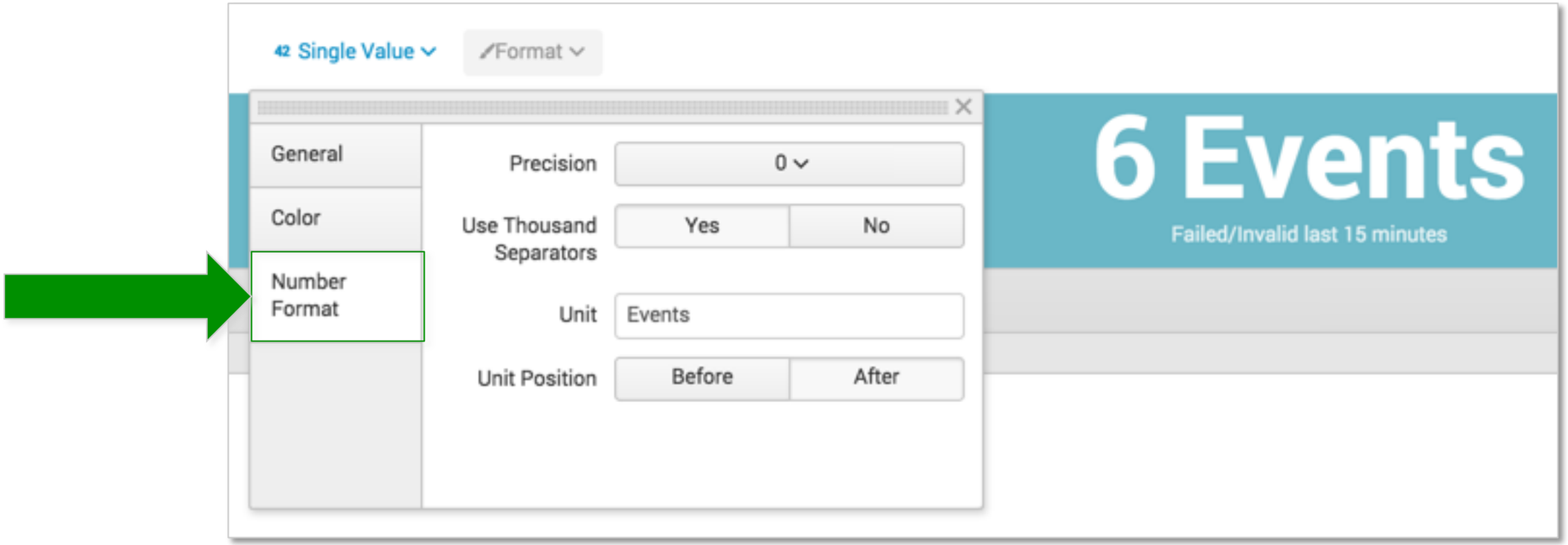


To resize the font, resize the pane

Single Value Visualizations: Formatting (cont.)

- If desired, specify number format information for the single value
- This is done by configuring settings on the **Number Format** tab

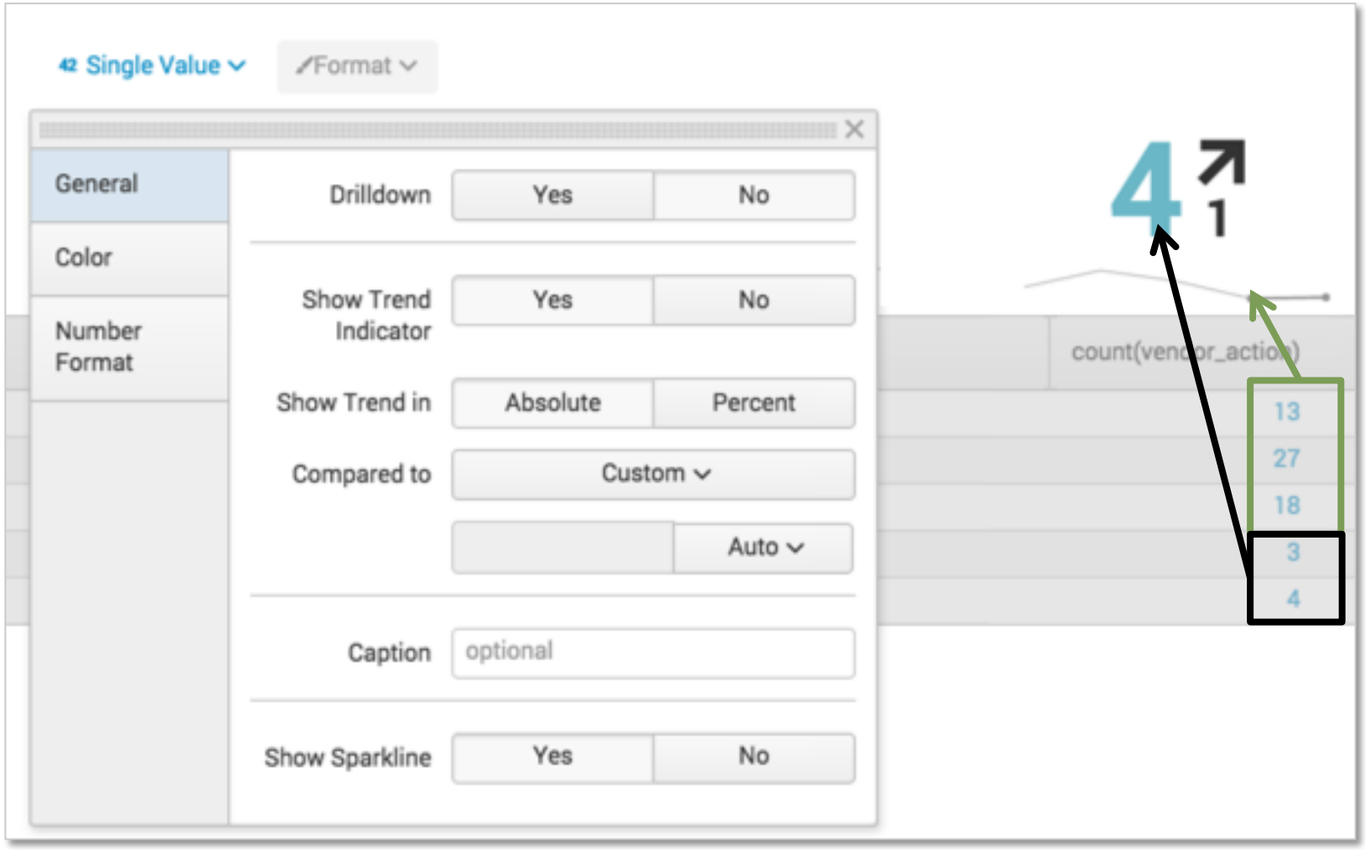
```
sourcetype=linux_secure (fail* OR invalid)
| stats count
```



Single Value Visualizations: `timechart`

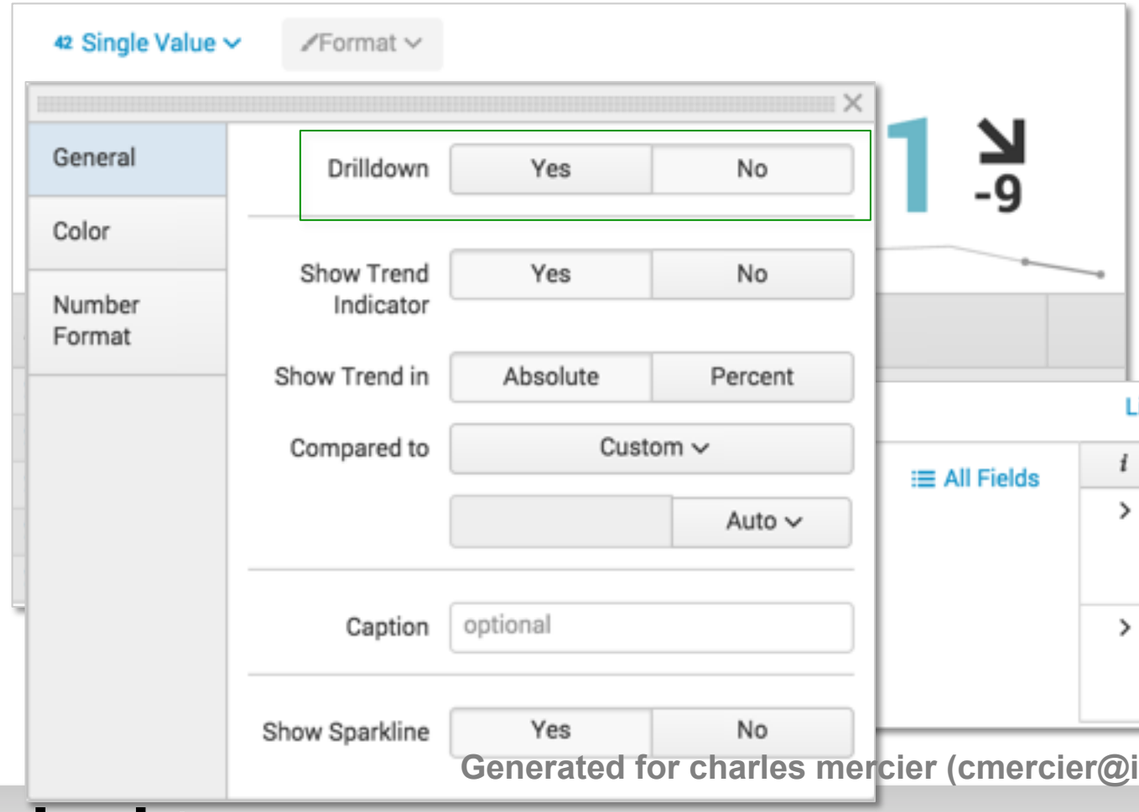
- With the `timechart` command, you can add a sparkline and a trend
- A **sparkline** is an inline chart, designed to display time-based trends associated with the primary key
- The **trend** shows the direction in which values are moving (appears to the right of the single value)

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

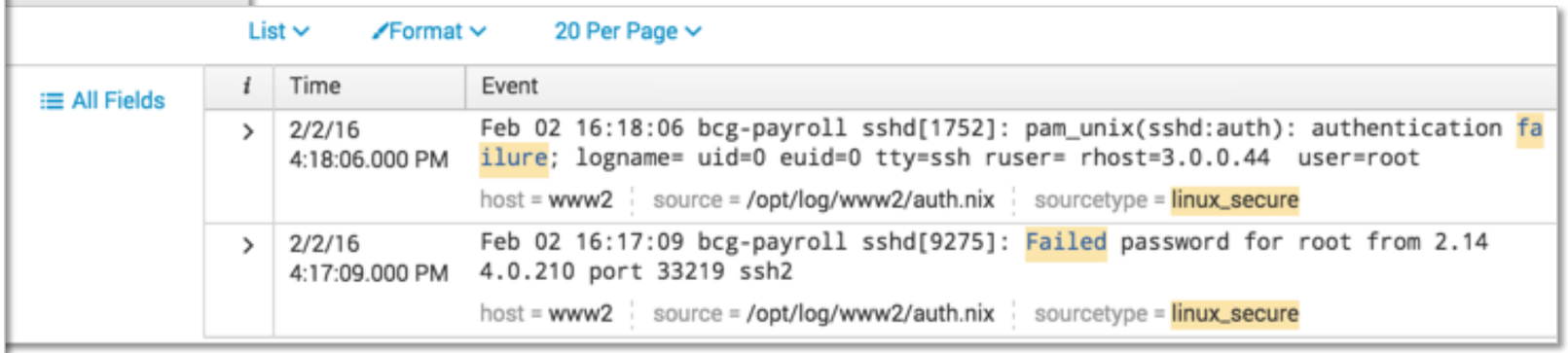


Single Value Visualizations: `timechart` (cont.)

- With the `timechart` command, you can also enable Drilldown
- When **Drilldown** is set to **Yes**, if you click on the single value, the dialog changes – to show the events used to construct the single value




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



addtotals Command: Overview

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

product_name	Indonesia	Norway	United States	Total Per Product
Curling 2014	0	0	1	1
Dream Crusher	0	0	4	4
Fire Resistance Suit of Provolone	0	0	2	2
Manganiello Bros.	0	0	1	1
Puppies vs. Zombies	0	0	3	3
SIM Cubicle	0	1	3	4
World of Cheese	1	0	2	3
Total Per Country	1	1	16	18

addtotals Command: Syntax

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

Row Options		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=*field* (Default=Total)	Defines a string used to create a field name for the totals column.	label=*string* (Default=Total)	Defines a string used to name the totals row.
		labelfield= *fieldname*	Defines where the label string is placed. (Generally, you should make this the first column.)

