Searching and Reporting with Splunk 5.0 class labs

Lab typographical conventions

`{student ID}` indicates you should replace this with your student number.

`{server-name}` indicates you should substitute the server name assigned to this class.

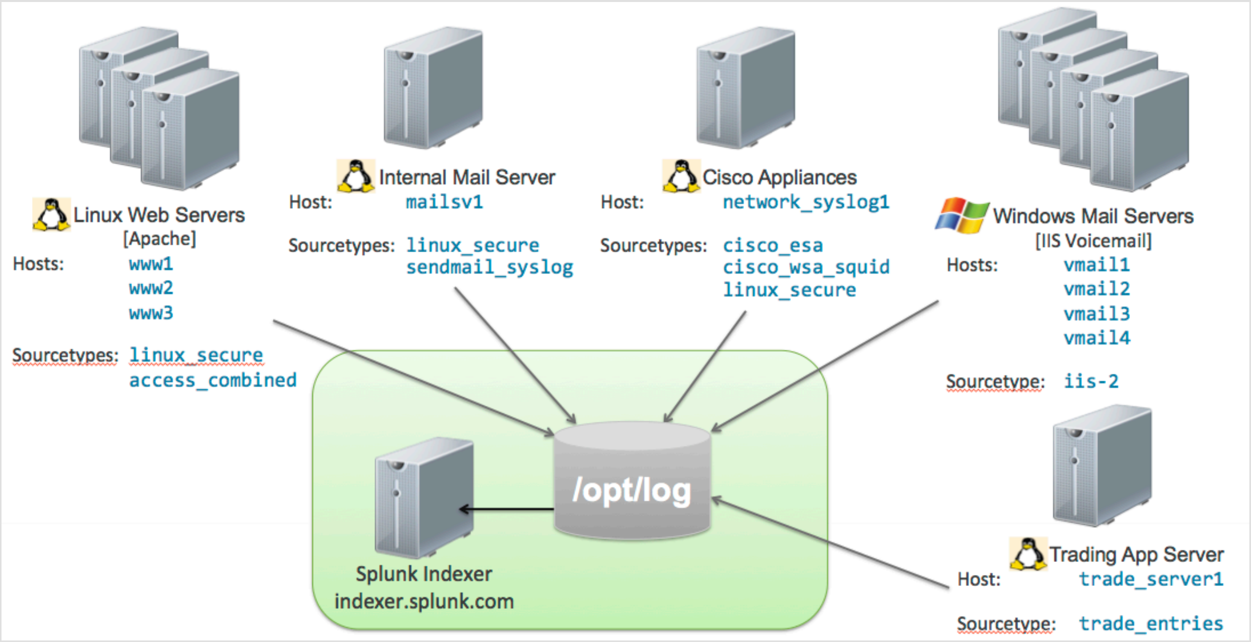
There are three source types used in the labs. The lab instructions refer to these source types by the types of data they represent. The data types are as follows:

Apache log data – `access_*` or `access_combined`

Firewall data – `cisco_w*` or `cisco_wsa_squid`

Email data – `cisco_e*` or `cisco_esa`

Training Lab Environment



Lab 1 – Fields Overview

Description

This is a short lab to familiarize you with the data used in this course.

Steps

Task: Log into Splunk on classroom server.

1. Direct your web browser to the class lab system (for example, `http://{server-name}.splunk.com`)
2. Log in with the credentials your instructor assigned.
3. Take a minute to examine the data sources on the Summary page.

Task: Change your account time zone setting to reflect your local time.

4. Click your login name next to the App menu.
5. Select your local time zone from the **Time zone** menu, and then click **Save**.
6. Return to the Search app.

Task: Perform basic searches on the apache log data and familiarize yourself with the table command.

7. Search for all events with the `access_*` source type over the **last 24 hours**.
8. Take a few moments to examine the fields that were automatically extracted.
9. Create a **table** that includes the `clientip` and `action` fields.

Results Example:

	clientip	action
1	192.1.2.40	addtocart
2	192.1.2.40	remove
3	67.230.133	purchase
4	104.255.109.201	
5	…	…

10. Modify your search to return only events where `action=purchase`.
11. Alter the **table** to display the `clientip` and **`status`** fields.
12. Rename the `clientip` field to `customer`.

Results Example:

	customer	status
1	192.1.2.40	200
2	192.1.2.40	200
3	67.230.133	404
4	…	…

13. To clear the previous search, click search in the App navigation bar.

Task: Perform basic searches on the firewall data.

14. Search for all events in the **last 24 hours** for the `cisco_w*` source type (firewall data).

15. Take a few moments to examine the fields that were automatically extracted.
16. Create a **table** that displays the `cs_username` and `usage` fields.

Results Example:

	cs_username	**usage**
1	grumpy@demo.com	Business
2	grumpy@demo.com	Personal
3	grumpy@demo.com	Business
4	…	…

17. To clear the previous search, click search in the App navigation bar.

****CHALLENGE LAB**

Use the rex command to extract a field called `threat` in the email data and then display the top threats.

18. Search for all events in the **Last 7 days** for the `cisco_esa` source type (email data).
19. Take a few moments to examine the fields that were automatically extracted.
20. Search for the term `OUTBREAK_*`.
21. Add the `rex` command to extract a new field called `threat` for the threat information.
22. Add the `top` command to display the top values of the `threat` field.

Results Example:

	threat	**count**	**percent**
1	OUTBREAK_0002499 has threat level 3	91	2.199662
2	OUTBREAK_0002476 has threat level 3	91	2.199662
3	…	…	…

Lab 2 – Basic Statistics

Description

This lab reinforces the commands you learned for basic statistics.

Steps

Task: Report on top and rare values.

1. Search the `sourcetype=access_*` for all events in the **last 24 hours** where the `referer_domain` is **not** `*myflowershop*`.
2. Use the `top` command to display the **top three** “referer” domains.

3. Add the `fields` command to remove the `percent` field from the results.

Results Example:

	referer_domain	count
1	http://www.google.com	2842
2	http://www.yahoo.com	154
3	http://www.bing.com	147
4	…	…

4. Enter a new search `sourcetype=access_combined` for the **top** status codes for each host over the **last 24 hours**.
   Hint: Use the fields `status` and `host`.
5. Add the `sort` command to sort by the `count` field in descending order.

Results Example:

	host	status	count	percent
1	www2	200	907	77.987962
2	www1	200	900	78.809107
3	www3	400	774	8.168530
4	…	…	…	…

6. Enter a new search `sourcetype=cisco_w*` for all events in the **last 24 hours**.
7. Use the `top` command to display the top usage types, grouped by user.
   Hint: Use the `usage` and `cs_username` fields.
8. Add the `sort` command to sort by the `count` field in descending order.
9. Rename the `cs_username` field to `User Name`.

