

## Using Fields – Lab Solutions 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
			action, app, dest, process, src_ip, src_port, user, vendor_action
			dcid, icid, mailfrom, mailto, mid
security	Web server	linux_secure	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
			bcg_ip, dept, Duration, fname, IP, lname, location, rfid, splunk_role, splunk_server, Username
network	Email security data	cisco_esa	
	Web security appliance data	cisco_wsa_squid	
	Firewall data	cisco_firewall	

## 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
<a href="#">sort</a>	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    <b>sort</b> 100 -src_ip
<a href="#">where</a>	command	Filters search results using eval-expressions.	Return events with a <code>count</code> value greater than 30    <b>where</b> count > 30
<a href="#">rename</a>	command	Renames one or more fields.	Rename <code>SESSIONID</code> to 'The session ID'    <b>rename</b> SESSIONID as "The session ID"
<a href="#">fields</a>	command	Keeps (+) or removes (-) fields from search results.	Remove the <code>host</code> field from the results    <b>fields</b> - host
<a href="#">stats</a>	command	Calculates aggregate statistics over the results set.	Calculate the total sales, i.e. the sum of <code>price</code> values.    <b>stats</b> sum(price)
<a href="#">eval</a>	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"    <b>eval</b> full_name=first_name." ".last_name
<a href="#">table</a>	command	Returns a table.	Output <code>vendorCountry</code> , <code>vendor</code> , and <code>sales</code> values to a table    <b>table</b> vendorCountry, vendor, sales
<a href="#">sum()</a>	statistical function	Returns the sum of the values of a field. Can be used with <b>stats</b> , <b>timechart</b> , and <b>chart</b> commands.	Calculate the sum of the <code>bytes</code> field    <b>stats</b> sum(bytes)
<a href="#">count or count()</a>	statistical function	Returns the number of occurrences of all events or a specific field. Can be used with <b>stats</b> , <b>timechart</b> , and <b>chart</b> commands.	Count all events as "events" and count all events that contain a value for <code>action</code> as "action"    <b>stats</b> 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 Fields in Searches

### Description

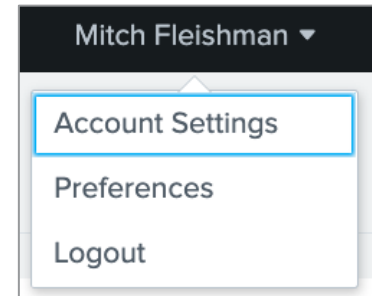
Configure the lab environment user account. Then, explore how using fields and fields with operators can change search results. Additional tasks will test your knowledge of the **rename** and **fields** commands.

### 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.

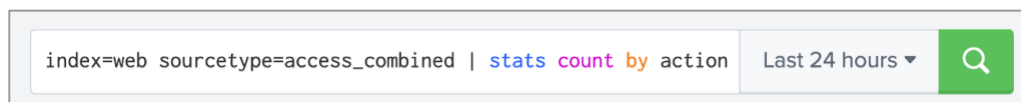
1. Log into your Splunk lab environment using the username and password provided to you.
2. You may see a pop-up window welcoming you to the lab environment. You can click **Continue to Tour** but this is not required. Click **Skip** to dismiss the window.
3. Click on the username you logged in with (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**.
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**.)



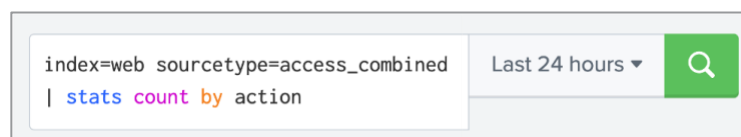
*After you complete step 6, you will see your name in the web interface.*

**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**.
10. (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.

11. In the top left corner of Splunk Web, select **Apps > Search & Reporting**. This sets our app context to the search app.
12. 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**.  
`index=web sourcetype=access_combined action=purchase`

**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.

13. 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 fields, you may need to open the **All Fields** window from the link at the top of the Fields sidebar.

14. 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.

