splunk>

Comparing Values – Lab Solutions Guide

Overview

Welcome to the Splunk Education lab environment. These lab exercises will test your knowledge of comparison commands within Splunk.

Scenario

You will use data from the international video game company, Buttercup Games. A list of source types is provided below.

NOTE: This is a lab environment driven by data generators with obvious limitations. This is not a production environment. Screenshots approximate what you should see, not the **exact** output.

Index	Type	Sourcetype	Interesting Fields
web	Online sales	**access_combined**	action, bytes, categoryId, clientip, itemId, JSESSIONID, price, productId, product_name, referer, referer_domain, sale_price, status, user, useragent
sales	Retail sales	**vendor_sales**	categoryId, product_name, productId, sale_price, Vendor, VendorCity, VendorCountry, VendorID, VendorStateProvince
	Business Intelligence server	**sales_entries**	AcctCode, CustomerID, TransactionID
network	Web security appliance data	**cisco_wsa_squid**	action, cs_method, cs_mime_type, cs_url, cs_username, sc_bytes, sc_http_status, sc_result_code, severity, src_ip, status, url, usage, x_mcafee_virus_name, x_wbrs_score, x_webcat_code_abbr

splunk>

Common Commands and Functions

These commands and statistical functions are commonly used in searches but may not have been explicitly discussed in the course. Please use this table for quick reference. Click on the hyperlinked SPL (Search Processing Language) to be taken to the Search Manual for that command or function.

SPL	Type	Description	Example
sort	command	Sorts results in descending or ascending order by a specified field. Can limit results to a specific number.	*Sort the first 100 `src_ip` values in descending order* `| sort 100 -src_ip`
where	command	Filters search results using eval-expressions.	*Return events with a `count` value greater than 30* `| where count > 30`
rename	command	Renames one or more fields.	*Rename `SESSIONID` to 'The session ID'* `| rename SESSIONID as "The session ID"`
fields	command	Keeps (+) or removes (-) fields from search results.	*Remove the `host` field from the results* `| fields - host`
stats	command	Calculates aggregate statistics over the results set.	*Calculate the total sales, i.e. the sum of `price` values.* `| stats sum(price)`
eval	command	Calculates an expression and puts the resulting value into a new or existing field.	*Concatenate `first_name` and `last_name` values with a space to create a field called "`full_name`"* `| eval full_name=first_name." ".last_name`
table	command	Returns a table.	*Output `vendorCountry`, `vendor`, and `sales` values to a table* `| table vendorCountry, vendor, sales`
sum()	statistical function	Returns the sum of the values of a field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Calculate the sum of the `bytes` field* `| stats sum(bytes)`
count or count()	statistical function	Returns the number of occurrences of all events or a specific field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Count all events as "events" and count all events that contain a value for `action` as "action"* `| stats count as events, count(action) as action`

Refer to the Search Reference Manual for a full list of commands and functions.

Lab Exercise 1 – Using `eval` to Compare

Description

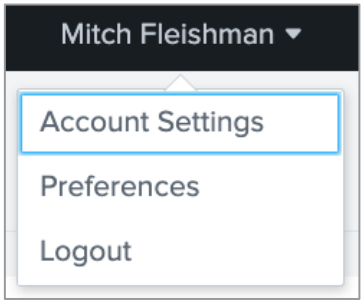
Configure the lab environment user account. Then, use the eval command to compare and conceal values in your results.

Steps

Task 1: Log into Splunk and change the account name and time zone.

Set up your lab environment to fit your time zone. This also allows the instructor to track your progress and assist you if necessary.

1. Log into your Splunk lab environment using the username and password provided to you.
2. Choose **Search & Reporting** from the list of **Apps**.
3. You may see a pop-up window welcoming you to the lab environment. You can click **Continue to Tour** but this is not required. Click **Skip** to dismiss the window.
4. Click on the username you logged in with (at the top of the screen) and then choose **Account Settings** from the drop-down menu.
5. In the **Full name** box, enter your first and last name.
6. Click **Save**.
7. Reload your browser to reflect the recent changes to the interface. (This area of the web interface will be referred to as ***user name***.)



After you complete step 6, you will see your name in the web interface.

NOTE: Sometimes there can be delays in executing an action like saving in the UI or returning results of a search. If you are experiencing a delay, please allow the UI a few minutes to execute your action.

