Comparing Values

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Goals

- Compare values
- Use conditional statements
- Use the **`eval`** command with comparison, conditional, and text functions
- Use multiple functions together

Course Outline

- Using **`eval`** to Compare
- Filtering with **`where`** & Managing Missing Data

Using `eval` to Compare

Topic Objectives

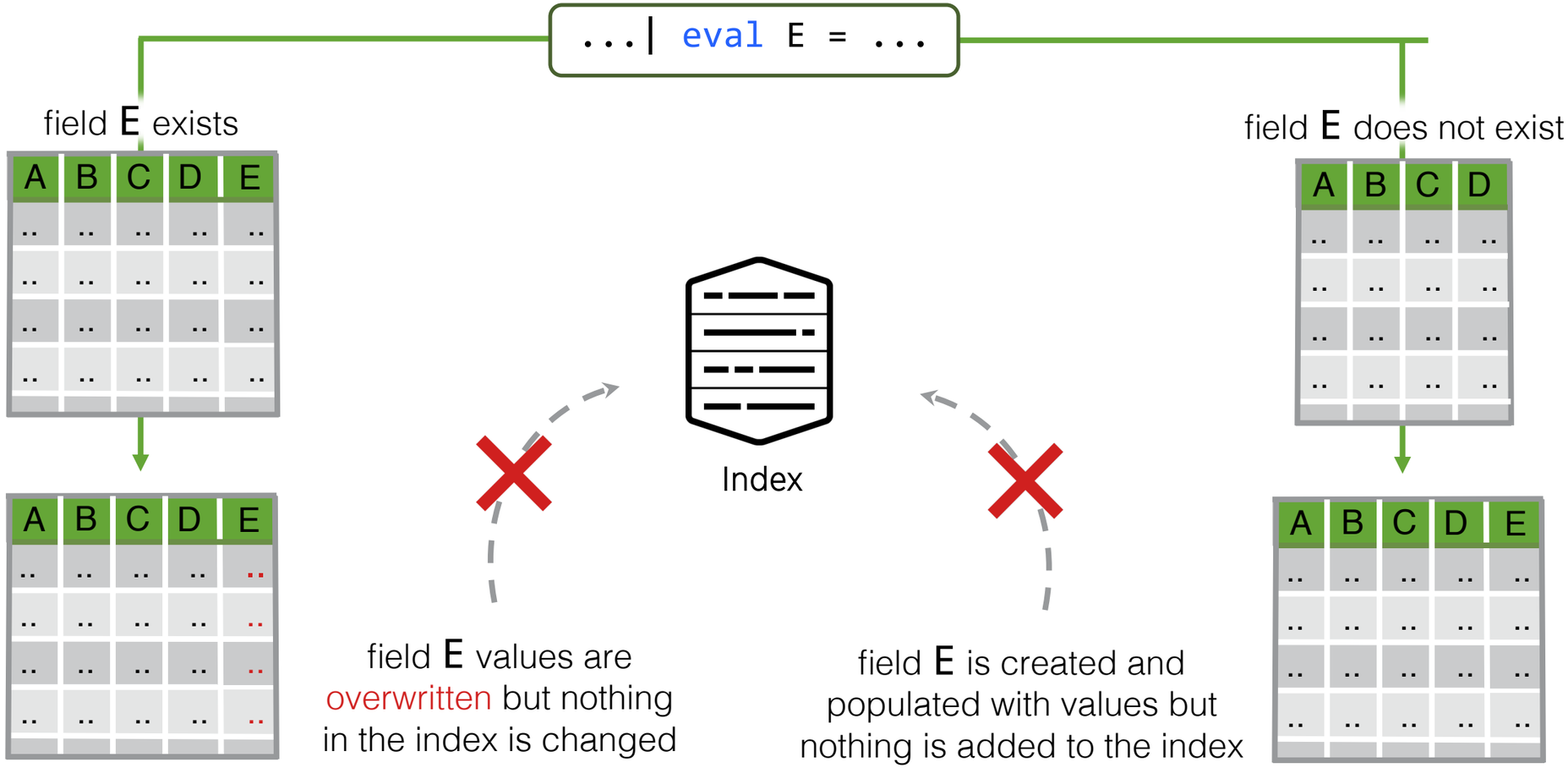
- Explore the **`eval`** command
- Identify and use comparison, conditional, and text functions
- Normalize data with the **`case`** function
- Use the **`fieldformat`** command to format field values

`eval` Command

```
...| eval <field1>=<expression1>[, <field2>=<expression2>]
```

- Calculates an expression and puts the resulting value into a new or existing field which can be reused in the search pipeline
- Extremely powerful and useful command that supports a vast assortment of functions
- Can exist as an expression

`eval` Command (cont.)



`eval` Command (cont.)

The `eval` command supports various operators

Type	Operators
arithmetic	`+ - * / %`
concatenation	`+ .`
Boolean	`AND OR NOT XOR`
comparison	`< > <= >= != = LIKE`

Note

Boolean operators and comparison operators should only be used within Boolean expressions. In other words, expressions that evaluate to TRUE or FALSE.

`eval` Command (cont.)

```
index=sales sourcetype=vendor_sales VendorCountry="United States"
| stats sum(price) as Sales by VendorStateProvince
| eval Performance = case(Sales<=500,"Needs immediate evaluation",
                         Sales<1000,"Underperformer",Sales>=1000,"Overperformer")
| eval Verdict = if(Performance IN("Underperformer","Needs immediate evaluation"), "Send to marketing",null())
| eval Sales = "$".tostring(Sales,"commas")
```

- Field values are treated in a case-sensitive manner
- String values must be "double-quoted"
- Field names must be unquoted or single quoted when they include a special character like a space
- Use a period (`.`) instead of (+) when concatenating strings and numbers to avoid conflicts

Note

`tostring` is a Text function that can be used by `eval`, `fieldformat`, `where`, and `eval` expressions.