General Options	
field-list=one or more numeric fields. (Default: all numeric fields)	Defines the numeric fields to be totaled.

addtotals Command – Example

Scenario

Display the retail products sold by country with totals by product and by country during the last 4 hours.

- row=t (default) counts the fields in each row under a column named "Total Per Product"
- col=t counts the fields in each row in a row named "Total Per Country"

```
sourcetype=vendor_sales
| chart count over product_name by VendorCountry
| addtotals
  fieldname="Total Per Product" A
  col=t B
  label="Total Per Country"  labelfield=product_name C
```

product_name C	Indonesia	Norway	B United States	A Total Per Product
Curling 2014	0	0	1	1
Dream Crusher	0	0	4	A 4
Fire Resistance Suit of Provolone	0	0	2	2
Manganiello Bros.	0	0	1	1
Puppies vs. Zombies	0	0	3	3
SIM Cubicle	0	1	3	4
World of Cheese	1	0	2	3
Total Per Country C	1	1	B 16	18

addtotals Command – Example 2

Scenario

Display the total number of events with total and average size (in bytes) by web server, and total the Bytes

```
sourcetype=access_combined
| stats sum(bytes) as Bytes,
  avg(bytes) as avgBytes,
  count as totalEvents by host
| addtotals row=f A col=t B label=totalBytes C
  labelfield=host D Bytes E
```

- A Do not total rows
- B Total columns
- C Add the label totalBytes
- D Place the label under the host column
- E Only total the Bytes column

D host	B Bytes	avgBytes	totalEvents
www1	444653	2039.692661	218
www2	470741	2120.454955	222
www3	537626	2133.436508	252
C totalBytes	E 1453020		

Lab Exercise 4

Time: 30 minutes

Scenarios:

- Show the failures and the trend on the web server during the last 7 days
- Display a choropleth map of American retail sales during the previous week
- Display a map of online sales by country during the previous week
- Count the retail sales units sold by country and include a grand total row

Module 5: Manipulating and Filtering Results

Module Objectives

- Use the `eval` command to:
  - Perform calculations
  - Convert values
  - Round values
  - Format values
  - Use conditional statements
- Use the `search` and `where` commands to filter calculated results
- Use `fillnull` command

`eval` Command – Overview

- `eval` allows you to calculate and manipulate field values in your report
  - Useful for calculations such as add, subtract, multiply, divide
  - Does not re-write event data into the index
- Supports a variety of functions
- Results of `eval` are written to a specified field
  - Can be a new or existing field
  - If the destination field exists, the values of the field are replaced by the results of `eval`
  - 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

eval Help More »

Calculates an expression and puts the resulting value into a field.

Examples

Set velocity to distance / time.

```
... | eval velocity=distance/time
```

Set lowuser to the lowercase version of username.

```
... | eval lowuser = lower(username)
```

Set full_name to the concatenation of first_name, a space, and last_name.

```
... | eval full_name = first_name." ".last_nameSearch
```

Type	Operators
Arithmetic	+ - * / %
Concatenation	+ .
Boolean	AND OR NOT XOR
Comparison	< > <= >= != = == LIKE

eval Command – Convert Values

- This example report displays the sum of bytes used for each usage category
  - It's hard to determine how much bandwidth is being used by looking at bytes
    - First, use `eval` to convert the `bytes` value into megabytes

Scenario

What types of websites used the most bandwidth in bytes during the previous month?

```
sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as Bytes by usage
```

Events | Patterns | Statistics (5) | Visualization

10 Per Page | Format | Preview

usage	Bytes
Borderline	3032638
Business	2151552
Personal	33467017
Unknown	4825323
Violation	198318

eval Command – Convert Values (cont.)

- Results of eval must be set to a new or existing field
- In this example:
  - A Calculate the number of bytes for each usage type
  - B Create a new field named bandwidth
  - C Convert the values of the Bytes field into MB by dividing Bytes field values by (1024*1024)

Scenario

What types of websites used the most bandwidth in megabytes during the previous month?

```
sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as Bytes A by usage
| eval bandwidth B = Bytes/(1024*1024) C
```

Events | Patterns | Statistics (5) | Visualization

10 Per Page | Format | Preview

usage	A Bytes	B bandwidth
Borderline	3032638	C 2.892149
Business	2151552	2.051880
Personal	33467017	31.916635
Unknown	4825323	4.601787
Violation	198318	0.189131

eval Command – Round Values

- However, the results 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)
  - Then round to 2 decimal points
- If number of decimals is not specified, then the result is a whole number

Scenario

What types of websites used the most bandwidth in megabytes, rounded to 2 decimal places, during the previous month? Sort by bandwidth.

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

A

Events | Patterns | Statistics (5) | Visualization

10 Per Page | Format | Preview

usage	Bytes	Bandwidth (MB)
Personal	33467017	31.92
Unknown	4825323	4.60
Borderline	3032638	2.89
Business	2151552	2.05
Violation	198318	0.19

Removing Fields

Now that you've calculated and formatted the results in the new "`Bandwidth (MB)`" field, you can remove the `bytes` field from the report as a final command

- It is safe to remove fields after their values have been used in previous parts of the search string

Scenario

What types of websites used the most bandwidth in megabytes, rounded to 2 decimal places, during the previous month? Sort by bandwidth.

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

Events | Patterns | Statistics (5) | Visualization

10 Per Page | Format | Preview

usage A	Bandwidth (MB)
Personal	31.92
Unknown	4.60
Borderline	2.89
Business	2.05
Violation	0.19

eval Command – Calculating Values

Can perform mathematical functions against fields with numeric field values

- A In this example, stats calculates the total list price and total sale price by product_name
- B eval calculates the discount percentage and formats the discount field
- C sort lists the highest discounted items first
- D rename provides user friendly headings

Scenario

Calculate total online sales for last week, include price, sales price, and discount percentage. Sort by descending discount value.

```
A sourcetype=access_combined product_name=* action=purchase
A | stats sum(price) as tp, sum(sale_price) as tsp by product_name
B | eval Discount = round(((tp - tsp)/ tp)*100)
  | sort -Discount
C | eval Discount = Discount."%"
D | 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	698.60	278.60	60%
Fire Resistance Suit of Provolone	602.49	300.49	50%
Holy Blade of Gouda	808.65	403.65	50%
Dream Crusher	7078.23	4423.23	38%
Manganiello Bros.	6798.30	4248.30	38%
Orvil the Wolverine	4638.84	2898.84	38%

stats Command – eval function

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

- Requires an as clause
- Double quotes are required for character field values
- Field values **are** case-sensitive

Scenario

Count the number of events that occurred yesterday where the vendor action was Accepted, Failed, or session opened.

```
sourcetype=linux_secure vendor_action=*
| stats
  count(eval(vendor_action="Accepted")) as Accepted, A
  count(eval(vendor_action="Failed")) as Failed, B
  count(eval(vendor_action="session opened")) as
  SessionOpened C
```

Events | Patterns | Statistics (1) | Visualization

10 Per Page | Format | Preview

A Accepted	B Failed	C SessionOpened
314	6219	351

eval Command – tostring Function

- tostring converts a numeric field value to a string

  tostring(field,"option")

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

Scenario

How much potential online sales revenue was lost the previous week, due to 503 server errors?

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

Events | Patterns | Statistics (1) | Visualization

20 Per Page | Format | Preview

A NumberOfLostSales	AverageLostSales B	TotalLostRevenue C
188	$22.19	$4,172.12

tostring Function – duration Option

This example shows "duration" option of tostring function

- A stats calculates sessionTime and JSESSIONID
- B sort 5 displays the top 5 most frequent values
- C The duration option formats the time as "hh:mm:ss"

Scenario

Identify the five longest client sessions over the last 4 hours in HH:MM:SS format

```
sourcetype=access_combined
| stats range(_time) as sessionTime by JSESSIONID A
| sort 5 -sessionTime B
| eval duration = tostring(sessionTime,"duration") C
```

Events | Patterns | Statistics (5) | Visualization

10 Per Page | Format | Preview

JSESSIONID A	A sessionTime	duration
SD4SL1FF3ADFF4960	5935	01:38:55
SD0SL9FF9ADFF4952	160	00:02:40
SD6SL10FF5ADFF4965	149	00:02:29
SD9SL9FF5ADFF4954	139	00:02:19
SD1SL2FF7ADFF4964	138	00:02:18

B C

Formatting and Sorting Values

- `eval` with added characters converts numeric field values to strings
- To order numerically, first `sort`, then use `eval`

```
sourcetype=access_combined price=*
| stats values(price) as price by product_name
| eval price = "$".price
| sort -price
```

product_name	price
Manganiello Bros. Tee	$9.99
World of Cheese Tee	$9.99
Holy Blade of Gouda	$5.99
Puppies vs. Zombies	$4.99
Dream Crusher	$39.99
Manganiello Bros.	$39.99
Orvil the Wolverine	$39.99
Fire Resistance Suit of Provolone	$3.99
Benign Space Debris	$24.99
Final Sequel	$24.99

Alpha

```
sourcetype=access_combined price=*
| stats values(price) as price by product_name
| sort -price
| eval price = "$".price
```

product_name	price
Dream Crusher	$39.99
Manganiello Bros.	$39.99
Orvil the Wolverine	$39.99
Benign Space Debris	$24.99
Final Sequel	$24.99
Mediocre Kingdoms	$24.99
World of Cheese	$24.99
Curling 2014	$19.99
SIM Cubicle	$19.99
Manganiello Bros. Tee	$9.99

Numeric

Multiple eval Commands

Each subsequent command references the results of previous commands

- A Based on the values of list_price and current_sale_price, calculate the current_discount percentage
- B Calculate the new_discount value by subtracting 5 from current_discount
- C Calculate the new_sale_price by applying the new_discount percentage

Scenario

Calculate a new sale price that is 5% less than the current discount percentage.

```
sourcetype=access_combined price=*
| stats values(price) as list_price, values(sale_price)
  as current_sale_price by product_name
| eval current_discount = round((list_price -  A
  current_sale_price)/list_price*100,2)
| eval new_discount = (current_discount - 5)  B
| eval new_sale_price = list_price - (list_price *
  (new_discount/100))  C
```

product_name	list_price	current_sale_price	current_discount	new_discount	new_sale_price
Benign Space Debris	24.99	19.99	20.01	15.01	21.24
Curling 2014	19.99	16.99	15.01	10.01	17.99
Dream Crusher	39.99	24.99	37.51	32.51	26.99
Final Sequel	24.99	16.99	32.01	27.01	18.24

`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

- Create a new field, SalesTerritory
- Evaluate VendorID
  - If < 4000 is TRUE, set result to "North America"
    - Remember, arguments must be enclosed in quotes
  - If it evaluates to FALSE, set result to "Rest of the World"

Scenario

Display retail sales for the previous week, broken down by North America and the Rest of the World.

```
sourcetype=vendor_sales
| eval SalesTerritory =
  if(VendorID < 4000,"North America", "Rest of the World")
          X                Y                  Z
| stats sum(price) as TotalRevenue by SalesTerritory
| eval TotalRevenue = "$" + tostring(TotalRevenue, "commas")
```

Events | Patterns | Statistics (2) | Visualization

10 Per Page | Format | Preview

SalesTerritory	TotalRevenue
North America	$10,263.06
Rest of the World	$4,950.60

Filtering Results

The `search` and `where` commands can be used at any point in the search pipeline to filter results

- `search` command
  - May be easier because you are familiar with basic search syntax
  - Treats field values in a case-insensitive manner
  - Can use the * (asterisk) as wildcard
  - Allows searching on keyword
- `where` command
  - Can compare values from two different fields
  - Can do a wildcard search on multiple characters (%) or simply on one character (_); must use the like operator with wildcards
  - Functions are available, example `isnotnull()`
  - Field values are case-sensitive

search Command

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

Scenario

Report which products during the last 24 hours have sold more than $500 online.

