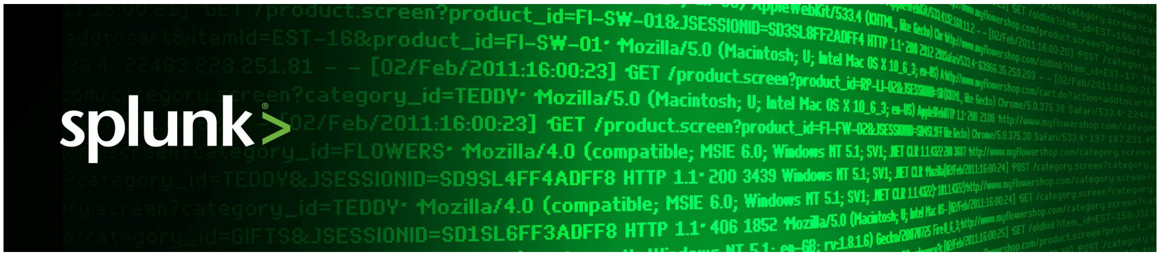


Splunk® Enterprise Search Reference 7.1.2

transaction

Generated: 8/29/2018 12:23 pm

transaction

Description

The transaction command finds transactions based on events that meet various constraints. Transactions are made up of the raw text (the `_raw` field) of each member, the time and date fields of the earliest member, as well as the union of all other fields of each member.

Additionally, the `transaction` command adds two fields to the raw events, `duration` and `eventcount`. The values in the `duration` field show the difference between the timestamps for the first and last events in the transaction. The values in the `eventcount` field show the number of events in the transaction.

See About transactions in the *Search Manual*.

Syntax

transaction [<field-list>] [name=<transaction-name>] [<txn_definition-options>...] [<memcontrol-options>...] [<rendering-options>...]

Required arguments

None.

Optional arguments

field-list

Syntax: <field> ...
Description: One field or more field names. The events are grouped into transactions based on the values of this field. If a quoted list of fields is specified, events are grouped together if they have the same value for each of the fields.

memcontrol-options

Syntax: <maxopentxn> | <maxopenevents> | <keepevicted>
Description: These options control the memory usage for your transactions. They are not required, but you can use 0 or more of the options to define your transaction.

name

Syntax: name=<transaction-name>

Description: Specify the stanza name of a transaction that is configured in the `transactiontypes.conf` file. This runs the search using the settings defined in this stanza of the configuration file. If you provide other transaction definition options (such as maxspan) in this search, they overrule the settings in the configuration file.

rendering-options

Syntax: <delim> | <mvlist> | <mvraw> | <nullstr>
Description: These options control the multivalue rendering for your transactions. They are not required, but you can use 0 or more of the options to define your transaction.

txn_definition-options

Syntax: <maxspan> | <maxpause> | <maxevents> | <startswith> | <endswith> | <connected> | <unifyends> | <keeporphans>
Description: Specify the transaction definition options to define your transactions. You can use multiple options to define your transaction.

Txn definition options

connected

Syntax: connected=<bool>
Description: Only relevant if a field or fields list is specified. If an event contains fields required by the transaction, but none of these fields have been instantiated in the transaction (added with a previous event), this opens a new transaction (connected=true) or adds the event to the transaction (connected=false).
Default: true

endswith

Syntax: endswith=<filter-string>
Description: A search or eval expression which, if satisfied by an event, marks the end of a transaction.

keeporphans

Syntax: keeporphans=true | false
Description: Specify whether the transaction command should output the results that are not part of any transactions. The results that are passed through as "orphans" are distinguished from transaction events with a `_txn_orphan` field, which has a value of 1 for orphan results.
Default: false

maxspan

Syntax: maxspan=<int>[s | m | h | d]
Description: Specifies the maximum length of time in seconds, minutes, hours, or days that the events can span. The events in the transaction must span less than integer specified for maxspan. If the value is negative, the maxspan constraint is disabled and there is no limit.
Default: -1 (no limit)

maxpause

Syntax: maxpause=<int>[s | m | h | d]
Description: Specifies the maximum length of time in seconds, minutes, hours, or days for the pause between the events in a transaction. If value is negative, the maxpause constraint is disabled and there is no limit.
Default: -1 (no limit)

maxevents

Syntax: maxevents=<int>
Description: The maximum number of events in a transaction. If the value is negative this constraint is disabled.
Default: 1000

startswith

Syntax: startswith=<filter-string>
Description: A search or eval filtering expression which if satisfied by an event marks the beginning of a new transaction.

unifyends

Syntax: unifyends= true | false
Description: Whether to force events that match startswith/endswith constraint(s) to also match at least one of the fields used to unify events into a transaction.
Default: false

Filter string options

<filter-string>

Syntax: <search-expression> | (<quoted-search-expression>) | eval(<eval-expression>)
Description: A search or eval filtering expression which if satisfied by an event marks the end of a transaction.