Ways to Write Multiple `eval`s

Expressions can be separate, nested, or linked with a comma

Separate **eval** pipeline segments

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = bytes/(1024*1024)
| eval bandwidth = round(bandwidth, 2)
```

Nested **eval** commands targeting the same field

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = round(bytes/(1024*1024), 2)
```

Combining **eval** commands with commas

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as bytes by usage
| eval bandwidth = bytes/(1024*1024),
  bandwidth = round(bandwidth, 2)
```

usage	bytes	bandwidth
Borderline	1298542	1.24
Business	2909449	2.77
Personal	9771346	9.32
Unknown	997092	0.95
Violation	495606	0.47

Note

round is a mathematical function that can be used by **eval**, **fieldformat**, **where**, and **eval** expressions.

Referencing `eval` Fields

Temporary fields created using **`eval`** can be referenced in the search pipeline by succeeding commands

```
index=network sourcetype=cisco_wsa_squid
| stats sum(sc_bytes) as Bytes by usage
| eval bandwidth = round(Bytes/pow(1024,2), 2)
| sort -bandwidth
| rename bandwidth as "Bandwidth (MB)"
```

usage	Bytes	Bandwidth (MB)
Personal	299584913	285.71
Unknown	77187989	73.61
Business	66844576	63.75
Borderline	54011022	51.51
Violation	3203231	3.05

Note

`pow` is a mathematical function that can be used by **`eval`**, **`fieldformat`**, **`where`**, and **`eval`** expressions.

Evaluation Functions

- Evaluates an expression based on your events and returns a result
- There are 11 categories of evaluation functions:
  - Conversion
  - Comparison and Conditional
  - Cryptographic
  - Informational
  - Statistical
  - Multivalue
  - etc.

Evaluation Functions (cont.)

Category	Function Syntax	Description
Comparison & Conditional	`case(X,"Y",…)`	Cycles through expressions and returns first value (Y) where the expression (X) evaluates to TRUE
	`cidrmatch("X",Y)`	Returns TRUE or FALSE based on whether an IP matches a CIDR notation
	`if(X,Y,Z)`	If X evaluates to TRUE, returns Y, otherwise returns Z
	`like(<string>,<pattern>)`	Returns TRUE if `<string>` matches `<pattern>`
	`in(<field>,<value-list>)`	Returns TRUE if a value in the `<value-list>` matches a value in `<field>`
	`match(SUBJECT,"<regex>")`	Returns TRUE or FALSE based on whether the `SUBJECT` matches the `<regex>`

Note

This is not a full list.

Evaluation Functions (cont.)

Category	Function Syntax	Description
Comparison & Conditional	`true()`	Returns TRUE
	`false()`	Returns FALSE
	`null()`	Takes no arguments and returns NULL
	`nullif(X,Y)`	Returns NULL if X=Y, otherwise returns X
	`validate(X,Y,…)`	Opposite of case function; returns Y for the first expression (X) that evaluates to FALSE
	`searchmatch(X)`	Returns TRUE if the search string X matches the event
Text	`replace(X,Y,Z)`	Returns a string by substituting Z for every instance of regex pattern Y in X

Note

This is not a full list.

Using Evaluation Functions

Most evaluation functions can be used in the **<eval-expression>** for **eval**, **fieldformat**, and **where** commands…

```
...| eval <field>=function(…)
```

```
...| where function(…)
```

```
...| fieldformat <field>=function(…)
```

…and as part of **eval** expressions

```
...| stats stats-func(eval(function(…)))
```

Note

The **where** command is discussed in the next topic. However, **eval** expressions are outside the scope of this module.

`true`, `false`, `null` and `nullif` Functions

- `true()`: always returns TRUE
- `false()`: always returns FALSE
- `null()`: always returns NULL
- `nullif(X,Y)`: returns NULL if `X=Y`, otherwise returns `X`
- These functions are commonly used within other functions

```
index=web sourcetype=access_combined
| eval engagement = if(isnull(action),"no engagement",action)
| table action engagement
```

action	engagement
	no engagement
purchase	purchase
purchase	purchase
addtocart	addtocart
	no engagement
	no engagement
	no engagement
view	view

Note

The `table` command puts specific fields in a table.

```
index=web sourcetype=access_combined
| eval engagement = case(isnotnull(action),action,true(),"no engagement")
| table action engagement
```

`if` Function

```
...| eval <field> = if(X,Y,Z)
```

If the expression **X** evaluates to TRUE, returns **Y**, otherwise returns **Z**

Scenario

Calculate total revenue for Asia (`VendorID=7000-7999`) and all other countries. Group all other countries as "Rest of the World" and format revenue as $x,xxx.

