



Creating Field Extractions

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Lab Exercise slides reference the hands-on lab exercise guide
- Do not distribute

Course Goals

- Determine when the field extraction process occurs
- Utilize the Field Extractor (FX)
- Compare regex and delimited field extractions
- Create and use a regex field extraction
- Create and use a delimited field extraction
- Modify extracted fields

Course Outline

- Use the Field Extractor
- Create Regex Field Extractions
- Create Delimited Field Extractions

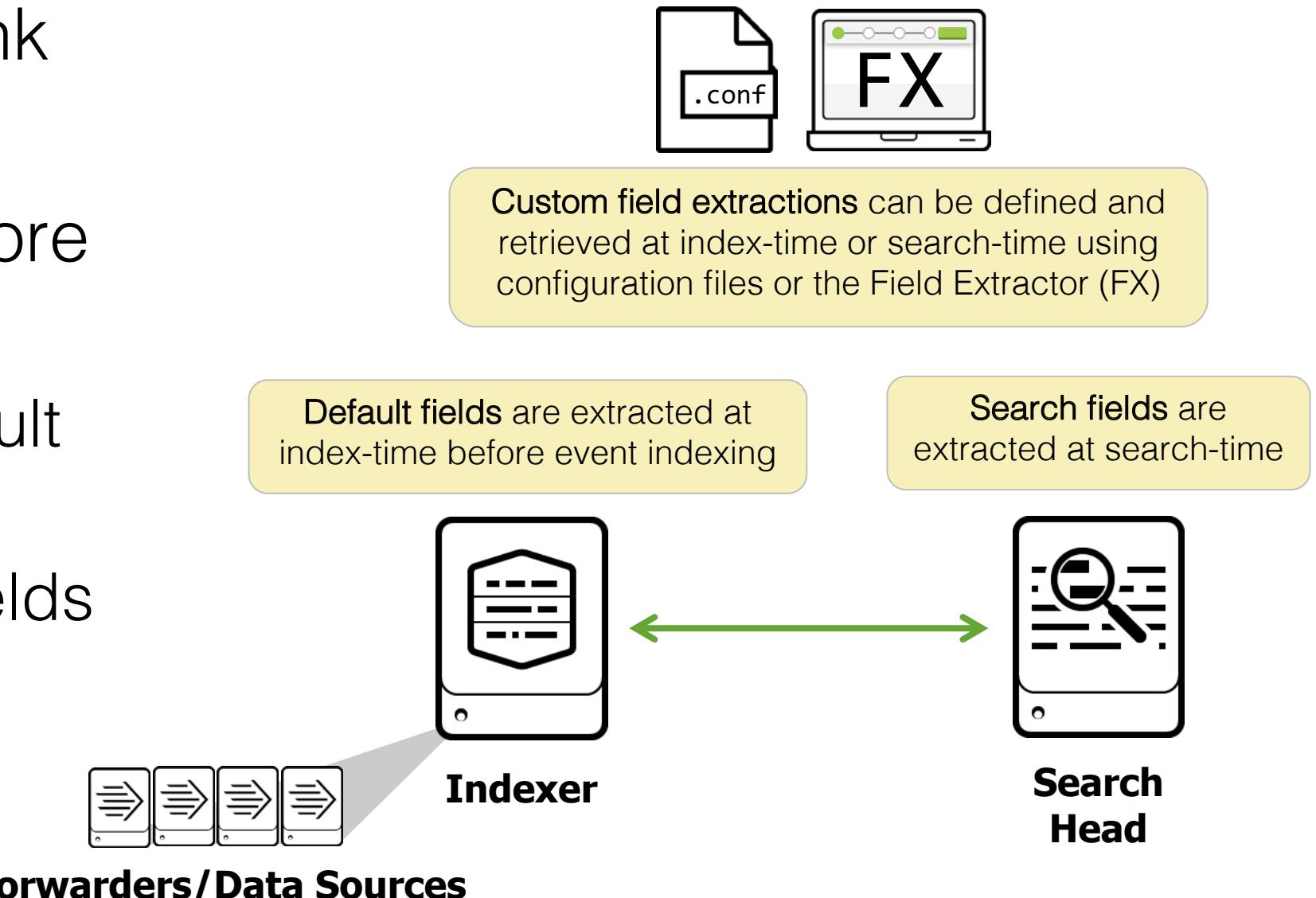
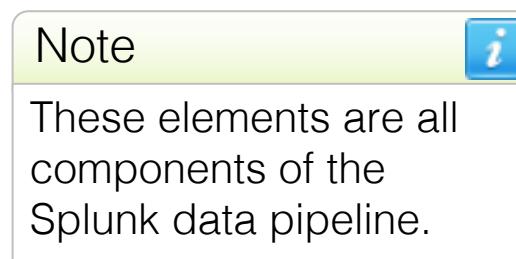
Use the Field Extractor

Topic Objectives

- Explore the different types of extracted fields and when they are extracted
- Define the Splunk Web Field Extractor (FX)

