splunk>

Creating Splunk 6.4 Knowledge Objects Class Lab Exercises with Solutions

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 a number of source types used in these lab exercises. The lab instructions refer to these source types by the types of data they represent:

Type	Sourcetype	Interesting Fields
AD/DNS	winauthentication_security (corporate network)	bcg_ip, bcg_workstation, fname, lname, location, rfid, splunk_role
	WinEventLog:Security (engineering network)	Account_Domain, Account_Name, action, app, Authentication_Package, Type, User
BI server	sales_entries	AcctCode, CustomerID, TransactionID
Email data	cisco_esa	dcid, icid, mailfrom, mailto, mid
Web appliance data	cisco_wsa_squid	action, bandwidth, 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
Online transactions	access_combined	action, bytes, categoryId, clientip, itemId, JSESSIONID, price, productId, product_name, referer, referer_domain, sale_price, status, user, useragent
Retail sales	vendor_sales	AcctID, categoryId, product_name, productId, sale_price, Vendor, VendorCity, VendorCountry, VendorID, VendorStateProvince
Web server	linux_secure	action, app, COMMAND, dest, process, src_city, src_country, src_ip, src_port, user, vendor_action
Windows server logs	win_audit	

****For all exercises, keep the permissions for your knowledge objects private.****

Module 3 Lab Exercise: Creating Lookups

Description

In this lab exercise, you create a new automatic lookup that provides additional information for the vendors selling Buttercup Games products. ****Note: this automatic lookup is required for Lab Exercise 9.**

Steps

Task 1: Log into Splunk on the classroom server.

1. Direct your web browser to the class lab system (for example, `{server-name}.splunk.com`)
2. Log in with the credentials assigned by your instructor.
3. Click on the **Search & Reporting** app. If you are prompted to take a tour, click **Skip**.
4. Take a minute to examine the data sources on the **Data Summary** page.

Task 2: Change your account name and time zone setting to reflect your local time.

5. From the Splunk bar, select your user name located to the left of the Messages menu. Choose **Edit Account**.
6. In the **Full name** field, type your name.
7. From the **Time zone** menu, select your local time zone.
8. From the **Default app** menu, select **search**. Click **Save**.
9. Return to the **Search & Reporting** app.

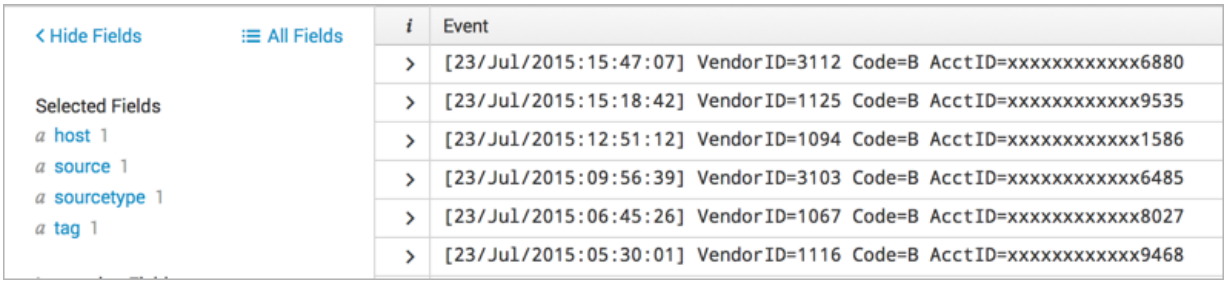
Scenario: The `vendor_sales` source type does not contain vendor locations. Reports need to be created to show how game sales are performing based on region, country, state, and city. A lookup is needed to provide this information when searches are performed.

Task 3: Search retail sales for vendor data.

10. Search the vendor data [`vendor_sales`] over the **last 30 days** specifically for the Dream Crusher product [`product_name="Dream Crusher"`].
    sourcetype=vendor_sales product_name="Dream Crusher"

    Note: As you can see, the raw data has a limited amount of useful and detailed information. For example, the vendor name or city is not listed.

    Results Example:



11. Save your search as report, **{student name}_DreamCrusherSales**.

Task 4: **Add a lookup file.**

12. Navigate to: **Settings > Lookups > Lookup table files**
13. Click **New**.
14. Save the lookup table file with these values:
    Destination app: **search**
    File: **vendor_lookup.csv**
    Destination filename: ***vendor_lookup.csv***

Task 5: **Create a lookup definition.**

15. Navigate to **Settings > Lookups > Lookup definitions**
16. Click **New**.
17. Save the lookup definition with these values:
    Destination app: **search**
    Name: **vendor_lookup**
    Type: **File-based**
    Lookup file: **vendor_lookup.csv**
18. Click **Save**.

Task 6: **Verify the lookup definition.**

19. Return to the Search view.
20. Use `inputlookup` to verify the lookup definition was created correctly.
    | inputlookup vendor_lookup

Results Example:

Vendor	VendorCity	VendorCountry	VendorID	VendorStateProvince
Anchorage Gaming	Anchorage	United States	1001	Alaska
Games of Salt Lake	Salt Lake City	United States	1002	Utah
New Jack Games	New York	United States	1003	New York
Seals Gaming	San Francisco	United States	1004	California
Lost Angels Games	Los Angeles	United States	1005	California
Flyin' Hawaiian Hobbyist	Honolulu	United States	1006	Hawaii
Flyin' Hawaiian Hobbyist	Kahului	United States	1007	Hawaii
Phoenix Games	Phoenix	United States	1008	Arizona
Mile High Games	Denver	United States	1009	Colorado
Beantown Games	Boston	United States	1010	Massachusetts
Seattle Games	Seattle	United States	1011	Washington

Task 7: **Use your lookup in a search. Search for all Dream Crusher games sold by each country in the last 30 days.**

21. Search the vendor data for the **last 30 days** for all Dream Crusher game sales worldwide. Create a table with the total of games sold by country. Use the `lookup` command and reference the file you just created. Use the OUTPUT function to output `VendorCountry`.

    sourcetype=vendor_sales product_name="Dream Crusher" | lookup vendor_lookup VendorID OUTPUT VendorCountry |stats sum(price) as sales by VendorCountry

Results Example:

VendorCountry	sales
United States	9997.50
India	479.88
Canada	439.89
China (PRC)	159.96
Austria	79.98
Bolivia	79.98
Brazil	79.98
Chile	79.98
Ecuador	79.98
Egypt	79.98
Estonia	79.98
Ethiopia	79.98

Task 8: Create an automatic lookup definition.

22. Navigate to **Settings > Lookups > Automatic lookups**
23. Click **New**.
24. Create the automatic lookup with these values:

Destination app:	**search**
Name:	**vendor_auto_lookup**
Lookup table:	**vendor_lookup**
Apply to:	**sourcetype**
named:	**vendor_sales**
Lookup input fields:	**VendorID**
Lookup output fields:	**Vendor**
	VendorCity
	VendorStateProvince
	VendorCountry

25. Click **Save**.

Task 9: Verify your automatic lookup is working.

26. Search the vendor sales data for the total amount of Manganiello Bros. games sold by country in the **last 30 days**. Sort your sales results in descending order.

    sourcetype=vendor_sales product_name="Manganiello Bros." | stats count, sum(price) as sales by VendorCountry | sort -sales

    Now, you will notice that the Vendor, VendorCity, VendorStateProvince, and VendorCountry fields appear in the fields sidebar when you perform a search on vendor_sales data.