```
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"
```

A

Product Name	Popular Products
Manganiello Bros.	$719.82
Dream Crusher	$679.83
Mediocre Kingdoms	$649.74

eval and where Commands

<eval-expression>

- Both commands use the same expression syntax
- Uses Booleans to filter search results and only keeps results that are True
- Quoted strings are interpreted as field values
  - Unquoted or single-quoted strings are treated as fields
  - Treats field values in a case-sensitive manner

Note

To view all of where's functions, please see:
docs.splunk.com/Documentation/Splunk/latest/SearchReference/Where

where Command - Example

- where <eval-expression>
- Compare results of calculated fields
- Keep rows where the number of removal actions exceeds the number change quantity actions

Scenario

Report which days over the previous week have seen more remove actions than change quantity actions.

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

A

_time	changes	removals
2014-10-24	121	127
2014-10-28	88	135
2014-10-29	121	130

A

`fillnull` Command

Use `fillnull` to replace null values in the specified fields

- Default is 0
- `value=string` is the string you want to display

fillnull Help More »

Replaces null values with a specified value.

Examples

For the current search results, fill all empty fields with NULL.

```
... | fillnull value=NULL
```

For the current search results, fill all empty fields with zero.

```
... | fillnull
```

Build a time series chart of web events by host and fill all empty fields with NULL.

```
sourcetype="web" | timechart count by host | fillnull value=NULL
```


fillnull Command – Examples

Scenario

Evaluate vendor sales by country for the last hour.

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

product_name	Ethiopia	United States
Final Sequel	0	24.99
Fire Resistance Suit of Provolone	0	3.99
Manganiello Bros.	0	39.99
Puppies vs. Zombies	0	0
World of Cheese	24.99	4.99

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

product_name	Ethiopia	United States
Final Sequel	No Value	24.99
Fire Resistance Suit of Provolone	No Value	3.99
Manganiello Bros.	No Value	39.99
Puppies vs. Zombies	No Value	4.99
World of Cheese	24.99	No Value

Lab Exercise 5

Time: 35 minutes

Scenarios:

- Chart the total daily volume (in MB) of the web servers during the previous week
- Calculate the ratio of GET requests to POST requests for each web server
- Identify users with more than 3 failed logins during the last 60 minutes, sort in descending order

****Challenge Exercises:**

- Classify and report employee web traffic by content type during the previous business week
- Report which days during the previous week the online store experienced more than three times the number of internal HTTP errors

Module 6: Correlating Events

Module Objectives

- Identify transactions
- Group events using fields
- Group events using fields and time
- Search with transactions
- Report on transactions
- Determine when to use `transaction` vs. `stats`

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

- <field-list>
  - One field or a list of field names
  - The events are grouped into transactions based on the values of this field list

```
... | transaction field1, field2
```

- Common constraints:

```
| <maxspan> | <maxpause> | <startswith> | <endswith>
```

transaction Help « Less

Groups events into transactions.

Syntax More »

transaction [*field-list*] name= [*string*] [(maxspan=*int* [s|m|h|d])|*maxpause-opt*|*maxevents-opt*|*field-list*|*start-opt*|*end-opt*|*connected-opt*|*unify-ends-opt*|*keeporphans-opt*]* [*memcontrol-opt*]* [*rendering-opt*]*

Examples

Group search results that have the same "host" and "cookie", occur within 30 seconds of each other, and do not have a pause greater than 5 seconds between each event into a transaction.

... | transaction host cookie maxspan=30s maxpause=5s

Events That Have the Same JSESSIONID

- Here you can see a number of events that share the same JSESSIONID value (SD6SL10FF6ADFF4961)
- However, it is difficult to view as a group or to gain insight to what is happening or know if there are others scattered in the results set

Scenario

Display customer transactions in the online store during the last 60 minutes.

```
sourcetype=access_combined
```

>	4/24/16 11:58:25.000 PM	217.132.169.69 - - [24/Apr/2016:23:58:25] "GET /cart.do?action=remove&itemId=EST-17&productId=WC-SH-G04&JSESSIONID=SD6SL10FF6ADFF4961 HTTP 1.1" 200 1661 "http://www.buttercupgames.com/oldlink?itemId=EST-17" "Googlebot/2.1 (http://www.googlebot.com/bot.html)" 613 JSESSIONID = SD6SL10FF6ADFF4961 host = www1 source = /opt/log/www1/access.log sourcetype = access_combined
>	4/24/16 11:58:11.000 PM	217.132.169.69 - - [24/Apr/2016:23:58:11] "GET /oldlink?itemId=EST-19&JSESSIONID=SD6SL10FF6ADFF4961 HTTP 1.1" 200 660 "http://www.buttercupgames.com/cart.do?action=view&itemId=EST-19&productId=SC-MG-G10" "Googlebot/2.1 (http://www.googlebot.com/bot.html)" 807 JSESSIONID = SD6SL10FF6ADFF4961 host = www1 source = /opt/log/www1/access.log sourcetype = access_combined
>	4/24/16 11:58:03.000 PM	217.132.169.69 - - [24/Apr/2016:23:58:03] "GET /product.screen?productId=WC-SH-G04&JSESSIONID=SD6SL10FF6ADFF4961 HTTP 1.1" 200 1966 "http://www.google.com" "Googlebot/2.1 (http://www.googlebot.com/bot.html)" 293 JSESSIONID = SD6SL10FF6ADFF4961 host = www1 source = /opt/log/www1/access.log sourcetype = access_combined

transaction Command – Example 1

- Use the transaction command to create a single event from a group of events that share the same value in a given field
- Transactions can cross multiple tiers (i.e., web server, application server) using a common field(s); such as JSESSIONID
- For example, you can easily view the events for JSESSIONID SD0SL6FF9ADFF4964

Note

Group together Buttercup games online store events, based on the JSESSIONID value.

```
sourcetype=access_combined
| transaction JSESSIONID
```

Time	Event
1/28/16 10:53:41.000 PM	87.194.216.51 - - [28/Jan/2016:22:53:41] "POST /oldlink?itemId=EST-19&JSESSIONID=SD0SL6FF5ADFF4964 HTTP 1.1" 200 1937 "http://www.yahoo.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 946 87.194.216.51 - - [28/Jan/2016:22:53:54] "GET /oldlink?itemId=EST-12&JSESSIONID=SD0SL6FF5ADFF4964 HTTP 1.1" 200 2253 "http://www.buttercupgames.com/oldlink?itemId=EST-12" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 439 87.194.216.51 - - [28/Jan/2016:22:54:08] "GET /product.screen?productId=SF-BVS-01&JSESSIONID=SD0SL6FF5ADFF4964 HTTP 1.1" 503 1162 "http://www.buttercupgames.com/cart.do?action=remove&itemId=EST-18" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 198 host = www3 source = /opt/log/www3/access.log sourcetype = access_combined

transaction Command – Example 2

- Use the search command at any point in the search pipeline to filter results
- Behaves exactly like search strings before the first pipe
  - search uses the "*" wildcard and treats field values in a case-insensitive manner
  - status=404 finds the errors
  - highlight highlights the terms you specify

Scenario

Display transactions that included a 404 error during the last 60 minutes.

```
sourcetype=access_combined
| transaction JSESSIONID A
| search status=404
| highlight JSESSIONID, 404 B
```

transaction Command – Example 3

Scenario

For failed network logins, display different users from the same IP during the last 60 minutes.

```
sourcetype=linux_secure failed
| transaction src_ip
```

i	Time	Event
>	2/9/16 7:37:58.000 PM	Feb 09 19:37:58 bcg-payroll sshd[17403]: Failed password for invalid user vpopmail from 175.45.176.98 port 52722 ssh2 Feb 09 19:46:57 bcg-payroll sshd[14883]: Failed password for invalid user guest from 175.45.176.98 port 34412 ssh2 host = www1 source = /opt/log/www1/auth.nix sourcetype = linux_secure
>	2/9/16 7:16:24.000 PM	Feb 09 19:16:24 bcg-fileserver sshd[9974]: Failed password for invalid user brooke from 41.32.0.85 port 58580 ssh2 Feb 09 19:18:11 bcg-fileserver sshd[10033]: Failed password for invalid user bruno from 41.32.0.85 port 53132 ssh2 Feb 09 19:22:36 bcg-fileserver sshd[10049]: Failed password for invalid user ftp from 41.32.0.85 port 33206 ssh2 Feb 09 19:33:28 bcg-fileserver sshd[10053]: Failed password for invalid user angel from 41.32.0.85 port 33572 ssh2 Feb 09 19:42:09 bcg-fileserver sshd[7653]: Failed password for invalid user addicted from 41.32.0.85 port 35502 ssh2 host = www1 host = www2 host = www3 source = /opt/log/www1/auth.nix source = /opt/log/www2/auth.nix source = /opt/log/www3/auth.nix sourcetype = linux_secure

`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)

Scenario

Display customer actions on the website during the last 4 hours.

```
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
121.9.245.177	00:02:57	addtocart purchase
173.44.37.226	00:02:37	addtocart purchase view
209.160.24.63	00:02:13	addtocart changequantity purchase view

Note

Assumptions: Transactions spanning more than 10 minutes with the same client IP are considered unrelated. Also there can be more than one 1 minute between any two related events.

transaction Command – startswith/endswith

- To form transactions based on terms, field values, or evaluations, use `startswith` and `endswith` options
- In this example, the first event in the transaction includes `addtocart` and the last event includes `purchase`

Scenario

Determine the length of time spent by customers in the online store to purchase.

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

clientip	JSESSIONID	duration	eventcount
74.82.57.172	SD3SL1FF2ADFF4954	4	2
74.82.57.172	SD3SL1FF2ADFF4954	49	5
192.162.19.179	SD10SL4FF2ADFF4953	5	2
192.162.19.179	SD10SL4FF2ADFF4953	26	3

Investigating with Transactions

- Transactions can be useful when a single event does not provide enough information
- This example searches email logs for the term “REJECT”
- Events that include the term don’t provide much information about the rejection

Scenario

Find emails that were rejected during the last 24 hours.

```
sourcetype=cisco_esa REJECT
```

i	Time	Event
>	1/28/16 11:19:02.000 PM	Thu Jan 28 23:19:02 2016 Info: ICID 744005 REJECT SG BLACKLIST match sbrs[10.0: 3.0] SBRS 10.0 host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	1/28/16 10:53:41.000 PM	Thu Jan 28 22:53:41 2016 Info: ICID 744003 REJECT SG BLACKLIST match sbrs[10.0: 3.0] SBRS 10.0 host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	1/28/16 9:53:42.000 PM	Thu Jan 28 21:53:42 2016 Info: ICID 744001 REJECT SG BLACKLIST match sbrs[10.0: 3.0] SBRS 6.8 host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	1/28/16 9:33:28.000 PM	Thu Jan 28 21:33:28 2016 Info: ICID 743999 REJECT SG BLACKLIST match sbrs[10.0: 3.0] SBRS 10.0 host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa

Investigating with Transactions (cont.)