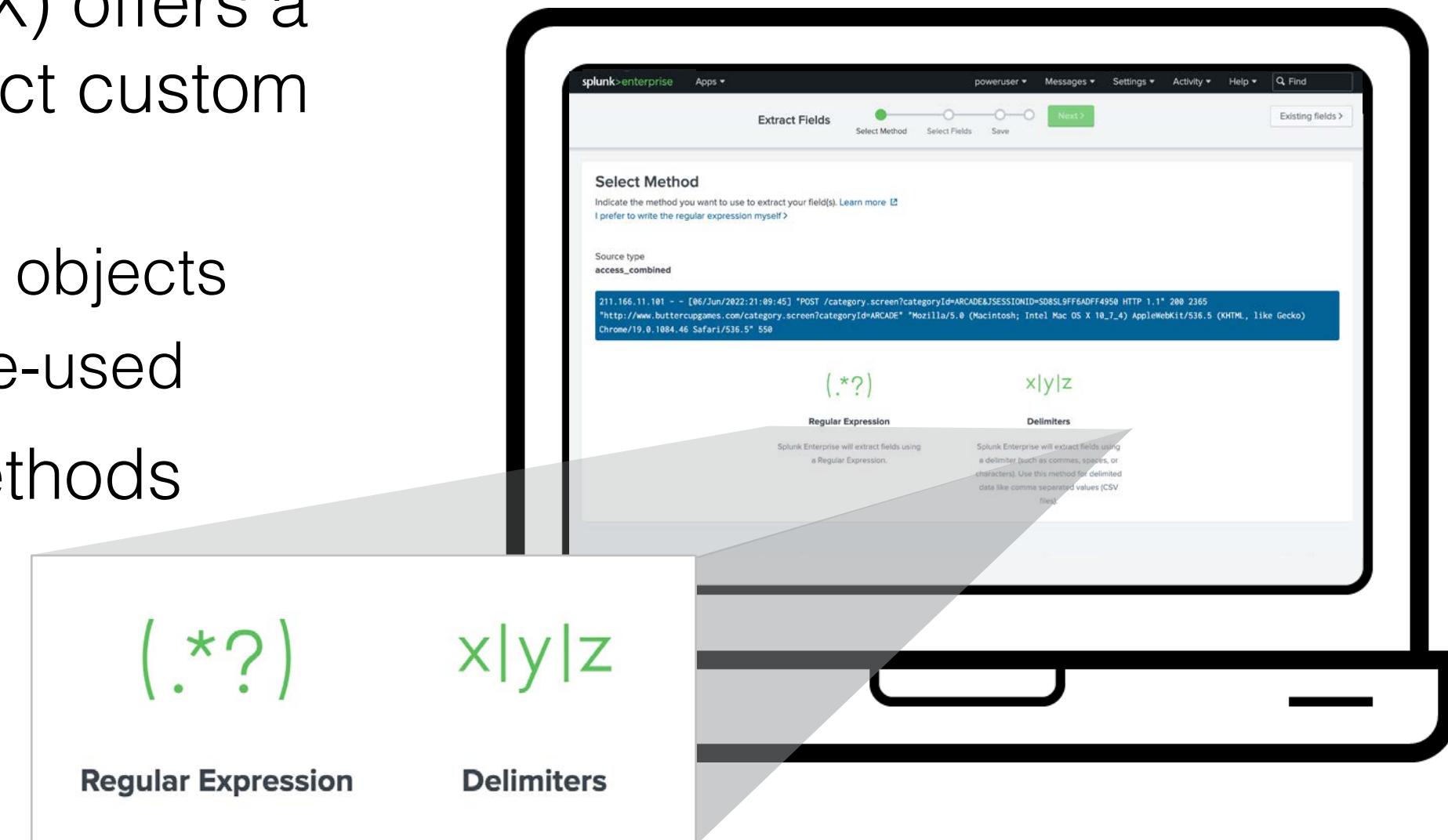
Explore Field Extraction

- The process by which Splunk pulls fields from event data
- Fields can be extracted before or after event indexing
 - Before: indexed fields (default and custom fields)
 - After: search and custom fields



The Field Extractor

- The Field Extractor (FX) offers a user interface to extract custom fields that:
 - Persist as knowledge objects
 - Can be shared and re-used
- The two extraction methods offered by the FX are **Regular Expression** and **Delimiters**



Access the Field Extractor

The FX user interface can be accessed via Settings, the Fields Sidebar, or the Event Actions menu

The screenshot illustrates three methods to access the Field Extractor:

- From the Settings page:** The top navigation bar shows "student1", "Messages", and "Settings". The "Fields" sidebar item is highlighted. The main content area displays a list of fields and their counts, with a callout pointing to the "+ Extract New Fields" link at the bottom right.
- From the Fields sidebar:** A callout points to the "Fields" sidebar item, which is currently selected.
- From the Event Actions menu:** An event table on the right shows a row for a log entry from "6/6/22". A context menu is open over this row, with the "Event Actions" option highlighted. A sub-menu titled "Build Event Type" is open, showing the "Extract Fields" option, which is also highlighted with a mouse cursor.

Create Regex Field Extractions

Topic Objectives

- Identify basics of regular expressions (regex)
- Explore the regex field extraction workflow
- Edit regex for field extractions

Field Extractor Method: Regex

- FX extracts fields using a Regular Expression (regex)
 - Generates regex from user-selected event data
 - Regex can be manually edited for more accuracy
- Use this option when your event contains unstructured data (e.g., a system log file)

What Is Regex?