8. Navigate to ***user name*** **> Preferences.**
9. Choose your local time zone from the **Time zone** drop-down menu.
10. Click **Apply**.
11. (Optional) Navigate to ***user name*** **> Preferences > SPL Editor > Search auto-format** and click on the toggle to activate auto-formatting. Then click **Apply**. When the pipe character is used in search, the SPL Editor will automatically begin the pipe on a new line.

Search auto-format disabled (default)

Search auto-format enabled

Scenario: The Sales department has requested a statistical analysis of sales worldwide and would like to review sales performance for every country over the last 7 days.

Task 2: Use the `case` and `if` functions to determine a country's sales performance and generate a verdict based on that performance.

12. In the top left corner of Splunk Web, select **Apps** > **Search & Reporting**. This sets the app context to the search app.
13. Search for (**`index=sales sourcetype=vendor_sales`**) over the **Last 7 days**.
```
index=sales sourcetype=vendor_sales
```
14. Pipe results to the following **`stats`** command.

```
| stats sum(price) as Sales by VendorStateProvince
```

The **`stats`** command calculates the sum of **`price`** values for each **`VendorStateProvince`**. The values of this calculation are assigned to a field called **`Sales`**.
```
index=sales sourcetype=vendor_sales
| stats sum(price) as Sales by VendorStateProvince
```

VendorStateProvince	Sales
Abu Dhabi	132.91
Al Wustaa	73.96
Alabama	461.74
Alaska	599.73
Alberta	204.93
Alexandria	69.96
Amman	27.97
Andhra Pradesh	320.87
Andorra la Vella	39.99
Arizona	545.73

15. Use the **`eval`** command to create a field called "Performance" that assigns the following values based on the value of **`Sales`**:
    a. Sales less than or equal to 200: “Needs performance appraisal”
    b. Sales greater than 200 and less than 500: “Underperformer”
    c. Sales greater than or equal to 500: “Overperformer”

```
index=sales sourcetype=vendor_sales
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=200,"Needs performance appraisal",
  Sales<500,"Underperformer",Sales>=500,"Overperformer")
```

VendorStateProvince	Sales	Performance
Abu Dhabi	132.91	Needs performance appraisal
Al Wustaa	73.96	Needs performance appraisal
Alabama	461.74	Underperformer
Alaska	599.73	Overperformer
Alberta	204.93	Underperformer
Alexandria	69.96	Needs performance appraisal
Amman	27.97	Needs performance appraisal
Andhra Pradesh	320.87	Underperformer
Andorra la Vella	39.99	Needs performance appraisal
Arizona	545.73	Overperformer

16. Use the **`eval`** command to create a field called "Verdict" that assigns a value of "Send to marketing" if the value of **`Performance`** is "Needs performance appraisal."

```
index=sales sourcetype=vendor_sales
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=200,"Needs performance appraisal",
  Sales<500,"Underperformer",Sales>=500,"Overperformer")
| eval Verdict = if(Performance IN("Needs performance appraisal"), "Send to
marketing",null())
```

VendorStateProvince	Sales	Performance	Verdict
Abu Dhabi	132.91	Needs performance appraisal	Send to marketing
Al Wustaa	73.96	Needs performance appraisal	Send to marketing
Alabama	461.74	Underperformer	
Alaska	599.73	Overperformer	
Alberta	204.93	Underperformer	
Alexandria	69.96	Needs performance appraisal	Send to marketing
Amman	27.97	Needs performance appraisal	Send to marketing
Andhra Pradesh	320.87	Underperformer	
Andorra la Vella	39.99	Needs performance appraisal	Send to marketing
Arizona	545.73	Overperformer	

17. Pipe results to the following **eval** command.

```
| eval Sales = "$".tostring(Sales,"commas")
```

The **eval** command formats **Sales** data to include a dollar sign ($) and commas.

```
index=sales sourcetype=vendor_sales
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=200,"Needs performance appraisal",
  Sales<500,"Underperformer",Sales>=500,"Overperformer")
| eval Verdict = if(Performance IN("Needs performance appraisal"), "Send to
marketing",null())
| eval Sales = "$".tostring(Sales,"commas")
```

