

Using Fields – Lab Guide

Overview

Welcome to the Splunk Education lab environment. These lab exercises will give you some practical experience of using fields in searches.

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
security	Web server	linux_secure	action, app, dest, process, src_ip, src_port, user, vendor_action
network	Email security data	cisco_esa	dcid, icid, mailfrom, mailto, mid
	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
	Firewall data	cisco_firewall	bcg_ip, dept, Duration, fname, IP, lname, location, rfid, splunk_role, splunk_server, Username
sales	Email security data	cisco_esa	dcid, icid, mailfrom, mailto, mid
	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
	Firewall data	cisco_firewall	bcg_ip, dept, Duration, fname, IP, lname, location, rfid, splunk_role, splunk_server, Username

Common Commands & Functions

These commands and statistical functions are commonly used in searches but may not have been explicitly discussed in the module. Please use this table for quick reference. Click on the hyperlinked SPL 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 <code>src_ip</code> 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 <code>SESSIONID</code> to 'The session ID' rename SESSIONID as "The session ID"
fields	command	Keeps (+) or removes (-) fields from search results.	Remove the <code>host</code> 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 <code>first_name</code> and <code>last_name</code> values with a space to create a field called "full_name" eval full_name=first_name." ".last_name
table	command	Returns a table.	Output <code>vendorCountry</code> , <code>vendor</code> , and <code>sales</code> 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 <code>action</code> 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 Connection Info

Access labs using the server URL, user name, and password shown in your lab environment.

SERVERS

LAB DOCUMENT

CHECK MY WORK

HELP

Lab Server Info:

SERVER URL	PUBLIC IP	SPLUNK USER NAME	PASSWORD	DOWNLOAD	STATUS
https://11-195-15-aio.class.splunk.com	3.23.114.109	powerUser	wrarug8hikoZuBa	link	DEPLOYED

Lab Exercise 1 – Use Fields in Searches

Description

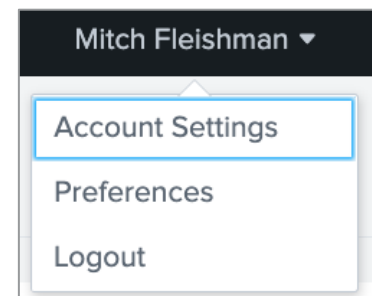
Configure the lab environment user account. Then, explore how using fields and fields with operators can change search results. Additional tasks will test your knowledge of the **rename** and **fields** commands.

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. 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.
3. Click on the username you logged in with (at the top of the screen) and then choose **Account Settings** from the drop-down menu.
4. In the **Full name** box, enter your first and last name.
5. Click **Save**. (Do not change the password)
6. 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.

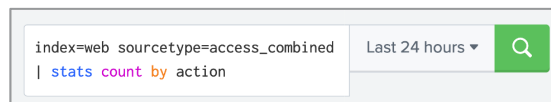
NOTE: Sometimes there can be delays in executing an action like saving in the UI or returning results of a search. If you are experiencing a delay, please allow the UI a few minutes to execute your action.

7. Navigate to **user name > Preferences**.
8. Choose your local time zone from the **Time zone** drop-down menu.
9. Click **Apply**.

- (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.



Search auto-format enabled.

Task 2: Use the Fields sidebar to examine search results.

- In the top left corner of Splunk Web, select **Apps > Search & Reporting**. This sets our app context to the search app.
- Search online sales data (**index=web sourcetype=access_combined**) for all events containing a purchase action (**action=purchase**.) Execute this search over the **Last 24 hours**.

NOTE: After the search finalizes, verify that the search executed in Smart Mode. The search mode is located under the time range picker. If the search did not execute in Smart Mode, change it to Smart Mode, and then re-execute the search.

- Examine the **Interesting Fields** list to the left of your events. Notice that the **product_name** is one of the fields returned by Splunk.

NOTE: To find some less common fields, you may need to open the **All Fields** window from the link at the top of the Fields sidebar.