- A regex (regular expression) is a case-sensitive sequence of characters defining a pattern
- Each character is either a regular character (with literal meaning) or a metacharacter (with special meaning)
- Widely used in programming and scripting languages for a variety of string processing tasks

regex example for U.S. email addresses

```
\b[a-zA-Z0-9._%+-]+@[a-zA-Z0-9_-]+\.[a-zA-Z]{2,}\b
```

Basics of Regex

regex

cat

Regular characters are treated literally

c.t

A **.** is treated as a wildcard and will match any character

c\t

A **** is used to “escape” characters so they can be treated as literal characters

matches

cat

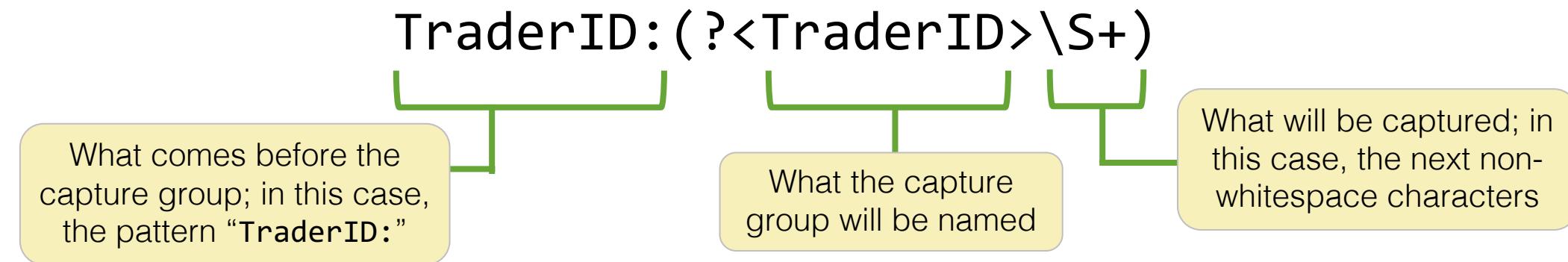
cat
cut
c1t
c#t

...any many others...

c.t

Regex Captures

- Regex can “capture” part of the matching pattern by using ()
- You can reference the capture by giving it a name using: ?<name>



- With some older versions of regex, a “P” must be inserted in order to perform named captures (?P<>)

Regex Examples

`user\s(\w+)` captures the word following user

Failed password for invalid user `fpass` from 211.24.4.4

Successfully captures `fpass`

Failed password for invalid user `jean-luc` from 211.25.4.4

Doesn't successfully capture `jean-luc` because “-” isn't a “word” character

`user\s(\S+)`

Failed password for invalid user `jean-luc` from 211.25.4.4

Successfully captures `jean-luc`

Regex Best Practices

> Best Practice

- Avoid “backtracking” by writing simple and concise regex
 - Backtracking occurs when the regex engine must return to a previously saved state to continue its search for a match, causing the engine to make multiple passes
- Quantifiers (e.g. *) and alternation constructs (e.g. |) are powerful but can slow performance by causing backtracking
 - Use + rather than *
 - Avoid using multiple .* matches
 - Avoid greedy operators (.*), use non-greedy (.*?) instead

Regex Field Extraction Workflow

The screenshot illustrates the process of creating a new field extraction in Splunk. It shows the navigation bar with 'student1', 'Messages', and 'Settings'. On the left, under 'KNOWLEDGE', the 'Fields' link is highlighted with a red circle containing the number 1. The main content area is titled 'Fields' with the sub-instruction: 'View, edit, and set permissions on field extractions. Define event workflow actions and field aliases. Rename sourcetypes.' Below this, there are five sections: 'Field aliases', 'Calculated fields', 'Field extractions' (which is selected and highlighted with a green box and the number 2), 'Field transformations', and 'Sourcetype renaming'. At the bottom, there are two green buttons: 'New Field Extraction' (with the number 3) and 'Open Field Extractor' (with a mouse cursor icon).

student1 ▾ Messages ▾ Settings ▾

Add Data

1 Fields

KNOWLEDGE

- Searches, reports, and alerts
- Data models
- Event types
- Tags
- 2 Field extractions
- Lookups
- User interface

Fields

View, edit, and set permissions on field extractions. Define event workflow actions and field aliases. Rename sourcetypes.

Field aliases

Edit or add one or more aliases to field names

+ Add new

Calculated fields

Edit or add one or more calculated fields

+ Add new

Field extractions

View and edit all field extractions. Add new field extractions and update permissions.

+ Add new

Field transformations

Edit or add transformations for field extractions that use a transform.

+ Add new

Sourcetype renaming

Rename a source type. Multiple source types can share the same name.

+ Add new

Field extractions

Fields » Field extractions

New Field Extraction

3 Open Field Extractor

Regex Field Extraction Workflow (cont.)

The screenshot shows the 'Extract Fields' workflow in Splunk. The current step is 'Select Sample'. The process consists of five steps: Select Sample (green dot), Select Method, Select Fields, Save, and Next >. A 'Existing fields >' link is also present.

Select Sample Event

Choose a source or source type, select a sample event, and click Next to go to the next step. The field extractor will use the event to extract fields. [Learn more](#)

Data Type: sourcetype ▾

Source Type: Select Source Type ▾

filter

4 Choose sourcetype or source

5 Use drop-down menu or search

Application: dmesg
Database: Output produced by the "dmesg" *nix command, printing the *nix kernel ring buffer
Email:
Log to Metrics:
Metrics:
Miscellaneous:
Network & Security:
Operating System: linux_audit
linux_audit: Output produced by the auditd system daemon used to track changes on a Linux machine
Structured:
Uncategorized:
Web:
linux_messages_syslog: Format found within the Linux log file /var/log/messages
linux_secure: Format for the /var/log/secure file containing

Regex Field Extraction Workflow (cont.)