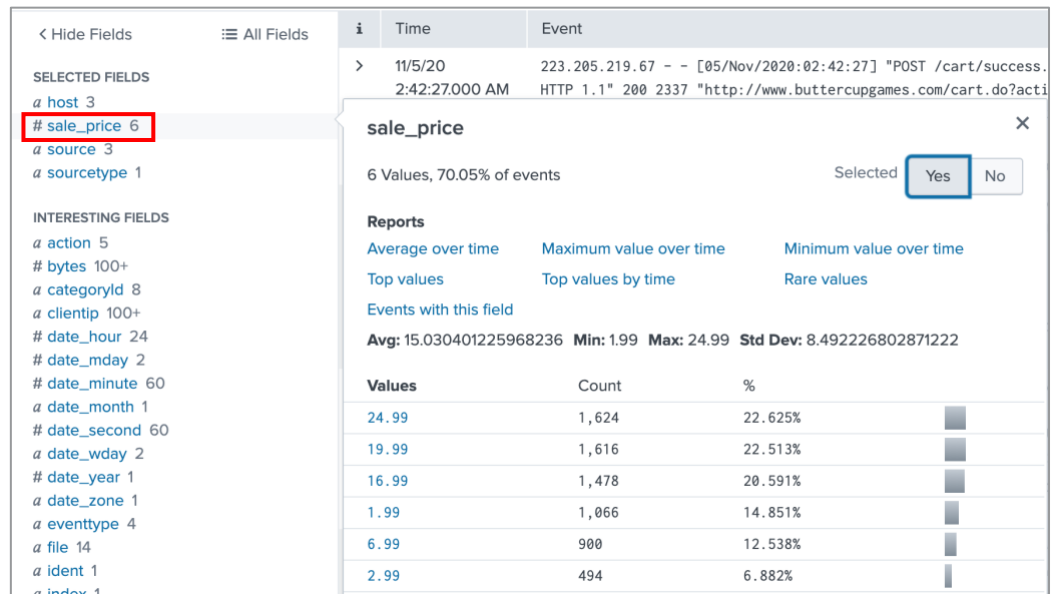
The screenshot shows the Splunk Fields sidebar on the left and a pop-up window for the `product_name` field on the right.

**Fields Sidebar (Left):** A list of fields is shown. The field `a product_name 14` is highlighted with a red box.

**product\_name Field Window (Right):** The window title is `product_name`. It shows 14 values, which is 70.05% of the events. There are 'Selected', 'Yes', and 'No' buttons. Below this, there are tabs for 'Reports', 'Top values', 'Top values by time', and 'Rare values'. The 'Top values' tab is selected, showing a table of the top 10 values.

Top 10 Values	Count	%
SIM Cubicle	657	9.153%
World of Cheese	648	9.028%
Dream Crusher	638	8.888%
Fire Resistance Suit of Provolone	605	8.428%
Mediocre Kingdoms	605	8.428%
Final Sequel	516	7.189%
Orvil the Wolverine	510	7.105%
Holy Blade of Gouda	494	6.882%
Manganiello Bros.	476	6.631%
Manganiello Bros. Tee	475	6.617%

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 and a field window for 'sale\_price' on the right. In the sidebar, under 'SELECTED FIELDS', the field '# sale\_price 6' is highlighted with a red box. The field window on the right shows the following information:

- sale\_price** (with a close button 'X')
- 6 Values, 70.05% of events
- Selected:
- Reports**
  - Average over time
  - Maximum value over time
  - Minimum value over time
  - Top values
  - Top values by time
  - Rare values
  - Events with this field
- Summary Statistics:** Avg: 15.030401225968236, Min: 1.99, Max: 24.99, Std Dev: 8.492226802871222
- Table of Values:**

Values	Count	%
24.99	1,624	22.625%
19.99	1,616	22.513%
16.99	1,478	20.591%
1.99	1,066	14.851%
6.99	900	12.538%
2.99	494	6.882%

16. In the Fields sidebar, under **Selected Fields**, click the **sale\_price** field.

- 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.
- Remove **sale\_price=<value>** from the search criteria (by deleting it from the search text) and re-execute the search.