<search-expression>

Description: A valid search expression that does not contain quotes.

<quoted-search-expression>

Description: A valid search expression that contains quotes.

<eval-expression>

Description: A valid eval expression that evaluates to a Boolean.

Memory constraint options

If you have Splunk Cloud, Splunk Support administers the settings in the `limits.conf` file on your behalf.

keepevicted

Syntax: keepevicted=<bool>
Description: Whether to output evicted transactions. Evicted transactions can be distinguished from non-evicted transactions by checking the value of the 'closed_txn' field. The 'closed_txn' field is set to '0', or false, for evicted transactions and '1', or true for non-evicted, or closed, transactions. The 'closed_txn' field is set to '1' if one of the following conditions is met: maxevents, maxpause, maxspan, startswith. For `startswith`, because the `transaction` command sees events in reverse time order, it closes a transaction when it satisfies the start condition. If none of these conditions is specified, all transactions are output even though all transactions will have 'closed_txn' set to '0'. A transaction can also be evicted when the memory limitations are reached.
Default: false or 0

maxopenevents

Syntax: maxopenevents=<int>
Description: Specifies the maximum number of events (which are) part of open transactions before transaction eviction starts happening, using LRU policy.
Default: The default value for this argument is read from the transactions stanza in the `limits.conf` file.

maxopentxn

Syntax: maxopentxn=<int>
Description: Specifies the maximum number of not yet closed transactions to keep in the open pool before starting to evict transactions, using LRU policy.
Default: The default value for this argument is read from the transactions stanza in the `limits.conf` file.

Multivalue rendering options

delim

Syntax: delim=<string>
Description: Specify a character to separate multiple values. When used in conjunction with mvraw=t, represents a string used to delimit the values of _raw.
Default: " " (whitespace)

mvlist

Syntax: mvlist= true | false | <field-list>
Description: Flag that controls how multivalued fields are processed. When set to `mvlist=true`, the multivalued fields in the transaction are a list of the original events ordered in arrival order. When set to `mvlist=false`, the multivalued fields in the transaction are a set of unique field values ordered alphabetically. If a comma or space delimited list of fields is provided, only those fields are rendered as lists.
Default: false

mvraw

Syntax: mvraw=<bool>
Description: Used to specify whether the `_raw` field of the transaction search result should be a multivalued field.
Default: false

nullstr

Syntax: nullstr=<string>
Description: A string value to use when rendering missing field values as part of multivalued fields in a transaction. This option applies only to fields that are rendered as lists.
Default: `NULL`

Usage

If there are more than 5 events in a transaction, the remaining events are collapsed. A message appears at the end of the transaction which gives you the option to show all of the events in the transaction.

Specifying multiple fields

The Splunk software does not necessarily interpret the transaction defined by multiple fields as a conjunction (`field1 AND field2 AND field3`) or a disjunction (`field1 OR field2 OR field3`) of those fields. If there is a transitive relationship between the fields in the fields list and if the related events appear in the correct sequence, each with a different timestamp, `transaction` command will try to use it. For example, if you searched for

```
... | transaction host cookie
```

You might see the following events grouped into a transaction:

```
event=1 host=a
event=2 host=a cookie=b
event=3 cookie=b
```

Descending time order required

The `transaction` command requires that the incoming events be in descending time order. Some commands, such as `eval`, might change the order or time labeling of events. If one of these commands precedes the `transaction` command, your search returns an error unless you include a `sort` command in your search. The `sort` command must occur immediately before the `transaction` command to reorder the search results in descending time order.

Basic Examples

1. Transactions with the same host, time range, and pause

Group search results that that have the same host and cookie value, occur within 30 seconds, and do not have a pause of more than 5 seconds between the events.

```
... | transaction host cookie maxspan=30s maxpause=5s
```

2. Transactions with the same "from" value, time range, and pause

Group search results that have 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 from maxspan=30s maxpause=5s
```

3. Transactions with the same field values

You have events that include an alert_level. You want to create transactions where the level is equal. Using the `streamstats` command, you can remember the value of the alert level for the current and previous event. Using the `transaction` command, you can create a new transaction if the alert level is different. Output specific fields to table.