Results Example:

	User Name	usage	count	percent
1	grumpy@demo.com	Personal	5189	57.191668
2	happy@demo.com	Personal	4590	66.919376
3	doc@demo.com	Unknown	3926	58.188825
4	…	…	…	…

10. Using the same source type, find the five most `rare` mime types.
    Hint: Use the field `cs_mime_type`.

Results Example:

	cs_mime_type	count	percent
1	application/x-elc	1	0.003685
2	audio/mpeg	1	0.003685
3	audio/x-ms-wma	1	0.003685
4	…	…	…

Task: Use the stats command and associated functions.

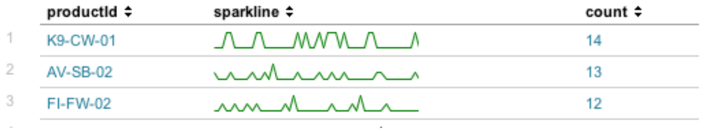
11. Enter a new search `sourcetype=access_*` for purchase events in the **last 24 hours**.
    Hint: `action=purchase`
12. Use the `stats` command to `count` the events by `productId`.
    Hint: Field names are case sensitive.
13. Add the `sort` command to sort by the `count` field in descending order.

Results Example:

	productId	count
1	AV-CB-01	14
2	AV-SB-02	13
3	…	…

14. Add the `sparkline` function to the `stats` command to display the trend in the table.

Results Example:



	productId ⬍	sparkline ⬍	count ⬍
1	K9-CW-01		14
2	AV-SB-02		13
3	FI-FW-02		12

15. Enter a new search `sourcetype=access_*` for the **last 24 hours**.
16. Use the `stats` command to determine a **distinct count** of `JSESSIONID`s for each `host`.

Results Example:

	host	dc(JSESSIONID)
1	www1	464
2	www2	557
3	www3	488

17. Alter the `stats` command to create a search that calculates a `sum` of `bytes` being served for each `file`.

Results Example:

	file	sum(bytes)
1	cart.do	951390
2	category.screen	976233
3	product.screen	827834
4	…	…

18. Modify the search to compute an **average** instead of a sum.

Results Example:

	file	avg(bytes)
1	cart.do	2111.488069
2	category.screen	2160.552463
3	product.screen	2097.279805
4	…	…

19. Enter a new search `sourcetype=cisco_w*` for events that include the term `BLOCK_*` in the **last 24 hours**.
20. Use the `stats` command to list the unique `values` of the `x_webroot_threat_name` field within the results.

Results Example:

	values(x_webroot_threat_name)
1	"Trojan-Backdoor-Zbot" "Trojan-Downloader-Suurch" "Trojan-Downloader.Gen" "Virus-Otwycal" "zhongsou zztoolbar" -

Task: Add the search you just created to a dashboard.

21. From the **Create** menu, select **Dashboard panel…**
22. Name the search **{student ID} Virus threats - last 24 hours**.
23. Click **Next**.
24. Create a new dashboard and name it **{student ID} - Operations**.
25. Verify the dashboard is shared with all users of the current app, then click **Next**.
26. Keep the default **Panel title**.
27. Verify that **Table** visualization is selected.
28. Select **Run search each time dashboard loads**.
    You learn about report acceleration in Module 7. For now, leave this unchecked.
29. Click **Finish**.
30. To view the dashboard, click the link in the confirmation dialog.
31. When you are finished viewing the dashboard, return to the **Search** view.

Lab 3 – Calculating and Formatting

Description

This lab reinforces the `eval`, `fieldformat`, and `where` commands.

Steps

Task: Use the eval command to convert field values.

1. Enter a new search `sourcetype=cisco_wsa*` for all events in the **last 24 hours**.
2. Use the `stats` command to calculate a **sum** of sc_bytes renamed `totalBytes` grouped by `cs_username` and rename the `cs_username` field to `User Name`.
   Hint: Use the `sc_bytes` field.

Results Example:

	User Name	totalBytes
1	grumpy@demo.com	2272853
2	bashful@demo.com	175084
3	doc@demo.com	185035786
4	…	…

3. Add the `eval` command to set a new field called `MB`. Divide the `totalBytes` field by /1024/1024 to populate the `MB` field. If there are not enough events, change the time span to 7 days.
   Hint: The format is `eval <new field> = (<field>/1024/1024)`

Results Example:

	User Name	totalBytes	MB
1	grumpy@demo.com	591434	0.564035
2	bashful@demo.com	1153845	1.100392
3	doc@demo.com	3580492	3.414623
4	…	…	…

Task: Round field values.

4. Using the search you just created, modify the `eval` command to round the field value for the `MB` field to two decimal points.

Results Example:

	User Name	totalBytes	MB
1	grumpy@demo.com	978198	0.93
2	bashful@demo.com	167052	0.16
3	doc@demo.com	2870207	2.74
4	…	…	…

5. Add the report to the **Operations** dashboard. From the **Create** menu, select **Dashboard panel**….
6. Name the search **{student ID} MB Per User - Last 24 hours**.
7. Click **Next**.
8. Select **Existing dashboard**. Choose your **Operations** dashboard from the list.
9. Click **Next**.
10. Set the schedule to run **every day at midnight**, then click **Finish**.
11. View the dashboard, then return to the Search view.

Task: Format field values.

In this task, you use the `fieldformat` command to make your data values more user-friendly.

12. Enter a new search the `access_*` source type for events with a `status` of `503` in the **last 24 hours**.
13. Calculate a sum of the price field and use an `as` clause to place the value in a field called `lost_revenue`.
14. Apply the `fieldformat` command to the `lost_revenue` field to prepend the value with a dollar sign ($) and apply commas appropriately using the `tostring` function.

Results Example:

	lost_revenue
1	$230,227

Task: Create a sales dashboard and add a panel.

15. Using the report you just created, create a new dashboard panel.
16. Name the search **{student ID} Lost Revenue - last 24 hours**.
17. Add the panel to a new dashboard named **{student ID} Sales Dashboard**.
18. Share the dashboard with all users of current app, and then click **Next**.
19. Select the **Single value** visualization and have the search run each time the dashboard loads.
20. Finish and view the new dashboard, then return to the Search view.

Task: Use conditional statements.

21. Enter a new search the `access_*` source type for all events in the **last 24 hours**.
22. Use the `eval` command to create a new field called `reqPerformance`. Use the `if` function to group all events with `status="200"` into a value called `"ok"`, and all other events into a value called `"failed"`.
    Hint: You must include the quotes around "ok" and "failed"
23. Add the `stats` command to get a `count` by `reqPerformance`.