Results Example:

VendorCountry	count	sales
United States	945	77055.30
Canada	91	7420.14
Germany	32	2609.28
Italy	29	2364.66
China (PRC)	28	2283.12
India	25	2038.50
France	24	1956.96
United Kingdom	23	1875.42
Spain	19	1549.26
Brazil	18	1467.72
Egypt	15	1223.10
Israel	15	1223.10
Japan	14	1141.56
Poland	14	1141.56
Australia	11	896.94
Belgium	10	815.40
Denmark	10	815.40
Hungary	10	815.40
Ireland	10	815.40
South Africa	9	733.86

Module 4 Lab Exercise: Working with Field Aliases and Calculated Fields

Description

This lab exercise walks you through the process of creating field aliases and calculated fields.

Steps

Scenario: The IT Ops team runs reports for all employee access but the user name field is not consistent across the different source types.

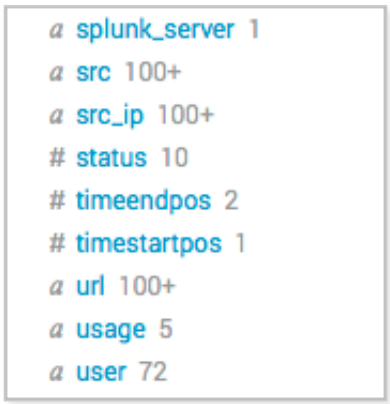
Task 1: Create a field alias to change *cs_username* to *user*.

1. Search for all events in the `cisco_wsa_squid` source type over the **last 24 hours**.

   sourcetype=cisco_wsa_squid
2. Note the `cs_username` field values.
3. Go to **Settings > Fields > Field aliases**. Create a field alias with the following values:

Destination app:	**search**
Name:	**cisco_wsa_squid_aliases**
Apply to:	**sourcetype**
Named:	**cisco_wsa_squid**
Field aliases:	**cs_username = user**
4. Click **Save**.
5. Return to the **Search & Reporting** app. Re-run your search and examine the user field and values.

   Results Example:

6. Perform a search on the `cisco_firewall` sourcetype for the `Username` field for the **last 24 hours.**

   sourcetype=cisco_firewall Username=*
7. Create a field alias for sourcetype `cisco_firewall` with the following values:

Destination app:	**search**
Name:	**cisco_firewall_aliases**
Apply to:	**sourcetype**
Named:	**`cisco_firewall`**
Field aliases:	**Username = user**
8. Perform the following search: `sourcetype=cisco* user=*`
9. Do you receive results from the `cisco_wsa_squid` and `cisco_firewall` sourcetypes?

Scenario: **The IT Ops team is monitoring bandwidth usage for all users for the last 30 days, but the data is reported in bytes. The team needs the usage to be measured in megabytes.**

Task 2: **Create a calculated field that converts bytes to MB.**

10. Search for all events in the **last 30 days** for the `cisco_wsa_squid` sourcetype (web appliance data).
11. Note the `sc_bytes` field. This field displays the amount of bytes used for that event.
12. Go to **Settings > Fields > Calculated fields**.
13. Create a calculated field named **bandwidth** that converts the value of `sc_bytes` to MB with the following values:

    Destination app: **search**

    Apply to: **sourcetype**

    Named: **cisco_wsa_squid**

    Name: **bandwidth**

    Eval expression: `sc_bytes/(1024*1024)`
14. Return to the Search & Reporting app. Perform a search on the `cisco_wsa_squid` sourcetype that shows the total bandwidth by usage.
    sourcetype=cisco_w* | stats sum(bandwidth) as "Bandwidth (MB)" by usage

    Results Example:

usage	Bandwidth (MB)
Borderline	9.133331
Business	14.596395
Personal	78.236745
Unknown	20.043213
Violation	1.063354

Supplemental Exercise:

Scenario: **The IT Ops team wants to correlate data from multiple source types using the `src` and `http_method` fields. However, these fields are called `clientip` and `method` in the `access_combined` source type.**

Task: **Create field aliases for `access_combined` so `src` and `http_method` can be used in searches.**

Module 5 Lab Exercise: Creating Field Extractions

Description

This lab exercise walks you through the process of creating regex and delimiters field extractions. ****Note: this field extraction is required for Lab Exercise 7.**

Steps

Scenario: Access to the Linux server needs to be monitored. However, the IP address and port number are not automatically extracted.

Task 1: Use the FX to extract the IP address and port fields using the Regular Expression method.

1. Search for all events in the **last 24 hours** for the `linux_secure` sourcetype that contain the keyword *port*.
   sourcetype=linux_secure port
2. View the event details to see all the extracted fields. Notice that the IP address and port fields are not extracted.
3. Use the Field Extractor to extract the IP address and port fields. Click the > arrow under the i icon in the first event.
4. Click **Event Actions** > **Extract Fields**.
5. Select the **Regular Expression** method and click **Next**.
6. Highlight the IP address value in the sample event.
7. In the **Field name** box, type **src_ip**.
8. Click **Add Extraction**.
9. Click on the *src_ip* tab and verify the correct information is extracted. Notice that **::** is extracted as a src_ip value. To exclude this from the src_ip field extraction, type the following in the filter field: **src_ip!=::** and click **Apply**.
10. Highlight the port value.
11. In the **Field name** box, type **port**.
12. Click **Add Extraction** and click **Next**.
13. Validate the proper fields are extracted and click **Next**.
14. Review the Extractions Name and click **Finish**.
15. Search for events in the `linux_secure` sourcetype in the **last 24 hours**. List the top ports by IP address.

    sourcetype=linux_secure | top port by src_ip

Results Example:

src_ip	port	count	percent
10.1.10.172	4717	3	0.273224
10.1.10.172	3567	3	0.273224
10.1.10.172	2558	3	0.273224
10.1.10.172	2080	3	0.273224
10.1.10.172	1713	3	0.273224
10.1.10.172	1676	3	0.273224
10.2.10.163	4673	3	0.303644
10.2.10.163	4541	3	0.303644
10.2.10.163	1063	3	0.303644
10.2.10.163	4884	2	0.202429
10.2.10.163	4824	2	0.202429

Scenario: The `win_audit` source type has been added to the Splunk environment and IT Ops needs to monitor events in the last 24 hours. However, the log file is in csv format, doesn't contain headers, and none of the fields are extracted.

Task 2: Use FX to extract the fields using the delimiters method.

16. Search for all events in the **last 24 hours** for the `win_audit` sourcetype.
17. View the event details to see which fields are extracted.
18. Click **Event Actions** > **Extract Fields**.
19. Select the **Delimiters** method and click **Next**.
20. For the Delimiter type, select **Comma**
21. Rename all the fields as follows (in this order):

field1	**Time**
field2	**EventCode**
field3	**EventType**
field4	**Type**
field5	**ComputerName**
field6	**LogName**
field7	**RecordNumber**

22. After all the fields are renamed, click **Next**.
23. For Extractions Name, enter **sysmon** and click **Finish>**.

Module 6 Lab Exercise: Creating Tags and Event Types

Description

This lab exercise walks you through the steps to create tags and event types.