- By creating a transaction, we can then search and see additional events related to the rejection, such as:
  - IP address of sender
  - Reverse DNS lookup results
  - Action taken by the mail system following the rejection
- `mid` – Message ID
- `dcid` – Delivery Connection ID
- `icid` – Incoming Connection ID

Scenario

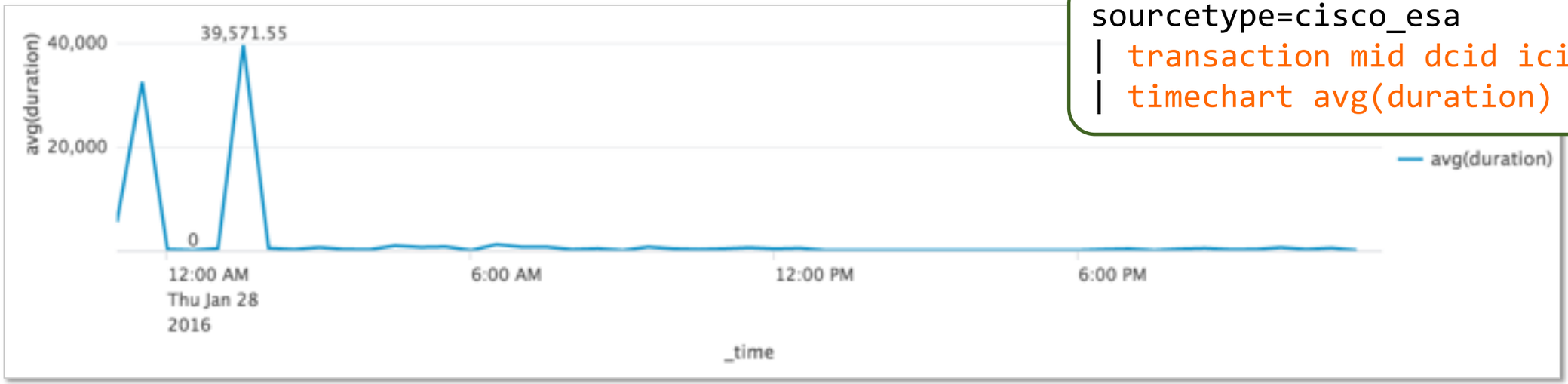
Find emails that were rejected.

```
sourcetype=cisco_esa
| transaction mid dcid icid
| search REJECT
```

i	Time	Event
>	1/27/16 11:24:37.000 PM	Wed Jan 27 23:35:55 2016 Info: New SMTP ICID 743914 interface Management (192.168.3.120) address 85.152.69.78 reverse dns host cm 85 152 69 78.telecable.es verified yes Wed Jan 27 23:36:00 2016 Info: ICID 743914 REJECT SG BLACKLIST match sbrs[10.0:3.0] SBRS 10.0 Wed Jan 27 23:36:08 2016 Info: ICID 743914 close host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	1/27/16 11:24:37.000 PM	Thu Jan 28 00:36:23 2016 Info: New SMTP ICID 743917 interface Management (192.168.3.120) address 216.102.155.100 reverse dns host adsl 216 102 155 100.dsl.lsan03.pacbell.net verified yes Thu Jan 28 00:36:37 2016 Info: ICID 743917 REJECT SG BLACKLIST match sbrs[10.0:3.0] SBRS 10.0 Thu Jan 28 00:36:49 2016 Info: ICID 743917 close host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa

Reporting on Transactions

- You can use statistics and reporting commands with transactions
- This example takes advantage of the `duration` field
  - It shows a trend of the mail queue slowing, then correcting, then slowing again
  - Adding events to the transaction from additional hosts or sources can uncover the cause of the slowdown




```
sourcetype=cisco_esa
| transaction mid dcid icid
| timechart avg(duration)
```

transaction vs. stats

- Use `transaction` when you:
  - Need to see events correlated together
  - Must define event grouping based on `start/end` values or chunk on time
  - Have less than 1,000 events for each correlated transaction
    - By default, `transaction` displays a maximum event count of 1,000
    - Admins can configure `max_events_per_bucket` in `limits.conf`
- 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`")
  - Have more than 1,000 events for each grouped set of events
- When you have a choice, always use `stats` as it is faster and more efficient, especially in large Splunk environments

transaction vs. stats: Example

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

Note

1. `transaction` has a limit of 1,000
2. Count of transactions vs count of IPs

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

src_ip	eventcount
3.0.0.44	1000
3.0.0.44	1000
3.0.0.44	1000
175.45.176.223	1000
2.144.0.210	784
3.0.0.44	608
41.32.0.85	473
23.16.0.181	316
175.45.176.98	233
175.45.176.223	121

src_ip	eventcount
3.0.0.44	3608
175.45.176.223	1121
2.144.0.210	784
41.32.0.85 .23	473
23.16.0.181	316
175.45.176.98	233
41.0.0.142	57
2.144.0.22	44
1.0.32.67	30
41.32.0.27	29

Lab Exercise 6

- **Time:**
  - 25 minutes
- **Tasks:**
  - Analyze transactions in the online store during the last 60 minutes
  - Display the online store purchase transactions lasting more than one minute and include the number of events in each transaction
  - Search for online store transactions that begin with an addtocart action and end with a purchase action

****CHALLENGE Exercise:**

  - Report common HTTP status errors that occurred during the previous week on the online sales web servers and the internal web appliance within a proximity of 30 seconds or less
    - Omit days with no common errors

Note

You may need to expand time on sourcetype=cisco_esa to the **Last 24 hours**.

What's Next?

		You are here						
Power User Certification	Using Splunk	Searching And Reporting	Creating Splunk Knowledge Objects	Infrastructure Overview (e-learning)	Certified Power User Online Test	Advanced Searching and Reporting	Analytics and Data Science	
Administrator Certification	Using Splunk	Searching And Reporting	Creating Splunk Knowledge Objects	Infrastructure Overview (e-learning)	Certified Power User Online Test	Splunk Administration	Certified Administrator Online Test	Splunk Cluster Administration
Architect Certification	Using Splunk	Searching And Reporting	Creating Splunk Knowledge Objects	Splunk Administration	Advanced Dashboards and Visualizations	Architecting and Deploying Splunk	Architect Certification Lab	
Splunk for App Developers	Using Splunk	Searching And Reporting	Creating Splunk Knowledge Objects	Advanced Searching and Reporting	Advanced Dashboards and Visualizations	Building Splunk Apps	Developing with Splunk Java and Python SDKs	

Required
Required E-learning
Exam
Recommended

For detailed course and certification information go to: http://splk.it/g8q
If you have further questions, send an email to: certification@splunk.com

What's Next? *Advanced Searching and Reporting*

- Gain experience searching in your own environment
- To learn to create more intricate searches, attend the *Advanced Searching and Reporting* class :
  - Use additional commands and functions
  - Include earliest and latest time fields, as well as other time modifiers
  - Incorporate a subsearch
  - Append results from a search to the results of another search
  - And more

Preview of AS&R Course – appendcols Command

Scenario

Calculate the daily online and retail sales for the previous 7 days. Include the total sales and the ratio of online sales vs. in-store sales.

```
sourcetype=vendor_sales
| timechart span=1d B sum(price) as "Store Sales" A
| appendcols [search sourcetype=access_combined action=purchase
| timechart span=1d sum(price) as "Online Sales"] C
| eval Day = strftime(_time, "%a, %b %e")
| eval "Total Sales"=('Store Sales' + 'Online Sales') D
| eval "Store Sales"="$"+tostring(round('Store Sales',2),"commas")
| eval "Online Sales"="$"+tostring(round('Online Sales',2),"commas")
| eval "Total Sales"="$"+tostring(round('Total Sales',2),"commas")
| eval "Online %"=tostring(round(('Online Sales'/'Total Sales')*100,2)) + "%" E
| table Day, "Online Sales", "Store Sales", "Total Sales", "Online %"
```

Day B	Online Sales C	Store Sales A	Total Sales D	Online % E
Thu, Oct 23	$1,160.39	$543.73	$1,704.12	68.09%
Fri, Oct 24	$6,199.07	$2,192.97	$8,392.04	73.87%
Sat, Oct 25	$5,496.44	$2,357.85	$7,854.29	69.98%
Sun, Oct 26	$5,310.32	$2,271.95	$7,582.27	70.04%
Mon, Oct 27	$5,711.48	$2,437.86	$8,149.34	70.09%
Tue, Oct 28	$5,356.47	$2,013.99	$7,370.46	72.67%
Wed, Oct 29	$4,730.70	$2,400.89	$7,131.59	66.33%
Thu, Oct 30	$4,163.06	$1,507.12	$5,670.18	73.42%

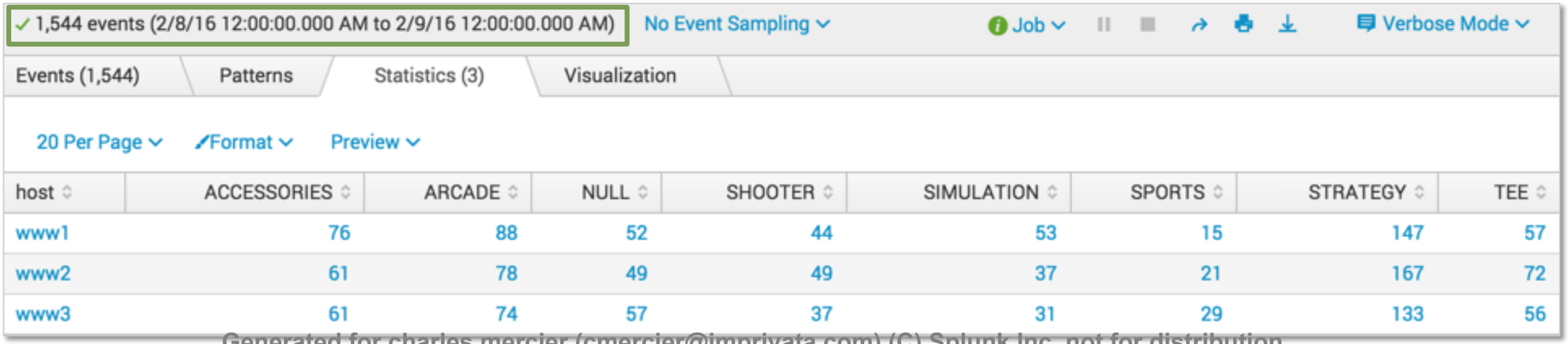
Preview of AS&R Course – return Command

Scenario

Display activity by sales category for the previous **business** day.

```
sourcetype=access_combined categoryId=*
[| stats count
 | eval day_of_week = strftime(now(), "%w")
A eval earliest = case(day_of_week == 0, "-2d@d",
  day_of_week == 1, "-3d@d", 1 == 1, "-1d@d")
B eval latest = case(day_of_week == 0, "-1d@d",
  day_of_week == 1, "-2d@d", 1 == 1, "@d")
 | return earliest, latest]
| chart count by host, categoryId
```

A B



host	ACCESSORIES	ARCADE	NULL	SHOOTER	SIMULATION	SPORTS	STRATEGY	TEE
www1	76	88	52	44	53	15	147	57
www2	61	78	49	49	37	21	167	72
www3	61	74	57	37	31	29	133	56

Support Programs

- **Community**
  - **Splunk Answers:** answers.splunk.com
    Post specific questions and get them answered by Splunk community experts.
  - **Splunk Docs:** docs.splunk.com
    These are constantly updated. Be sure to select the version of Splunk you are using.
  - **Wiki:** wiki.splunk.com
    A community space where you can share what you know with other Splunk users.
  - **IRC Channel:** #splunk on the EFNet IRC server Many well-informed Splunk users “hang out” here.
- **Global Support**
  Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365.
  - **Phone: (855) SPLUNK-S or (855) 775-8657**
  - **Web:** http://www.splunk.com/index.php/submit_issue
- **Enterprise Support**
  Access your customer support team by phone and manage your cases online 24 x 7 (depending on support contract.)

.conf2016: The 7th Annual Splunk Worldwide Users' Conference

- September 26-29, 2016
- The Disney Swan and Dolphin, Orlando
- 4400+ IT & Business Professionals

- 3 days of technical content
  - 175+ sessions

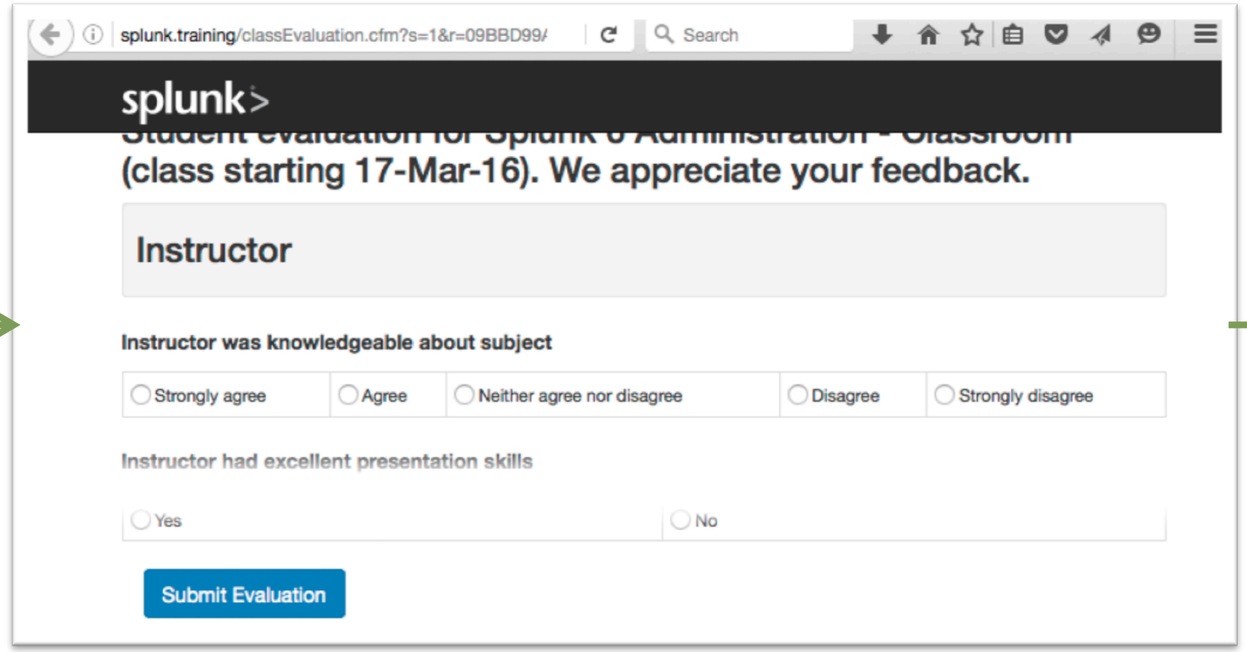
- 3 days of Splunk University
  - Sept 24-26, 2016
  - Get Splunk Certified!
  - Get CPE credits for CISSP, CAP, SSCP

- 80+ Customer Speakers
- 40+ Apps in Splunk Apps Showcase
- 70+ Technology Partners
- 1:1 networking: Ask The Experts and Security Experts, Birds of a Feather and Chalk Talks
- NEW hands-on labs!
- Expanded show floor, Dashboards Control Room & Clinic, and MORE!