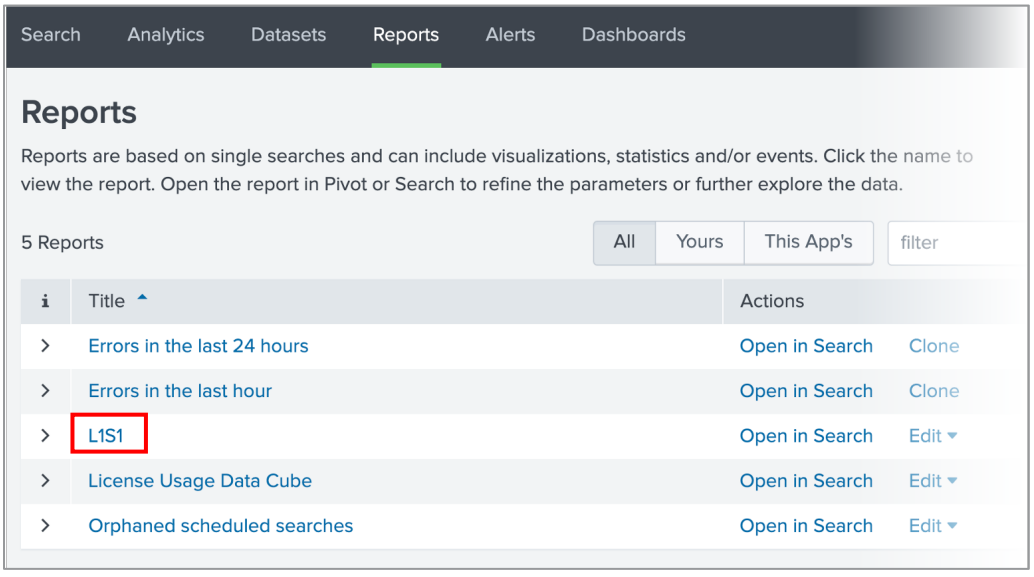
VendorStateProvince	Sales	Performance	Verdict
Abu Dhabi	$132.91	Needs performance appraisal	Send to marketing
Al Wustaa	$73.96	Needs performance appraisal	Send to marketing
Alabama	$461.74	Underperformer	
Alaska	$599.73	Overperformer	
Alberta	$204.93	Underperformer	
Alexandria	$69.96	Needs performance appraisal	Send to marketing
Amman	$27.97	Needs performance appraisal	Send to marketing
Andhra Pradesh	$320.87	Underperformer	
Andorra la Vella	$39.99	Needs performance appraisal	Send to marketing
Arizona	$545.73	Overperformer	

18. Use the **rename** command to rename **VendorStateProvince** as "Location."

```
index=sales sourcetype=vendor_sales
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=200,"Needs performance appraisal",
  Sales<500,"Underperformer",Sales>=500,"Overperformer")
| eval Verdict = if(Performance IN("Needs performance appraisal"), "Send to
marketing",null())
| eval Sales = "$".tostring(Sales,"commas")
| rename VendorStateProvince as Location
```

Location	Sales	Performance	Verdict
Abu Dhabi	$132.91	Needs performance appraisal	Send to marketing
Al Wustaa	$73.96	Needs performance appraisal	Send to marketing
Alabama	$461.74	Underperformer	
Alaska	$599.73	Overperformer	
Alberta	$204.93	Underperformer	
Alexandria	$69.96	Needs performance appraisal	Send to marketing
Amman	$27.97	Needs performance appraisal	Send to marketing
Andhra Pradesh	$320.87	Underperformer	
Andorra la Vella	$39.99	Needs performance appraisal	Send to marketing
Arizona	$545.73	Overperformer	

19. Save your search as a report with the name **L1S1**.
    a. Click **Save As** > **Report**
    b. For **Title**, enter L1S1.
    c. **Save**.
    d. You can **View** your report or exit out of the **Your Report Has Been Created** window by clicking the **X** in the upper-right corner.
    e. You can access your saved reports using the **Reports** tab in the application bar.



*Your recently saved **L1S1** report will be visible in the **Reports** tab.*

Scenario: Evaluate and classify the number of bytes associated with each web server event during the last 24 hours.

Task 3: Use the `case` function to categorize events.