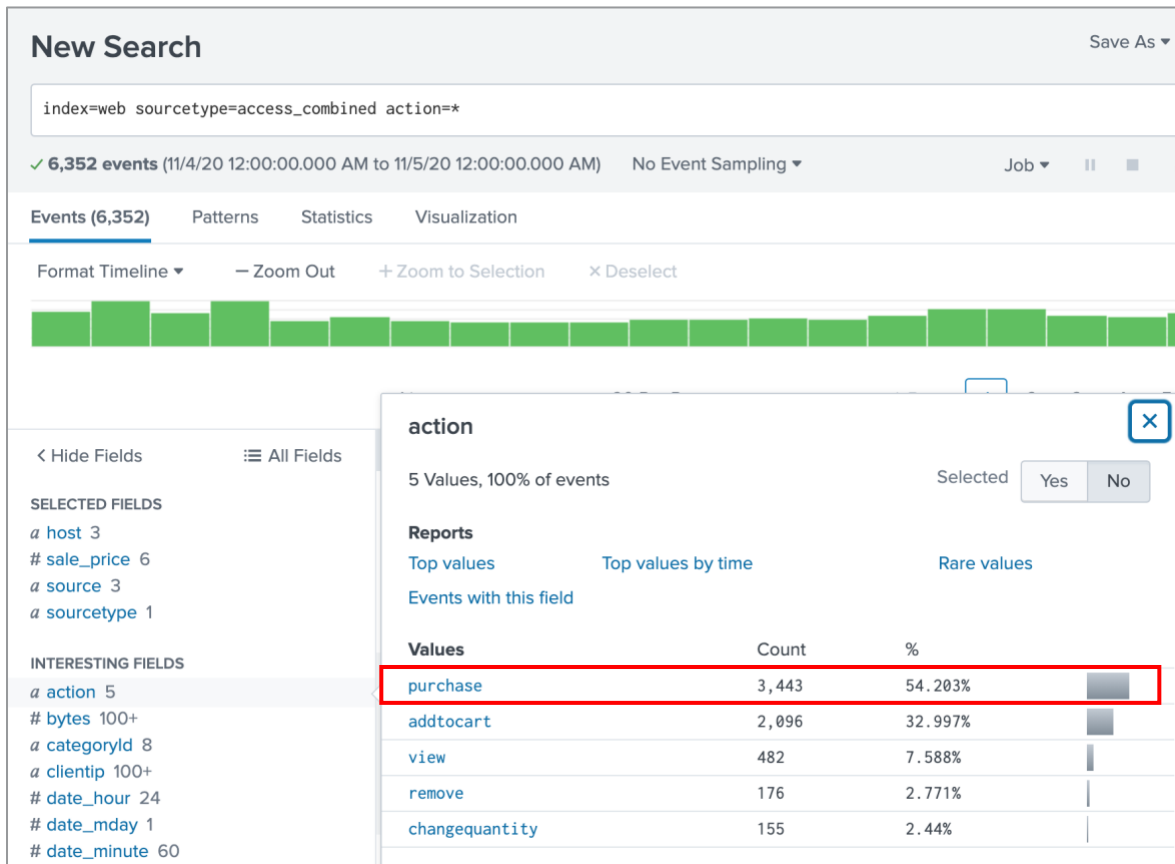
17. In the Fields sidebar, under **Interesting Fields**, click **categoryId** to see which types of products make up the most purchases. Close the window by clicking the **X** in the upper-right corner.

### 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**.  
How many events are returned? \_\_\_\_\_
19. Edit your search to find only events that have a value present for the **action** field. Run the search again.  
Now, how many events are returned? \_\_\_\_\_  
`index=web sourcetype=access_combined action=*`

This search will only find events that have some value present for the **action** field. Therefore, this search should return the same or fewer events than your previous search.