```
... | streamstats window=2 current=t latest(alert_level) AS last
earliest(alert_level) AS first | transaction endswith=eval(first!=last)
| table _time duration first last alert_level eventcount
```

Extended Examples

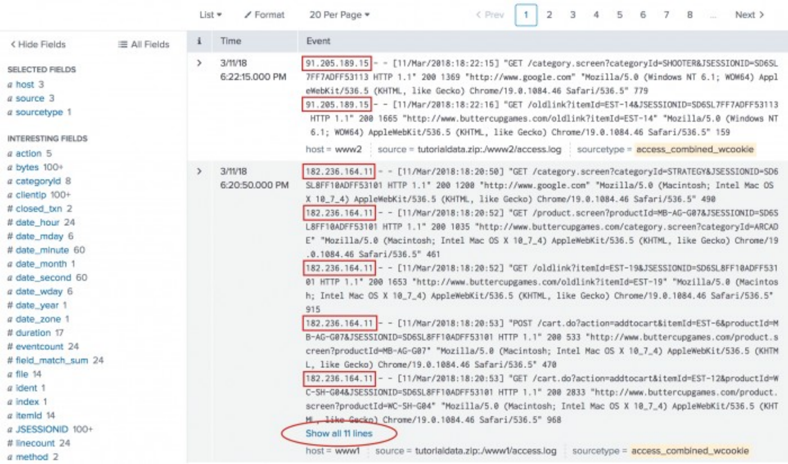
1. Transactions of Web access events based on IP address

This example uses the sample data from the Search Tutorial but should work with any format of Apache web access log. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to **get the tutorial data into Splunk**. Use the time range **Yesterday** when you run the search.

Define a transaction based on Web access events that share the same IP address. The first and last events in the transaction should be no more than thirty seconds apart and each event should not be longer than five seconds apart.

```
sourcetype=access_* | transaction clientip maxspan=30s maxpause=5s
```

This produces the following events list. The clientip for each event in the transaction is highlighted.



3/11/18
6:22:15.000 PM

```
91.205.189.15 - - [11/Mar/2018:18:22:15] "GET /category.screen?categoryId=SHOOTER&JSESSIONID=SD6SL
7FF7ADFF53113 HTTP 1.1" 200 1369 "http://www.google.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) Appl
eWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 779
91.205.189.15 - - [11/Mar/2018:18:22:16] "GET /oldlink?itemId=EST-14&JSESSIONID=SD6SL7FF7ADFF53113
 HTTP 1.1" 200 1665 "http://www.buttercupgames.com/oldlink?itemId=EST-14" "Mozilla/5.0 (Windows NT
 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 159
host = www2   source = tutorialdata.zip:./www2/access.log   sourcetype = access_combined_wcookie
```

3/11/18
6:20:50.000 PM