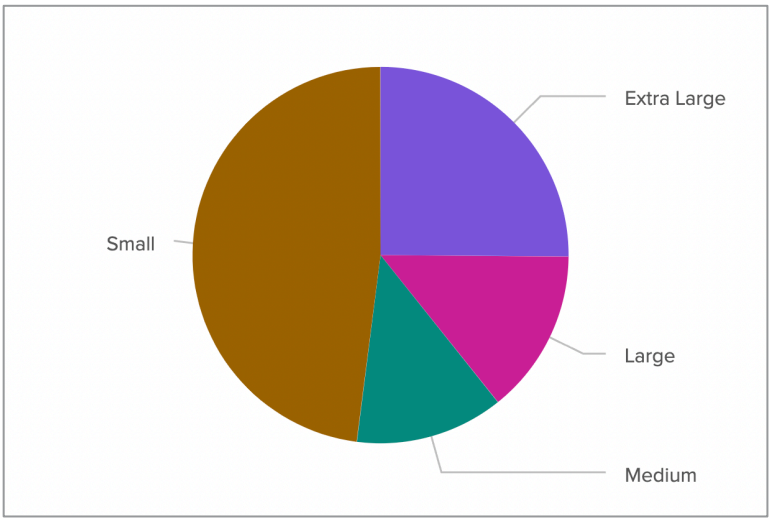
20. Navigate to the **Search & Reporting** app in the application bar.
21. Complete the **`<missing>`** portion of this search so that:
    a. Events with **`bytes`** less than 2000 are classified as "Small"
    b. Events with **`bytes`** greater than 2000 but less than 2500 are classified as "Medium"
    c. Events with **`bytes`** greater than 2500 but less than 3000 are classified as "Large"
    d. All other events are classified as "Extra Large"
    e. Values are stored in a field called "dataSize" (This field will be used by the following **`chart`** command.)
    f. Run this search over the **Last 24 hours**.

```
index=web sourcetype=access_combined
| <missing>
| chart count by dataSize
```

dataSize	count
Extra Large	621
Large	300
Medium	323
Small	1055

```
index=web sourcetype=access_combined
| eval dataSize = case(bytes<2000,"Small", bytes<2500,"Medium", bytes<3000,"Large",
  true(),"Extra Large")
| chart count by dataSize
```

22. Navigate to **Visualization** and visualize the results as a **Pie Chart**.



23. Save your search as a report with the name **L1S2**.

Scenario: Sales wants a count of all vendor retail sales over the last 4 hours but need to mask the customer account codes for security purposes.

Task 4: Use the `replace` function to conceal account codes.

24. Complete the **`<missing>`** portion of the **`eval`** command so that only the first number and the last 3 numbers of the account codes are visible. The regex needed for this is **`^(\d)\d{3}-\d(\d{3})`**. Search over the **Last 4 hours**. (Hint: There are two capture groups present in this regex.)

```
index=sales sourcetype=sales_entries AcctCode=*
| stats count by AcctCode
| eval AcctCode = <missing>
```

AcctCode	count
0xxx-x250	2
0xxx-x454	4
0xxx-x173	5
0xxx-x873	3
0xxx-x656	4
0xxx-x768	1
1xxx-x044	3

```
index=sales sourcetype=sales_entries AcctCode=*
| stats count by AcctCode
| eval AcctCode = replace(AcctCode,"^(\d)\d{3}-\d(\d{3})","\1xxx-x\2")
```

25. Save your search as a report with the name **L1S3**.

Lab Exercise 2 – Filtering & Managing Missing Data

Description

Use the **`where`** command to filter results. Optionally, complete a challenge exercise where you will use an **`eval`** expression with the **`stats`** command to filter results.

Steps

Scenario: The Buttercup Games Customer Support email has been receiving a lot of complaints that product names are not showing up when customers try to add items to their cart or checkout. Instead, customers are seeing only product ID numbers.

Task 1: Use the `where` command to find `addtocart` and `purchase` events that have a `productId` value but are missing a `product_name` value. Then use the `fillnull` command to mark these events for review.

1. Search online sales data for **`addtocart`** and **`purchase`** events (**`index=web sourcetype=access_combined action IN(addtocart, purchase)`**) over the **Last 24 hours**.
```
index=web sourcetype=access_combined action IN(addtocart,purchase)
```
2. Use the **`where`** command to find events where **`product_name`** values are NULL and where **`productId`** values are not NULL.
```
index=web sourcetype=access_combined action IN(addtocart,purchase)
| where isnull(product_name)
| where isnotnull(productId)
```
3. Pipe results to the table command to see the values for **`JSESSIONID`**, **`action`**, **`product_name`**, and **`productId`**.
```
| table JSESSIONID, action product_name productId
```