Steps

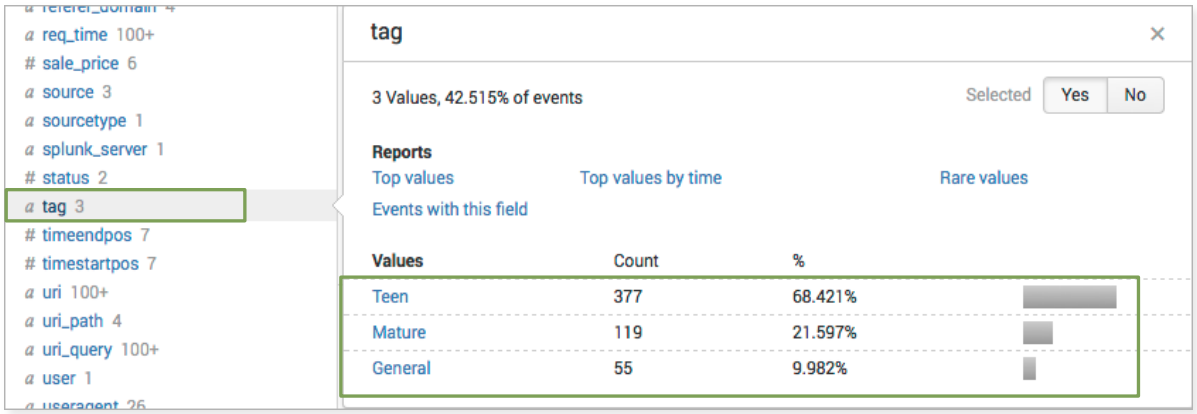
Scenario: The SVP of Marketing wants to easily identify products by a rating system that is not currently tracked in the data. The ratings of General, Teen, and Mature need to be applied to the games within the different categories.

Task 1: Create tags to identify a product rating.

1. Run a search for the **Last 24 hours** for all events with the `access_combined` sourcetype and `categoryId` field with valid values.

   sourcetype=access_combined categoryId!=null
2. In the Fields sidebar, click the `categoryId` field and note all the categories that are returned from the search. You should see eight categories.
3. Run a search for `categoryId=sports`
4. For the first event in the results, view the event details.
5. Find the row for the **categoryId** field. Click the **down arrow** under the **Actions** column and select **Edit Tags**.
6. Tag categoryId `sports` with the value **General** and click **Save**.
7. Run a search over the **Last 24 hours** for `categoryId=strategy`
8. For the first event in the results, view the event details.
9. Find the row for the **categoryId** field. Click the **down arrow** under the **Actions** column and select **Edit Tags**.
10. Tag categoryId `strategy` with the value **Teen** and save it**.**
11. Run a search over the **Last 24 hours** for `categoryId=shooter`
12. For the first event in the results, view the event details.
13. Find the row for the **categoryId** field. Click the **down arrow** under the **Actions** column and select **Edit Tags**.
14. Tag categoryId `shooter` with the value **Mature**.
15. Perform a search and verify the tags are created.

Results Example:



Task 2: Use tags in a search.

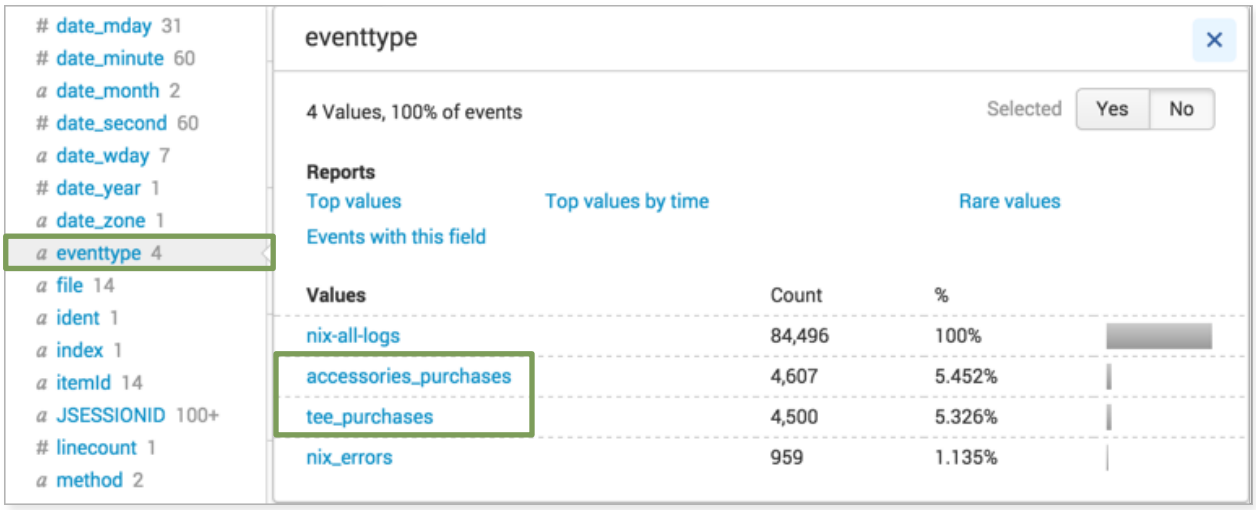
16. Search for `sourcetype=access_combined`.
17. Modify the search to limit results to only game categories tagged as **Teen**.
    Hint: `tag=Teen`. Also note that tags are case sensitive. A search for `tag=teen` produces no results.

Scenario: The Sales team wants to track monthly online sales. However, they want to easily identify purchases that are categorized by item.

Task 3: Use the Search interface to create event types for accessories and tee purchase events.

18. Search the `access_combined` source type for all purchase events in the **last 24 hours** where `categoryId=accessories`.
19. Select **Save As** > **Event Type**.
20. Name your event type: **accessories_purchases**
21. Optionally, select a color to flag the event type and a priority, then click **Save**.
22. Search the `access_combined` sourcetype for all purchase events in the **last 24 hours** where `categoryId=tee`.
23. Save the second event type as **tee_purchases**
24. Select a color to flag the event type and a priority, then click **Save.**
25. Perform a search for purchase events with `categoryId` values.
26. Verify your event types were created, by clicking on the `eventtype` field in the sidebar.

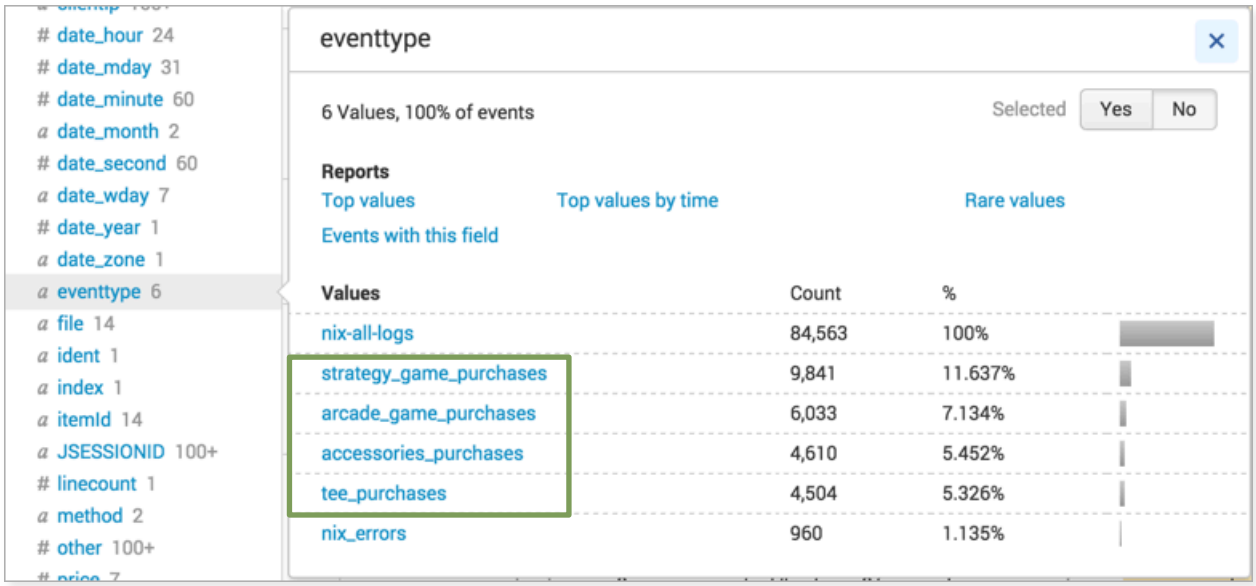
Results Example:



Task 4: Use the Event Type Settings page to create event types for strategy and arcade games purchase events.