The chosen sample event appears in a blue box

Extract Fields

Select Sample Select Method Select Fields Save

Next > 7 Existing fields >

Select Sample Event

Choose a source or source type, select a sample event, and click Next to go to the next step. The field extractor will use the event to extract fields. [Learn more](#)

I prefer to write the regular expression myself >

Data Type: sourcetype

Source Type: linux_secure

Time Range: Last 90 days

Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2

Events

✓ 1,000 events (3/8/22 12:00:00.000 AM to 6/6/22 9:47:29.000 PM)

20 per page ▾ < Prev 1 2 3 4 5 6 7 8 ... Next >

filter Apply Sample: 1,000 events ▾ All events ▾

_raw ▾

Jun 06 2022 21:47:28 mailsv1 sshd[5676]: Failed password for squid from 59.36.99.70 port 3521 ssh2

Jun 06 2022 21:47:14 mailsv1 su: pam_unix(su:session): session closed for user root

Regex Field Extraction Workflow (cont.)

The screenshot shows the 'Extract Fields' workflow in Splunk. The current step is 'Select Method', which is highlighted by a green dot in the progress bar. The steps are: Select Sample, Select Method (highlighted), Select Fields, Validate, Save. A 'Next >' button is at the end of the bar, with a red circle containing the number '9' above it. To the right of the bar is a link 'Existing fields >'. The main area is titled 'Select Method' and contains instructions: 'Indicate the method you want to use to extract your field(s). [Learn more](#)' and '[I prefer to write the regular expression myself >](#)'. Below this, the 'Source type' is set to 'linux_secure'. A sample log entry is shown: 'Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2'. On the left, there are two options: 'Regular Expression' (selected) and 'Delimiters'. The 'Regular Expression' section contains a box with the pattern '(.*?)' and a cursor icon pointing to it. A red circle with the number '8' is positioned above the box. The 'Delimiters' section shows the text 'x|y|z'.

Extract Fields

Select Sample Select Method Select Fields Validate Save

< Back Next > 9

Existing fields >

Select Method

Indicate the method you want to use to extract your field(s). [Learn more](#)

[I prefer to write the regular expression myself >](#)

Source type
linux_secure

Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2

8

(.*?)

Regular Expression

Splunk Enterprise will extract fields using a Regular Expression.

x|y|z

Delimiters

Splunk Enterprise will extract fields using a delimiter (such as commas, spaces, or characters). Use this method for delimited data like comma separated values (CSV files).

Regex Field Extraction Workflow (cont.)

The screenshot shows the 'Extract Fields' workflow in Splunk, specifically the 'Select Fields' step. The workflow steps are: Select Sample, Select Method, Select Fields, Validate, Save, and Next >. The 'Select Fields' step is currently active, indicated by a green dot on the progress bar.

Select Fields

Highlight one or more values in the sample event to create fields. You can indicate one value is required, meaning it must exist in an event for the regular expression to match. Click on highlighted values in the sample event to modify them. To highlight text that is already part of an existing extraction, first turn off the existing extractions.

Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2

Use cursor to highlight value

10

Field Name: port

Sample Value: 3057

11 Type in Field Name

12 Add Extraction

Existing fields >

Regex Field Extraction Workflow (cont.)

Select Fields

Highlight one or more values in the sample event to create fields. You can indicate one value is required, meaning it must exist in an event for the regular expression to match. Click on highlighted values in the sample event to modify them. To highlight text that is already part of an existing extraction, first turn off the existing extractions. [Learn more](#)

```
Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2
```

Show Regular Expression > View in Search

Preview

If you see incorrect results below, click an additional event to add it to the set of sample events. Highlight its values to improve the extraction. You can remove incorrect values in the next step.

Events port

✓ 1,000 events (3/8/22 12:00:00.000 AM to 6/6/22 10:07:14.000 PM) 20 per page ▾ < Prev 1 2 3 4 5 6 7 8 ... Next >

filter Apply Sample: 1,000 events ▾ All events ▾ All Events Matches Non-Matches

_raw	port
✓ Mon Jun 06 2022 22:07:06 www2 sshd[26269]: Accepted password for djohnson from 10.3.10.46 port 5931 ssh2	5931
✓ Mon Jun 06 2022 22:06:48 www2 sshd[5643]: Failed password for invalid user whois from 10.2.10.163 port 1356 ssh2	1356
✓ Mon Jun 06 2022 22:06:30 www2 sshd[5525]: Failed password for invalid user whois from 10.2.10.163 port 2491 ssh2	2491
✓ Mon Jun 06 2022 22:06:18 www2 sshd[5981]: Failed password for invalid user fpass from 10.2.10.163 port 2020 ssh2	2020
✓ Mon Jun 06 2022 22:06:06 www2 sshd[4631]: Failed password for sync from 10.2.10.163 port 3313 ssh2	3313
✗ Mon Jun 06 2022 22:05:53 www2 sudo: djohnson ; TTY=pts/0 ; PWD=/home/djohnson ; USER=root ; COMMAND=/bin/su	
✓ Mon Jun 06 2022 22:05:39 www2 sshd[5870]: Failed password for invalid user admin from 10.2.10.163 port 2013 ssh2	2013
✓ Mon Jun 06 2022 22:05:31 www2 sshd[3813]: Failed password for root from 10.2.10.163 port 3984 ssh2	3984

Selected fields appear highlighted in the events

Regex Field Extraction Workflow (cont.)

The screenshot shows the 'Extract Fields' workflow in Splunk, specifically the 'Select Fields' step. The top navigation bar includes 'Extract Fields', a progress bar with steps 'Select Sample' (green), 'Select Method' (green), 'Select Fields' (gray), 'Validate' (light gray), and 'Save' (light gray), and buttons for '< Back', 'Next > 16', and 'Existing fields >'.