JSESSIONID	action	product_name	productId
SD5SL8FF3ADFF4955	addtocart		SF-BVS-01
SD1SL2FF9ADFF193117	purchase		SF-BVS-G01
SD3SL8FF8ADFF193026	addtocart		SF-BVS-G01
SD3SL8FF8ADFF193026	purchase		SF-BVS-G01
SD1SL7FF9ADFF192360	addtocart		SF-BVS-G01
SD10SL9FF3ADFF192352	addtocart		SF-BVS-G01

```
index=web sourcetype=access_combined action IN(addtocart,purchase)
| where isnull(product_name)
| where isnotnull(productId)
| table JSESSIONID, action product_name productId
```

4. Use the **`fillnull`** command to add “NEEDS REVIEW” for all NULL values.

JSESSIONID	action	product_name	productId
SD5SL8FF3ADFF4955	addtocart	NEEDS REVIEW	SF-BVS-01
SD1SL2FF9ADFF193117	purchase	NEEDS REVIEW	SF-BVS-G01
SD3SL8FF8ADFF193026	addtocart	NEEDS REVIEW	SF-BVS-G01
SD3SL8FF8ADFF193026	purchase	NEEDS REVIEW	SF-BVS-G01
SD1SL7FF9ADFF192360	addtocart	NEEDS REVIEW	SF-BVS-G01
SD10SL9FF3ADFF192352	addtocart	NEEDS REVIEW	SF-BVS-G01

```
index=web sourcetype=access_combined action IN(addtocart,purchase)
| where isnull(product_name)
| where isnotnull(productId)
| table JSESSIONID, action product_name productId
| fillnull product_name value="NEEDS REVIEW"
```

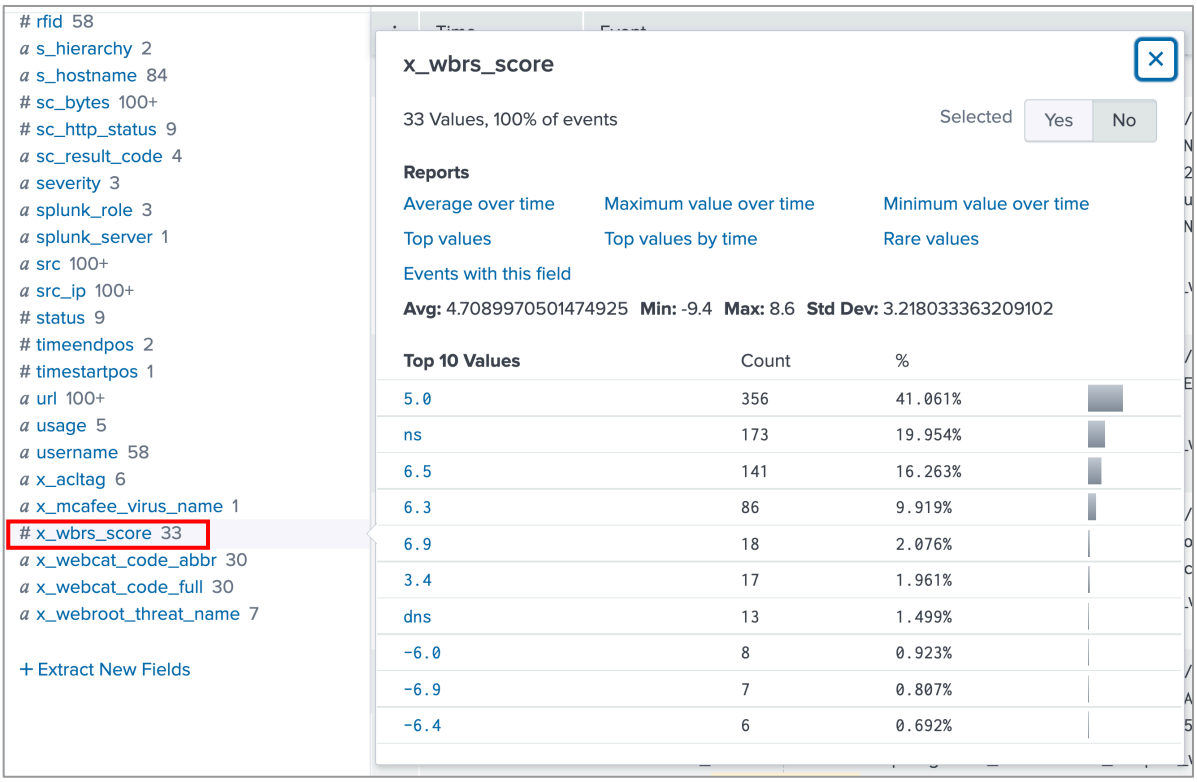
5. Save your search as a report with the name **L2S1**.