27. Search the `access_combined` sourcetype for all purchase events in the **last 24 hours** for `STRATEGY` games.
28. After the search returns results, copy your search string.
29. Go to **Settings > Event types** and create a new event type.
30. Name the event type **strategy_game_purchases** and paste your search string in the **Search string** field. Click **Save**.
31. Repeat the above steps for purchased `ARCADE` game events and name the event type: **arcade_game_purchases**
32. Return to the **Search & Reporting** app and run a search to verify that your event types are being returned.

Results Example:



Note: Based on add-ons or apps you have installed, additional event types may be displayed. In this example, nix-all-logs is added by the *NIX app.

Supplemental Exercise:

Task: Tag these event types as purchases. Perform a search for the **purchases** tag. What types of events do you receive?

Module 7 Lab Exercise: Creating and Using Workflow Actions

Description

These steps create GET and Search workflow actions. You will use the src_ip field from the field extraction lab exercise.

**Note: You must have successfully completed Lab Exercise 5 to complete this lab exercise.

Steps

Scenario: Hackers are continually trying to log into the Linux server. IT Ops analysts need to track ongoing attempts by external sources trying to log in with invalid credentials.

Task 1: Create a GET workflow action that opens a new browser window with information about the source IP address.

1. Navigate to **Settings > Fields > Workflow actions**.
2. Click **New** to create a workflow action.
3. For the **Destination App**, select **search**.
4. For **Name**, type: **get_whois_info**
5. For **Label**, type: **Get info for IPaddress: src_ip**
6. For **Apply only to the following fields**, type: **src_ip**
7. For **Action type**, make sure **link** is selected.
8. For **URI**, type: **http://who.is/whois-ip/ip-address/src_ip**
9. From the **Open link in** dropdown menu, verify **New window** is selected.
10. From the **Link Method** dropdown menu, verify **get** is selected.
11. **Save** your workflow action.
12. Verify your workflow action works as expected. Return to the **Search & Reporting** app and search for `sourcetype=linux_secure src_ip=*` over the **last 24 hours**.

    HINT: Click the search menu option to refresh your browser.

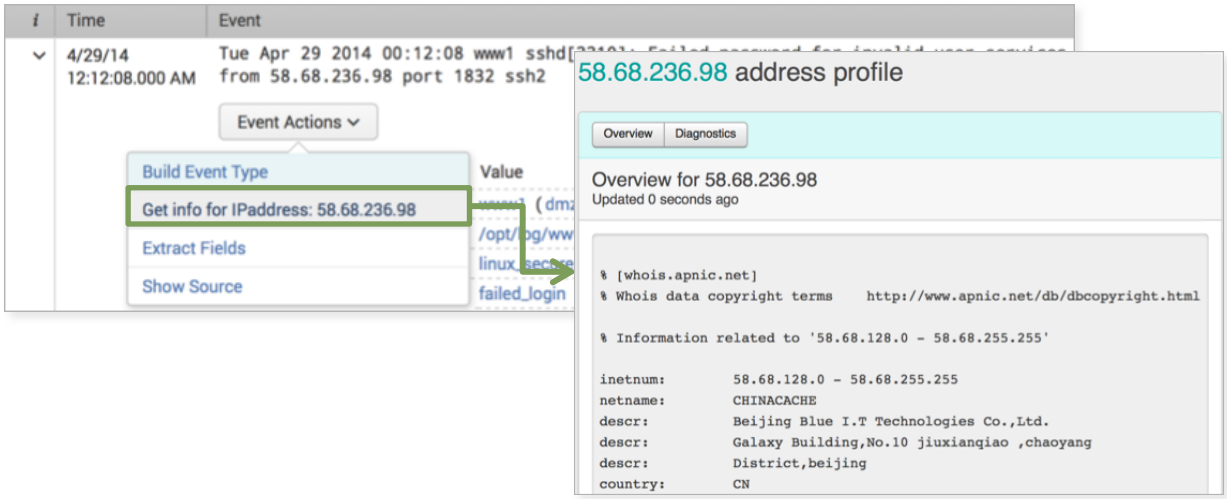
15. Expand the first event and click **Event Actions**.

13. Click **Get info for IPaddress: {src_ip}**.

**Note: If who.is is not behaving as expected, try http://whois.domaintools.com/src_ip.

14. A secondary browser window should open to the URI and display the IP address information.

Results Example:

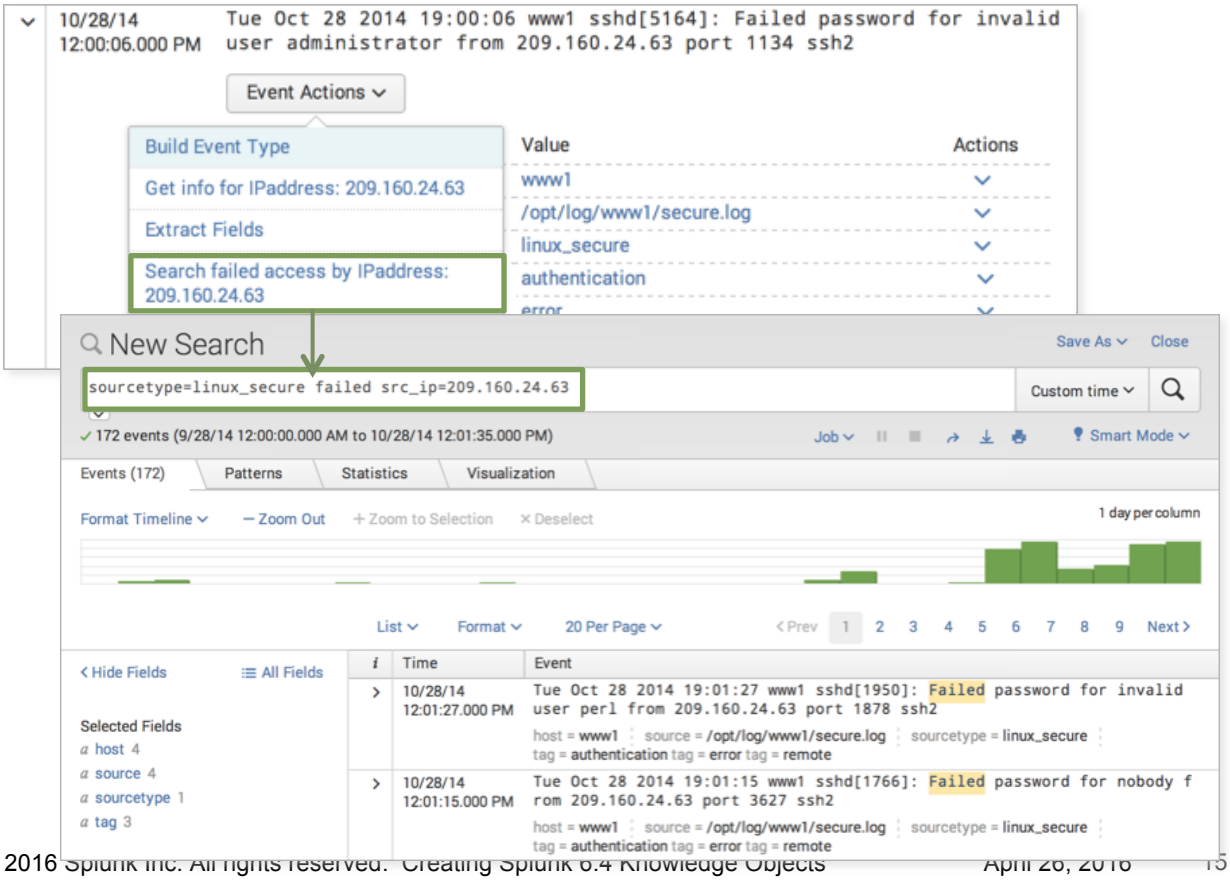


Task 2: **Create a Search workflow action that performs a search for all failed password events associated with a specific IP address.**