Visit conf.splunk.com for more information

Thank You

- Complete the Class Evaluation to be in this month's drawing for a $100 Splunk Store voucher
  1. Look for the invitation email, *What did you think of your Splunk Education class*, in your inbox
  2. Click the link or go to the specified URL in the email



Thank You

Appendix A: erex, rex, multikv Commands

Fields Extraction Commands

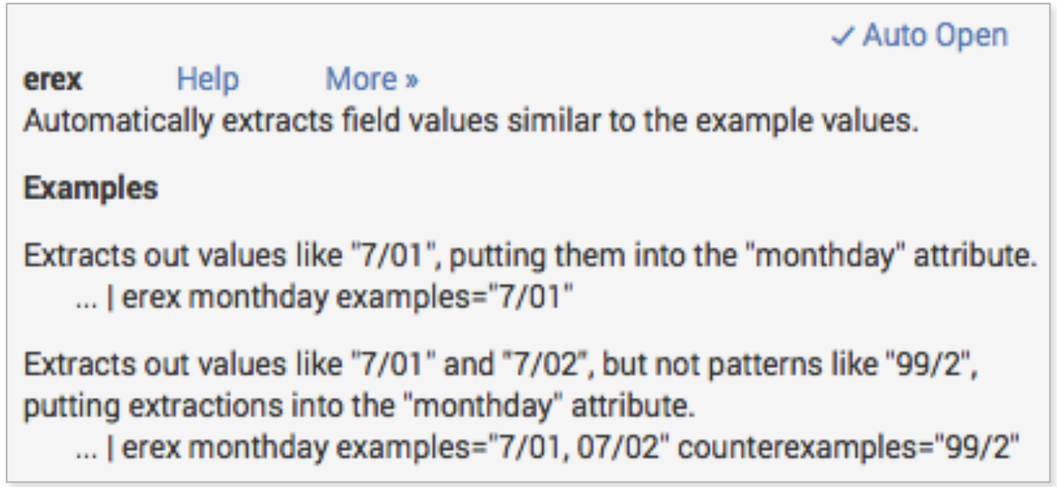
- `erex` command
  - You do not know the regular expression to use
  - You have example values in your retrieved events
- `rex` command
  - NO UI; you must write regex
  - Only persists for the duration of the search
  - Does not persists as a knowledge object
  - Good for rarely used fields

Note

You can define automatic field extractions using methods described in *Creating Splunk Knowledge Objects* class.

erex Command

- Instead of using `regex`, the `erex` command allows you to extract a field at search time by providing examples
- `examples=<erex-examples>` comma-separated list of example values for the information to be extracted and saved into a new field



✓ Auto Open

erex Help More »

Automatically extracts field values similar to the example values.

Examples

Extracts out values like "7/01", putting them into the "monthday" attribute.

... | erex monthday examples="7/01"

Extracts out values like "7/01" and "7/02", but not patterns like "99/2", putting extractions into the "monthday" attribute.

... | erex monthday examples="7/01, 07/02" counterexamples="99/2"

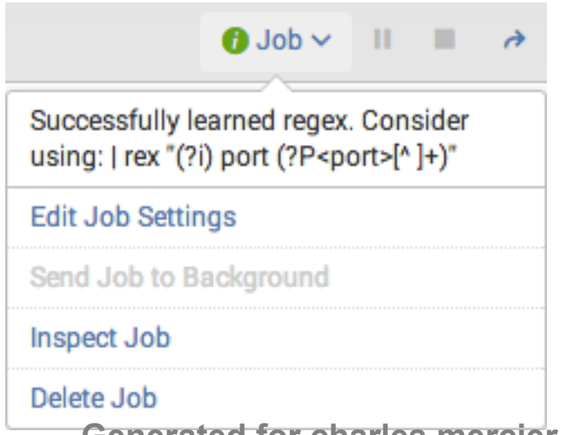
Note

The examples used must be in the returned results. For example, first run `sourcetype=linux_secure port` and use some of the resulting ports as examples. Then run the `erex` command.

erex Command – Example

- A Creates a new field called, port
- B Extracts values using the examples, which exist in the data
  - 4987, 4549

- To view the regex generated by your search, click the Job dropdown



Scenario

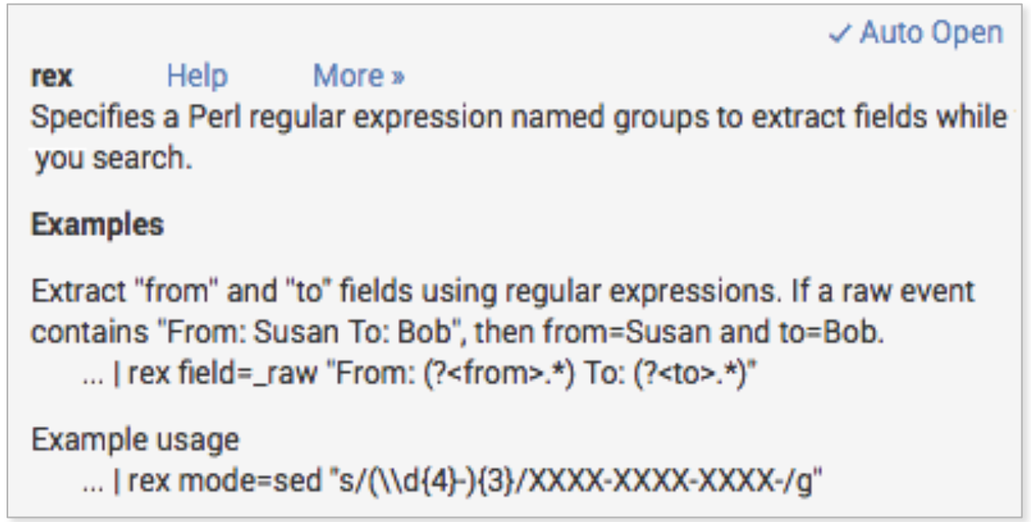
Display IP address and port of potential attackers.

```
sourcetype=linux_secure port "failed password"
| erex port A examples="4945,3136" B
| table src_ip, port
```

src	A port
66.69.195.226	1605
66.69.195.226	2143
10.3.10.46	B 4945
10.3.10.46	B 3136
10.3.10.46	4707
10.3.10.46	3883
10.3.10.46	3475
10.3.10.46	B 4945

rex Command

- The rex command allows you to extract fields at search time
- Matches the value of the field against unanchored regex
  - Defaults to field=_raw
- regex specifies both the match and a named capture that creates the new field



rex Command – Example 1

Scenario

Display the user name of potential email attackers.

```
sourcetype=cisco_esa mailfrom=*
| rex  "\<(?<potentialAttacker>.*)@"
| table potentialAttacker
```

- The Cisco router server contains the email addresses of those sending email to the company
- `mailfrom` contains the entire address
- Use `rex` to extract just the user name at search time

potentialAttacker
OSGIMR5CJLB22OKK5SVWVNDD7NHO746JAQ.1.7
saver
saver

i	Time	Event
>	2/7/16 11:08:11.000 PM	Sun Feb 07 23:08:11 2016 Info: MID 245461 ICID 744500 From: <OSGIMR5CJLB22OKK5SVWVNDD7NHO746JAQ.1.7@b.mypoints.com> host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	2/7/16 10:00:37.000 PM	Sun Feb 07 22:00:37 2016 Info: MID 245459 ICID 0 From: <saver@ingdirect.com> host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	2/7/16 9:59:10.000 PM	Sun Feb 07 21:59:10 2016 Info: MID 245458 ICID 744497 From: <saver@ingdirect.com> host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa

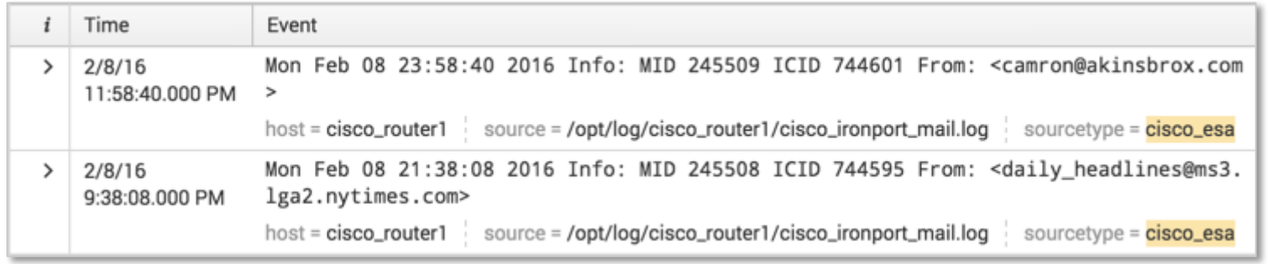
rex Command – Example 2

Scenario

Display the user name and mail domains from which our employees are receiving email.

```
sourcetype=cisco_esa mailfrom=*
| rex  "\<(?<potentialAttacker>.*)@(?<domain>.*)>"
| table mailfrom, potentialAttacker, domain
```

- Email domain is not extracted as a separate field, but as part of `mailfrom`
- Use `rex` to extract it at search time
- Can perform multiple extractions



i	Time	Event
>	2/8/16 11:58:40.000 PM	Mon Feb 08 23:58:40 2016 Info: MID 245509 ICID 744601 From: <camron@akinsbrox.com > host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa
>	2/8/16 9:38:08.000 PM	Mon Feb 08 21:38:08 2016 Info: MID 245508 ICID 744595 From: <daily_headlines@ms3.lga2.nytimes.com> host = cisco_router1 source = /opt/log/cisco_router1/cisco_ironport_mail.log sourcetype = cisco_esa

mailfrom	potentialAttacker	domain
camron@akinsbrox.com		
daily_headlines@ms3.lga2.nytimes.com		

Note

To limit the scope of the `rex` command, use the `field=mailfrom` argument. This can significantly reduce the complexity of your regex code.

erex vs. rex – Example

```
sourcetype=linux_secure port
 "failed password"
| erex port examples="3890,3703"
| table src_ip, port
```

Scenario

Display IP address and port of potential attackers.