Scenario: SecOps found a potential virus on a user's machine. Find and classify the number of internet visits by risk and workstation during the past 7 days.

Task 2: Use the `eval` command and `case` function to classify events using the `x_wbrs_score` field.

6. Search web security appliance data (**`index=network sourcetype=cisco_wsa_squid`**) during the **Last 7 days** to find events that contain a web-based reputation score (**`x_wbrs_score=*`**).

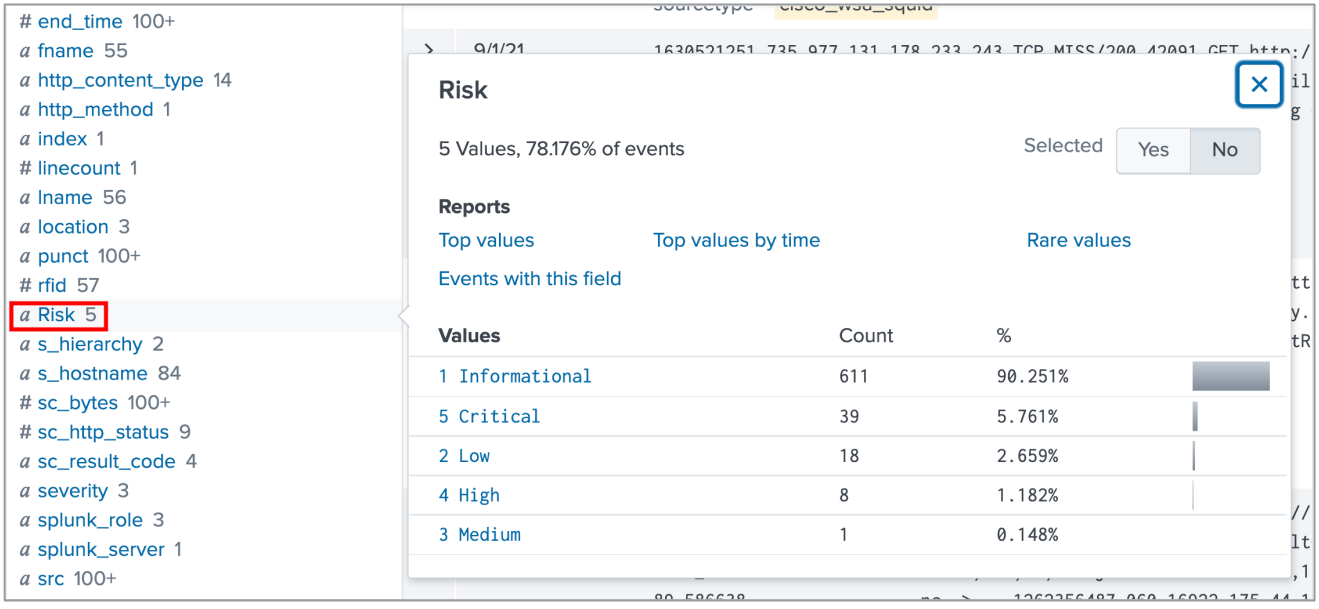
```
index=network sourcetype=cisco_wsa_squid x_wbrs_score=*
```



NOTE: The web-based reputation score (WBRS) assigned to URLs determines the likelihood that the webpage contains URL-based malware. The Cisco Web Security appliance uses this information to stop malware attacks before they occur. Scores range from 10.0 to -10.0 and anything under 6.0 is scanned or blocked.

7. Use a single **eval** command to create a field named "Risk" and assign a string value for each event based on the value of **x_wbrs_score**:
   - Events with a **x_wbrs_score** greater than or equal to 5 will be assigned “1 Informational”
   - Events with a **x_wbrs_score** greater than or equal to 3 should be assigned ”2 Low”
   - Events with a **x_wbrs_score** greater than or equal to 0 should be assigned "3 Medium"
   - Events with a **x_wbrs_score** greater than or equal to -5 should be assigned “4 High”
   - Events with a **x_wbrs_score** less than -5 should be assigned “5 Critical”

```
index=network sourcetype=cisco_wsa_squid x_wbrs_score=*
| eval Risk = case(x_wbrs_score >= 5,"1 Informational",
  x_wbrs_score >= 3,"2 Low",
  x_wbrs_score >= 0,"3 Medium",
  x_wbrs_score >= -5,"4 High",
  x_wbrs_score < -5, "5 Critical")
```