15. Navigate to **Settings > Fields > Workflow actions**.
16. Click **New**.
17. For the **Destination App**, select **search**.
18. For **Name**, type: **search_access_by_ipaddress**
19. For **Label**, type: **Search failed access by IPaddress: src_ip**
20. For **Apply only to the following fields**, type: **src_ip**
21. From the **Action Type** drop down menu, select **search**.
22. In the **Search string** field, type: **sourcetype=linux_secure failed src_ip=src_ip**
23. From the **Run in app** dropdown, select **search**.
24. From the **Run search in** dropdown menu, verify **New window** is selected.
25. Select the **Use the same time range as the search that created the field listing** check box.
26. **Save** your workflow action.
27. Verify your workflow action works as expected. Return to the **Search & Reporting** app and search for `sourcetype=linux_secure src_ip=*` over the **last 24 hours**.

    HINT: Click the search menu option to refresh your browser.
28. Expand an event with an IP Address field and click **Event Actions**.
29. Select **Search failed access by IPaddress: {src_ip}**
30. A secondary search window should open with the search results for the IP address.

    Results Example:



Module 8 Lab Exercise: Creating Alerts

Description

You learn to create an alert.

Steps

Scenario: **For security reasons, you need to monitor failed login attempts into the web servers. You are only interested in failed logins from known user accounts. You need to track these attempts because they can be more dangerous than unknown users. To gain access, attackers need a user name and a password. You want to be notified when there is more than one failed login attempt within one minute.**

Task 1: **Create a search to identify specific types of failed logins.**

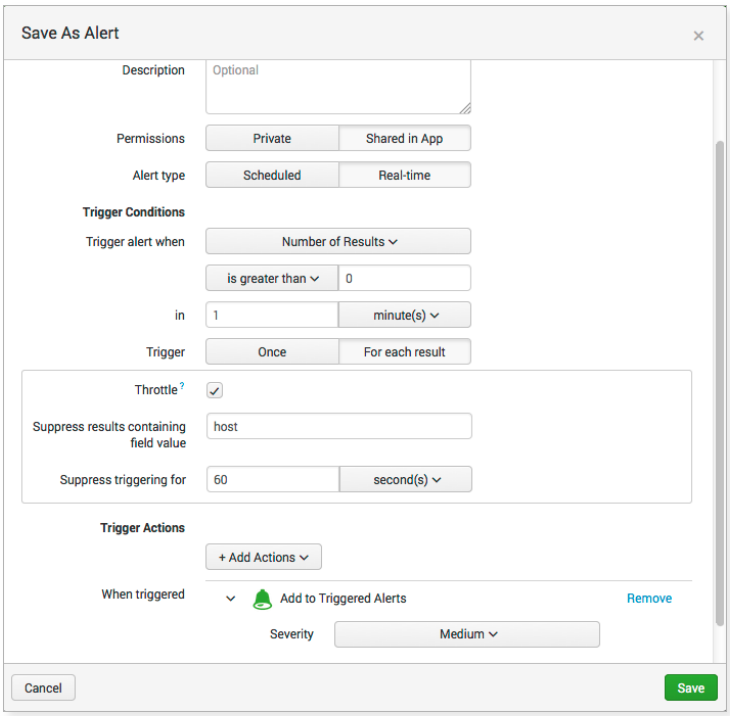
1. Search for the Linux secure logs on all web servers in the **Last 60 minutes**.
   sourcetype=linux_secure
2. Add the keywords `failed password NOT invalid`. Re-run the search.

Task 2: **Create and view an alert.**

3. From the **Save As** menu, select **Alert.**
4. Name the alert: **<student name> - Login Attempts**
5. For **Permissions**, select **Shared in App.**
6. For **Alert type**, select **Real-time.**
7. For **Trigger alert when**, select **Number of Results.**
8. Set the number of results to: **is greater than 0.**

   Note: This setting is set to 0 for testing. Once the alert is verified, you can change this value.
9. The **in** field should be set to **1 minute**.
10. For **Trigger**, select **For each result**.
11. Check the **Throttle** checkbox.
12. For **Suppress results containing field value**, type: **host**
13. Make sure **Suppress triggering for** is set to **60 second(s)**.
14. Click **Add Actions** and select **Add to Triggered Alerts**.
15. Set the **Severity** to **High.**

Example:



16. Click **Save**.
17. Click the **Permissions** link to examine details about the permissions you set.
18. Click **Cancel**. You should see an overview screen describing your new alert.
19. From the Splunk bar, click **Activity** > **Triggered Alerts**.
20. Select your student ID from the **Owner** menu and view the triggered alerts.
    Note: It may take a few minutes for your alert to appear.
21. Click the **View results** link on a triggered alert to see the event(s) that caused the alert.

Task 3: Disable the alert.

22. In the App navigation bar, click **Alerts**.
24. For the row containing your alert, click **Edit**, then **Disable**.
25. When the Disable dialog appears, click **Disable**.

Module 9 Lab Exercise: Creating and Using Macros

Description

This lab exercise walks you through the steps for creating a basic macro and a macro with arguments. You will use the VendorCountry field that was added from the lookup lab exercise.

**Note: You must have successfully completed Lab Exercise 3 to complete this lab exercise.

Steps

Scenario: The VP of Sales wants to run ad-hoc searches to determine how much product is being sold in a given month in various countries. He also wants to easily convert the sales to US Dollars based on the current exchange rates.

Task 1: Create a basic macro that lists the monthly total sales in the US.

1. Navigate to **Settings > Advanced search > Search macros**.
2. Click **New**.
3. Verify the **Destination app** is set to **search**.
4. Name the macro: **US_sales**
5. In the **Definition** field, type the following search string:

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

6. Save the macro.

Task 2: Use a basic macro.

7. Return to the **Search & Reporting** app.
8. In the search bar, type `` `US_sales` `` and search over the **Last 30 days**. Examine the results.

   Results Example:

product_name	USD
Benign Space Debris	$2,848.86
Curling 2014	$3,518.24
Dream Crusher	$21,514.62
Final Sequel	$8,596.56
Fire Resistance Suit of Provolone	$1,695.75
Holy Blade of Gouda	$1,743.09
Manganiello Bros.	$11,717.07
Manganiello Bros. Tee	$3,066.93
Mediocre Kingdoms	$5,322.87

Task 3: Create a macro with currency, currency symbol, and rate as arguments.