- In the Fields sidebar, under **Interesting Fields**, click **product_name**. Notice the pop-up window shows the top ten purchased products. Close the window by clicking the **X** in the upper-right corner.

The screenshot shows the Splunk Fields sidebar on the left with a list of fields. The field `a product_name 14` is highlighted with a red box. To the right, the `product_name` field window is open, showing 14 values (50.499% of events). The window includes a 'Selected' toggle set to 'Yes' and a close button (X). Below the toggle, there are sections for 'Reports' (Top values, Top values by time, Rare values) and 'Events with this field'. A table titled 'Top 10 Values' displays the following data:

Top 10 Values	Count	%
SIM Cubicle	45	11.111%
World of Cheese	40	9.876%
Dream Crusher	37	9.136%
Curling 2014	34	8.395%
Mediocre Kingdoms	33	8.148%
Final Sequel	26	6.42%
Fire Resistance Suit of Provolone	26	6.42%
Manganiello Bros. Tee	26	6.42%
Puppies vs. Zombies	25	6.173%
Benign Space Debris	24	5.926%

15. In the Fields sidebar, under **Interesting Fields**, click `sale_price`. This field contains the product's discounted price for each purchase event.
 - a. Make the `sale_price` field a selected field. From the `sale_price` field window, click **Yes** in the upper right corner next to **Selected**. Close the `sale_price` field window by clicking the **X** in the upper-right corner.
 - b. Notice `sale_price` is now a selected field in the Fields sidebar.
 - c. Now, each event with a value present for `sale_price` will have `sale_price=<value>` in the last line of the event.

The screenshot shows the Splunk Fields sidebar on the left. Under the 'INTERESTING FIELDS' section, the field `# sale_price 6` is highlighted with a red box. To the right, the `sale_price` field window is open, showing 6 values (50.499% of events). The window includes a 'Selected' toggle set to 'Yes' and a close button (X). Below the toggle, there are sections for 'Reports' (Average over time, Maximum value over time, Minimum value over time, Top values, Top values by time, Rare values) and 'Events with this field'. A summary line shows: **Avg:** 15.483827160493826 **Min:** 1.99 **Max:** 24.99 **Std Dev:** 8.00452139680036. A table titled 'Values' displays the following data:

Values	Count	%
16.99	105	25.926%
19.99	97	23.951%
24.99	82	20.247%
1.99	51	12.592%
6.99	47	11.605%
2.99	23	5.679%

16. In the Fields sidebar, under **Selected Fields**, click the `sale_price` field.

- a. From the field window, click the value with the highest number of purchases (listed at the top.) Notice the field and value have been added to the search criteria in the search bar. Also, this selection causes a new search to execute using the new search criteria.

```
index=web sourcetype=access_combined action=purchase sale_price="16.99"
```

- b. Remove **sale_price=<value>** from the search criteria (by deleting it from the search text) and re-execute the search by pressing ENTER or by clicking the magnifying glass to the right of the search bar.
17. In the Fields sidebar, under **Interesting Fields**, click **categoryId** to see which types of products make up the most purchases. Close the window by clicking the **X** in the upper-right corner.

Task 3: Compare results from searches using the !=, NOT, and *= field expressions.

18. Search for **index=web sourcetype=access_combined** with a time range of **Yesterday**. Note the number of returned events.
19. Edit your search to find only events that have a value present for the **action** field. Run the search again. Again, note the number of return events.
20. In the Fields sidebar, under **Interesting Fields**, click **action**. Notice that the events contain five different values for **action**. Close the window by clicking the **X** in the upper-right corner.

