splunk> Search CheatSheet

Here are some examples illustrating some useful things you can do with the search language.

Learn more about the commands used in these examples by referring to the search command reference.

Add fields

Extract data from events into fields so that you can analyze and run reports on it in a meaningful way.

Extract field/value pairs and reload field extraction settings from disk.	* | extract reload=true
Extract field/value pairs that are delimited by "|;", and values of fields that are delimited by "=:".	* | extract pairdelim="|;", kvdelim="=:", auto=f
Extract the COMMAND field when it occurs in rows that contain "splunkd".	* | xmlkv
Add the field: comboIP. Values of comboIP = "sourceIP + "/" + destIP".	* | multikv fields COMMAND filter splunkd
Extract "from" and "to" fields using regular expressions. If a raw event contains "From: Susan To: Bob", then from=Susan and to=Bob.	* | rex field=_raw "From: (?<from>.*) To: (?<to>.*)"
Add the field: comboIP. Values of comboIP = "sourceIP + "/" + destIP".	* | strcat sourceIP "/" destIP comboIP
Add the field: velocity. Values of velocity = distance field value / time field value (using an SQLite evaluation).	* | eval velocity=distance/time
Add location information (based on IP address) to the first twenty events that contain "404" and are from from webserver1.	404 host=webserver1 | head 20 | iplocation

Convert fields

Change the names of fields, the units of values stored in fields, the types of data stored in fields, or the attributes of fields.

Convert every field value to a number value except for values in the field "foo" (use the {{none}} argument to specify fields to ignore).	* | convert auto(*) none(foo)
Change all memory values in the virt field to Kilobytes.	* | convert memk(virt)
Change the sendmail syslog duration format (D+HH:MM:SS) to seconds. For example, if delay="00:10:15", the resulting value will be delay="615".	* | convert dur2sec(delay)
Convert values of the duration field into number value by removing string values in the field value. For example, if duration="212 sec", the resulting value will be duration="212".	* | convert rmunit(duration)}}
Rename the _ip field as IPAddress.	* | rename _ip as IPAddress
Change any host value that ends with "localhost" to "localhost".	* | replace *localhost with localhost in host

Filter and order fields

Filter and re-arrange how Splunk displays fields within search results.

Keep only the host and ip fields, and display them in the order: host, ip.	* | fields host, ip
Keep only the host and ip fields, and remove all internal fields (for example, _time, _raw, etc.) that may cause problems in Splunk Web.	* | fields + host, ip
Remove the host and ip fields.	* | fields - host, ip

Filter results

Filter search result sets by removing duplicate events, using regular expressions, or by searching within a result set.

Keep only search results that have matching src or dst values.	* | search src="10.9.165.*" OR dst="10.9.165.8"
Keep only search results whose _raw field contains IP addresses in the non-routable class A (10.0.0.0/8).	* | regex _raw=(?<!\d)10.\d{1,3}\.\d{1,3}\.\d {1,3}(?!\d
Remove duplicates of results with the same host value.	* | dedup host

Order results

Sort, re-order, or return a portion of a search result set.

Sort results by ip value in ascending order and then by url value in descending order.	* | sort ip, -url
Reverse the order of a result set.	* | reverse
Return the first 20 results.	* | head 20
Return the last 20 results (in reverse order).	* | tail 20

Group results

Group search results into a transaction (a single observation of any event stretching over multiple logged events) based on related pieces of information, or group results by statistical correlation.

Group search results that have the same host and cookie, occur within 30 seconds of each other, and do not have a pause greater than 5 seconds between each event into a transaction.	* | transaction fields="host,cookie" maxspan=30s maxpause=5s
Group search results that share the same value of from, with a maximum span of 30 seconds, and a pause between events no greater than 5 seconds into a transaction.	* | transaction fields=from maxspan=30s maxpause=5s
Group search results into 4 clusters based on the values of the date_hour and date_minute fields.	* | kmeans k=4 date_hour date_minute
Cluster events together, sort them by their cluster_count values, and then return the 20 largest	* | cluster t=0.9 showcount=true | sort - cluster_count | head 20}}|

Classify Events

Classify events as a type (event type), or have Splunk automatically classify events.