```
index=sales sourcetype=vendor_sales
| eval SalesTerritory = if((VendorID>=7000 AND VendorID<8000),
  "Asia", "Rest of the World")
| stats sum(price) as TotalRevenue by SalesTerritory
| eval TotalRevenue = "$".tostring(TotalRevenue, "commas")
```

SalesTerritory	TotalRevenue
Asia	$13,091.82
Rest of the World	$121,571.00

`case` Function

```
...| eval <field> = case(X1,Y1,X2,Y2,…)
```

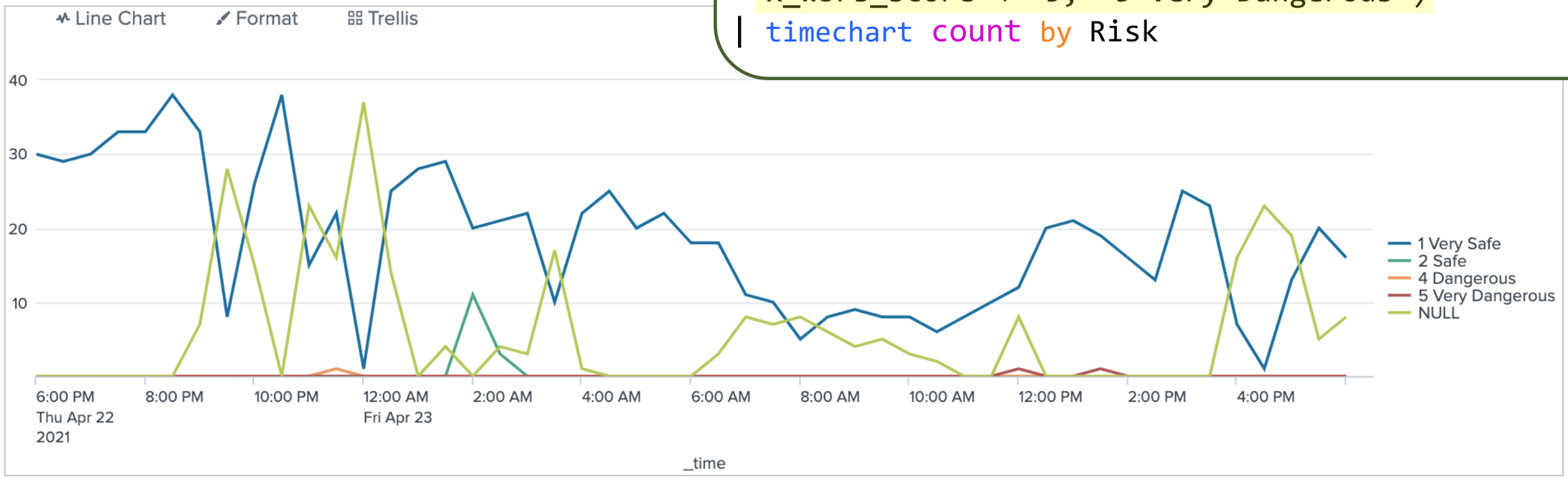
- Evaluates Boolean expressions **X** and returns **Y** for the first expression to evaluate to TRUE
  - If **X1=TRUE**, returns **Y1**
  - If **X1=FALSE**, then the next expression, **X2**, is evaluated, etc.
- Returns NULL if no expressions evaluate to TRUE

`case` Function Example

Scenario

SecOps found a potential virus on a user's machine. Find and classify the number of internet visits by risk during the past 24 hours.

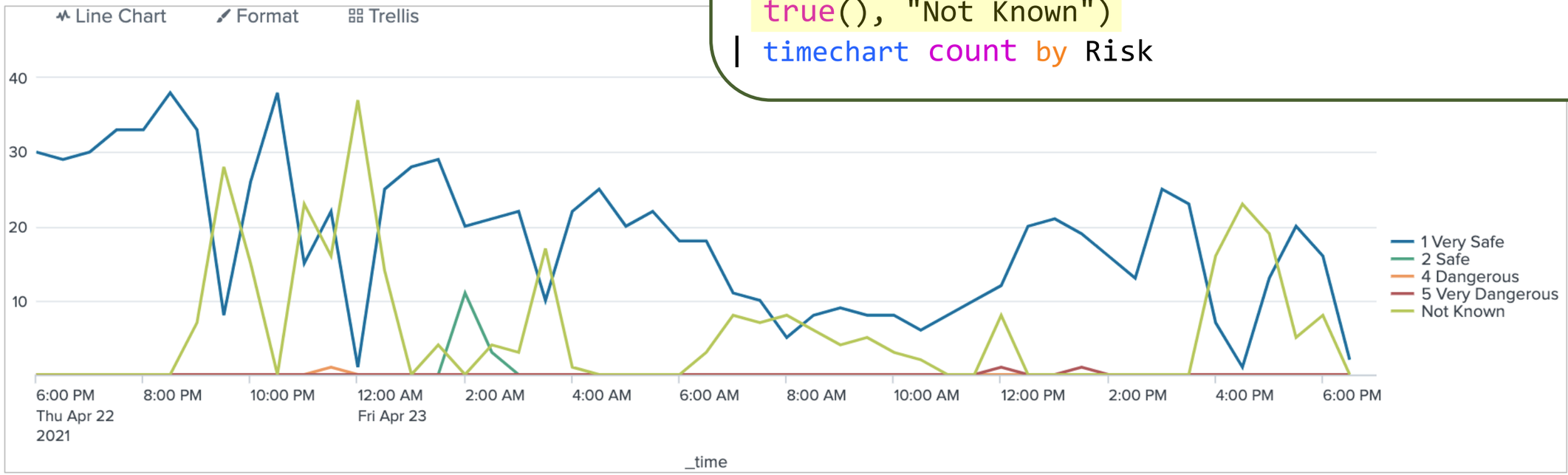
```
index=network sourcetype=cisco_wsa_squid
| eval Risk = case(x_wbrs_score >= 5,"1 Very Safe",
  x_wbrs_score >= 3,"2 Safe",
  x_wbrs_score >= 0,"3 Neutral",
  x_wbrs_score >= -5,"4 Dangerous",
  x_wbrs_score < -5, "5 Very Dangerous")
| timechart count by Risk
```



`case` Function Example (cont.)

Test for a condition you know is true (e.g., `0=0`) if you want an "otherwise" clause or use `true()`