The main area is titled 'Select Fields' and contains instructions: 'Highlight one or more values in the sample event to create fields. You can indicate one value is required, meaning it must exist in an event for the regular expression to match. Click on highlighted values in the sample event to modify them. To highlight text that is already part of an existing extraction, first turn off the existing extractions.' Below this is a sample event:

```
Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2
```

Below the event, there's a 'Show Regular Expression >' link and a modal window for extraction configuration. The modal has fields for 'Field Name' (srcl), 'Sample' (10.1.10.172), and 'Value' (10.1.10.172). It includes 'Extract' and 'Require' buttons, and an 'Add Extraction' button. A yellow callout bubble labeled '15' says 'Add extractions for additional sample events'.

The 'Preview' section below the modal shows a list of events with a 'port' filter applied. It displays 1,000 events from 3/8/22 to 6/6/22. A callout bubble labeled '14' says 'Use the Preview pane to add additional events to refine regex'.

At the bottom, there's a table of events with columns for '_raw', '_index', and '_score'. The table shows four events related to failed password attempts for users 'operator', 'whois', and 'fpass' from various IP addresses and ports.

_raw	_index	_score
✓ Mon Jun 06 2022 22:07:06 www2 sshd[26269]: Accepted password for djohnson from 10.3.10.46 port 5931 ssh2		
✓ Mon Jun 06 2022 22:06:48 www2 sshd[5643]: Failed password for invalid user whois from 10.2.10.163 port 1356 ssh2		
✓ Mon Jun 06 2022 22:06:30 www2 sshd[5525]: Failed password for invalid user whois from 10.2.10.163 port 2491 ssh2		2491
✓ Mon Jun 06 2022 22:06:18 www2 sshd[5981]: Failed password for invalid user fpass from 10.2.10.163 port 2020 ssh2		2020

Regex Field Extraction Workflow (cont.)

The screenshot shows the 'Extract Fields' workflow in Splunk. The current step is 'Validate', indicated by a green dot on the progress bar. A yellow callout box points to the validate section with the text: 'Use this screen to verify that the proper field values are selected'. The 'Events' tab is selected, showing 1,000 events from March 8, 2022, to June 6, 2022. The 'src' field is highlighted in blue. The interface includes a 'Show Regular Expression >' button, a 'View in Search' button, and a 'filter' input field with an 'Apply' button. Below the event list, there are dropdowns for 'Sample: 1,000 events', 'All events', and 'All Events / Matches / Non-Matches'. The event list itself has columns for '_raw', 'src', and 'port'. Three log entries are shown:

_raw	src	port
✓ Mon Jun 06 2022 22:18:18 www2 sshd[4397]: Failed password for invalid user testing from 88.191.145.142 X port 3803 X ssh2	88.191.145.142	3803
✓ Mon Jun 06 2022 22:18:03 www2 sshd[1075]: Failed password for invalid user uni from 88.191.145.142 X port 2430 X ssh2	88.191.145.142	2430
✓ Mon Jun 06 2022 22:17:48 www2 sshd[2571]: Failed password for invalid user admin from 88.191.145.142 X port 4710 X ssh2	88.191.145.142	4710

A yellow callout box with orange border and number 18 points to the 'X' button next to the port value '3803' in the first log entry, with the text: 'Click the X to update the regex to remove invalid values'.

Regex Field Extraction Workflow (cont.)

Extract Fields

Select Sample Select Method Select Fields Validate Save < Back Finish > (21)

Save

Name the extraction and set permissions.

Extractions Name EXTRACT- src,port

Owner poweruser

App search

Permissions Owner App All apps (20) Set Permissions

Source type linux_secure

Sample event Mon Jun 06 2022 21:44:09 mailsv1 sshd[1415]: Failed password for invalid user operator from 10.1.10.172 port 3057 ssh2

Fields src,port

Regular Expression ^\w+\s+\w+\s+\d+\s+\d+\s+\d+:\d+:\d+\s+\w+\d+\s+\w+[\d+]:\s+\w+\s+\w+\s+\w+\s+\w+\s+\w+\s+\w+\s+(?P<src>[^]+) port (?P<port>\d+)

19 Review Extractions Name

Use the Extracted Fields

< Hide Fields All Fields Time Event

SELECTED FIELDS

- [a host](#) 4
- [a source](#) 4
- [a sourcetype](#) 1

INTERESTING FIELDS

- [a action](#) 2
- [a app](#) 2
- [# date_hour](#) 14
- [# date_mday](#) 2
- [# date_minute](#) 60
- [a date_month](#) 1
- [# date_second](#) 60
- [a date_wday](#) 2
- [# date_year](#) 1
- [a date_zone](#) 1
- [a dest](#) 4
- [a index](#) 1
- [# linecount](#) 1
- [# pid](#) 100+
- [# port](#) 100+
- [a process](#) 3
- [a punct](#) 9
- [a splunk_server](#) 1
- [a src](#) 58
- [a src_ip](#) 60
- [# src_port](#) 100+
- [a sshd_protocol](#) 1
- [# timeendpos](#) 1
- [# timestamppos](#) 1
- [a user](#) 100+

src

58 Values, 58.898% of events

Selected Yes No

Reports

[Top values](#) [Top values by time](#) [Rare values](#)

Events with this field

Top 10 Values	Count	%
10.2.10.163	50	9.746%
10.1.10.172	33	6.433%
87.194.216.51	28	5.458%
175.45.177.66	23	4.483%
10.3.10.46	22	4.288%
192.162.19.179	18	3.509%
211.166.11.101	18	3.509%
182.236.164.11	16	3.119%
212.114.31.255	15	2.924%
223.213.255.255	15	2.924%