New Search

index=web sourcetype=access_combined action=*

✓ 2,607 events (5/19/22 12:00:00.000 AM to 5/20/22 12:00:00.000 AM) No Event Sampling ▼

Events (2,607) Patterns Statistics Visualization

Format Timeline ▼ — Zoom Out + Zoom to Selection × Deselect

List ▼ ✓ Format 20 Per Page ▼

< Hide Fields All Fields

SELECTED FIELDS

- a host 3
- # sale_price 6
- a source 3
- a sourcetype 1

INTERESTING FIELDS

- a action 5
- # bytes 100+
- a categoryId 8
- a clientip 100+
- # date_hour 24
- # date_mday 2
- # date_minute 60
- a date_month 1

action

5 Values, 100% of events

Selected Yes No

Reports

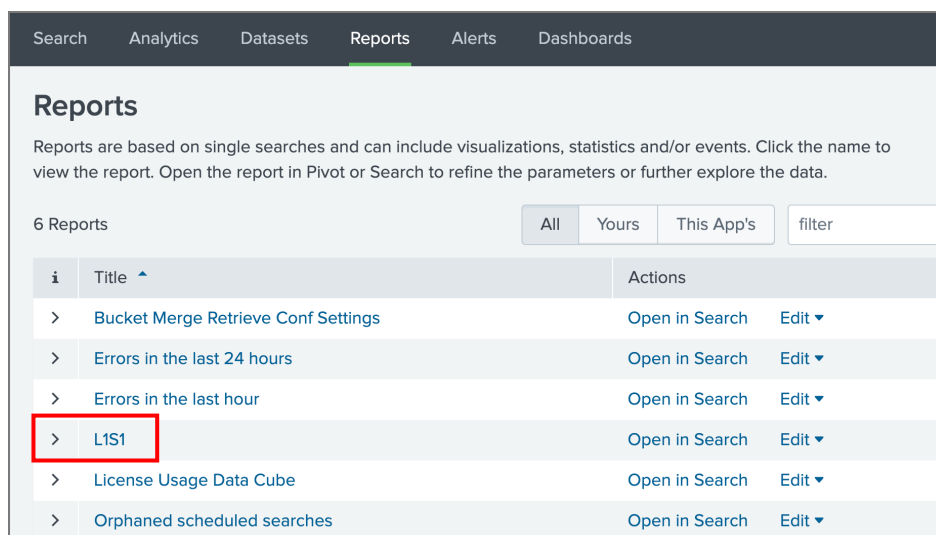
- Top values
- Top values by time
- Rare values

Events with this field

Values	Count	%
addtocart	776	29.766%
purchase	773	29.651%
view	670	25.7%
changequantity	196	7.518%
remove	192	7.365%

21. Edit your search to find events that do not have a value present for the **action** field. Run the search again. Note the number of returned events.
22. In the Fields sidebar, under **Interesting Fields**, try to find the **action** field.

23. Add the number of events that were returned from your last two searches—the events that contain an **action** value and the events that don't. Does the sum equal the total number of events returned from your first search? (Hint: If this is not the case, try running all three searches again and be sure the time range is set to **Yesterday** for each search.)
24. Edit the search to find only those events where the **action** field contains the value, **purchase**.
25. In the Fields sidebar, under **Interesting Fields**, click **action**. Notice that you now see only one possible value, **purchase**. Close the window by clicking the **X** in the upper right corner.
26. Edit the search to find events where the **action** field contains some value other than **purchase**. Note the number of returned events.
27. In the Fields sidebar, under **Interesting Fields**, click **action**. Notice that you now see all possible values except **purchase**. Close the window by clicking on the **X** in the upper right corner.
28. Edit the search to find events where:
 - a. The **action** field contains some value other than **purchase**.
 - b. The **action** field contains no value at all (See step 21 above).
 Note the number of returned events.
29. 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: SecOps wants a list of authentication failure events associated with admin roles over the last 60 minutes.

Task 4: Use keywords, field expressions, and the fields command to filter for specific events.

30. Search the web server (`index=security sourcetype=linux_secure`) for events during the **Last 60 minutes**.

31. Modify your search to include the following:

- Failed password attempts by invalid users by adding `failed invalid` to your basic search.
- Events associated with the administrator user, i.e. user accounts that begin with `admin`.

