Splunk® Enterprise Search Manual 7.1.1

Identify and group events into transactions

Generated: 6/30/2018 9:44 pm

Identify and group events into transactions

You can search for related events and group them into one single event, called a *transaction* (sometimes referred to as a session).

Transactions can include:

- Different events from the same source and the same host.
- Different events from different sources from the same host.
- Similar events from different hosts and different sources.

Search for transactions using the `transaction` command either in Splunk Web or at the CLI. The `transaction` command yields groupings of events which can be used in reports. To use `transaction`, either call a transaction type (that you configured via transactiontypes.conf), or define transaction constraints in your search by setting the search options of the `transaction` command.

Transaction search options

Transactions returned at search time consist of the raw text of each event, the shared event types, and the field values. Transactions also have additional data that is stored in the fields: `duration` and `transactiontype`.

- **`duration`** contains the duration of the transaction (the difference between the timestamps of the first and last events of the transaction).
- **`transactiontype`** is the name of the transaction (as defined in `transactiontypes.conf` by the transaction's stanza name).

You can add `transaction` to any search. For best search performance, craft your search and then pipe it to the transaction command. For more information see the topic on the `transaction` command in the Search Reference manual.

Follow the `transaction` command with the following options. **Note:** Some `transaction` options do not work in conjunction with others.

`name=<transaction-name>`

- Specifies the name of a stanza from `transactiontypes.conf`. Use this to invoke a **transaction type** that you have already configured for reuse. If other arguments are provided, they overule values specified for the same arguments in the transaction rule. For example, if `web_purchase`, the transaction rule you're invoking, is configured with `maxevents=10`, but you'd

like to run it with a different value for `maxevents`, add `maxevents` to the search string with the value you want:

```
sourcetype=access_* | transaction name=web_purchase maxevents=5
```

[field-list]

- This is a comma-separated list of fields, such as `...| transaction host,cookie`
- If set, each event must have the same field(s) to be considered part of the same transaction.
- Events with common field names and different values will not be grouped.
  - For example, if you add `...| transaction host`, then a search result that has `host=mylaptop` can never be in the same transaction as a search result with `host=myserver`.
  - A search result that has no `host` value can be in a transaction with a result that has `host=mylaptop`.

`match=closest`

- Specify the matching type to use with a transaction definition.
- The only value supported currently is closest.

`maxspan=[<integer>s | m | h | d]`

- Set the maximum duration of one transaction.
- Can be in seconds, minutes, hours or days.
  - For example: 5s, 6m, 12h or 30d.
- Defaults to `maxspan=-1`, for an "all time" timerange.

`maxpause=[<integer> s|m|h|d]`

- Specifies the maximum pause between transactions.
- Requires there be no pause between the events within the transaction greater than maxpause.
- If the value is negative, the maxspause constraint is disabled.
- Defaults to `maxpause=-1`.

`startswith=<string>`

- A search or eval-filtering expression which, if satisfied by an event, marks the beginning of a new transaction.
- For example:
  - `startswith="login"`
  - `startswith=(username=foobar)`
  - `startswith=eval(speed_field < max_speed_field)`
  - `startswith=eval(speed_field < max_speed_field/12)`
- Defaults to `""`.

`endswith=<transam-filter-string>`

- A search or eval-filtering expression which, if satisfied by an event, marks the end of a transaction.
- For example:
  - `endswith="logout"`
  - `endswith=(username=foobar)`
  - `endswith=eval(speed_field < max_speed_field)`
  - `endswith=eval(speed_field < max_speed_field/12)`
- Defaults to `""`.

For `startswith` and `endswith`, `<transam-filter-string>` is defined with the following syntax: `"<search-expression>" | (<quoted-search-expression>) | eval(<eval-expression>`

- `<search-expression>` is a valid search expression that does not contain quotes.
- `<quoted-search-expression>` is a valid search expression that contains quotes.
- `<eval-expression>` is a valid eval expression that evaluates to a boolean.

Examples:

- search expression: `(name="foo bar")`
- search expression: `"user=mildred"`
- search expression: `("search literal")`
- eval bool expression: `eval(distance/time < max_speed)`

Example transaction search

Run a search that groups together all of the web pages a single user (or client IP address) looked at over a time range.

This search takes events from the access logs, and creates a transaction from events that share the same `clientip` value that occurred within 5 minutes of each other (within a 3 hour time span).

```
sourcetype=access_combined | transaction clientip maxpause=5m
maxspan=3h
```

Refer to the transaction command topic in the Search Reference Manual for more examples.