> 6/6/22 Mon Jun 06 2022 22:24:50 www2 sshd[51603]: Failed password for djohnson from 10.3.10.46 port 9139 ssh2
host = www2 | source = /opt/log/www2/secure.log | sourcetype = linux_secure

> 6/6/22 Mon Jun 06 2022 22:24:45 www2 sshd[1697]: Failed password for invalid user nginx from 91.208.184.24 port 2699 ssh2
host = www2 | source = /opt/log/www2/secure.log | sourcetype = linux_secure

> 6/6/22 Mon Jun 06 2022 22:24:35 www2 sshd[2714]: Failed password for invalid user info from 91.208.184.24 port 3292 ssh2
host = www2 | source = /opt/log/www2/secure.log | sourcetype = linux_secure

... (more log entries)

> 6/6/22 Mon Jun 06 2022 22:23:20 www2 sshd[5764]: Failed password for invalid user zabbix from 99.61.68.230 port 2224 ssh2
host = www2 | source = /opt/log/www2/secure.log | sourcetype = linux_secure

Edit Regex for Field Extractions

Regular Expression can be edited during the **Select Fields** step

Edit Regex for Field Extractions (cont.)

Extract Fields [Back](#)

Warning  After you edit the regular expression, you cannot go back to the Field Extractor UI.

Regular Expression [Regular Expression Reference](#) [View in Search](#)

```
^\w+\s+\w+\s+\d+\s+\d+\s+\d+:\d+:\d+\s+\w+\d+\s+\w+\|\d+\|:\s+\w+\s+\w+\s+\w+\s+\w+\s+\w+\s+(?P<src>[^ ]+) port (?P<port>\d+)
```

Use Preview to verify your custom expression

[Preview](#) [Save](#)

Create Regex Field Extractions Lab Exercise

Time: 15 minutes

Tasks:

- Use the Field Extractor (FX) to extract fields using the Regular Expression method

Create Delimited Field Extractions

Topic Objectives

- Identify delimited field values in event data
- Explore the delimited field extraction workflow

Delimited Field Values in Event Data

Delimited field values exist in consistently structured events and are separated by spaces, commas, or characters (i.e. delimiters)

"2018-01-30T22:43:39000-0400",29,1>Error,HOST0167,System,772103058

Values separated by commas

i	Time	Event
>	1/30/18 10:43:39.000 PM	"2018-01-30T22:43:39000-0400",29,1>Error,HOST0167,System,772103058 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit
>	1/30/18 10:43:37.000 PM	"2018-01-30T22:43:37000-0400",35,4,Information,HOST0201,System,507701378 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit
>	1/30/18 10:43:36.000 PM	"2018-01-30T22:43:36000-0400",35,4,Information,HOST0201,System,753380719 host = adldapsv1 source = /opt/log/adldapsv1/sysmonitor.log sourcetype = win_audit

7036 4 Information HOST0167 System 89278068

Values separated by spaces

i	Time	Event
>	8/21/19 4:55:52.000 PM	7036 4 Information HOST0167 System 89278068 host = bcg1 source = sysmonitor.log sourcetype = win_audit
>	8/21/19 4:55:50.000 PM	17 1 Error HOST0167 System 758575060 host = bcg1 source = sysmonitor.log sourcetype = win_audit
>	8/21/19 4:55:49.000 PM	35 4 Information HOST0201 System 855753635 host = bcg1 source = sysmonitor.log sourcetype = win_audit

Delimited Field Extraction Workflow

New Search

index=games sourcetype=SimCubeBeta 1

✓ 66 events (5/8/22 12:00:00.000 AM to 6/7/22 3:31:15.000 PM) Last 30 days 🔍

Events (66) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect 1 day per column

List ▾ Format 20 Per Page ▾ 1 2 3 4 Next >

< Hide Fields All Fields

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
date_hour 2
date_mday 1
date_minute 49
a date_month 1
date_second 40
a date_wday 1
date_year 1
a date_zone 1
a index 1

Time Event

6/7/22 07/Jun/2022:15:31:15 , 183.60.133.18 , v2.003B , User:'quint@msn.com' CharacterName:'quetee' Action:'Filed TPS Report' CurrentStanding:'Office Joke'

Event Actions ▾

Build Event Type

Value	Actions
sim_cube_server	▼
/opt/log/SIMlog/simgame.log	▼
<input checked="" type="checkbox"/> sourcetype ▾	SimCubeBeta
Time + _time ▾	2022-06-07T15:31:15.000+00:00
Default index ▾	games
linecount ▾	1
punct ▾	//::_,_,...,_,_,':@'.':_!_!_!_!

Extract Fields 2

Show Source

Delimited Field Extraction Workflow (cont.)

Extract Fields  Next > 

Select Method

Indicate the method you want to use to extract your field(s). [Learn more](#) 
[I prefer to write the regular expression myself](#) >

Source type
SimCubeBeta

07/Jun/2022:15:31:15 , 183.60.133.18 , v2.003B , User:'quint@msn.com' CharacterName:'quetee' Action:'Filed TPS Report' CurrentStanding:'Office Joke'



Regular Expression

Splunk Enterprise will extract fields using
a Regular Expression.



Delimiters

Splunk Enterprise will extract fields using
a delimiter (such as commas, spaces, or
characters). Use this method for delimited
data like comma separated values (CSV
files).



Delimited Field Extraction Workflow (cont.)