Results Example:

	reqPerformance	count
1	ok	712
2	failed	2566

Task: Filter results with the where command.

24. Run the saved search you created earlier called **{student ID} MB Per User - Last 24 Hours**
25. Add the `where` command to only display results if the value of the `MB` field is greater than `.2`

Results Example:

	cs_username	totalBytes	MB
1	doc@demo.com	1324219	1
2	bashful@demo.com	1153845	1
3	grumpy@demo.com	3580492	3
4	…	…	…

Lab 4 – Charting

Description

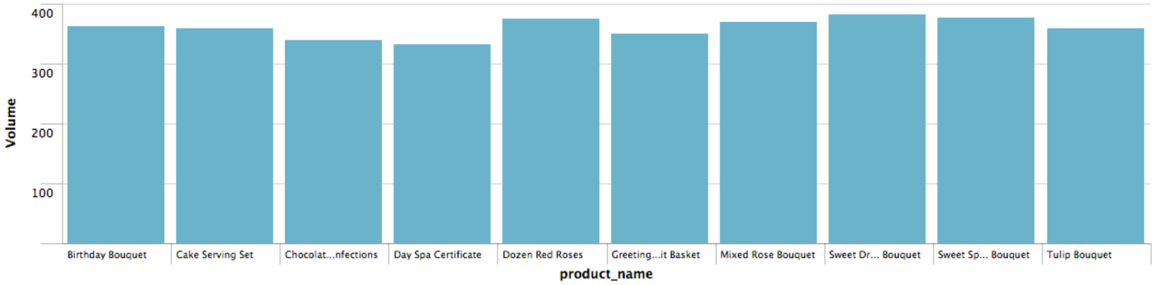
Create charts and time charts.

Steps

Task: Create a basic column chart.

1. Enter a new search for `purchase` actions in the `access_combined` source type in the **last 24 hours**.
2. Use the `chart` command to display a `count` of events by `product_name`.
3. Switch to the **Results Chart** view.
4. Click formatting options and be sure that the **Chart type** is set to **column**.
5. To display and configure options for the Y-axis, click the **Y-axis** link and label it **Volume**.

Chart Example:

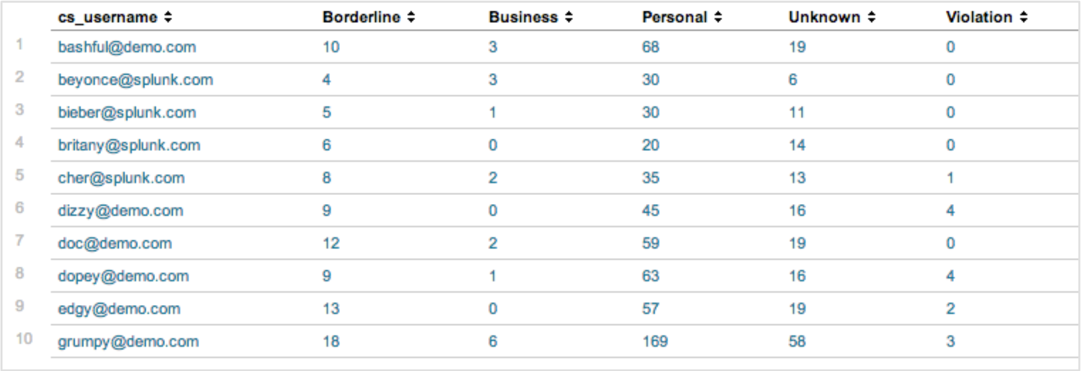


6. Add the chart to your **{student ID} Sales Dashboard** and name the search **{student ID} Daily Product Volume**.
7. View your dashboard.

Task: Create a multi-series chart and work with formatting options.

8. Return to the **Search** view and create a search for `sourcetype=cisco_w*` that displays each user's Internet usage types in the **last 24 hours**.
9. Use the `chart` command to initially count events by `cs_username`, then by `usage`.

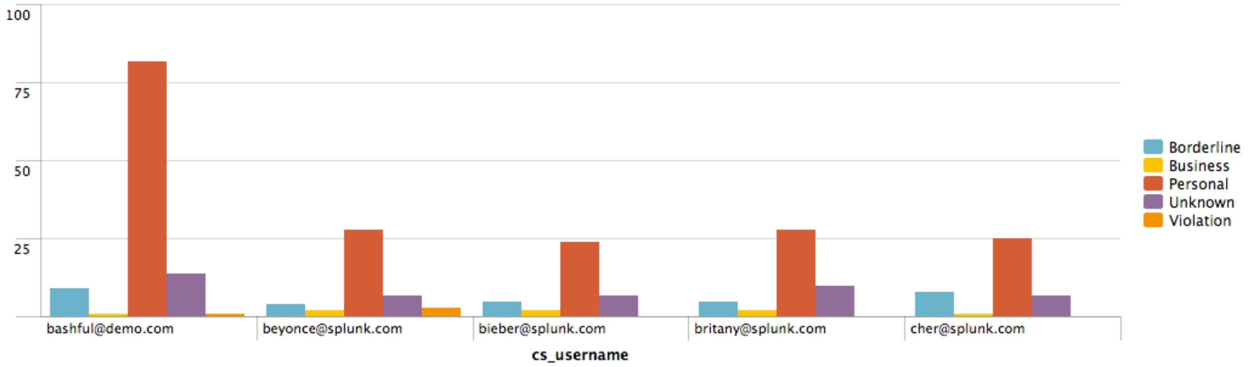
Table Example:



	cs_username ⇕	Borderline ⇕	Business ⇕	Personal ⇕	Unknown ⇕	Violation ⇕
1	bashful@demo.com	10	3	68	19	0
2	beyonce@splunk.com	4	3	30	6	0
3	bieber@splunk.com	5	1	30	11	0
4	britany@splunk.com	6	0	20	14	0
5	cher@splunk.com	8	2	35	13	1
6	dizzy@demo.com	9	0	45	16	4
7	doc@demo.com	12	2	59	19	0
8	dopey@demo.com	9	1	63	16	4
9	edgy@demo.com	13	0	57	19	2
10	grumpy@demo.com	18	6	169	58	3