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** Save As ▾

index=web sourcetype=access\_combined action=\*

✓ 6,352 events (11/4/20 12:00:00.000 AM to 11/5/20 12:00:00.000 AM) No Event Sampling ▾ Job ▾ || ■

Events (6,352) Patterns Statistics Visualization

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

< 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 1
- # date\_minute 60

**action** ×

5 Values, 100% of events Selected Yes No

**Reports**

- Top values
- Top values by time
- Rare values
- Events with this field

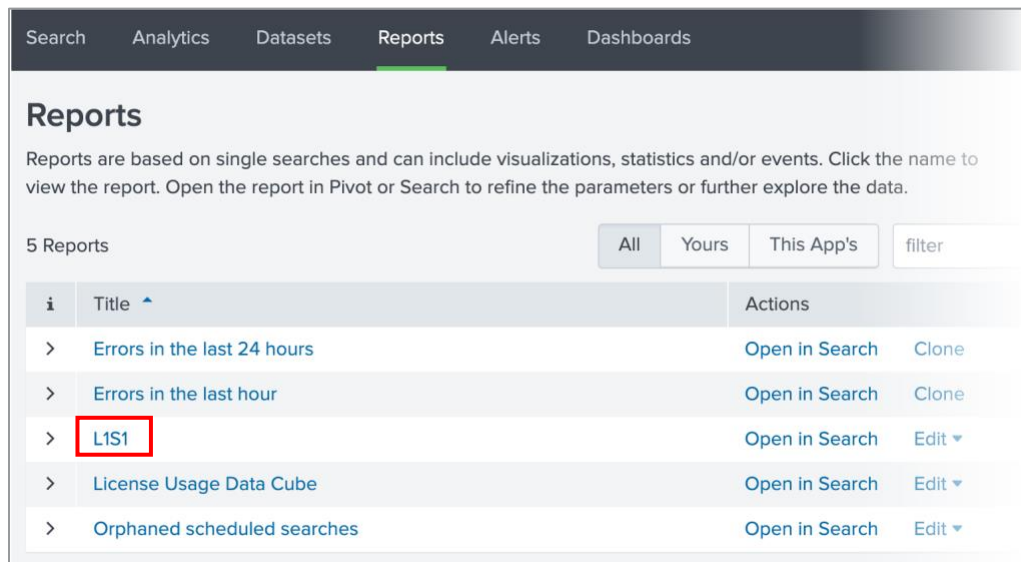
Values	Count	%
purchase	3,443	54.203%
addtocart	2,096	32.997%
view	482	7.588%
remove	176	2.771%
changequantity	155	2.44%

21. Edit your search to find events that do not have a value present for the **action** field. Run the search again.  
How many events are returned? \_\_\_\_\_  
`index=web sourcetype=access_combined NOT action=*`

This search will find events that do not have some value in the action field.

22. In the Fields sidebar, under **Interesting Fields**, try to find the **action** field.  
You should not see the action field because none of the returned events should contain that 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**.  
`index=web sourcetype=access_combined action=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**.  
How many events are returned? \_\_\_\_\_  
`index=web sourcetype=access_combined action!=purchase`
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.
 How many events are returned? \_\_\_\_\_  
`index=web sourcetype=access_combined NOT action=purchase`  
 The number of events returned from this search should be greater than the number of events returned from the previous search. If this is not the case, you should check your syntax, run the searches again, and be sure the time range is set to **Yesterday** for each search.
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**.

```
index=security sourcetype=linux_secure
```

31. Modify your search to look for:

a. Failed password attempts by invalid users by adding **failed invalid** to your basic search.

b. Events associated with the administrator user, i.e. user accounts that begin with **admin**.

```
index=security sourcetype=linux_secure failed invalid user=admin*
```