9. Navigate to **Settings** > **Advanced search** > **Search macros**.
10. Click **New**.
11. Verify the **Destination app** is set to **search**.
12. Name the macro: **monthly_sales(3)**
13. Enter the following search string:

```
| stats sum(price) as USD by product_name |  eval $currency$ = "$symbol$"
+ tostring(USD*$rate$, "commas") | eval USD = "$" + tostring(USD,
"commas")
```

14. In the Arguments field, type the arguments, separated by commas.
    Hint: currency,symbol,rate (order of variables must match the order of the values you enter in the search string)
15. Save the macro.

Task 4: **Use your macro with arguments in a search.**

16. Return to the **Search & Reporting** app.
17. Perform a search for `sourcetype=vendor_sales` where the `VendorCountry` is Germany, France, or Italy. Use the macro and pass the arguments `euro, €,` and `.79` for results in the **Last 30 days**. Copy/paste the `€` symbol from this document.
    Hint: `` `monthly_sales(currency,symbol,rate)` ``

    sourcetype=vendor_sales VendorCountry=Germany OR VendorCountry=France OR VendorCountry=Italy `monthly_sales(euro,€,.79)`
18. Run the search again for sales in the UK with the following arguments `GBP, £,` and `.64`. Copy/paste the `€` symbol from this document.

    sourcetype=vendor_sales VendorCountry="United Kingdom" `monthly_sales(GBP,£,.64)`

    Results Example:

product_name	USD	GBP
Benign Space Debris	$174.93	£112
Curling 2014	$219.89	£141
Dream Crusher	$119.97	£77
Final Sequel	$49.98	£32
Fire Resistance Suit of Provolone	$35.91	£23
Holy Blade of Gouda	$41.93	£27
Manganiello Bros.	$319.92	£205
Manganiello Bros. Tee	$109.89	£70
Mediocre Kingdoms	$99.96	£64
Orvil the Wolverine	$399.90	£256
SIM Cubicle	$359.82	£230
World of Cheese	$199.92	£128
World of Cheese Tee	$129.87	£83

Supplemental Exercise:

Task: Edit your macro and use the macro validation fields. Use the **isnum** expression to validate the rate field and provide an appropriate error message. Then test your macro by entering a string value for the rate.

Module 10 Lab Exercise: Creating a Data Model

Description

This exercise walks you through the process of creating a data model. After the data model is created, create a pivot to verify your data model provides the expected results.

Steps

Scenario: The VP of Sales wants to run reports based on daily activity from the online store.

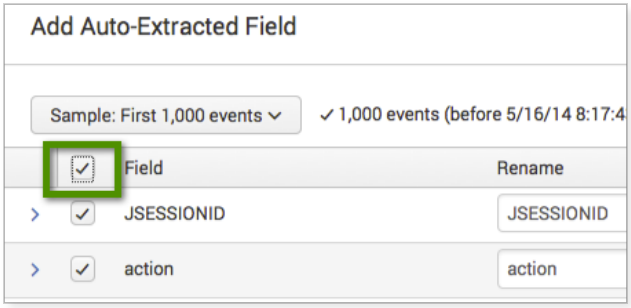
Task 1: Add the Web Requests root event. The root event will be the base search for all child events.

1. Navigate to **Settings > Data models**.
2. Click **New Data Model**.
3. In the **Title** field, type: **Buttercup Games Site Activity**
4. For **App**, make sure **Search & Reporting** is selected.
5. Create the data model.
6. Click **Add Object** and select **Root Event**
7. In the **Object Name** field, type: **Web Requests**
8. In the **Constraints** field, type: `sourcetype=access_combined`
9. Click **Preview** to see a sampling of the events.
10. After the data has been verified, **save** the root event.

Task 2: Add auto-extracted fields attributes.

11. Make sure the root **Web Requests** object is selected.
12. Click **Add Attribute** and select **Auto-Extracted**. A dialog box opens and displays all auto-extracted fields.
13. For this exercise, check the checkbox to the left of the **Field** column header. Checking this box selects all auto-extracted fields.

    Example:



14. Rename the following fields for pivot users:

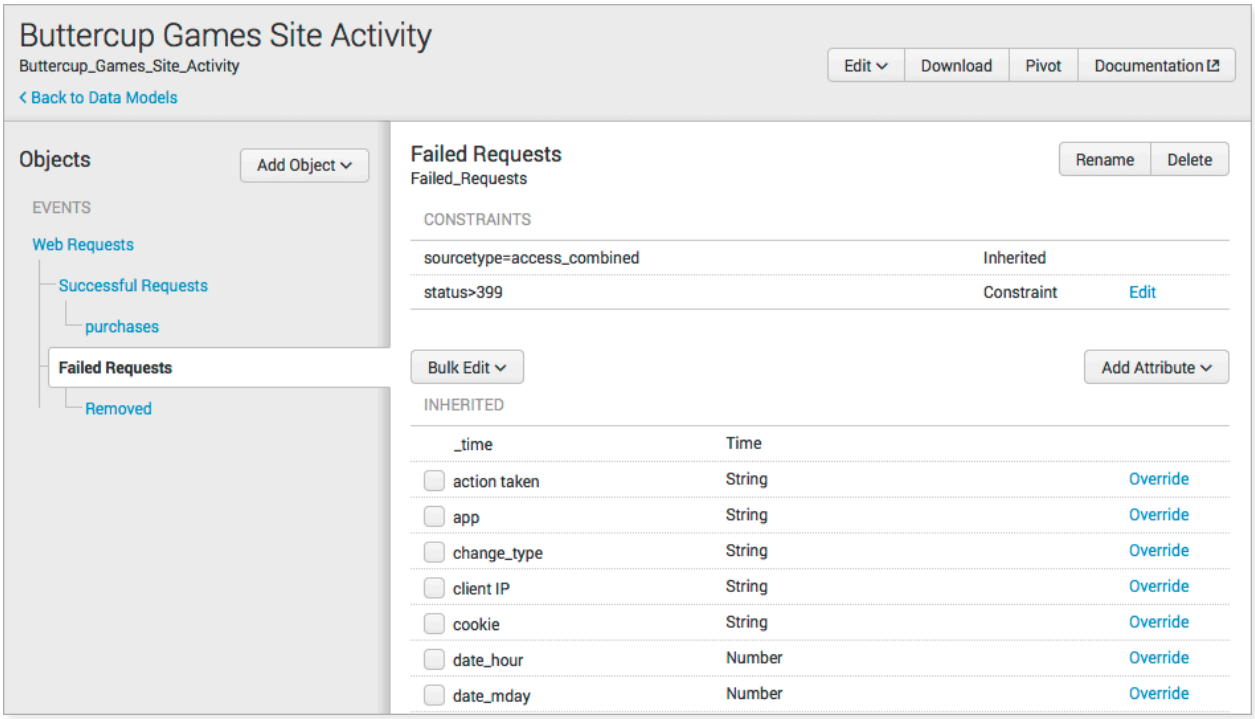
    action > action taken
    bytes > size
    categoryId > product category
    clientip > client IP
    productId > product ID
    product_name > product name
    req_time > request time

15. Click **Save**.

Task 3: **Add a child event for actions that were successful.**

16. Click **Add Object** and select **Child**.
17. In the **Object Name** field, type: **Successful Requests**
18. In the **Additional Constraints** field, type: `status<400`
19. Click **Preview** to see a test sample of your results.
20. **Save** the child object.
21. Select the **Successful Requests** object. Add a child object called **purchases** with an **Additional Constraints** value of `action=purchase productId=*`. Remember to click **Save**.
22. Select the **Web Requests** event and add child object named: **Failed Requests**
23. In the **Additional Constraints** field, type: `status>399`
24. Click **Preview** to receive a test sample of your results.
25. **Save** the child object.
26. Under the **Failed Requests** child object, add a child object named **removed** with an **Additional Constraints** value of `action=remove productId=*`. Remember to click **Save**.