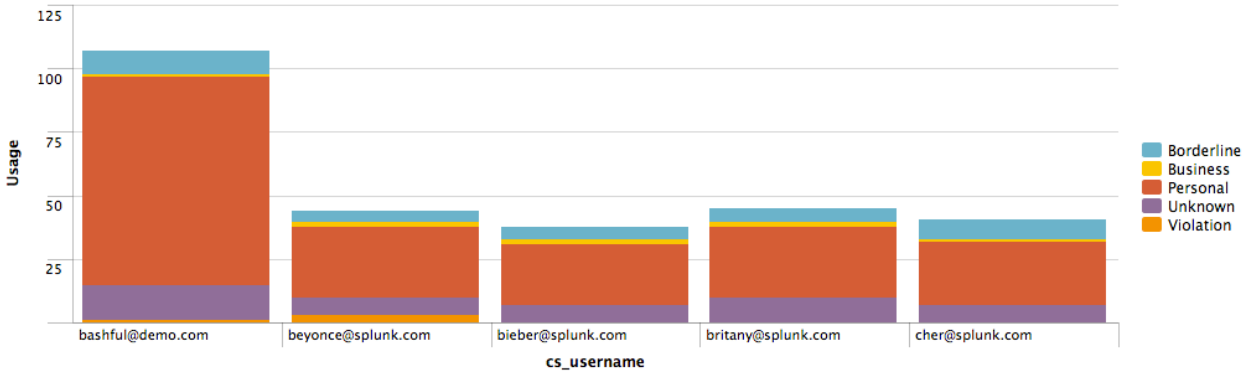
10. Switch to the **Results Chart** view.
    Limit the number of values to plot on the X-axis by piping to: `sort 5 cs_username`

    Chart Example:



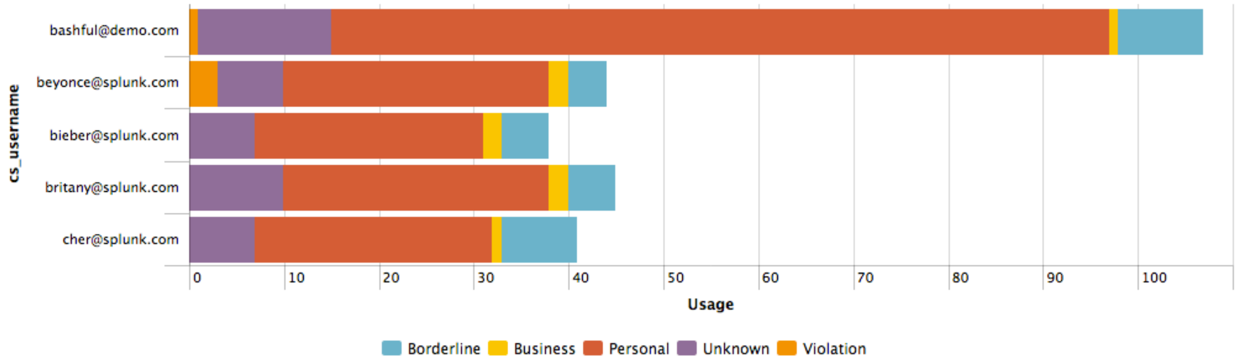
11. Change the **Stack Mode** to **Stacked**.
12. Label the **Y-axis: Usage**

    Stacked Chart Example:



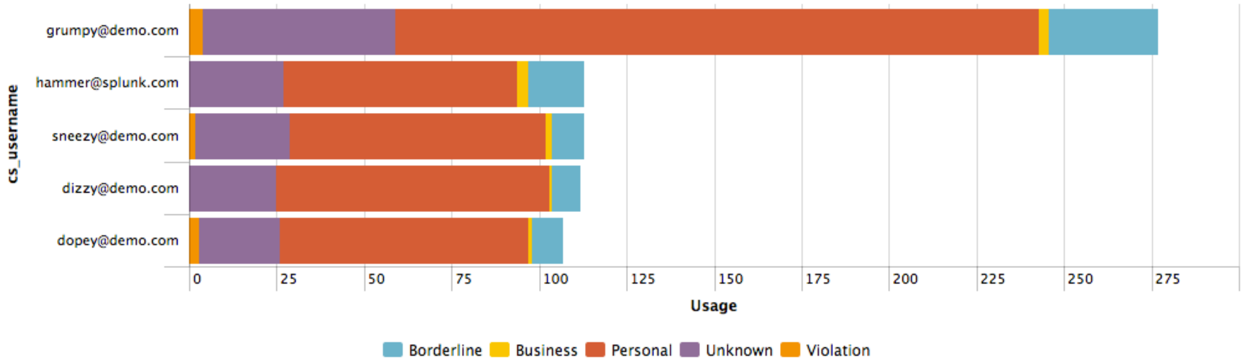
13. Return to the **General** options.
14. Change the **Chart type** to **bar**.
15. For **Legend placement**, select **Bottom**.

    Bar Chart Example:



16. Optional challenge: Modify the search to display the five most active users in descending order.
    Hint: Use the `addtotals` command.

    Challenge Example:

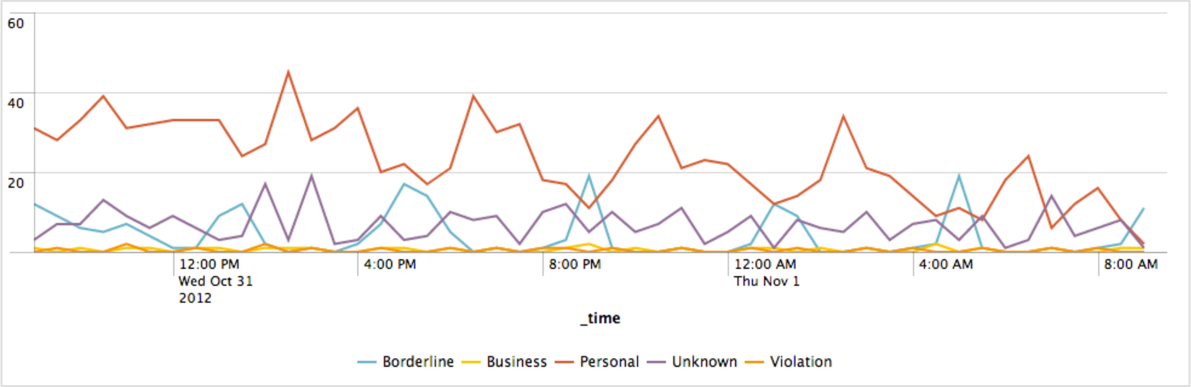


17. Add your chart to the **{student ID} Operations** dashboard and name the search **{student ID} Internet Usage by User**
18. View the dashboard.

Task: Create a basic time chart.

19. Return to the **Search** view and create a timechart for `sourcetype=cisco_w*` that displays a count of Internet usage types over time for the **last 24 hours**.
20. Switch to **Results Chart** view and set the **Chart type** to **line**, **Multi-series mode** to **combined**, and **Missing values** to **omit**.