32. Use the **fields** command to extract only the **user**, **src\_ip**, and **app** fields.

<div>&lt; Hide Fields</div> <div>≡ All Fields</div>		i	Time	Event
INTERESTING FIELDS				
a app 1				
a src_ip 21				
a user 2				
+ Extract New Fields				
	>	10/9/19	7:44:41.000 PM	Thu Oct 10 2019 02:44:41 mailsv1 sshd[3354]: Failed password for invalid user administrator from 211.245.24.3 port 3993 ssh2
	>	10/9/19	7:44:41.000 PM	Thu Oct 10 2019 02:44:41 www3 sshd[5837]: Failed password for invalid user admin from 198.28.212.52 port 4919 ssh2
	>	10/9/19	7:40:58.000 PM	Thu Oct 10 2019 02:40:58 mailsv1 sshd[4039]: Failed password for invalid user admin from 170.192.178.10 port 2244 ssh2
	>	10/9/19	7:36:57.000 PM	Thu Oct 10 2019 02:36:57 www3 sshd[3025]: Failed password for invalid user admin from 85.62.218.82 port 4765 ssh2

```
index=security sourcetype=linux_secure failed invalid user=admin*
| fields user, src_ip, app
```

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>
```

Customer IP	Web Server	HTTP Status
49.212.64.138	www1	503
49.212.64.138	www1	505
60.220.218.88	www3	503
201.3.120.132	www3	503
58.68.236.98	www2	503

```
index=web sourcetype=access_combined action=purchase status>399
| table clientip host status
| rename clientip as "Customer IP", host as "Web Server", status as "HTTP Status"
```

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



## Lab Exercise 2 – Comparing 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”.

**index=security sourcetype=linux\_secure port**

2. Scroll through the list of events. Notice how many events have “Accepted password for...” and “Failed password for...”.

```
sshd[4399]: Failed password for invalid user syst
g/www1/secure.log | sourcetype = linux_secure

sshd[28822]: Accepted password for nsharpe from 15
g/www2/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.

**index=security sourcetype=linux\_secure port**  
**| erex event\_description examples="Failed password , Accepted password "**

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**.

event_description	count(src_port)
Accepted password	121
Failed password	2522
Server listening	0

**index=security sourcetype=linux\_secure port**  
**| erex event\_description examples="Failed password , Accepted password "**  
**| stats count(src\_port) by event\_description**

5. Observe your results. There appears to be no results for “Server listening”. What is wrong? (If you don’t see unusual results, expand your time range.)

When you view events for "Server listening" (click on Server listening in your results > **View events**) you will see that the associated port is a two-digit number. If you look at the **Interesting Fields** sidebar, you will also notice that there is no **src\_port** field indicating that Splunk does not recognize 22 as a port

number. (NOTE: "port" is highlighted because it shows up in the basic search.)

i	Time	Event
>	9/11/20 5:26:22.000 PM	Fri Sep 11 2020 17:26:22 www1 sshd[61899]: Server listening on :: port 22. host = www1   source = /opt/log/www1/secure.log   sourcetype = linux_secure

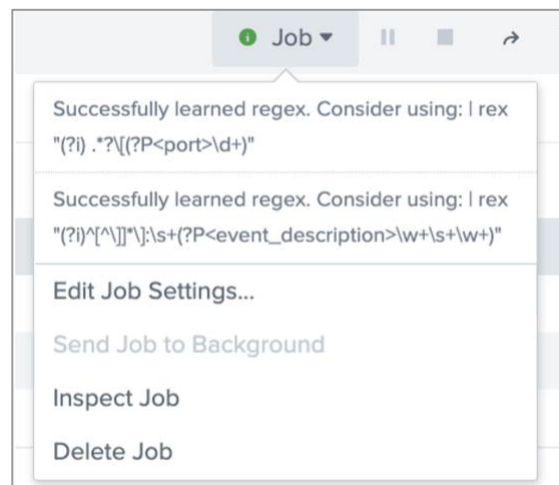
- Save your search as a report with the name **L2S1**. In Task 2, you will fix your search.

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

- 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.

```
index=security sourcetype=linux_secure port
| erex port examples="22,1229,1268"
| erex event_description examples="Failed password , Accepted password "
| stats count(port) by event_description
```

- 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**.)



- Replace your **erex** commands with the suggested **rex** commands.

event_description	count(port)
Accepted password	125
Failed password	2550
Server listening	43

```
index=security sourcetype=linux_secure port
| rex "(?i).*?\\[(?P<port>\\d+)"
| rex "(?i)^[^\\]]*\\]:\\s+(?P<event_description>\\w+\\s+\\w+)"
| stats count(port) by event_description
```

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