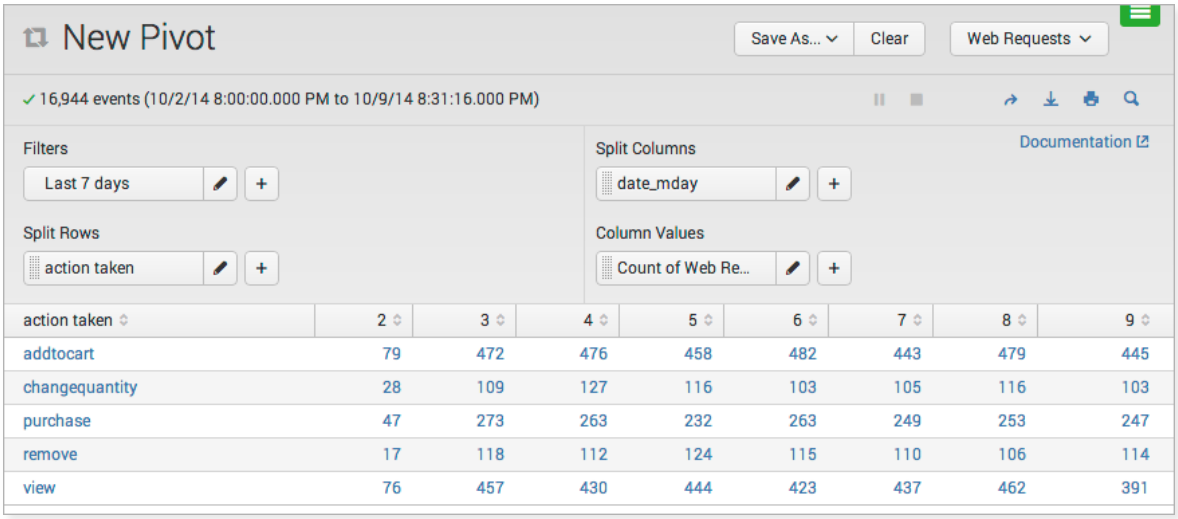
    Results Example:



Task 4: Test your data model by creating a pivot.

27. Click **Pivot**.
28. Select the **Web Requests** object.
29. In the **New Pivot** window, change the following:
    - **Filter** on the **Last 7 days**
    - **Split Rows** by **action taken** and click **Add To Table**
    - **Split Columns** by **date_mday** and click **Add To Table**

Results Example:

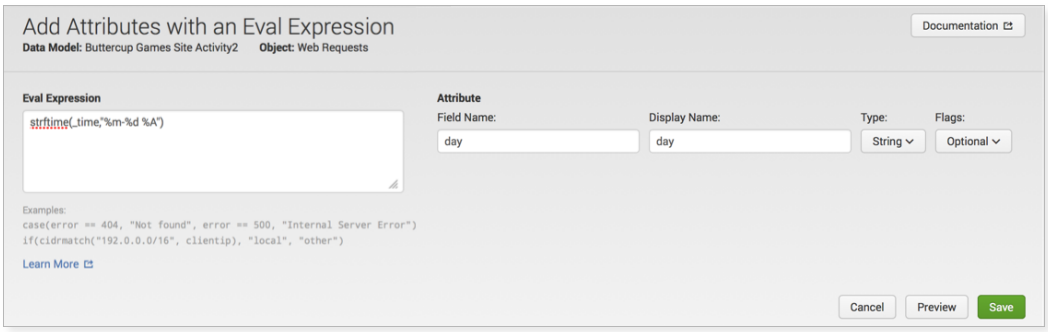


action taken	2	3	4	5	6	7	8	9
addtocart	79	472	476	458	482	443	479	445
changequantity	28	109	127	116	103	105	116	103
purchase	47	273	263	232	263	249	253	247
remove	17	118	112	124	115	110	106	114
view	76	457	430	444	423	437	462	391

Task 5: Add an attribute that uses an eval expression. The eval expression will list events chronologically by date and day.

30. Click the **Web Requests** button to go back to the Buttercup Games Site Activity data model.
31. Select **Edit Object.**
32. Make sure **Web Requests** is selected.
33. From the **Add Attribute** menu, select **Eval Expression**.
34. In the **Eval Expression** field, type:
    `strftime(_time,"%m-%d %A")`
35. For **Field Name**, type: **day**
36. For **Display Name**, type: **day**
37. Click **Preview** to verify your eval expression returns results.
38. Save the eval expression.

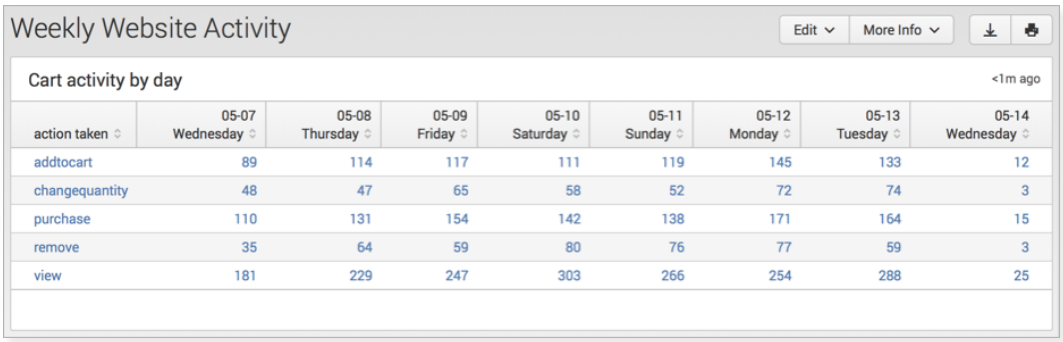
Results Example:



Task 6: Verify the eval expression works as expected by using Pivot to create a dashboard.

39. Click **Pivot**.
40. Select the **Web Requests** object.
41. Change the time filter to the **Last 7 days**.
42. **Split Rows** by **action taken**.
43. Click **Add To Table**.
44. **Split Columns** by **day**.
45. Click **Add To Table**.
46. Click **Save As** and select **Dashboard Panel**.
47. For **Dashboard Title**, type: **Weekly Website Activity**
48. For **Panel Title**, type: **Cart activity by day**
49. Click **Save**.
50. Click **View Dashboard**. You should see the web requests categorized and counted by day.

    Results Example:



action taken	05-07 Wednesday	05-08 Thursday	05-09 Friday	05-10 Saturday	05-11 Sunday	05-12 Monday	05-13 Tuesday	05-14 Wednesday
addtocart	89	114	117	111	119	145	133	12
changequantity	48	47	65	58	52	72	74	3
purchase	110	131	154	142	138	171	164	15
remove	35	64	59	80	76	77	59	3
view	181	229	247	303	266	254	288	25

Task 7: Add attributes from a lookup. The lookup table will provide descriptions for status codes.

51. Navigate to **Settings > Data models**.
52. Select the **Buttercup Games Site Activity** data model.
53. Make sure the **Web Requests** root object is selected.
54. Click **Add Attribute** and select **Lookup**.
55. From the **Lookup Table** dropdown list, select **http_status_lookup**.
56. From the **Field in Lookup** dropdown, select **code**.
57. From the **Attribute** dropdown**,** select **status**. This maps the `status` field in your indexed data to the `code` column in the lookup table.
58. For the lookup **Output** section in the **Field in Lookup** field, check the **description** checkbox.