```
index=network sourcetype=cisco_wsa_squid
| eval Risk = case(x_wbrs_score >= 5,"1 Very Safe",
  x_wbrs_score >= 3,"2 Safe",
  x_wbrs_score >= 0,"3 Neutral",
  x_wbrs_score >= -5,"4 Dangerous",
  x_wbrs_score < -5, "5 Very Dangerous",
  true(), "Not Known")
| timechart count by Risk
```

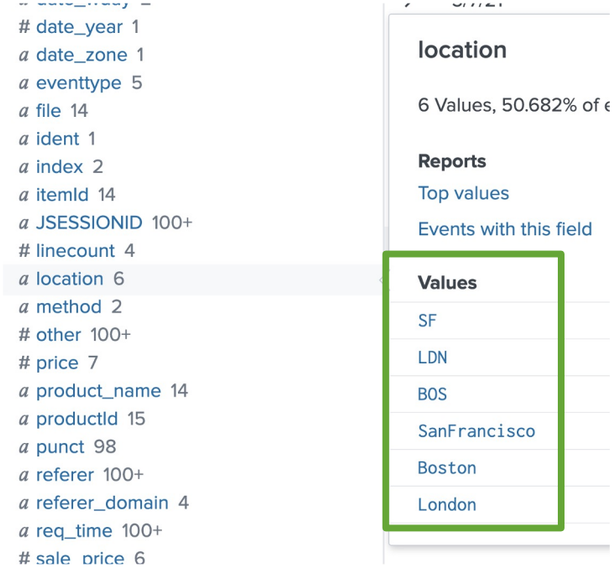


Data Normalization with `case`

The values for **`location`** are not normalized and therefore, this search generates incomplete results

Scenario

Compare total sales to staff attendance at each of the three development offices over the last 24 hours.




```
sourcetype=access_c* OR sourcetype=winauth*
| stats sum(price) as total_revenue, count(rfid) as staff_count by location
```

location	total_revenue	staff_count
BOS	40919.91	0
Boston		51
LDN	43521.94	0
London		44
SF	54995.10	0
SanFrancisco		112

Data Normalization with `case` (cont.)

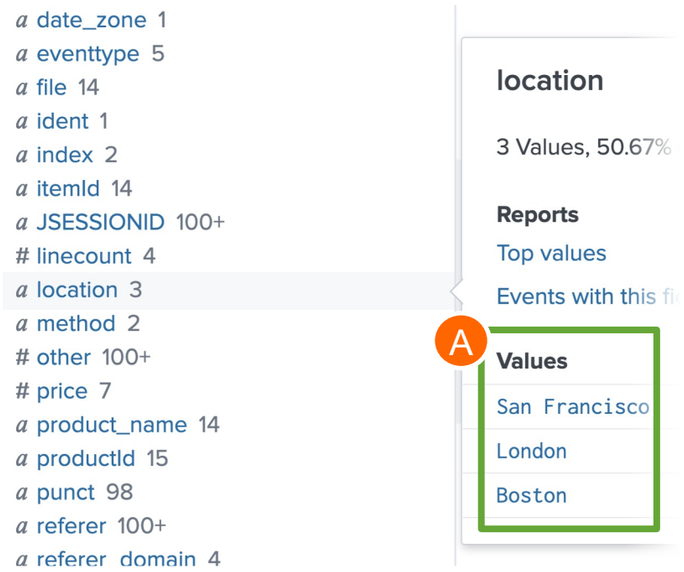
Normalizing after `stats` does not generate the desired results

```
sourcetype=access_c* OR sourcetype=winauth*
| stats sum(price) as total_revenue, count(rfid) as staff_count by location
| eval location = case(location="BOS" OR location="Boston","Boston",
      location="LDN" OR location="London","London",
      location="SF" OR location="SanFrancisco","San Francisco")
```

location	total_revenue	staff_count
Boston	41374.77	0
Boston		51
London	43806.82	0
London		44
San Francisco	55180.02	0
San Francisco		113

Data Normalization with `case` (cont.)

Normalize the `location` field before data transforms




```
sourcetype=access_c* OR sourcetype=winauth*
A| eval location = case(location="BOS" OR location="Boston","Boston",
       location="LDN" OR location="London","London",
       location="SF" OR location="SanFrancisco","San Francisco")
B| stats sum(price) as total_revenue, count(rfid) as staff_count by location
```

B

location	total_revenue	staff_count
Boston	40974.87	51
London	43646.89	44
San Francisco	55035.09	113

Data Normalization with `case` (cont.)

Use the `true()` function to keep correct values and create a simpler expression

```
sourcetype=access_c* OR sourcetype=winauth*
| eval location = case(location="BOS","Boston", location="LDN","London",
      location="SF" OR location="SanFrancisco","San Francisco",
      true(),location)
| stats sum(price) as total_revenue, count(rfid) as staff_count by location
```

location	total_revenue	staff_count
Boston	40974.87	51
London	43646.89	44
San Francisco	55035.09	113

`validate` Function

```
...| eval <field> = validate(X1,Y1,X2,Y2,…)
```

- Returns **Y** for the first expression **X** that evaluates to FALSE
- If all expressions evaluate to TRUE, returns NULL
- Opposite of the **`case`** function

```
index=security sourcetype=linux_secure
| eval portWarning=validate(src_port > 0, "ERROR: Port is not a positive integer",
      port >= 1 AND port <= 9900, "WARNING: Port is out of range")
```

`in` Function