Extract Fields Select Method Rename Fields Save < Back Next > 9

Rename Fields

Select a delimiter. In the table, click on fields by clicking on field names or values. Learn more ↗

5 Select delimiter

Delimiter

Space Comma Tab Pipe Other

field1 field2 field3 field4

07/Jun/2022:15:31:15 183.60.133.18

Field Name field4

6 7 8

CharacterName:'quetee' Action:'Filed TPS Report' CurrentStanding:'Office Joke'

Preview (4 fields)

Events field1 field2 field3 field4

✓ 66 events (3/9/22 12:00:00.000 AM to 6/7/22 3:56:05.000 PM)

20 per page ▾ < Prev 1 2 3 4 Next >

filter Apply Sample: 1,000 events ▾ All events ▾ All Events Matches Non-Matches

Delimited Field Extraction Workflow (cont.)

Extract Fields 

Save
Name the extraction and set permissions.

Extractions Name **REPORT- simgame_log** 10 Name the extraction

Owner **poweruser**

App **search**

Permissions Owner App All apps

Source type **SimCubeBeta**

Sample event **07/Jun/2022:15:31:15 , 183.60.133.18 , v2.003B , User:'quint@msn.com' CharacterName:'quetee'
Action:'Filed TPS Report' CurrentStanding:'Office Joke'**

Fields **time,src,version,misc** Fields

Delimiter **comma**

Use a Delimited Field Extraction

New Search

index=_* OR index==* sourcetype=SimCubeBeta

99 events (6/6/22 4:00:00.000 PM to 6/7/22 4:03:53.000 PM) No Event Sampling ▾

Events (99) Patterns Statistics Visualization

Format Timeline ▾ — Zoom Out + Zoom to Selection × Deselect 1 hour per column

List ▾ Format 20 Per Page ▾

1 2 3 4 5 Next >

Time	Event																																												
6/7/22 4:03:12.000 PM	07/Jun/2022:16:03:12 , 170.192.178.10 , v2.003B , User:'rita_gagnon@hotmail.com' CharacterName:'mario' Action:'Stole From Petty Cash' CurrentStanding:'Office Joke' Event Actions ▾ <table border="1"><thead><tr><th>Type</th><th>Field</th><th>Value</th><th>Actions</th></tr></thead><tbody><tr><td>Selected</td><td>host</td><td>sim_cube_server</td><td>▼</td></tr><tr><td></td><td>source</td><td>/opt/log/SIMlog/simgame.log</td><td>▼</td></tr><tr><td></td><td>sourcetype</td><td>SimCubeBeta</td><td>▼</td></tr><tr><td>Event</td><td>misc</td><td>User:'rita_gagnon@hotmail.com' CharacterName:'mario' Action:'Stole From Petty Cash' CurrentStanding:'Office Joke'</td><td>▼</td></tr><tr><td></td><td>src</td><td>170.192.178.10</td><td>▼</td></tr><tr><td></td><td>time</td><td>07/Jun/2022:16:03:12</td><td>▼</td></tr><tr><td></td><td>version</td><td>v2.003B</td><td>▼</td></tr><tr><td>Time</td><td>_time</td><td>2022-06-07T16:03:12.000+00:00</td><td>▼</td></tr><tr><td>Default</td><td>index</td><td>games</td><td>▼</td></tr><tr><td></td><td>linecount</td><td>1</td><td>▼</td></tr></tbody></table>	Type	Field	Value	Actions	Selected	host	sim_cube_server	▼		source	/opt/log/SIMlog/simgame.log	▼		sourcetype	SimCubeBeta	▼	Event	misc	User:'rita_gagnon@hotmail.com' CharacterName:'mario' Action:'Stole From Petty Cash' CurrentStanding:'Office Joke'	▼		src	170.192.178.10	▼		time	07/Jun/2022:16:03:12	▼		version	v2.003B	▼	Time	_time	2022-06-07T16:03:12.000+00:00	▼	Default	index	games	▼		linecount	1	▼
Type	Field	Value	Actions																																										
Selected	host	sim_cube_server	▼																																										
	source	/opt/log/SIMlog/simgame.log	▼																																										
	sourcetype	SimCubeBeta	▼																																										
Event	misc	User:'rita_gagnon@hotmail.com' CharacterName:'mario' Action:'Stole From Petty Cash' CurrentStanding:'Office Joke'	▼																																										
	src	170.192.178.10	▼																																										
	time	07/Jun/2022:16:03:12	▼																																										
	version	v2.003B	▼																																										
Time	_time	2022-06-07T16:03:12.000+00:00	▼																																										
Default	index	games	▼																																										
	linecount	1	▼																																										

Create Delimited Field Extractions Lab Exercise

Time: 15 minutes

Tasks:

- Use the Field Extractor (FX) to extract fields using the delimiters method
- Use the regex method to extract additional fields that were not captured using the delimiters method

Wrap-up Slides

Community

- Splunk Community Portal
community.splunk.com
 - Answers
 - Discussions
 - Splunk Trust
 - User Groups
 - Ideas
- Splunk Blogs
splunk.com/blog/
- Splunk Apps
splunkbase.com
- Splunk Dev Google Group
groups.google.com/forum/#!forum/splunkdev
- Splunk Docs on Twitter
twitter.com/splunkdocs
- Splunk Dev on Twitter
twitter.com/splunkdev
- Splunk Live!
splunklive.splunk.com
- .conf
conf.splunk.com

Support Programs

- Web
 - Documentation: dev.splunk.com and docs.splunk.com
- Splunk Lantern
 - Guidance from Splunk experts
 - lantern.splunk.com
- Global Support
 - Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365
 - Web: splunk.com/index.php/submit_issue
- Enterprise, Cloud, ITSI, Security Support
 - Web: splunk.com/en_us/about-splunk/contact-us.html#tabs/customersupport
 - Phone: (855) SPLUNK-S or (855) 775-8657

Support ^

Support Portal

Submit a case ticket

Splunk Answers

Ask Splunk experts questions

Contact Us

Contact our customer support

Product Security Updates

Keep your data secure

System Status

Learning Paths (cont.)