8. Pipe results to the following **timechart** command.

```
| timechart count(bcg_workstation) by Risk usenull=f useother=f
```

The **timechart** command will count events that contain a value for **bcg_workstation** for each day in your time range. The **by Risk** clause will further split these values for each value of **Risk** and create a multi-series data series. The **usenull** and **useother** options are included to remove NULL and OTHER columns from your results.

```
index=network sourcetype=cisco_wsa_squid x_wbrs_score=*
| eval Risk = case(x_wbrs_score >= 5,"1 Informational",
  x_wbrs_score >= 3,"2 Low",
```

```
  x_wbrs_score >= 0,"3 Medium",
  x_wbrs_score >= -5,"4 High",
  x_wbrs_score < -5, "5 Critical")
| timechart count(bcg_workstation) by Risk usenull=f useother=f
```

_time	1 Informational	2 Low	3 Medium	4 High	5 Critical
2021-08-25	39	0	0	0	0
2021-08-26	79	0	0	0	0
2021-08-27	59	13	0	0	5
2021-08-28	116	0	1	0	1
2021-08-29	94	0	0	3	4
2021-08-30	63	5	0	3	1
2021-08-31	84	0	0	0	6
2021-09-01	77	0	0	2	23

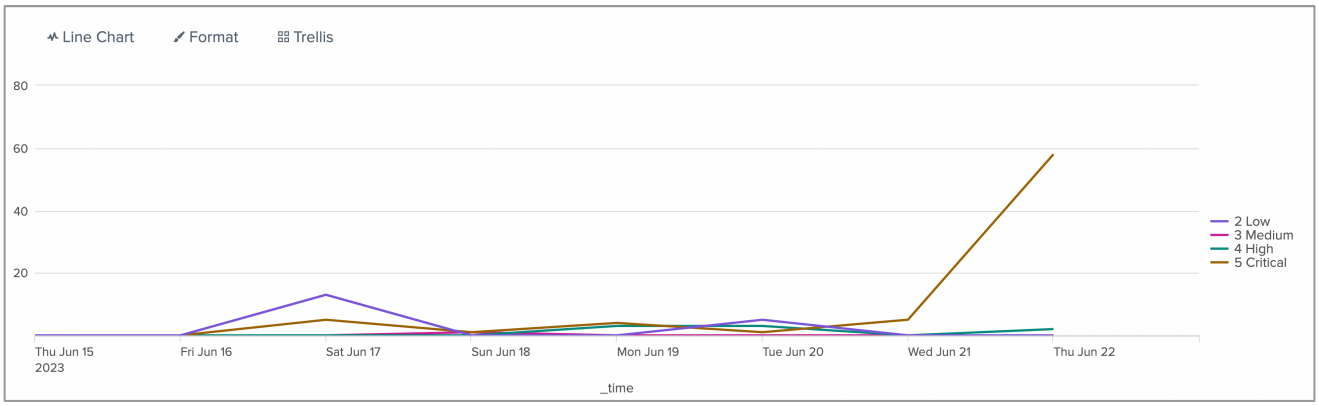
9. There are too many **`1 Informational`** events skewing the results. Remove some of these events by limiting your results to only those events with a web-based reputation score less than 4.

```
index=network sourcetype=cisco_wsa_squid x_wbrs_score=*
| eval Risk = case(x_wbrs_score >= 5,"1 Informational",
  x_wbrs_score >= 3,"2 Low",
  x_wbrs_score >= 0,"3 Medium",
  x_wbrs_score >= -5,"4 High",
  x_wbrs_score < -5, "5 Critical")
| where x_wbrs_score < 4
| timechart count(bcg_workstation) by Risk usenull=f useother=f
```

If you include the **`where`** command after the **`timechart`** command, you will not see results. Once the data is transformed by the **`timechart`** command the only fields available to you are **`_time`** and **`Risk`** and the values produced by the **`count(bcg_workstation)`** calculation. Therefore, you must add the **`where`** command before the **`timechart`** command to filter results.

_time	2 Low	3 Medium	4 High	5 Critical
2021-08-25	0	0	0	0
2021-08-26	0	0	0	0
2021-08-27	13	0	0	5
2021-08-28	0	1	0	1
2021-08-29	0	0	3	4
2021-08-30	5	0	3	1
2021-08-31	0	0	0	6
2021-09-01	0	0	2	25

10. Select **Visualization** and select a **Line Chart**.



11. Save your search as a report with the name **L2S2**.

Challenge: The sales department would like to find out the results of product sales during the previous 7 days as a trellis visualization. Use `stats` command with an `eval` expression to filter results.