```
...| eval <field> = if(in(<field>,<value-list>),Y,Z)
```

- Returns TRUE if one of the values in **`<value-list>`** matches a value from **`<field>`**
- Must be used within the **`if`** function or **`case`** function with **`eval`**

```
sourcetype=access_*
| eval error = if(in(status, "404","500","503"),"true","false")
| stats count by error
```

error	count
false	9683
true	383

searchmatch Function

```
...| eval <field> = if(searchmatch(X),Y,Z)
```

- Returns TRUE if an event matches the search string, **X**
- Must be used within the **if** function or **case** function with **eval**

```
sourcetype=win_audit
| eval eventInfo = if(searchmatch("642 8 SuccessAudit
      BUSDEV-003 Security 224542437"),_raw,"not found")
| where eventInfo!="not found"
| table eventInfo _time
```

eventInfo	_time
642 8 SuccessAudit BUSDEV-003 Security 224542437	2021-04-23 18:11:48
642 8 SuccessAudit BUSDEV-003 Security 224542437	2021-04-23 17:20:22
642 8 SuccessAudit BUSDEV-003 Security 224542437	2021-04-23 16:18:53
642 8 SuccessAudit BUSDEV-003 Security 224542437	2021-04-23 14:36:31

`cidrmatch` & `match` Functions

- `cidrmatch("X",Y)`: returns TRUE/FALSE based on whether provided IP address `Y` matches subnet specified by `X`

```
index=network sourcetype=cisco_wsa_squid
| eval isLocal = if(cidrmatch("10.2/16",bcg_ip), "IS local sub2", "NOT local sub2")
```

- `match(SUBJECT,"<regex>")`: returns TRUE/FALSE based on whether the `SUBJECT` matches `<regex>` pattern

```
index=network sourcetype=cisco_wsa_squid
| eval proper_ip_address = if(match(src,"^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$"), "true", "false")
```

`replace` Function

```
...| eval field1=replace(X,Y,Z)
```

- Returns a string by substituting `Z` for every occurrence of `Y` in `X`
  - `X` is a literal string or field
  - `Y` contains regex to identify a pattern in `X`
  - `Z` specifies the substitution
- Useful for masking data such as account numbers and IP addresses

`replace` Function Example 1

Scenario

Calculate the number of events for each account on the Business Intelligence server. Mask the last 4 digits of each account number.

```
index=sales sourcetype=sales_entries
A | stats count by AcctCode
B | eval AcctCode = replace(AcctCode,"(\d{4}-)\d{4}","\1xxxx")
```

A

AcctCode	count
0012-4250	5
0182-4713	1
0322-4454	3
0508-4173	5

B

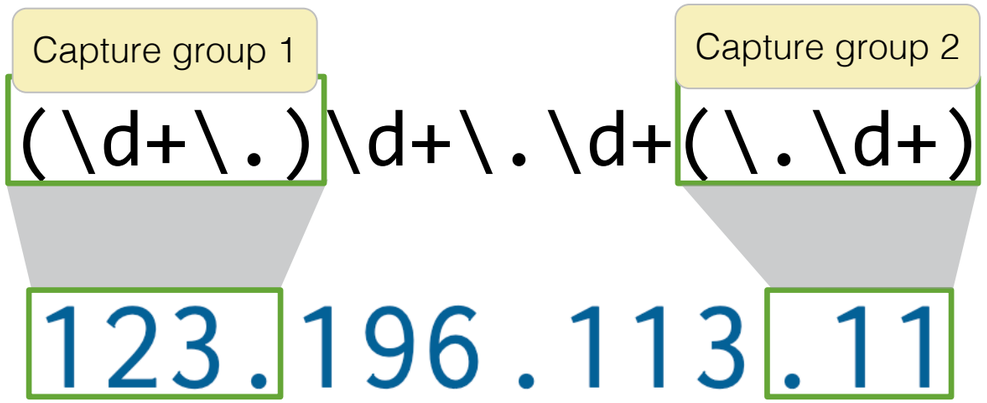
AcctCode	count
0012-xxxx	5
0182-xxxx	2
0322-xxxx	3
0508-xxxx	4

- A Count results by **`AcctCode`**
- B Mask the last 4 digits of each account code
  - The parenthesis **`( )`** around **`\d{4}-`** indicates a regex capture group
    - Reference this group using **`\1`** because this is the first and only capture group indicated in the regex

`replace` Function Example 2

Use multiple capture groups to create complex substitutions

```
...
| eval clientip_new = replace(clientip, "(\d+\.)\d+\.\d+(\.\d+)","\1xxx.xxx\2")
| table clientip, clientip_new
```



clientip	clientip_new
193.33.170.23	193.xxx.xxx.23
88.12.32.208	88.xxx.xxx.208
123.196.113.11	123.xxx.xxx.11
24.185.15.226	24.xxx.xxx.226
46.251.224.66	46.xxx.xxx.66
95.163.78.227	95.xxx.xxx.227
91.205.40.22	91.xxx.xxx.22

`fieldformat` Command

```
...| fieldformat <field>=<eval-expression>
```

- Changes the format of a field's values with an **`<eval-expression>`**
- Only changes the appearance, not the underlying value
- Accepts a wide variety of evaluation functions
- Should be used late in the pipeline because formatted results cannot be modified by other commands

`fieldformat` Command Example

Multiple values of a field can be targeted using `fieldformat`

Scenario

ITOps wants to know how many events were logged for each sourcetype in the security index. Reformat values so they have commas.