src	port
10.2.10.163	1668
10.2.10.163	2980
10.2.10.163	1281
10.2.10.163	1720
10.2.10.163	2175
10.2.10.163	1460
10.1.10.172	3560
10.1.10.172	3560

```
sourcetype=linux_secure port
"failed password"
| rex "port\s(?<port>\d+)"
| table src_ip, port
```

erex

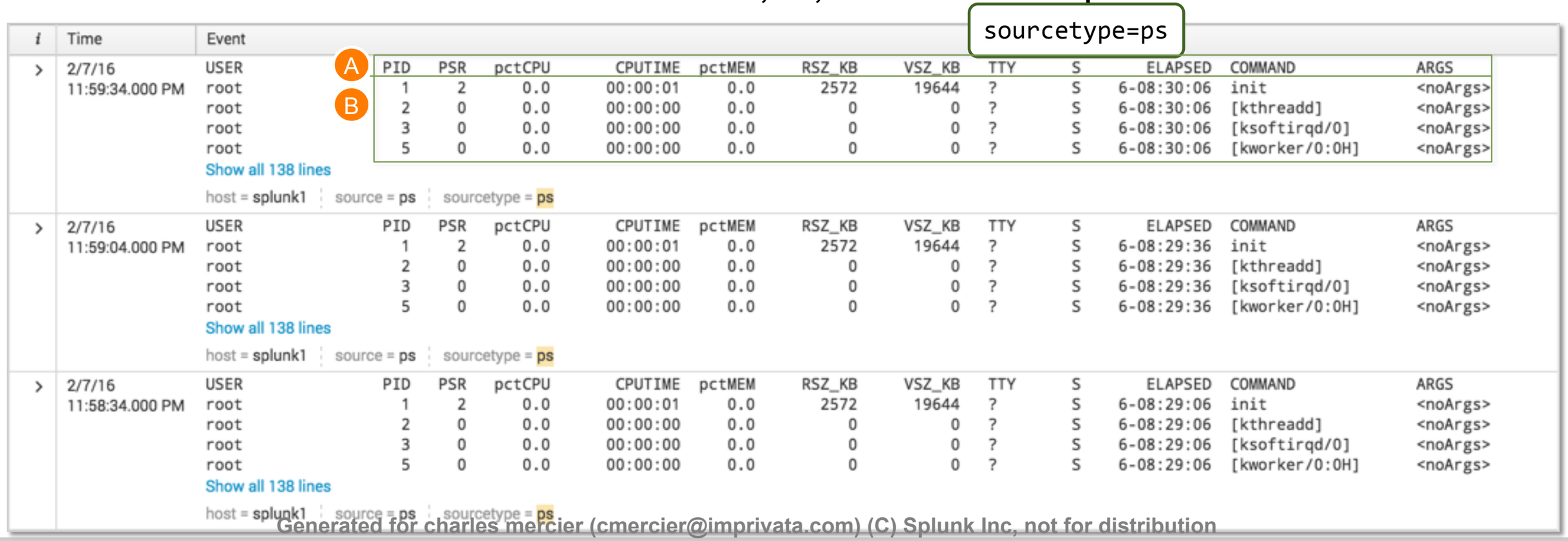
- Easier to use
  (Do not have to know regex)
- Provides regex code
- Must constantly provide examples
- Should not use in saved reports

rex

- More difficult to use
  (Must know regex)
- Do not have to provide examples
- Can use regex from erex
- Can use in saved reports

Extracting Fields from a Table-Formatted Event

- Many data types are formatted as large single events in a table
- Each event contains titles with tabular values
  - Fieldnames are derived from the title row, A; all other rows represent values B



multikv Command

- For table-formatted events, multikv creates an event for each row
- Field names are from the first row of each event

Interesting Fields

- a ARGS 72
- a COMMAND 98
- a CPUTIME 100+
- a ELAPSED 100+
- a eventtype 3
- # pctCPU 100+
- # pctMEM 10
- # PID 100+
- # PSR 4
- a punct 1
- # RSZ_KB 100+
- a S 4
- a tag 8
- a tag::eventtype 8
- a timestamp 1
- a TTY 9
- a USER 7
- # VSZ_KB 100+

sourcetype=ps

i	Time	Event
>	4/29/15 11:59:41.000 PM	USER PID PSR pctCPU CPUTIME pctMEM RSZ_KB VSZ_KB TTY S ELAPSED COMMAND ARGS root 1 3 0.0 00:00:00 0.0 1604 19488 ? S 07:53:36 init <noArgs> root 2 2 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [kthreadd] <noArgs> root 3 0 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [ksoftirqd/0] <noArgs> root 5 0 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [kworker/0:0H] <noArgs> Show all 106 lines host = splunk1 index = main linecount = 107 source = ps sourcetype = ps splunk_server = ip-10-222-134-157

sourcetype=ps | multikv

i	Time	Event
>	4/29/15 11:59:41.000 PM	root 1 3 0.0 00:00:00 0.0 1604 19488 ? S 07:53:36 init <noArgs> host = splunk1 index = main linecount = 1 source = ps sourcetype = ps splunk_server = ip-10-222-134-157
>	4/29/15 11:59:41.000 PM	root 2 2 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [kthreadd] <noArgs> host = splunk1 index = main linecount = 1 source = ps sourcetype = ps splunk_server = ip-10-222-134-157
>	4/29/15 11:59:41.000 PM	root 3 0 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [ksoftirqd/0] <noArgs> host = splunk1 index = main linecount = 1 source = ps sourcetype = ps splunk_server = ip-10-222-134-157
>	4/29/15 11:59:41.000 PM	root 5 0 0.0 00:00:00 0.0 0 0 ? S 07:53:36 [kworker/0:0H] <noArgs> host = splunk1 index = main linecount = 1 source = ps sourcetype = ps splunk_server = ip-10-222-134-157

multikv Command – fields

- To make the fields easier to read, this example pipes to the table command
- Use the fields option to limit the fields returned in the table
  - Only fields includes in this option are included, all others are ignored

```
sourcetype=ps
| multikv fields USER pctCPU COMMAND
| table USER, pctCPU, COMMAND
```

USER	pctCPU	COMMAND
root	0.0	init
root	0.0	[kthreadd]
root	0.0	[ksoftirqd/0]
root	0.0	[kworker/0:0H]
root	0.0	[rcu_sched]
root	0.0	[rcu_bh]

Lab Appendix A

- **Time:**
  - 20 minutes
- **Tasks:**
  - View the process activity on the Splunk indexer for the last 60 minutes
  - Extract the domain name from the email data into a field called domain
  - To gather the threat information, use the rex command to extract a field called threat in the email data

Appendix B: Accelerating Reports

Module Objectives

- Describe report acceleration
- Create summaries
- Search against summaries

Comparison of Data Summary Creation Methods

Report Acceleration	• Uses automatically created summaries to speed completion times for qualified reports • Easier to create than summary indexes and backfill automatically • Depending on the defined time span, periodically ages out data • Can correct gaps and overlaps from the UI 'rebuild' feature • Cannot create a "data-cube" and report on smaller subsets
Summary Indexing	• Useful for speeding up searches that don't qualify for report acceleration • Can persist after underlying events have been frozen by controlling retention period or index size • Backfill is a manual (scripted) process
Data Model Acceleration	• Uses automatically created summaries to speed completion times for pivots • Takes the form of time-series index files

Report Acceleration – Overview

- Reports that span a large volume of data can:
  - Take a long time to complete
  - Consume a great deal of system resources
- You can ‘accelerate’ a qualifying report when you:
  - Save it
  - Create a dashboard panel based on it
  - Edit a qualifying saved report

Report Acceleration – Conditions

You

- Your role has the `schedule search` capability
- You have `write` permissions for the report you want to accelerate

The report

- The report was not created via Pivot
- The search that the report is based upon is qualified for acceleration

Search Mode

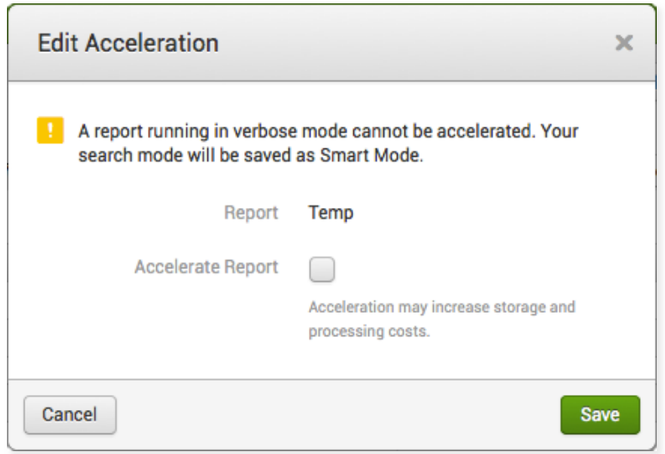
- You can accelerate a qualified report if the underlying search uses verbose mode, but Splunk automatically changes the search mode to smart or fast
- You cannot change search mode of an accelerated report to verbose

Report Acceleration – Common Use Cases

- Common use cases include:
  - More efficiently run reports for large datasets over long time ranges
    - Show the number of page views and visitors for each of your websites over the past 30 days, broken out by site
  - A rolling report that shows aggregated statistics over long periods of time
    - Display a running count of downloads for a specific file on a website
    - Calculate the average amount spent per purchase over a year

Report Acceleration – Acceleration Summaries

- To accelerate a report, Splunk creates an acceleration summary
- Acceleration summaries
  - Efficiently report on large volumes of data
  - Qualify future searches against the summary
- To accelerate a report, search mode must be set to either smart or fast
  - If in verbose mode, the report is saved with smart mode
  - Neither the timeline, nor the Fields sidebar displays
- By default, only power users can accelerate reports
- If you delete all the searches that use a summary, the summary is deleted
- If an acceleration summary is created from a shared report, reports that can use it, will use it



Populating Search Requirements

- Qualifying searches
  - Search must include a reporting command
    - For example: `chart`, `timechart`, `stats`, `top`, and `rare`
  - Any command before the reporting command must be a streaming command; that is, a command that applies a transformation to each event returned by the search
    - For example: `eval`, `fields`, `multikv`, `rex`, `rename`, and `replace`

Search Examples

- Qualifying search examples:

```
sourcetype=access_combined action=purchase status=200
| stats sum(price) as revenue by productId
| eval revenue="$" + revenue
```

```
sourcetype=access*
| fields price action host
| chart sum(price) over action by host
```

- Non-qualifying search examples

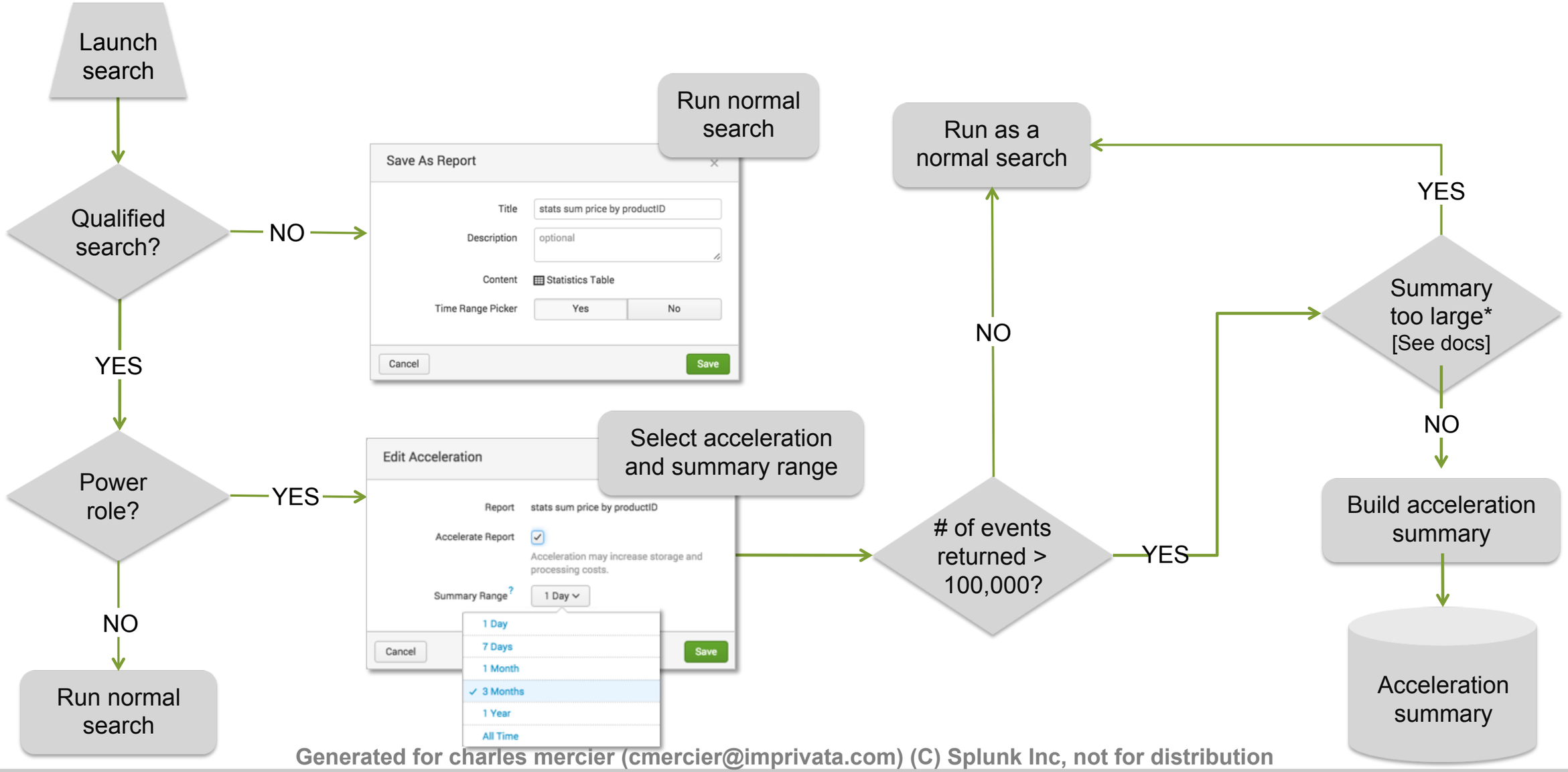
```
sourcetype=access_combined action=purchase status=404
```