12. Search the web server data for events involving an action (**`index=web sourcetype=access_combined action=*`**) over the **Last 7 days**.

```
index=web sourcetype=access_combined action=*
```

13. Use the **`stats`** command to count events by **`action`**.

```
index=web sourcetype=access_combined action=*
| stats count by action
```

action	count
addtocart	4214
changequantity	948
purchase	4323
remove	944
view	3910

14. Edit the **`stats`** command so that the following **`action`** values are counted and renamed. (**Hint**: Use the **`count`** function with multiple **`eval`** expressions to count these values and use **`as`** clauses to rename the fields. Review the examples provided in "Filtering with where" topic of the instruction document, the "Where Command" video module (eLearning), or the Search Manual for additional assistance.)

a. **changequantity** as "Changed"
b. **remove** as "Removed"
c. **purchase** as "Purchased"
d. **view** as "Viewed"

Include a **by** clause so that the results of the **stats** command are split by **product_name**.

```
index=web sourcetype=access_combined
| stats count(eval(action="changequantity")) as Changed, count(eval(action="remove"))
  as Removed, count(eval(action="purchase")) as Purchased, count(eval(action="view"))
  as Viewed by product_name
```

product_name	Changed	Removed	Purchased	Viewed
Benign Space Debris	36	35	128	177
Curling 2014	40	31	117	179
Dream Crusher	73	90	180	253
Final Sequel	56	52	169	245
Fire Resistance Suit of Provolone	66	61	181	229
Holy Blade of Gouda	55	55	145	223
Manganiello Bros.	51	71	153	262

15. Rename **product_name** as "Product."

```
index=web sourcetype=access_combined
| stats count(eval(action="changequantity")) as Changed, count(eval(action="remove"))
  as Removed, count(eval(action="purchase")) as Purchased, count(eval(action="view"))
  as Viewed by product_name
| rename product_name as Product
```

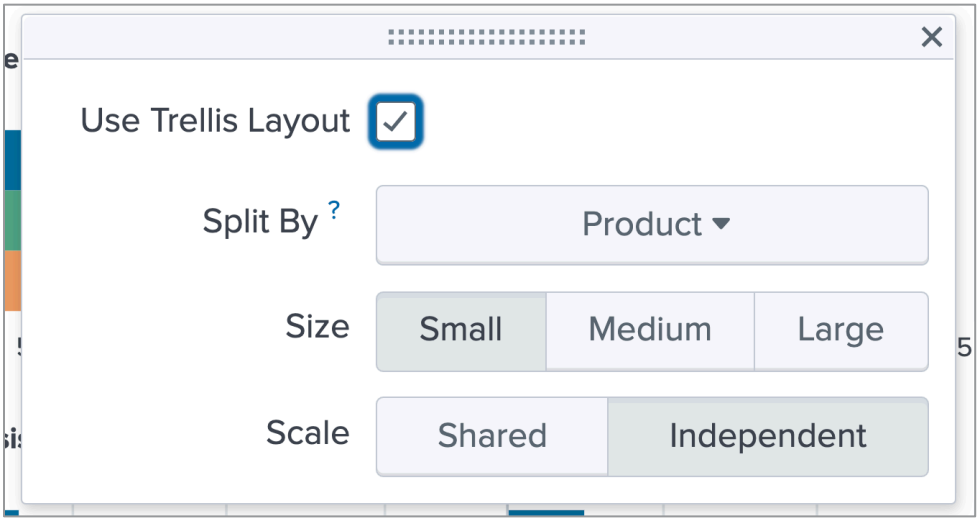
Product	Changed	Removed	Purchased	Viewed
Benign Space Debris	36	35	128	177
Curling 2014	40	31	117	179
Dream Crusher	73	90	180	253
Final Sequel	56	52	169	245
Fire Resistance Suit of Provolone	66	61	181	229
Holy Blade of Gouda	55	55	145	223
Manganiello Bros.	51	71	153	262

16. Navigate to the **Visualization** tab and select **Bar Chart**.
17. Select **Trellis** and configure the following settings:
    a. Check **Use Trellis Layout**

b. **Split** By: choose **Product**

c. **Size**: choose **Small**

d. **Scale**: choose **Independent**



18. Save your search as a report with the name **L2X**.