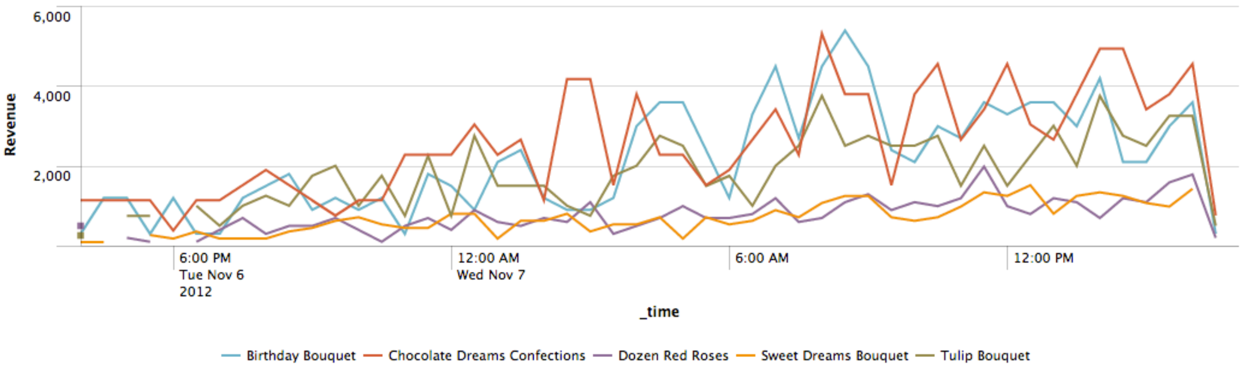
    Chart Example:



21. Create a timechart that plots purchases for `sourcetype=access_combined` by calculating a sum of the `price` field split by `product_name` for the **last 7 days**.
22. Limit the number of products to five and eliminate the Other plot line from the chart.
    sourcetype=access_* action=purchase | timechart sum(price) by product_name limit=5 useother=f

23. Rename the **Y-axis** to **Revenue**.

Chart Example:



24. Add the chart to your **{student ID} Sales Dashboard** and name the search **{student ID} Daily Product Sales**.

Lab 5 – Correlating Events

Description

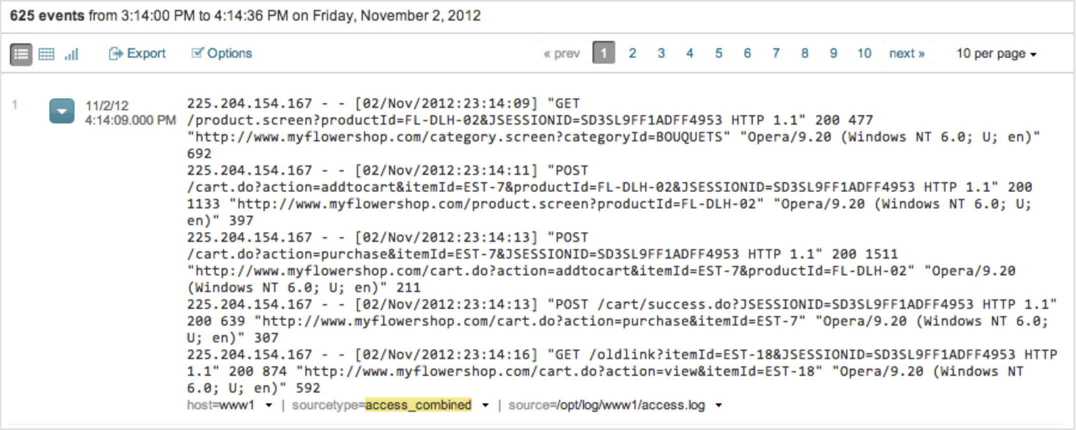
Use the transaction command to correlate events.

Steps

Task: Create a transaction using a common field.

1. Return to **Search**.
2. Enter a new search for all flower shop events (`sourcetype=access_combined`) in the **Last 60 minutes**. Note the number of events.
3. Add the `transaction` command to the search and use the Java session ID field, `JSESSIONID,` to create the transactions. Note the number of events.
   Now, to view only transactions that contain at least one purchase event, add the `search` command to search within the transactions for `action=purchase`.

Chart Example:



625 events from 3:14:00 PM to 4:14:36 PM on Friday, November 2, 2012

1 11/2/12 4:14:09.000 PM

```
225.204.154.167 - - [02/Nov/2012:23:14:09] "GET
/product.screen?productId=FL-DLH-02&JSESSIONID=SD3SL9FF1ADFF4953 HTTP 1.1" 200 477
"http://www.myflowershop.com/category.screen?categoryId=BOUQUETS" "Opera/9.20 (Windows NT 6.0; U; en)"
692
225.204.154.167 - - [02/Nov/2012:23:14:11] "POST
/cart.do?action=addtocart&itemId=EST-7&productId=FL-DLH-02&JSESSIONID=SD3SL9FF1ADFF4953 HTTP 1.1" 200
1133 "http://www.myflowershop.com/product.screen?productId=FL-DLH-02" "Opera/9.20 (Windows NT 6.0; U;
en)" 397
225.204.154.167 - - [02/Nov/2012:23:14:13] "POST
/cart.do?action=purchase&itemId=EST-7&JSESSIONID=SD3SL9FF1ADFF4953 HTTP 1.1" 200 1511
"http://www.myflowershop.com/cart.do?action=addtocart&itemId=EST-7&productId=FL-DLH-02" "Opera/9.20
(Windows NT 6.0; U; en)" 211
225.204.154.167 - - [02/Nov/2012:23:14:13] "POST /cart/success.do?JSESSIONID=SD3SL9FF1ADFF4953 HTTP 1.1"
200 639 "http://www.myflowershop.com/cart.do?action=purchase&itemId=EST-7" "Opera/9.20 (Windows NT 6.0;
U; en)" 307
225.204.154.167 - - [02/Nov/2012:23:14:16] "GET /oldlink?itemId=EST-18&JSESSIONID=SD3SL9FF1ADFF4953 HTTP
1.1" 200 874 "http://www.myflowershop.com/cart.do?action=view&itemId=EST-18" "Opera/9.20 (Windows NT
6.0; U; en)" 592
```