[No reporting command]

```
sourcetype=access_combined
| transaction startswith="view" endswith="purchase"
| stats avg(duration)
```

[Transaction is not a streaming command]

Creating Acceleration Summaries

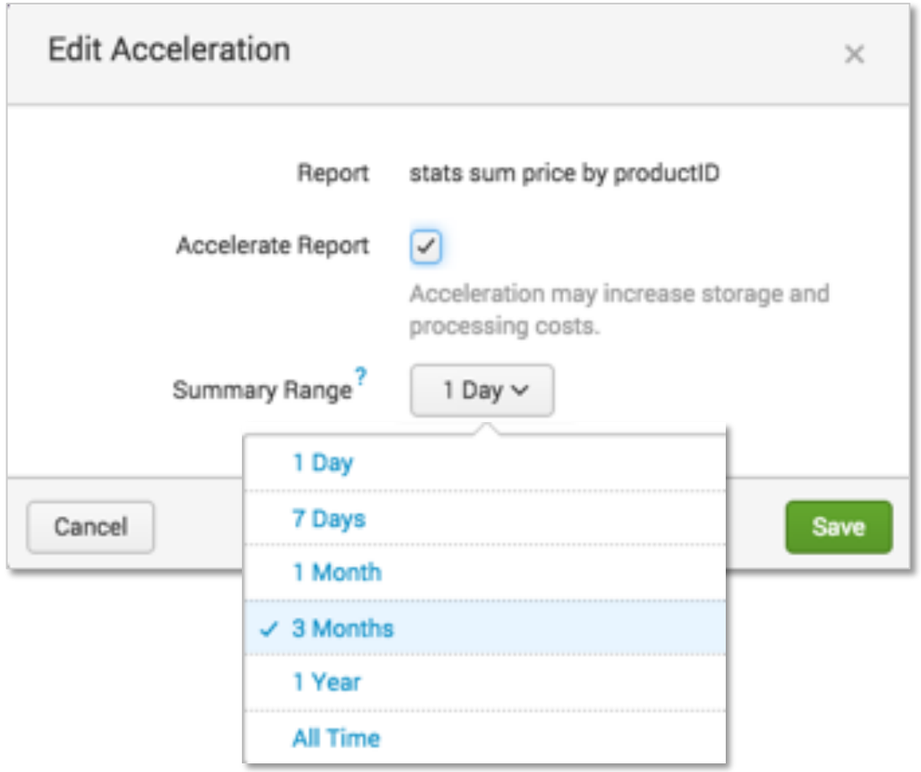


When Splunk Does NOT Build a Summary

- There are cases where Splunk allows you to "accelerate" a search, but a summary won't be created
- Splunk knows what's most efficient and **generally** won't generate a summary if:
  - There are fewer than 100K events in the summary range –
    It's faster executing the search without a summary
  - Summary size is projected to be too large –
    It's faster executing the search because the main index is smaller
- If a summary is defined and not created for the above reasons, Splunk continues to check periodically, then automatically creates a summary after it meets the requirements

Acceleration Summary Time Ranges

- Summary spans the specified range, relative to now
- Periodically removes older summary data that passes out of the range

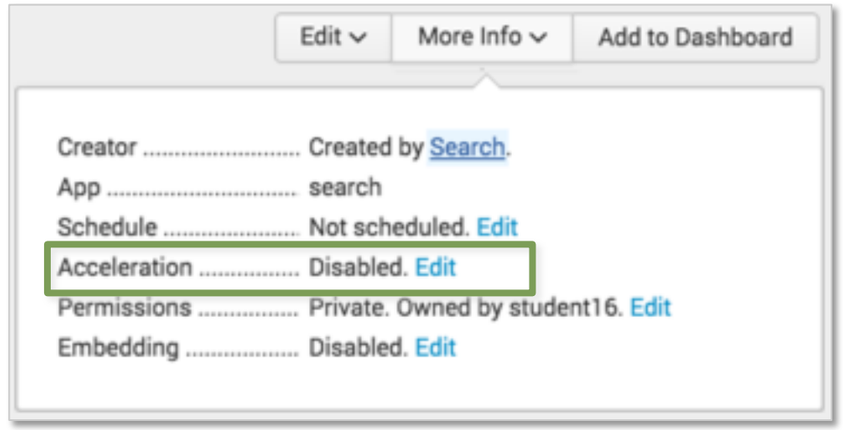


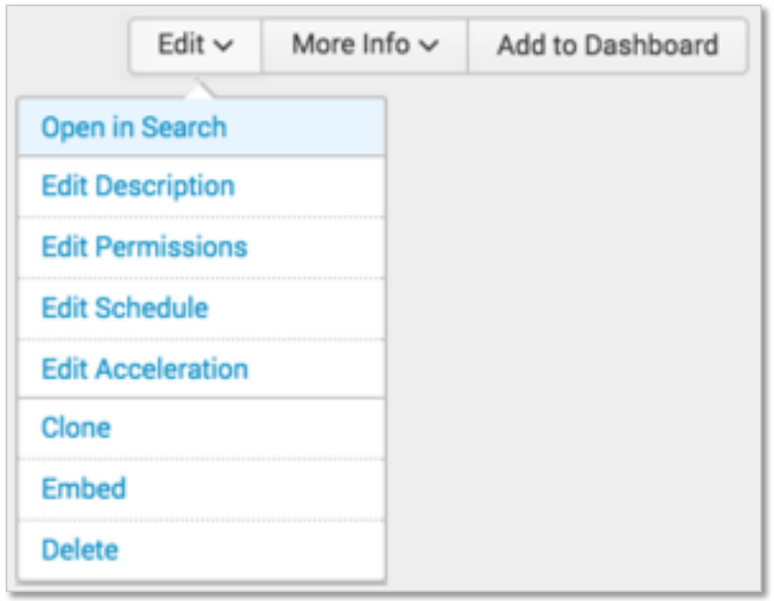
Note

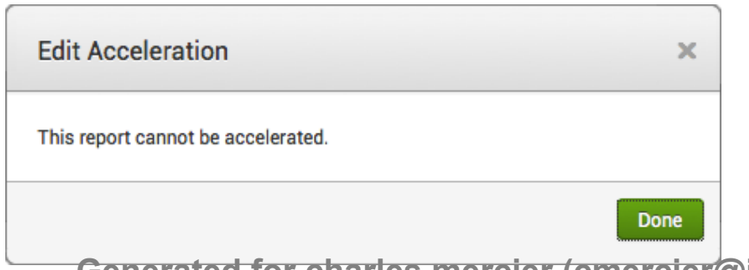
If 'old' [historical] data is important, then use summary indexing as all data is kept and not aged out.

Accelerating a Saved Report

- From `Settings > Searches, Reports and Alerts`, select a saved report
- The Accelerate option is always available, whether the search qualifies or not
  - Splunk determines if it can be accelerated when you click `Save`
- If you try to accelerate an unqualified search, an error message displays:







Searching Against a Summary

- In addition to saved accelerated reports, ad-hoc searches can use the summary when:
  - Search criteria matches the populating saved search
  - The time span is greater than or equal to the summary span
    - For time spans that are greater than the span of the summary, Splunk uses as much of the summary as it can
- You can also append the search string with additional reporting commands
  - Example:
    populating search –

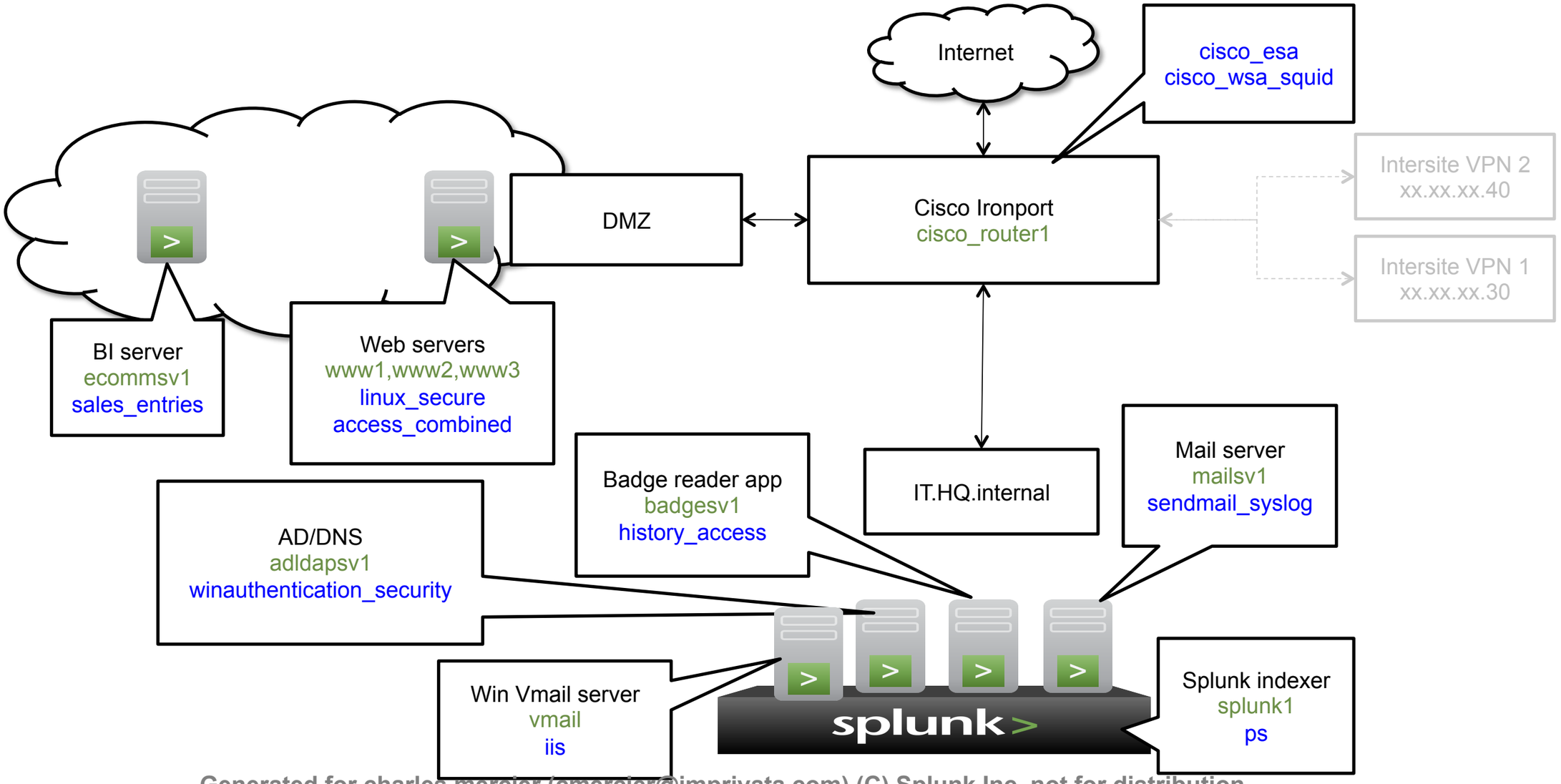
```
sourcetype=access_combined | stats count by price
```