```
index=security
| stats count as totalCount by sourcetype
| fieldformat totalCount=tostring(totalCount,"commas")
```

sourcetype	totalCount
history_access	1,738
linux_secure	84,377
winauthentication_security	2,262

Using `eval` to Compare Lab Exercise

Time: 15 minutes

Tasks:

- Use the `case` and `if` functions to determine a country's sales performance and generate a verdict based on that performance
- Use the `case` function to categorize web server events based on the number of bytes consumed
- Use the `replace` function to conceal vendor account codes

Filtering with `where` & Managing Missing Data

Topic Objectives

- Use the **`where`** command to filter results
- Use wildcards with the **`where`** command
- Filter fields with **`isnull`** and **`isnotnull`** informational functions
- Manage missing data with the **`fillnull`** command

`where` Command

```
...| where <eval-expression>
```

- Acts as a filter on search results by removing results that do not match the **`<eval-expression>`**
  - Uses mathematical and Boolean operators to evaluate values in the expression and return TRUE or FALSE
  - The **`where`** command only returns results that evaluate to TRUE
- Interprets unquoted or single-quoted strings as fields and double-quoted strings as field values
- Treats field values in case-sensitive manner

`where` Command: Operators

Mathematical Operators
+
-
*
/
%

Boolean Operators
AND
OR
NOT
<
>
<=
>=
!=
= or ==
LIKE

`where` Command Example

Scenario

SalesOps wants to know which days during the previous week have seen more "remove" actions than "change quantity" actions.

```
index=web sourcetype=access_combined
| timechart count(eval(action="changequantity"))
  as changes, count(eval(action="remove")) as removals
| where removals > changes
```

1. Results after transformation by the **`timechart`** command
2. **`where`** command filters results and only returns events where **`removals`** are greater than **`changes`**

1

_time	changes	removals
2021-04-11	171	171
2021-04-12	182	168
2021-04-13	171	166
2021-04-14	149	180
2021-04-15	151	159
2021-04-16	171	161
2021-04-17	142	154

2

_time	changes	removals
2021-04-14	149	180
2021-04-15	151	159
2021-04-17	142	154

Note

The **`timechart`** command evaluates events and groups results by time.

Boolean Expression Evaluation Order

- Boolean operators: **`AND`**, **`OR`**, and **`NOT`** (case sensitive)
- When the **`<eval-expression>`** uses a Boolean operator, it is an **`<eval-boolean-expression>`** and evaluates in a specific order:

Boolean Expression Order of Evaluation	
First	Expressions within parenthesis
Second	`NOT` clauses
Third	`AND` clauses
Fourth	`OR` clauses

Note

This order applies to **`where`** and **`eval`** commands.

`where` Command: Case Sensitivity Example

```
index=sales sourcetype=vendor_sales VendorCountry="United States"
| where categoryId="STRATEGY"
```

i	Time	Event
>	4/17/21 11:45:49.000 PM	[17/Apr/2021:23:45:49] VendorID=1098 Code=C AcctID=xxxxxxxxxxx5855 categoryId = STRATEGY host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	4/17/21 11:43:25.000 PM	[17/Apr/2021:23:43:25] VendorID=1041 Code=C AcctID=xxxxxxxxxxx5167 categoryId = STRATEGY host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	4/17/21 11:40:26.000 PM	[17/Apr/2021:23:40:26] VendorID=1187 Code=C AcctID=xxxxxxxxxxx4927 categoryId = STRATEGY host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	4/17/21 11:29:13.000 PM	[17/Apr/2021:23:29:13] VendorID=1297 Code=A AcctID=xxxxxxxxxxx7932 categoryId = STRATEGY host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales

```
index=sales sourcetype=vendor_sales VendorCountry="United States"
| where categoryId="strategy"
```

No results found. Try expanding the time range.

Note

Splunk returns no results if you run the search with correct case but no double-quotes.

`where` Command: Wildcards

- Specify a wildcard by using the **`LIKE`** operator or the **`like`** function

```
...| where <string> LIKE <pattern>
```

```
...| where like(<string>,<pattern>)
```

- Do not use * as a wildcard, the **`where`** command will interpret this as a literal character or mathematical operator
- Use **%** for multiple characters or _ for a single character

`where` Command: Wildcards Example

Both searches return the same results

Scenario

IT wants a list of user accounts that are like admin (`adm%`).

```
index=security sourcetype=linux_secure
| where like(user,"adm%")
| dedup user
| table user
```

```
index=security sourcetype=linux_secure
| where user like "adm%"
| dedup user
| table user
```

user
admin
administrator
adm

Note

Notice that you still must use double quotes if your pattern references a string.

`where` Command: `isnull` and `isnotnull`

- Filter fields with null values using `isnull` or `isnotnull`
- Useful for troubleshooting

Scenario

A sales campaign manager wants to know which 1-hour periods contained no sales over the last 24 hours in Canada.

```
index=sales sourcetype=vendor_sales
| timechart span=1h sum(price) as sum
| where isnull(sum)
```

_time	sum
2021-04-21 22:00	
2021-04-22 00:00	
2021-04-22 01:00	
2021-04-22 07:00	
2021-04-22 10:00	
2021-04-22 12:00	

Note

`isnull` and `isnotnull` are Informational functions. Informational functions return information about a value.

fillnull Command

```
...| fillnull [value=<string>] [<field-list>]
```

- Replaces null values in fields
- Specify what to replace a null value with using **value=<string>**
  - If not specified, defaults to **value=0**
- Restrict which fields to apply **fillnull** to using **<field-list>**

bar	foo
foobar	
	barfoo