```
182.236.164.11 - - [11/Mar/2018:18:20:50] "GET /category.screen?categoryId=STRATEGY&JSESSIONID=SD6
SL8FF10ADFF53101 HTTP 1.1" 200 1200 "http://www.google.com" "Mozilla/5.0 (Macintosh; Intel Mac OS
X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 490
182.236.164.11 - - [11/Mar/2018:18:20:52] "GET /product.screen?productId=MB-AG-G07&JSESSIONID=SD6S
L8FF10ADFF53101 HTTP 1.1" 200 1035 "http://www.buttercupgames.com/category.screen?categoryId=ARCAD
E" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19
.0.1084.46 Safari/536.5" 461
182.236.164.11 - - [11/Mar/2018:18:20:52] "GET /oldlink?itemId=EST-19&JSESSIONID=SD6SL8FF10ADFF531
01 HTTP 1.1" 200 1653 "http://www.buttercupgames.com/oldlink?itemId=EST-19" "Mozilla/5.0 (Macintos
h; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5"
915
182.236.164.11 - - [11/Mar/2018:18:20:53] "POST /cart.do?action=addtocart&itemId=EST-6&productId=M
B-AG-G07&JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1" 200 533 "http://www.buttercupgames.com/product.s
creen?productId=MB-AG-G07" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTM
L, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 470
182.236.164.11 - - [11/Mar/2018:18:20:53] "GET /cart.do?action=addtocart&itemId=EST-12&productId=W
C-SH-G04&JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1" 200 2833 "http://www.buttercupgames.com/product.
screen?productId=WC-SH-G04" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHT
ML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 968
host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie
```

This search groups events together based on the IP addresses accessing the server and the time constraints. The search results might have multiple values for some fields, such as `host` and `source`. For example, requests from a single IP could come from multiple hosts if multiple people are shopping from the same office. For more information, read the topic About transactions in the *Knowledge Manager Manual*.

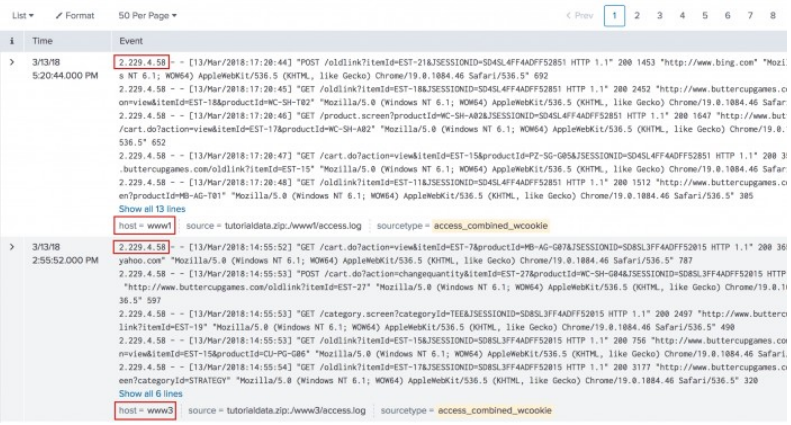
2. Transaction of Web access events based on host and client IP

This example uses the sample data from the Search Tutorial but should work with any format of Apache web access log. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to **get the tutorial data into Splunk**. Use the time range **Yesterday** when you run the search.

Define a transaction based on Web access events that have a unique combination of `host` and `clientip` values. The first and last events in the transaction should be no more than thirty seconds apart and each event should not be longer than five seconds apart.

```
sourcetype=access_* | transaction clientip host maxspan=30s maxpause=5s
```

This search produces the following events list.