Force Splunk to apply event types that you have configured (Splunk Web automatically does this when you view the eventtype field).	* | typer
Have Splunk automatically discover and apply event types to events that contain the string "error".	error | typelearner

Change display formatting

Change how Splunk displays events by highlighting terms, displaying summarized raw data, showing the differences between events, or unescaping XML characters.

Highlight the terms "login" and "logout".	* | highlight login,logout
Search for events from "xml_escaped", and unescape XML characters.	source="xml_escaped" | xmlunescape
Show a summary of up to 5 lines for each search result.	* | abstract maxlines=5
Compare the ip values of the first and third search results.	* | diff pos1=1 pos2=3 attribute=ip

Generate data

Generate search results from your data using commands other than search.

Note: You must use a pipe (\|) before any data-generating command that isn't the search command.

Read in results from the CSV file: $SPLUNK_HOME/var/run/splunk/all.csv, keep any that contain the file: $SPLUNK_HOME/var/run/splunk/error.csv	| inputcsv all.csv | search error | outputcsv errors.csv
Display events from the file messages.1 as if the events were indexed in Splunk.	| file /var/log/messages.1
Run the mysecurityquery saved search, and email any results to user@domain.com.	| savedsearch mysecurityquery AND _count > 0 or | sendemail to=user@domain.com

Report

Summarize the results of any search as a report by performing statistical operations, and graphing functions.

Return the least common values of the url field.	* | rare url
Return the 20 most common values of the url field.	* | top limit=20 url
Remove duplicates of results with the same host value and return the total count of the remaining results.	* | stats dc(host)
Return the average for each hour, of any unique field that ends with the string "lay" (for example, delay, xdelay, relay, etc).	* | stats avg(*lay) BY date_hour
Search the access logs, and return the number of hits from the top 100 values of referer_domain.	sourcetype=access_combined | top limit=100 referer_domain | stats sum(count)
Search the access logs, and return the results associated with each other (that have at least 3 references to each other).	sourcetype=access_combined | associate supcnt=3
Return the average (mean) size for each distinct host.	* | chart avg(size) by host
Return the the maximum delay by size, where size is broken down into a maximum of 10 equal sized buckets.	* | chart max(delay) by size bins=10
Graph the average thruput of hosts over time.	* | timechart span=5m avg(thruput) by host
Create a timechart of average cpu_seconds by host, and remove data (outlying values) that may distort the timechart's axis.	* | timechart avg(cpu_seconds) by host | outlier action=TR
Search for all ps events, extract values, and calculate the average value of CPU each minute for each host.	sourcetype=ps | multikv | timechart span=1m avg(CPU) by host
Create a timechart of the count of from web sources by host, and fill all null values with "NULL".	sourcetype=web | timechart count by host | fillnull value=NULL
Build a contingency table of datafields from all events.	* | contingency datafield1 datafield2 maxrows=5 maxcols=5 usetotal=F
Calculate the co-occurrence correlation between all fields.	* | correlate type=cocur
Calculate the sums of the numeric fields of each result, and put the sums in the field sum.	* | addtotals fieldname=sum
Return events with uncommon values.	* | anomalousvalue action=filter pthresh=0.02
Bucket search results into 10 bins, and return the count of raw events for each bucket.	* | bucket size bins=10 | stats count(_raw) by size
Return the average thruput of each host for each 5 minute time span.	* | bucket _time span=5m | stats avg(thruput) by=_time host

Administrative

Perform administration tasks using search commands. Crawl your servers to discover more data to index, view configuration settings, or see audit information.

Crawl root and home directories and add all possible inputs found (adds configuration information to inputs.conf).	| crawl root="/;/Users/" | input add
View processing properties stored in props.conf - time zones, breaking characters, etc.	| admin props
View audit trail information stored in the local audit index. Also decrypt signed audit events while checking for gaps and tampering.	index=audit | audit

Subsearch

Use subsearches to use search results as an argument to filter search result sets with more granularity.

Return values of URL that contain the string "404" or "303" but not both.	* | set diff [search 404 | fields url] [search 303 | fields url]
Search for events around events associated with "root" and "login", and then search each of those time ranges for "failure".	login root | localize maxspan=5m maxpause=5m | map search= "search failure starttimeu=$starttime$ endtimeu=$endtime$"
Create a search string from the values of the host, source and sourcetype fields.	[* | fields + source, sourcetype, host | format]