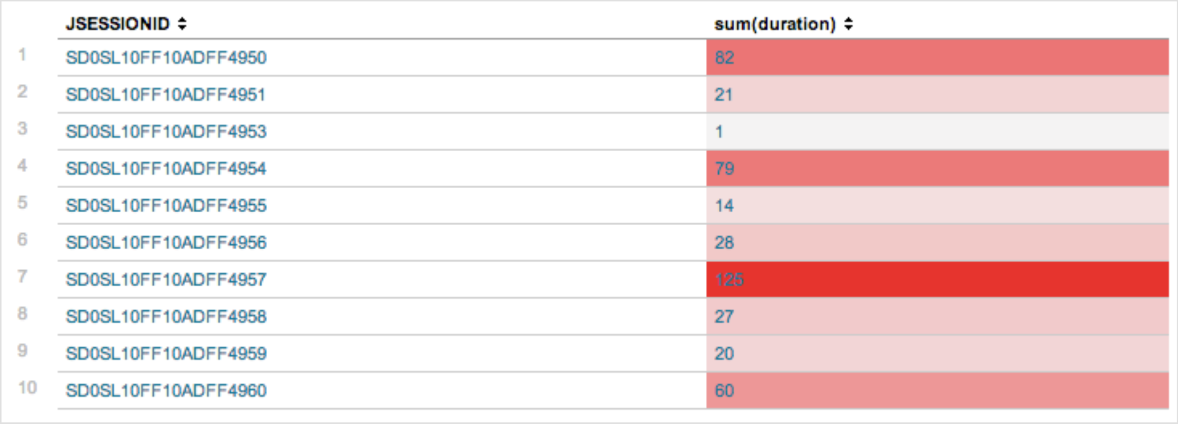
host=www1 | sourcetype=access_combined | source=/opt/log/www1/access.log

Task: Create a transaction using common fields values, maxspan, and maxpause.

4. Enter a new search `sourcetype=access_*` for the **Last 24 hours**.
5. Create a `transaction` based on the `JSESSIONID` field with a max span of 10 minutes and max pause of 2 minutes.
6. Add the `stats` command to get a `sum` of the `duration` for each `JSESSIONID`.

7. To easily view potential problem areas, change the **Overlay** to **Heat map**.

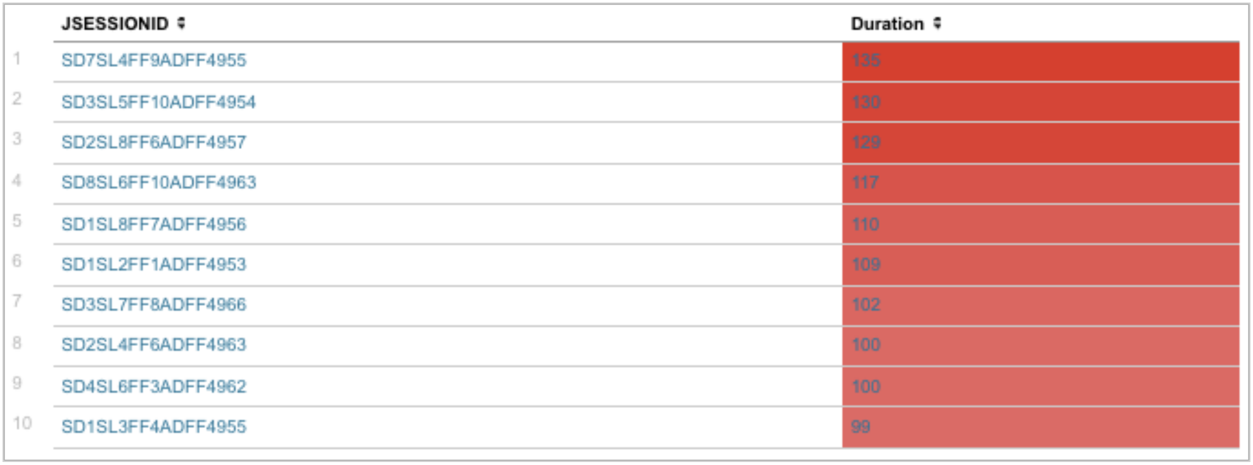
Chart Example:



	JSESSIONID	sum(duration)
1	SD0SL10FF10ADFF4950	82
2	SD0SL10FF10ADFF4951	21
3	SD0SL10FF10ADFF4953	1
4	SD0SL10FF10ADFF4954	79
5	SD0SL10FF10ADFF4955	14
6	SD0SL10FF10ADFF4956	28
7	SD0SL10FF10ADFF4957	125
8	SD0SL10FF10ADFF4958	27
9	SD0SL10FF10ADFF4959	20
10	SD0SL10FF10ADFF4960	60

8. Optional challenge: Rename the `sum(duration)` column to `Duration` and sort the output by the ten longest sessions in descending order.

Challenge Example:



	JSESSIONID	Duration
1	SD7SL4FF9ADFF4955	135
2	SD3SL5FF10ADFF4954	130
3	SD2SL8FF6ADFF4957	129
4	SD8SL6FF10ADFF4963	117
5	SD1SL8FF7ADFF4956	110
6	SD1SL2FF1ADFF4953	109
7	SD3SL7FF8ADFF4966	102
8	SD2SL4FF6ADFF4963	100
9	SD4SL6FF3ADFF4962	100
10	SD1SL3FF4ADFF4955	99

Lab 6 – Creating and Using Lookups and Workflows

Description

Create and use a new lookup that will identify a browser, version, and OS based on the `useragent` field in the store data. You then create a workflow action to perform a Whois lookup of an IP address.

Steps

Task: Add a lookup table file.

1. Save the file `browser_lookup.csv` to your computer. (Provided by your instructor)
2. Navigate to **Manager > Lookups > Lookup table files**.
3. Click **New**.
4. Verify the **Destination app** is **Search**.
5. Click **Browse** and navigate to and select the `browser_lookup.csv` file you saved in Step 1.
6. In the **Destination filename** field, type: **browser_lookup.csv**
7. Click **Save**.
   Note: A message indicating success appears below the Splunk logo.

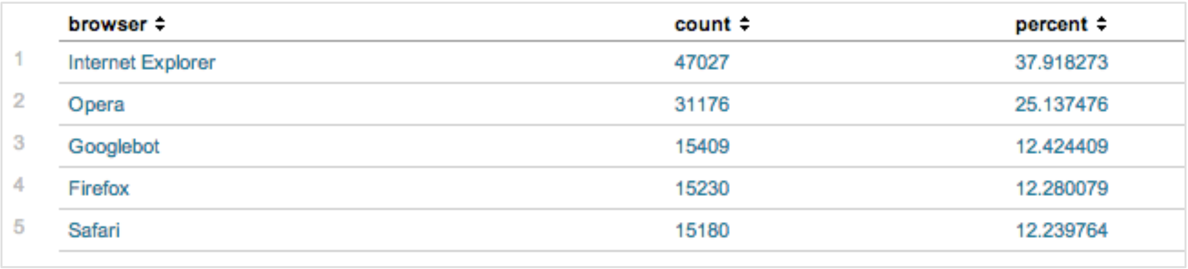
Task: Create a lookup definition.