```
i   Time               Event
>   3/13/18            2.229.4.58 - - [13/Mar/2018:17:20:44] "POST /oldlink?itemId=EST-21&JSESSIONID=SD4SL4FF4ADFF52851 HTTP 1.1" 200 1453 "http://www.bing.com" "Mozi
    5:20:44.000 PM     s NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 692
                       2.229.4.58 - - [13/Mar/2018:17:20:45] "GET /oldlink?itemId=EST-18&JSESSIONID=SD4SL4FF4ADFF52851 HTTP 1.1" 200 2452 "http://www.buttercupgames.c
                       on=view&itemId=EST-18&productId=WC-SH-T02" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safar
                       2.229.4.58 - - [13/Mar/2018:17:20:46] "GET /product.screen?productId=WC-SH-A02&JSESSIONID=SD4SL4FF4ADFF52851 HTTP 1.1" 200 1647 "http://www.but
                       /cart.do?action=view&itemId=EST-17&productId=WC-SH-A02" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.
                       536.5" 652
                       2.229.4.58 - - [13/Mar/2018:17:20:47] "GET /cart.do?action=view&itemId=EST-15&productId=PZ-SG-G05&JSESSIONID=SD4SL4FF4ADFF52851 HTTP 1.1" 200 3
                       .buttercupgames.com/oldlink?itemId=EST-15" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safar
                       2.229.4.58 - - [13/Mar/2018:17:20:48] "GET /oldlink?itemId=EST-11&JSESSIONID=SD4SL4FF4ADFF52851 HTTP 1.1" 200 1512 "http://www.buttercupgames.c
                       en?productId=MB-AG-T01" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 305
                       Show all 13 lines
                       host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie
>   3/13/18            2.229.4.58 - - [13/Mar/2018:14:55:52] "GET /cart.do?action=view&itemId=EST-7&productId=MB-AG-G07&JSESSIONID=SD8SL3FF4ADFF52015 HTTP 1.1" 200 36
    2:55:52.000 PM     yahoo.com" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 787
                       2.229.4.58 - - [13/Mar/2018:14:55:53] "POST /cart.do?action=changequantity&itemId=EST-27&productId=WC-SH-G04&JSESSIONID=SD8SL3FF4ADFF52015 HTTP
                        "http://www.buttercupgames.com/oldlink?itemId=EST-27" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1
                       36.5" 597
                       2.229.4.58 - - [13/Mar/2018:14:55:53] "GET /category.screen?categoryId=TEE&JSESSIONID=SD8SL3FF4ADFF52015 HTTP 1.1" 200 2497 "http://www.butterc
                       link?itemId=EST-19" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 490
                       2.229.4.58 - - [13/Mar/2018:14:55:53] "GET /oldlink?itemId=EST-15&JSESSIONID=SD8SL3FF4ADFF52015 HTTP 1.1" 200 756 "http://www.buttercupgames.co
                       n=view&itemId=EST-15&productId=CU-PG-G06" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari.
                       2.229.4.58 - - [13/Mar/2018:14:55:54] "GET /oldlink?itemId=EST-17&JSESSIONID=SD8SL3FF4ADFF52015 HTTP 1.1" 200 3177 "http://www.buttercupgames.c
                       een?categoryId=STRATEGY" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 320
                       Show all 6 lines
                       host = www3   source = tutorialdata.zip:./www3/access.log   sourcetype = access_combined_wcookie
```

Each of these events have a distinct combination of the IP address (`clientip`) values and `host` values within the limits of the time constraints specified in the search.

3. Purchase transactions based on IP address and time range

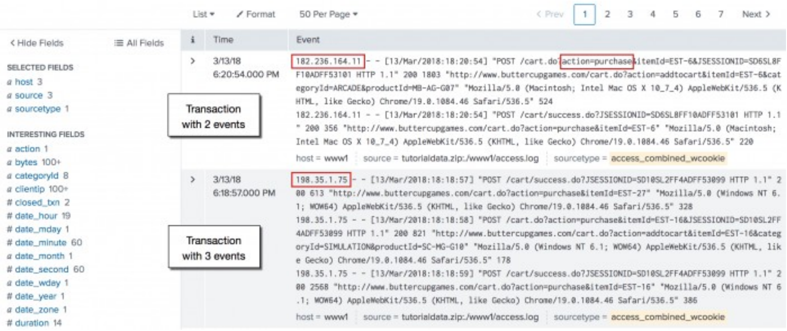
This example uses the sample data from the Search Tutorial but should work with any format of Apache web access log. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to **get the tutorial data into Splunk**. Use the time range **Yesterday** when you run the search.

This search defines a purchase transaction as 3 events from one IP address which occur in a 10 minute span of time.

```
sourcetype=access_* action=purchase | transaction clientip maxspan=10m
maxevents=3
```

This search defines a purchase event based on Web access events that have the `action=purchase` value. These results are then piped into the `transaction` command. This search identifies purchase transactions by events that share the same `clientip`, where each session lasts no longer than 10 minutes, and includes no more than 3 events.

This search produces the following events list:



```
3/13/18
6:20:54.000 PM

182.236.164.11 - - [13/Mar/2018:18:20:54] "POST /cart.do?action=purchase&itemId=EST-6&JSESSIONID=SD6SL8F
F10ADFF53101 HTTP 1.1" 200 1803 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-6&cat
egoryId=ARCADE&productId=MB-AG-G07" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (K
HTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 524
182.236.164.11 - - [13/Mar/2018:18:20:54] "POST /cart/success.do?JSESSIONID=SD6SL8FF10ADFF53101 HTTP 1.1
" 200 356 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-6" "Mozilla/5.0 (Macintosh;
Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 220
host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie

3/13/18
6:18:57.000 PM

198.35.1.75 - - [13/Mar/2018:18:18:57] "POST /cart/success.do?JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 2
00 613 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-27" "Mozilla/5.0 (Windows NT 6.
1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 328
198.35.1.75 - - [13/Mar/2018:18:18:58] "POST /cart.do?action=purchase&itemId=EST-16&JSESSIONID=SD10SL2FF
4ADFF53099 HTTP 1.1" 200 821 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-16&categ
oryId=SIMULATION&productId=SC-MG-G10" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, lik
e Gecko) Chrome/19.0.1084.46 Safari/536.5" 178
198.35.1.75 - - [13/Mar/2018:18:18:59] "POST /cart/success.do?JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 2
00 2568 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-16" "Mozilla/5.0 (Windows NT 6
.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 386
host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie
```

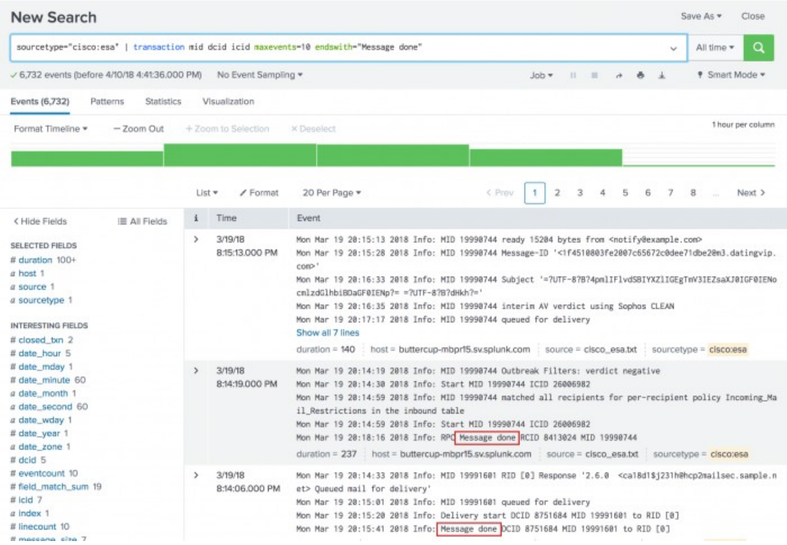
4. Email transactions based on maxevents and endswith

This example uses sample email data. You should be able to run this search on any email data by replacing the `sourcetype=cisco:esa` with the `sourcetype` value and the `mailfrom` field with email address field name in your data. For example, the email might be `To`, `From`, or `Cc`).

This example defines an email transaction as a group of up to 10 events. Each event contains the same value for the `mid` (message ID), `icid` (incoming connection ID), and `dcid` (delivery connection ID). The last event in the transaction contains a **Message done** string.

```
sourcetype="cisco:esa" | transaction mid dcid icid maxevents=10
endswith="Message done"
```

This search produces the following list of events:



```
sourcetype="cisco:esa" | transaction mid dcid icid maxevents=10 endswith="Message done"
```

✓ 6,732 events (before 4/10/18 4:41:36.000 PM) No Event Sampling

SELECTED FIELDS
# duration 100+
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
# closed_txn 2
# date_hour 5
# date_mday 1
# date_minute 60
a date_month 1
a date_second 60
a date_wday 1
# date_year 1
a date_zone 1
# dcid 5
# eventcount 10
# field_match_sum 19
# icid 7
a index 1
# linecount 10
# message_size 7

3/19/18 8:15:13.000 PM

```
Mon Mar 19 20:15:13 2018 Info: MID 19990744 ready 15204 bytes from <notify@example.com>
Mon Mar 19 20:15:28 2018 Info: MID 19990744 Message-ID '<1f4510803fe2007c65672c0dee71dbe20m3.datingvip.
com>'
Mon Mar 19 20:16:33 2018 Info: MID 19990744 Subject '=?UTF-8?B?4pmlIFlvdSBIYXZlIGEgTmV3IEZsaXJ0IGF0IENo
cmlzdGlhbiBDaGF0IENp?= =?UTF-8?B?dHkh?='
Mon Mar 19 20:16:35 2018 Info: MID 19990744 interim AV verdict using Sophos CLEAN
Mon Mar 19 20:17:17 2018 Info: MID 19990744 queued for delivery
```

Show all 7 lines

duration = 140 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa

3/19/18 8:14:19.000 PM

```
Mon Mar 19 20:14:19 2018 Info: MID 19990744 Outbreak Filters: verdict negative
Mon Mar 19 20:14:30 2018 Info: Start MID 19990744 ICID 26006982
Mon Mar 19 20:14:59 2018 Info: MID 19990744 matched all recipients for per-recipient policy Incoming_Ma
il_Restrictions in the inbound table
Mon Mar 19 20:14:59 2018 Info: Start MID 19990744 ICID 26006982
Mon Mar 19 20:18:16 2018 Info: RPC Message done RCID 8413024 MID 19990744
```

duration = 237 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa

3/19/18 8:14:06.000 PM