```
...| fillnull
```

bar	foo
foobar	0
0	barfoo

fillnull Command Examples

```
...
| fillnull
```

product_name	Brazil	Canada	Colombia	Germany	Iceland
Benign Space Debris					
Curling 2014		39.98			
Dream Crusher		39.99	39.99	39.99	
Final Sequel					
Fire Resistance Suit of Provolone				3.99	

product_name	Brazil	Canada	Colombia	Germany	Iceland
Benign Space Debris	0	0	0	0	0
Curling 2014	0	39.98	0	0	0
Dream Crusher	0	39.99	39.99	39.99	0
Final Sequel	0	0	0	0	0
Fire Resistance Suit of Provolone	0	0	0	3.99	0

```
...
| fillnull value="NULL"
```

product_name	Brazil	Canada	Colombia	Germany	Iceland
Benign Space Debris	NULL	NULL	NULL	NULL	NULL
Curling 2014	NULL	39.98	NULL	NULL	NULL
Dream Crusher	NULL	39.99	39.99	39.99	NULL
Final Sequel	NULL	NULL	NULL	NULL	NULL
Fire Resistance Suit of Provolone	NULL	NULL	NULL	3.99	NULL

Filtering & Managing Missing Data Lab Exercise

Time: 20 minutes

Tasks:

- Use the **`where`** command to find events with a **`productId`** value but missing a **`product_name`** value. Then use the **`fillnull`** command to mark these events for review.
- Use the **`eval`** command and **`case`** function to classify events using the **`x_wbrs_score`** field

Wrap-up Slides

Wrap-Up

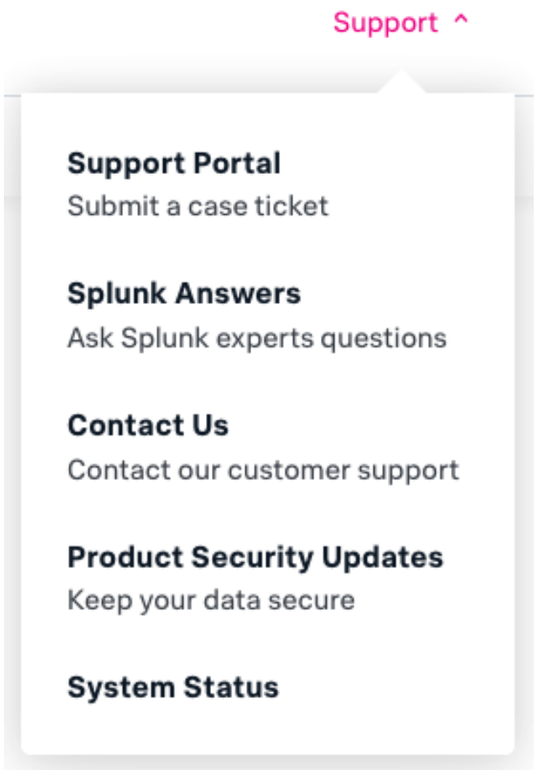
- You should now be able to:
  - Use comparison, conditional, and text functions
  - Format field values with the **`fieldformat`** command
  - Filter results with the **`where`** command
  - Use the **`isnull`** and **`isnotnull`** informational functions to find null values in your data
  - Use the **`fillnull`** command to replace null values

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
  - Wiki: wiki.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



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 *

Learning Paths (cont.)

Knowledge Manager - Recommended Courses

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

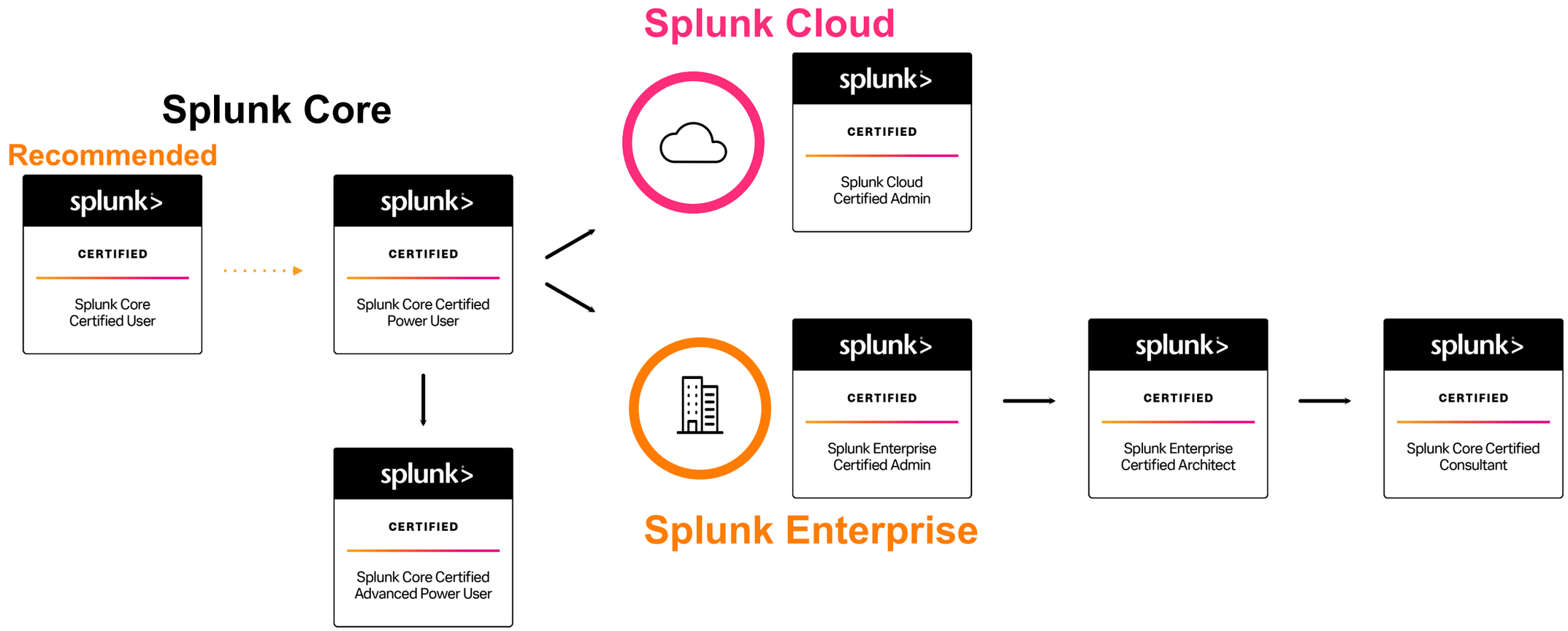
- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth
- Search Optimization *

Splunk Certification

Offerings & Requirements

Splunk Core and Beyond

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



App-Specific Offerings

For Splunk Add-Ons

App Developer

ES Administration

ITSI Administration

SOAR Automation Developer

Splunk Core Certified User

This entry-level certification demonstrates an individual's basic ability to navigate and use Splunk software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 1 **or** the following courses:

- What is Splunk?
- Intro to Splunk
- Using Fields
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Leveraging Lookups and Subsearches
- Search Optimization
- Enriching Data with Lookups
- Data Models

See here for registration assistance.

Congratulations! You are a...

Recommended Next Step

- Splunk Core Certified Power User

 | 31 October 2022

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...

splunk>

CERTIFIED

Splunk Core Certified Advanced Power User

Recommended Next Steps

- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

 |

Splunk Cloud Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Cloud environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Cloud Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete **either** the Splunk Cloud Administration **or** the Transitioning to Splunk Cloud course.

Both courses will equally prepare candidates for the exam, but are tailored to meet the needs of the individual based on prior Splunk experience.

Splunk Cloud Administration is designed for net-new administrators working in a Splunk Cloud environment. **Transitioning to Splunk Cloud** is for experienced Enterprise administrators looking to maximize their success in migrating to a Cloud environment.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Certified Developer

Splunk Enterprise Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Enterprise environment

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Enterprise Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Splunk System Administration
- Splunk Data Administration

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Architect
- Splunk Certified Developer

Splunk Certified Architect

This certification demonstrates an individual's ability to deploy, manage, and troubleshoot complex Splunk Enterprise environments

Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin

Prerequisite Course(s):

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Splunk Enterprise Certified Architect Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Candidates who are **Splunk Enterprise Certified Admin** and have completed all of the above courses will automatically receive an exam authorization for the Splunk Enterprise Certified Architect exam within 5-7 business days of receiving their passing lab results.

See here for registration assistance.

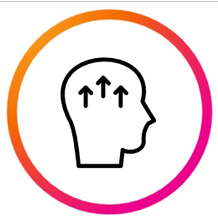
Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Consultant

Splunk Core Certified Consultant

This certification demonstrates an individual's ability to properly size, install, and implement Splunk environments and to advise others on how to utilize the product and maximize its value for their needs



Prerequisite Certification(s):

- Splunk Core Certified Power User
- Splunk Enterprise Certified Admin
- Splunk Enterprise Certified Architect

Prerequisite Course(s):

- *Advanced Power User courses **or** digital badge**
- Core Consultant Labs
  - Indexer Cluster Implementation
  - Distributed Search Migration
  - Implementation Fundamentals
  - Architect Implementation 1-3
- Services Core Implementation

Splunk Core Certified Consultant Exam

Time to study! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- *Fundamentals 3*, *Creating Dashboards*, *Advanced Searching & Reporting**
- Core Consultant Labs
- Services Core Implementation

Candidates who are **Splunk Enterprise Certified Architects** and have completed all of the above courses must contact certification@splunk.com to request their Core Consultant exam authorization.

See here for registration assistance.

These Advanced Power User courses can be replaced with a Splunk Certified Advanced Power User badge **or completion of the following courses:*

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

Congratulations! You are a...

Recommended Next Steps

- None

 | 31 October 2022

Splunk Certified Developer

This certification demonstrates an individual's expertise in drilldowns, advanced behaviors and visualizations, planning, creating, and packaging apps, and REST endpoints

Prerequisite Certification(s):

- Splunk Core Certified Power User

AND

- Splunk Enterprise Certified Admin

OR

- Splunk Cloud Certified Admin

Prerequisite Course(s):

- None

Splunk Certified Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Creating Dashboards with Splunk*
- Advanced Dashboards & Visualizations
- Building Splunk Apps
- Developing with Splunk's REST API

This course may also be substituted with the following newly-launched courses:

- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Splunk Enterprise Security Certified Admin

This certification demonstrates an individual's ability to install, configure, and manage a Splunk Enterprise Security deployment

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Enterprise Security Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Administering Splunk Enterprise Security

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Phantom Certified Admin

 | 31 October 2022

Splunk IT Service Intelligence Certified Admin

This certification demonstrates an individual's ability to deploy, manage, and utilize Splunk ITSI to monitor mission-critical services

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk IT Service Intelligence Certified Admin Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following course:

- Implementing Splunk IT Service Intelligence

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Courses on Observability

Splunk SOAR Certified Automation Developer

This certification demonstrates an individual's ability to install and configure a SOAR server, integrate it with Splunk, and plan, design, create, and debug playbooks

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk SOAR Certified Automation Developer Exam

Time to study! We suggest candidates looking to prepare for this exam complete the following courses:

- Administering SOAR (Phantom)
- Developing SOAR (Phantom) Playbooks
- Advanced SOAR (Phantom) Implementation

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- None

Thank You

 |