Knowledge Manager - Recommended Courses

Free eLearning courses are in [blue](#) and courses with an * are present in both learning paths.

- [What is Splunk *](#)
- [Introduction to Splunk *](#)
- [Using Fields *](#)
- [Introduction to Knowledge Objects](#)
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- [Introduction to Dashboards](#)
- Dynamic Dashboards
- Using Choropleth
- Search Optimization *

Learning Paths

Search Expert - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

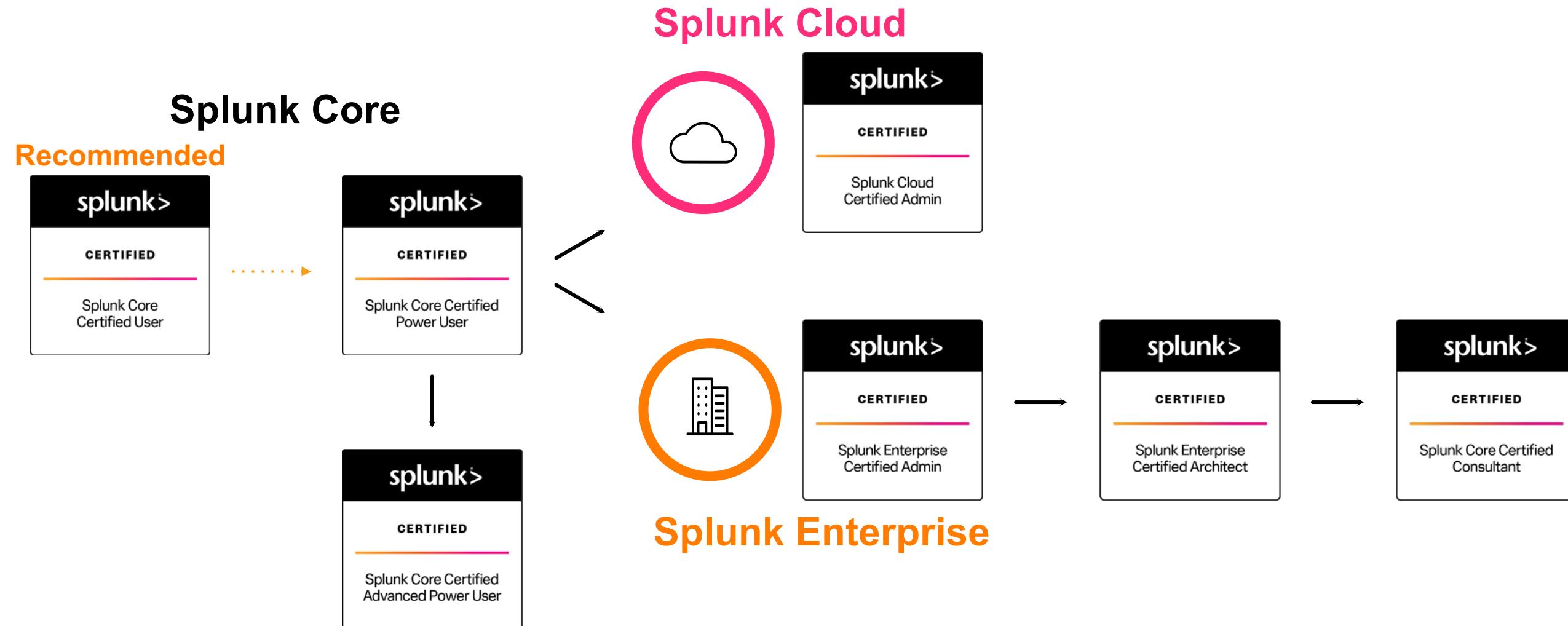
- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization *

Splunk Certification

Offerings & Requirements

Splunk Core and Beyond

Regardless of which Splunk product you use, it all starts with Splunk Core



Splunk Core Certified Power User

This entry-level certification demonstrates an individual's foundational competence of Splunk's core software



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

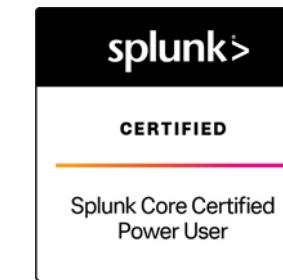
Splunk Core Certified Power User Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete Fundamentals 2 **or** the following courses:

- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Correlation Analysis
- Search Under the Hood
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Data Models
- Using Choropleth

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Core Certified Advanced Power User](#)
- [Splunk Enterprise Certified Admin](#)
- [Splunk Cloud Certified Admin](#)

Splunk Core Certified Advanced Power User

This certification demonstrates an individual's ability to generate complex searches, reports, and dashboards with Splunk's core software to get the most out of their data



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)

Prerequisite Course(s):

- None

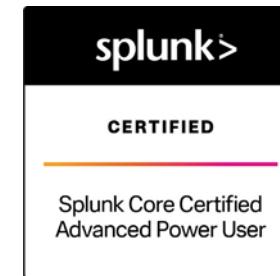
Splunk Core Certified Advanced Power User Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete Fundamentals 3, Creating Dashboards, and Advanced Searching & Reporting **or** the following courses:

- Using Fields
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Using Choropleth
- Introduction to Dashboards
- Dynamic Dashboards

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Enterprise Certified Admin](#)
- [Splunk Cloud Certified Admin](#)

Thank You

splunk>