```
Mon Mar 19 20:14:33 2018 Info: MID 19991601 RID [0] Response '2.6.0  <ca18d1$j231h@hcp2mailsec.sample.n
et> Queued mail for delivery'
Mon Mar 19 20:15:01 2018 Info: MID 19991601 queued for delivery
Mon Mar 19 20:15:20 2018 Info: Delivery start DCID 8751684 MID 19991601 to RID [0]
Mon Mar 19 20:15:41 2018 Info: Message done DCID 8751684 MID 19991601 to RID [0]
```

By default, only the first 5 events in a transaction are shown. The first transaction contains 7 events and the last event is hidden. The second and third transactions show the **Message done** string in the last event in the transaction.

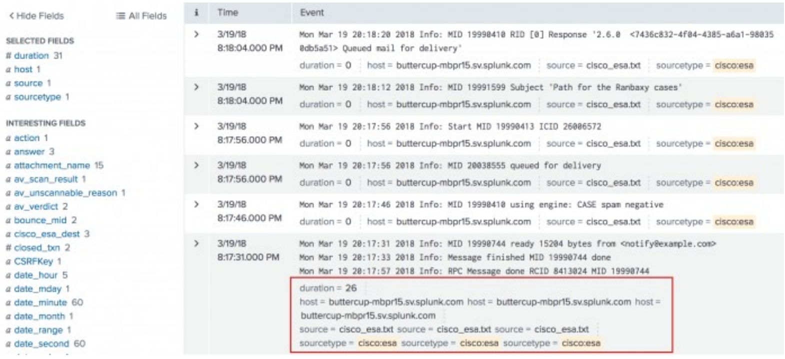
5. Email transactions based on maxevents, maxspan, and mvlist

This example uses sample email data. You should be able to run this search on any email data by replacing the `sourcetype=cisco:esa` with the `sourcetype` value and the `mailfrom` field with email address field name in your data. For example, the email might be `To`, `From`, or `Cc`).

This example defines an email transaction as a group of up to 10 events. Each event contains the same value for the `mid` (message ID), `icid` (incoming connection ID), and `dcid` (delivery connection ID). The first and last events in the transaction should be no more than thirty seconds apart.

```
sourcetype="cisco:esa" | transaction mid dcid icid maxevents=10
maxspan=30s mvlist=true
```

By default, the values of multivalue fields are suppressed in search results with the default setting for `mvlist`, which is false. Specifying `mvlist=true` in this search displays all of the values of the selected fields. This produces the following events list:



SELECTED FIELDS
# duration 31
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a action 1
a answer 3
a attachment_name 15
a av_scan_result 1
a av_unscannable_reason 1
a av_verdict 2
a bounce_mid 2
a cisco_esa_dest 3
# closed_txn 2
a CSRFKey 1
a date_hour 5
a date_mday 1
a date_minute 60
a date_month 1
a date_range 1
a date_second 60

i	Time	Event
>	3/19/18 8:18:04.000 PM	Mon Mar 19 20:18:20 2018 Info: MID 19990410 RID [0] Response '2.6.0 <7436c832-4f04-4385-a6a1-98035 0db5a51> Queued mail for delivery' duration = 0 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa
>	3/19/18 8:18:04.000 PM	Mon Mar 19 20:18:12 2018 Info: MID 19991599 Subject 'Path for the Ranbaxy cases' duration = 0 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa
>	3/19/18 8:17:56.000 PM	Mon Mar 19 20:17:56 2018 Info: Start MID 19990413 ICID 26006572 duration = 0 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa
>	3/19/18 8:17:56.000 PM	Mon Mar 19 20:17:56 2018 Info: MID 20038555 queued for delivery duration = 0 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa
>	3/19/18 8:17:46.000 PM	Mon Mar 19 20:17:46 2018 Info: MID 19990410 using engine: CASE spam negative duration = 0 | host = buttercup-mbpr15.sv.splunk.com | source = cisco_esa.txt | sourcetype = cisco:esa

>	3/19/18
	8:17:31.000 PM
		Mon Mar 19 20:17:31 2018 Info: MID 19990744 ready 15204 bytes from <notify@example.com>
		Mon Mar 19 20:17:33 2018 Info: Message finished MID 19990744 done
		Mon Mar 19 20:17:57 2018 Info: RPC Message done RCID 8413024 MID 19990744
		duration = 26
		host = buttercup-mbpr15.sv.splunk.com host = buttercup-mbpr15.sv.splunk.com host = buttercup-mbpr15.sv.splunk.com
		source = cisco_esa.txt source = cisco_esa.txt source = cisco_esa.txt
		sourcetype = cisco:esa sourcetype = cisco:esa sourcetype = cisco:esa

Here you can see that each transaction has a duration that is less than thirty seconds. Also, if there is more than one value for a field, each of the values is listed.

6. Transactions with the same session ID and IP address

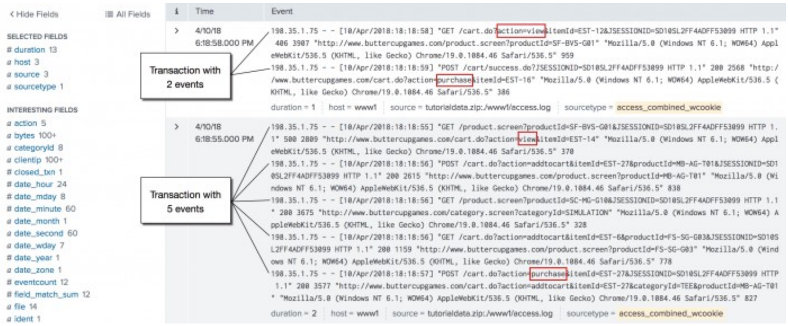
This example uses the sample data from the Search Tutorial but should work with any format of Apache web access log. To try this example on your own Splunk instance, you must download the sample data and follow the instructions to **get the tutorial data into Splunk**. Use the time range **All time** when you run the search.

Define a transaction as a group of events that have the same session ID, `JSESSIONID`, and come from the same IP address, `clientip`, and where the first event contains the string, "view", and the last event contains the string, "purchase".