59. In the **Display Name** field, type: **status description**
60. Click the **Preview** button. You should see a **description** column in the results.
61. Click **Save**.

Task 8: Verify the lookup works properly by creating a Pivot report.

62. Click **Pivot**.
63. Select the **Web Requests** object.
64. Change the **Filter** to **Last 7 days**.
65. From **Split Rows**, add the **status description** attribute and click **Add To Table**.
66. Click the **+** button to split by another row and add the **status** attribute. Click **Add To Table**.

    Note: This is a double row split, not a column split.

    Results Example:

status description	status	Count of Web Requests
Bad Request.	400	577
Forbidden.	403	182
HTTP Version Not Supported.	505	357
Internal Server Error.	500	533

67. **Split Columns** by **day** and click **Add To Table**.
68. Click **Save As** and select **Dashboard Panel**.
69. Select **Existing** Dashboard and select **Weekly Website Activity**.
70. For the **Panel Title**, type: **Web Requests Summary**
71. Click **Save**.
72. Click **View Dashboard**.

    Results Example:

Weekly Website Activity

Edit More Info

Cart activity by day 1m ago

action taken	05-07 Wednesday	05-08 Thursday	05-09 Friday	05-10 Saturday	05-11 Sunday	05-12 Monday	05-13 Tuesday	05-14 Wednesday
addtocart	79	114	117	111	119	145	133	15
changequantity	40	47	65	58	52	72	74	4
purchase	95	131	154	142	138	171	164	18
remove	30	64	59	80	76	77	59	4
view	163	229	247	303	266	254	288	36

Web Requests Summary 1m ago

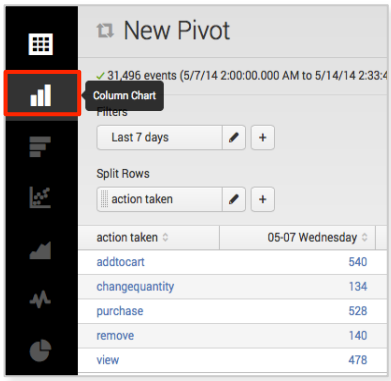
status description	status	05-07 Wednesday	05-08 Thursday	05-09 Friday	05-10 Saturday	05-11 Sunday	05-12 Monday	05-13 Tuesday	05-14 Wednesday
Bad Request.	400	127	193	201	213	202	234	203	24
Forbidden.	403	50	69	60	66	73	71	75	6
HTTP Version Not Supported.	505	69	130	124	164	121	136	130	15
Internal Server Error.	500	113	175	206	188	209	238	224	19
Not Acceptable.	406	132	164	199	197	188	224	207	26
Not Found.	404	114	177	223	197	213	210	219	20
OK.	200	6371	9028	9779	10281	9952	10379	10860	1352
Request Timeout.	408	118	175	193	217	199	195	219	25
Service Unavailable.	503	190	245	265	278	276	298	309	47

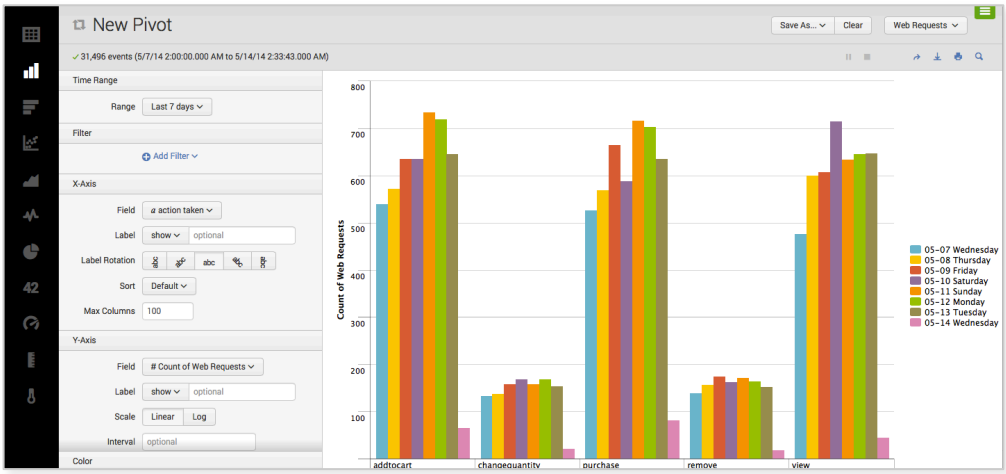
Supplemental Exercise:

Task 1: **From the pivot editor, add an attribute as a filter that displays all shopping cart activity except changequantity and remove.**

1. Hover your mouse in the lower right corner of the **Cart Activity by day** dashboard panel. Click the **Open in Pivot** icon .
2. Refine your search results by selecting the **Column chart** icon from the table formats on the left.

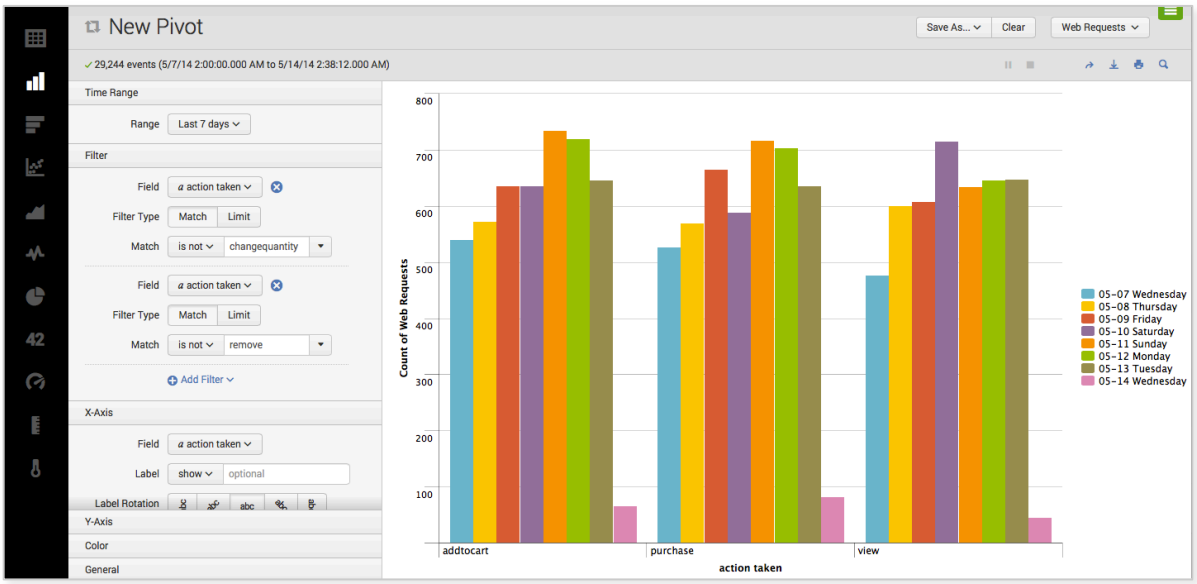
   Results Examples:





3. Click **Add Filter** and choose **action taken**.
4. For **Filter Type**, select **Match**.
5. For **Match,** change the operator to **is not**, then select **changequantity**.
6. Add another filter and choose **action taken**.
7. For the **Filter Type** select **Match**.
8. For **Match**, change the operator to **is not** and then select **remove**.

Results Example:



9. Click **Save As** and select **Dashboard Panel**.
10. Save to the **Weekly Website Activity** dashboard.
11. For **Panel Title**, type: **Add Purchase View**
12. Save and view your dashboard.
13. Rearrange the panels to your liking and admire your work!