8. Navigate to **Manager > Lookups > Lookup definitions**.
9. Click **New**.
10. Verify the **Destination app** is **Search**.
11. In the **Name** field, type: **browser_lookup**
12. Verify the **Type** is **File-based**.
13. If not already selected, from the **Lookup file** menu, select **browser_lookup.csv**.
14. Click **Save**.
15. Verify the lookup table data by returning to the Search view and searching:
    ```
    | inputlookup browser_lookup
    ```
    Note: You may have to clear the Heat map overlay from the previous exercise.

Task: Use the lookup in a report.

16. Enter a new search for all events in `sourcetype=access_*` for the **last 24 hours**.
17. Pipe to the `lookup` command to call the `browser_lookup` and reference the `useragent` field as the input field. `OUTPUT` the `browser`, `version`, and `os` fields.
    Note the new fields are now available in the field picker.
18. Add the `top` command to display the top browsers.

Results Example:



	browser ⇕	count ⇕	percent ⇕
1	Internet Explorer	47027	37.918273
2	Opera	31176	25.137476
3	Googlebot	15409	12.424409
4	Firefox	15230	12.280079
5	Safari	15180	12.239764

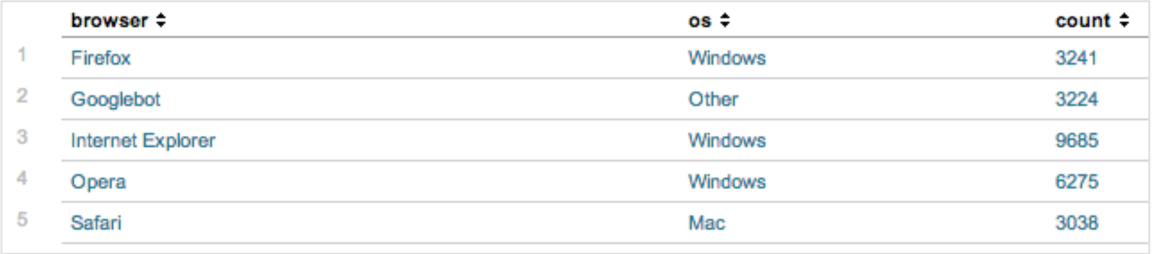
Task: Configure the lookup to run automatically so that the lookup fields are always returned in the events.

19. Navigate to **Manager > Lookups > Automatic lookups**.
20. Click **New**.
21. Verify the **Destination app** is **Search**.
22. In the **Name** field, type: **auto_browser_lookup**
23. From the **Lookup table** menu, select **browser_lookup**.
24. Verify that **sourcetype** is selected in the **Apply to** menu.
25. In the **named** field, type: **access_combined**
26. In the **Lookup input fields**, type **useragent** in the left field.
27. In the **Lookup output fields**, type **browser** in the left field.
28. Click **Add another field**.
29. Type **version** in left field.
30. Click **Add another field**.
31. Type **os** in the left field.
32. Check the **Overwrite field values** checkbox.
33. Click **Save**.

Task: Use the automatic lookup.

34. Return to **Search**.
35. Enter a new search `sourcetype=access_*` for all events in the **last 24 hours**.
36. In the Fields sidebar, view all fields. Notice that `browser`, `os`, and `version` fields are now automatically extracted.
37. Use the `stats` command to create a report that displays a count for each `browser` / `os` combination.

    Results Example:



	browser ⇕	os ⇕	count ⇕
1	Firefox	Windows	3241
2	Googlebot	Other	3224
3	Internet Explorer	Windows	9685
4	Opera	Windows	6275
5	Safari	Mac	3038

Task: Create a workflow action to do a Whois lookup for the source IP field.

38. Navigate to **Manager > Fields > Workflow actions**.
39. Verify the **Destination app** is **search**.
40. Create a new workflow action for the Search app and name it **{student ID}_whois**.
41. For **Label**, type **Whois lookup for: src_ip** and apply it only to the `src_ip` field.
42. Show the action in the **Event menu** and make it a **link** action type.
43. In the **URI*** field, type: `http://www.whois.net/ip-address-lookup/$src_ip$`
44. For **Open link in**, select **New window**.
45. For **Link method**, select **get** and click **Save**.
46. The Workflow action you created should appear in the list.
    Note: By default, workflow actions are private. You did not have an option to make it public while you were creating it. You can only do so from the Manager's Workflow action page.

Task: Use a workflow action to do a Whois lookup for the source IP field.

47. Return to Search and look for events over the **Last 4 hours** where source IP is not blank.
    Hint: src_ip=* or src_ip!=" "

48. Choose any event and from its Event menu, select `Whois lookup for: {source IP value}`.
    Note: Using `$src_ip$` automatically inserted the value from the event's `src_ip` field.
49. The results of the Whois search for the IP address displays in another browser tab.

Lab 7 – Report Acceleration

Description

Create, manage, and use accelerated reports.

Steps

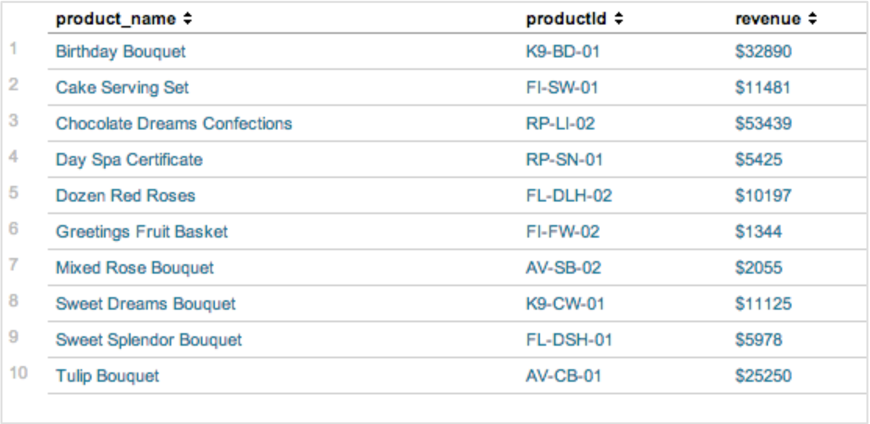
Task: Create a report that cannot be accelerated.