```
sourcetype=access_* | transaction JSESSIONID clientip startswith="view"
endswith="purchase" | where duration>0
```

The search defines the first event in the transaction as events that include the string, "view", using the `startswith="view"` argument. The `endswith="purchase"` argument does the same for the last event in the transaction.

This example then pipes the transactions into the `where` command and the `duration` field to filter out all of the transactions that took less than a second to complete. The `where` filter cannot be applied before the `transaction` command because the `duration` field is added by the `transaction` command.



```
4/10/18 6:18:58.000 PM
198.35.1.75 - - [10/Apr/2018:18:18:58] "GET /cart.do?action=view&itemId=EST-12&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 406 3907 "http://www.buttercupgames.com/product.screen?productId=SF-BVS-G01" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 959
198.35.1.75 - - [10/Apr/2018:18:18:59] "POST /cart/success.do?JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 200 2568 "http://www.buttercupgames.com/cart.do?action=purchase&itemId=EST-16" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 386
duration = 1   host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie

4/10/18 6:18:55.000 PM
198.35.1.75 - - [10/Apr/2018:18:18:55] "GET /product.screen?productId=SF-BVS-G01&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 500 2809 "http://www.buttercupgames.com/cart.do?action=view&itemId=EST-14" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 370
198.35.1.75 - - [10/Apr/2018:18:18:56] "POST /cart.do?action=addtocart&itemId=EST-27&productId=MB-AG-T01&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 200 2615 "http://www.buttercupgames.com/product.screen?productId=MB-AG-T01" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 838
198.35.1.75 - - [10/Apr/2018:18:18:56] "GET /product.screen?productId=SC-MG-G10&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 200 3675 "http://www.buttercupgames.com/category.screen?categoryId=SIMULATION" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 328
198.35.1.75 - - [10/Apr/2018:18:18:56] "GET /cart.do?action=addtocart&itemId=EST-6&productId=FS-SG-G03&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 200 1159 "http://www.buttercupgames.com/product.screen?productId=FS-SG-G03" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 778
198.35.1.75 - - [10/Apr/2018:18:18:57] "POST /cart.do?action=purchase&itemId=EST-27&JSESSIONID=SD10SL2FF4ADFF53099 HTTP 1.1" 200 3577 "http://www.buttercupgames.com/cart.do?action=addtocart&itemId=EST-27&categoryId=TEE&productId=MB-AG-T01" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5" 827
duration = 2   host = www1   source = tutorialdata.zip:./www1/access.log   sourcetype = access_combined_wcookie
```

You might be curious about why the transactions took a long time, so viewing these events might help you to troubleshoot. You won't see it in this data, but some transactions may take a long time because the user is updating and removing items from his shopping cart before he completes the purchase. Additionally, this search is run over all events. There is no filtering before the `transaction` command. Anytime you can filter the search before the first pipe, the faster the search runs.

See also

stats, concurrency

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has using the transaction command.