    ad hoc search –

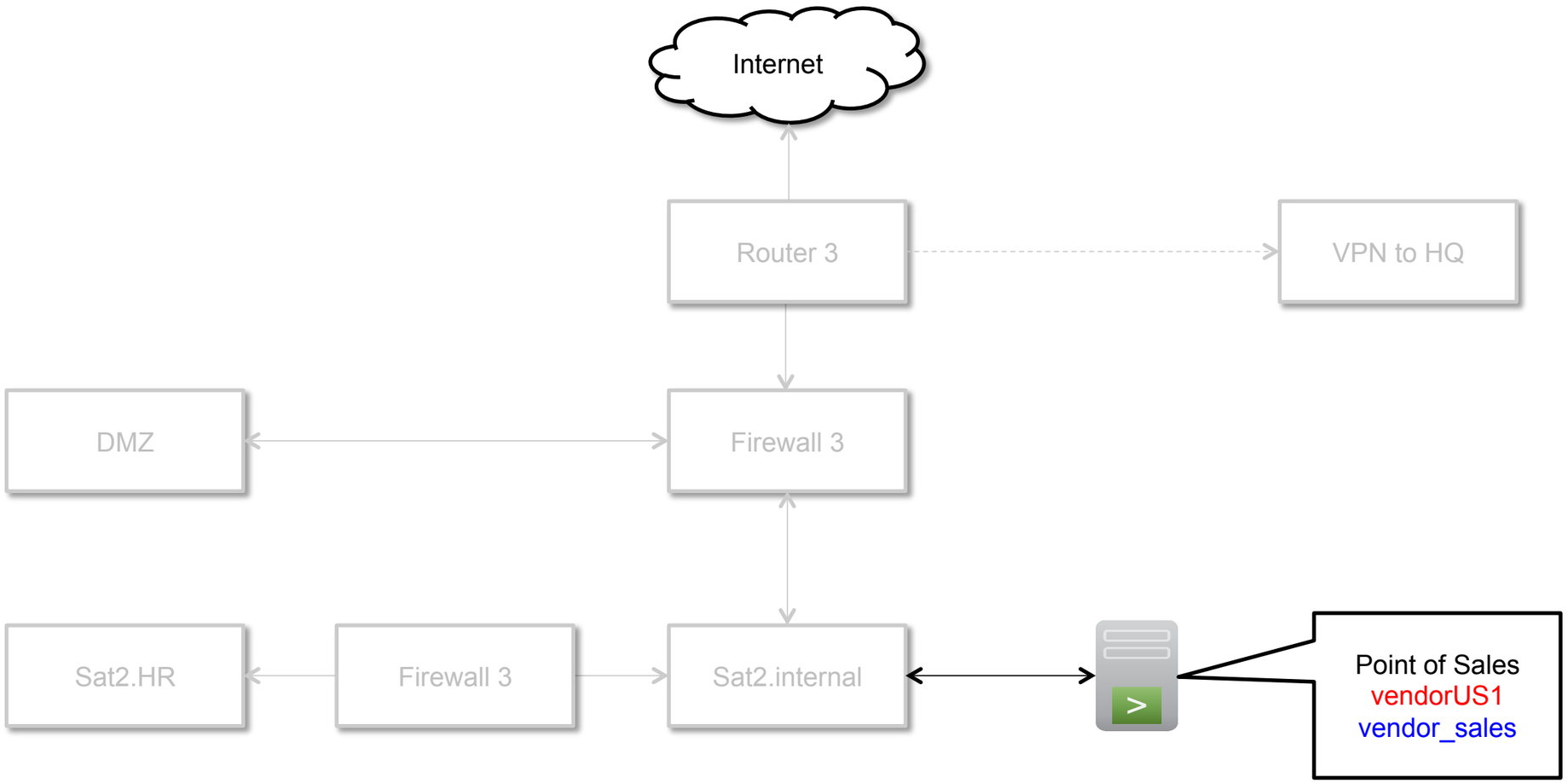
```
sourcetype=access_combined | stats count by price | eval discount = price/2
```

Appendix C: Buttercup Games

San Francisco – Headquarters



Boston – Satellite Office 2



Buttercup Games Environment

Data	host	sourcetype
AD/DNS data	adldapsv1	WinEventLog:Security
Badge reader data	badgesv1	history_access
BI server data	ecommsv1	sales_entries
Email data	cisco_router1	cisco_esa
Online transactions & Web server	www1	access_combined linux secure
	www2	
	www3	
Retail sales data	vendorUS1	vendor_sales
Splunk indexer data	splunk1	ps
Web appliance data	cisco_router1	cisco_wsa_squid

Buttercup Games – HQ Employees

emp_no	birth_date	first_name	last_name	gender	hire_date	location
1	1963-12-25	Suzanne	Flaemmchen	F	2008-05-03	San Francisco
2	1960-04-20	Huang	Sham	M	2008-05-03	San Francisco
3	1950-06-09	Stefano	Pahkthecah	M	2008-05-03	San Francisco
4	1962-01-01	Shawn	Scallion	M	2008-05-03	San Francisco
5	1992-02-29	Shane	Youngin	M	2008-05-03	San Francisco
11	1969-08-19	Placido	Toscani	M	2009-06-09	San Francisco
12	1988-12-06	Meng	Yuan	F	2009-06-09	San Francisco
13	1963-09-29	Amanda	Curry	F	2009-06-09	San Francisco
14	1978-10-31	Bao	Lu	M	2009-06-09	San Francisco
			:			
			:			
68	1978-09-19	Pat	Leuchs	NR	2011-02-04	San Francisco
70	1964-05-19	Patricia	dAbbeville	F	2009-03-14	San Francisco
72	1978-07-10	Saran	Wrappe	F	2011-04-16	San Francisco
73	1988-12-01	Thomasina	Cugina	F	2012-05-19	San Francisco
75	1963-06-28	Frazer	Ullian	M	2013-12-13	San Francisco
76	1964-05-19	Mitsuko	Oh	F	2008-07-04	San Francisco
77	1962-04-01	Yurij	Schonegge	M	2010-01-11	San Francisco
81	1970-01-01	Buttercup	Pony	P	2008-05-03	San Francisco

Buttercup Games – Satellite Employees

emp_no	birth_date	first_name	last_name	gender	hire_date	location
7	1963-06-07	Daniil	Piazza	M	2009-06-09	Boston
8	1961-05-02	Enrique	Dutra	M	2009-06-09	Boston
9	1974-06-19	Louis	Sagers	M	2009-06-09	Boston
23	1978-02-19	Saniya	Kalloufi	M	2009-09-15	Boston
			:			
69	1962-09-18	Kish	Perna	F	2008-10-21	Boston
79	1973-10-18	Debatosh	Khasidashvili	M	2009-01-30	Boston

emp_no	birth_date	first_name	last_name	gender	hire_date	location
10	1986-02-12	Cosima	Quinn	F	2009-06-09	London
32	1977-05-23	Tzvetan	Zielinski	F	2010-02-10	London
34	1986-02-12	Berni	Genin	M	2010-03-11	London
35	1966-11-14	Cedric	Munson	M	2010-03-18	London
37	1983-09-02	Gianpaolo	Facello	M	2010-06-26	London
			:			
71	1977-01-27	Gioia	Bottazzi	F	2013-05-12	London
74	1963-06-07	Moses	Adeyemi	M	2013-05-11	London
78	1984-05-27	Santino	Sbarro	M	2009-11-06	London
80	1975-07-22	Giancarlo	Rao	M	2008-10-21	London

Buttercup Games – Employee Information

emp no	RFID	IP	user	host	email	splunk role
1	417852300683	10.1.10.201	sflaemmchen	BG01-sflaemmchen	sflaemmchen@buttercupgames.com	user
2	542830538161	10.1.10.231	hsham	BG01-hsham	hsham@buttercupgames.com	user
3	520156890727	10.1.10.230	spahkthecah	BG01-spahkthecah	spahkthecah@buttercupgames.com	user
4	564931543224	10.1.10.216	sscallion	BG01-sscallion	sscallion@buttercupgames.com	user
5	534931200268	10.1.10.241	syoungin	BG01-syoungin	syoungin@buttercupgames.com	power
6	768166372290	10.1.10.290	lhaddadi	BG01-lhaddadi	lhaddadi@buttercupgames.com	power
7	659636929855	10.2.10.38	dpiazza	BG02-dpiazza	dpiazza@buttercupgames.com	user
8	559129672655	10.2.10.77	edutra	BG02-edutra	edutra@buttercupgames.com	power
9	960318676000	10.2.10.45	lsagers	BG02-lsagers	lsagers@buttercupgames.com	power
10	513908343176	10.3.10.28	cquinn	BG03-cquinn	cquinn@buttercupgames.com	admin
11	125179529264	10.1.10.234	ptoscani	BG01-ptoscani	ptoscani@buttercupgames.com	power
12	382839148784	10.1.10.238	myuan	BG01-myuan	myuan@buttercupgames.com	power
13	713929421175	10.1.10.246	acurry	BG01-acurry	acurry@buttercupgames.com	power
14	900191452102	10.1.10.252	blu	BG01-blu	blu@buttercupgames.com	user
				:		
				:		
81	999999999999	10.1.10.1	bpony	BG01-bpony	bpony@buttercupgames.com	user

Vendor Sales – Sample

Vendor	VendorCity	VendorCountry	VendorID	VendorLatitude	VendorLongitude	VendorStateProvince
Frozen Fun General Store	Amundsen-Scott Station	Antarctica	9999	90.0000	139.2667	Antarctica
Jeremy's House of Hobbies	Fort-Lamy	Chad	9116	12.134846	15.055742	Chari-Baguirmi
Pan-African RC and Toys	Ouagadougou	Burkina Faso	9115	12.364637	-1.533864	Nord Region
Passe-Temps	Yamoussoukro	Cote d'Ivoire	9114	6.816667	-5.283333	Lacs
Kahled's Amusements	Tripoli	Libya	9113	5.560735	-0.193087	Tripoli
Mburo Games	Kampala	Uganda	9112	0.313611	32.581111	Kampala
Pan-African RC and Toys	Yaounde	Cameroon	9111	3.866667	11.516667	Centre Region
Comics and Games	Dar es Salaam	Tanzania	9110	-6.822921	39.269661	Dar es Salaam
Pan-African RC and Toys	Mombasa	Kenya	9109	-4.043477	39.668207	Mombasa
Pan-African RC and Toys	Accra	Ghana	9108	5.555717	-0.196306	Greater Accra
Seminna-Werq Games Warehouse	Addis Ababa	Ethiopia	9107	9.022736	38.746799	Oromia
RTL Boutique de Train Miniature	Tunis	Tunisia	9106	36.81881	10.16596	Tunis
Rick's Toy Shop and Cafe	Casablanca	Morocco	9105	33.533333	-7.583333	Grand Casablanca
Laval's Joke and Toy Store	Oran	Algeria	9104	35.696944	-0.633056	Oran
Sweepstake Games	Lagos	Nigeria	9103	6.441158	3.417977	Lagos
Lightening Games of Johannesburg	Johannesburg	South Africa	9102	-26.204103	28.047305	Gauteng
Natal Games of Pietermaritzburg	Pietermaritzburg	South Africa	9101	-29.600607	30.379412	KwaZulu-Natal
Peers Games of Cape Town	Cape Town	South Africa	9100	-33.924868	18.424055	Western Cape
Kiwi Game Warehouse	Auckland	New Zealand	7045	-36.84846	174.763332	Auckland
Kiwi Game Warehouse	Christchurch	New Zealand	7044	-43.529854	172.637888	Canterbury

Buttercup Games – Products

productId	product_name	categoryId
WC-SH-T02	World of Cheese Tee	TEE
WC-SH-G04	World of Cheese	SHOOTER
WC-SH-A02	Fire Resistance Suit of Provolone	ACCESSORIES
WC-SH-A01	Holy Blade of Gouda	ACCESSORIES
SC-MG-G10	SIM Cubicle	SIMULATION
PZ-SG-G05	Puppies vs. Zombies	STRATEGY
MB-AG-T01	Manganiello Bros. Tee	TEE
MB-AG-G07	Manganiello Bros.	ARCADE
FS-SG-G03	Final Sequel	STRATEGY
FI-AG-G08	Orvil the Wolverine	ARCADE
DC-SG-G02	Dream Crusher	STRATEGY
DB-SG-G01	Mediocre Kingdoms	STRATEGY
CU-PG-G06	Curling 2014	SPORTS
BS-AG-G09	Benign Space Debris	ARCADE