1. Enter a new search for the `access_combined` source type for successful **purchases** over the **last 30 days**.
   Hint: `action=purchase`
2. From the **Save** menu, select **Save Search**.
   Does the **Accelerate this search** checkbox appear? _______ Why or why not?
   ____
   Keep the search private and save it as: **{student ID} Can't accelerate search**
3. Navigate to **Manager > Searches and reports**.
4. Open the search you just saved.
5. Check the **Accelerate this search** checkbox.
6. For **Summary range**, select **1 Month** and click **Save**.
   Notice the "This search cannot be accelerated" error message. Even though the **Accelerate this search** checkbox appears in the form, Splunk validates the search string during the save operation.
7. Click **Cancel** and return to the Search view.

Task: Create an accelerated report.

8. Run your **{student ID} Can't accelerate search** search.
9. Add the necessary reporting commands to calculate the sum of the `price` field split by `product_name` and `productId`. Rename the calculated field **revenue** and pipe it to the `fieldformat` command to display values with prepended dollar signs as shown below.

Results Example:



	product_name ⇕	productId ⇕	revenue ⇕
1	Birthday Bouquet	K9-BD-01	$32890
2	Cake Serving Set	FI-SW-01	$11481
3	Chocolate Dreams Confections	RP-LI-02	$53439
4	Day Spa Certificate	RP-SN-01	$5425
5	Dozen Red Roses	FL-DLH-02	$10197
6	Greetings Fruit Basket	FI-FW-02	$1344
7	Mixed Rose Bouquet	AV-SB-02	$2055
8	Sweet Dreams Bouquet	K9-CW-01	$11125
9	Sweet Splendor Bouquet	FL-DSH-01	$5978
10	Tulip Bouquet	AV-CB-01	$25250

10. Access the Save Search dialog.
11. Name the search **{student ID} Accelerated search**.
12. Keep the search private.
    Can you accelerate this search? _______ Why or why not? ____
13. Check the **Accelerate this search** checkbox.
14. For **Summary Range**, select **1 month** and click **Finish**.

Task: Create a non-accelerated report.

15. Re-run your **{student ID} Accelerated search** search.
16. Access the Save Search dialog.
17. Name the search **{student ID} NOT accelerated search**.
18. Keep the search private.
19. Do **not** check the **Accelerate this search** checkbox and click **Finish**.

Task: Add an accelerated search to a dashboard panel.

20. From the **Searches & Reports** menu, run your **Accelerated search** search.
21. Select **Create > Dashboard** panel…
22. Name the search: **{student ID} Revenue by Product**
23. Add it to your existing **Sales Dashboard**.
24. On the Panel page, turn on Acceleration with a time span of **1 month** and click **Finish**.
25. Click **OK** to close the dialog.

Task: Access the summary management pages.

26. Navigate to **Manager > Report Acceleration Summaries**.
27. You should see the **Report Acceleration Summaries** page.
    Note your accelerated reports are using the same Summary ID.
28. To view the details of a summary, click a **Summary ID**.
    The Summary Details page displays.

When you are done, the instructor will show you the administrator's view of the Report Acceleration Summaries page.

Lab 8 – Creating and Using Macros

Description

Create and use macros.

Steps

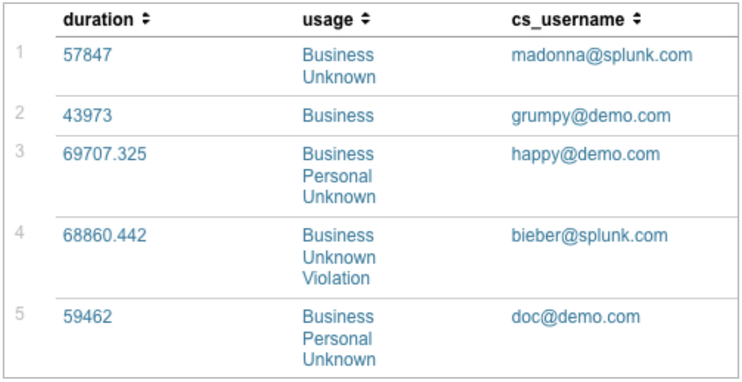
Task: Create a basic macro.

1. Navigate to **Manager > Advanced search > Search macros**.
2. Click **New**.
3. Verify the **Destination app** is set to **Search**.
4. Name the macro **webusage**.
5. In the **Definition** field, type the following search string:
   ```
   sourcetype=cisco_w* | transaction s_hostname, cs_username
   ```
6. **Save** the macro.

Task: Use a basic macro.

7. Return to the **Search** view.
8. In the search bar, type `` `webusage` `` and search over the **Last 24 hours**. Examine the transactions.
9. Add the `where` command. Filter the results to only return transactions where `usage="Business"` and `duration > 0`.
   Hint: Enclose each argument for the where command in parenthesis, and separate with `AND`.
   Hint: You must use quotes when indicating the field/value, i.e., usage="Business"
   Hint: Use the Job inspector to check that your macro is expanding as intended.
10. Add the `table` command to create a report that displays `duration`, `usage` and `cs_username`.

Results Example:



	duration	usage	cs_username
1	57847	Business Unknown	madonna@splunk.com
2	43973	Business	grumpy@demo.com
3	69707.325	Business Personal Unknown	happy@demo.com
4	68860.442	Business Unknown Violation	bieber@splunk.com
5	59462	Business Personal Unknown	doc@demo.com

Task: Create a macro with arguments.

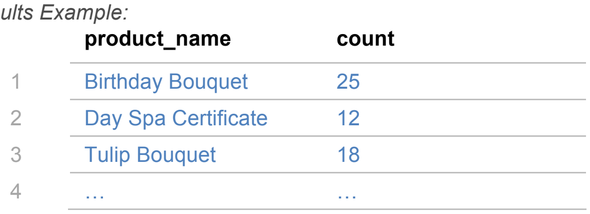
11. Navigate to **Manager > Advanced search > Search macros**.
12. Click **New**.
13. Verify the **Destination app** is set to **Search**.
14. Name the macro: **activityByHost(2)**
15. Enter a search string that searches `sourcetype=access_*` for variable `action` and `host` values.
    Hint: Format is `fieldname=$argument$`
16. Add the `stats` command to get a `count` by `product_name`.
17. In the **Arguments** field, enter the arguments, separated by a comma.
    Hint: argument,argument (no $'s)

18. **Save** the macro.

Task: Use the macro with arguments in a search.

19. Return to the **Search** view.
20. Use the macro and pass the arguments `action=purchase` and `host=www2`.
    Hint: `` `macroname(value,value)` ``
21. Run the search again with the following arguments `remove` and `www1`.

Results Example:



	product_name	count
1	Birthday Bouquet	25
2	Day Spa Certificate	12
3	Tulip Bouquet	18
4	…	…