32. Use the `fields` command to extract only the `user`, `src_ip`, and `app` fields.

< Hide Fields	≡ All Fields	i	Time	Event
INTERESTING FIELDS		>	5/20/22 10:30:39.000 AM	Fri May 20 2022 16:30:39 www3 sshd[2893]: Failed password for invalid user administrator h2
a app 1		>	5/20/22 10:25:41.000 AM	Fri May 20 2022 16:25:41 www3 sshd[4246]: Failed password for invalid user admin from 10.
a src_ip 10		>	5/20/22 10:24:09.000 AM	Fri May 20 2022 16:24:09 www3 sshd[3703]: Failed password for invalid user administrator h2
a user 2		>	5/20/22 10:22:29.000 AM	Fri May 20 2022 16:22:29 www3 sshd[2879]: Failed password for invalid user administrator ssh2
+ Extract New Fields				

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

Task 5: Complete the missing portion of a search with the rename command.

34. This search finds purchase events from the online sales data that encountered a server problem (`status>399`.) Complete the `<missing>` portion of this search so that the `clientip` field is renamed to "Customer IP", the `host` field is renamed to "Web Server", and the `status` field is renamed to "HTTP Status." Run this search over the **Last 4 hours**.

```
index=web sourcetype=access_combined action=purchase status>399
| table clientip host status
| <missing>
```

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

Lab Exercise 2 – Compare Temporary vs Persistent Fields

Description

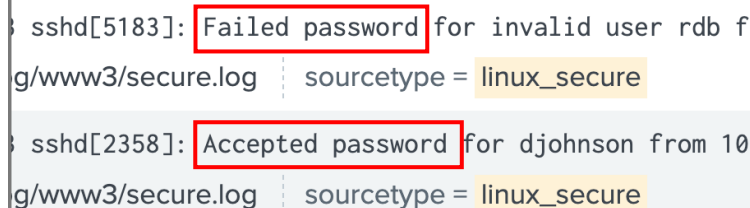
Perform search-time field extractions using the **erex** and **rex** commands.

Steps

Scenario: SecOps wants to see a count of event descriptions by port from all web server events over the past 7 days.

Task 1: Use the **erex** command to extract temporary fields and include events based on pattern matching.

1. Search for all web server events (**index=security sourcetype=linux_secure**) over the **Last 7 days** that contain the keyword “port”.
2. Scroll through the list of events. Notice how many events have “Accepted password for...” and “Failed password for...”.



```
sshhd[5183]: Failed password for invalid user rdb f
g/www3/secure.log | sourcetype = linux_secure

sshhd[2358]: Accepted password for djohnson from 10
g/www3/secure.log | sourcetype = linux_secure
```

3. Use the **erex** command to create a field called **event_description**. Provide the phrases "Accepted password " and "Failed password " as examples.
4. Pipe your search to the following **stats** command:

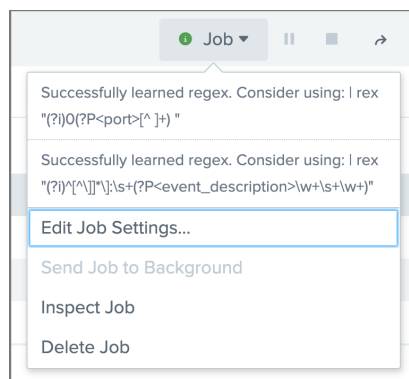
```
| stats count(src_port) by event_description
```

The **stats** command uses the **count** function to count the number of ports (**src_port**) for each value of **event_description**.

5. Observe your results. There appears to be no results for “Server listening”. Why?
6. Save your search as a report with the name **L2S1**. In Task 2, you will fix your search.

Task 2: Use the **rex** command to improve your search results from Task 1.

7. Insert an **erex** command after the basic search that will create a new field called “port”. Provide it with three port examples including 22. Edit the **stats** command so that it now counts **port** values instead of **src_port** values.
8. Click on the **Job** dropdown and view the messages. You will see that Splunk is recommending you use the **rex** command with regex it has automatically generated. (Note: Your regex may be slightly different for **port**.)



9. Replace your two **erex** commands with the suggested **rex** commands.
10. Save your search as a report with the name **L2S2**.