Table of Contents

Configure Streams ............................................................................................................ 1
 Configure streams to capture network data ................................................................. 1
 Stream field details ....................................................................................................... 18
 Stream aggregation methods ...................................................................................... 19

Global IP Filters .......................................................................................................... 23
 Use Global IP filters .................................................................................................... 23

Distributed Forwarder Management ............................................................................ 24
 Distributed Forwarder Management ............................................................................. 24

Stream Estimate .......................................................................................................... 28
 Use Stream Estimate to preview index volume ............................................................ 28

Dashboards .................................................................................................................. 31
 Stream Informational Dashboards ............................................................................... 31
 Stream Admin dashboards .......................................................................................... 40
Configure Streams

Configure streams to capture network data

Splunk Stream supports passive capture of network data for a variety of protocols. This page shows you how to create stream configurations that determine the characteristics of network data capture, which Stream forwarder (streamfwd binary) performs on the network interface.

What is a stream?

In Splunk Stream, a grouping of network event data is called a "stream." You can use the Configure Streams UI to create any number of unique streams for supported network protocols. Stream forwarder (streamfwd binary) retrieves your stream configurations, and captures data on the network interface based on those configurations.

When you create a stream, depending on the stream type, you can:

- Specify a network protocol and target protocol fields.
- Create filters to constrain data capture and minimize indexer requirements.
- Apply aggregation methods for statistical analysis of captured event data.
- Use content extraction rules to capture subsets of data.
- Use file extraction to capture files for analysis.
- Capture full network packets for detailed inspection.

Supported stream types

Splunk Stream supports these stream types:

<table>
<thead>
<tr>
<th>Stream Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata stream</td>
<td>Captures network traffic metadata generated by network and system devices.</td>
</tr>
<tr>
<td>Packet stream</td>
<td>Captures full network packets based on specific target fields. Enables searches against raw packet data. Supports extraction of packet contents and download of raw packets for detailed inspection.</td>
</tr>
</tbody>
</table>
Stream Type | Description
--- | ---
Ephemeral stream | Monitor ephemeral (time-limited) streams in Splunk apps that support ephemeral streams via Stream REST API.

**Where are stream configurations stored?**

The streams that you configure using the Configure Streams UI are stored in the KV store. You cannot access them from the file system. You can, however, access individual stream configurations in the KV store using the Stream REST API. See /streams/{stream_id} in the Splunk Stream REST API reference.

**Configure metadata streams**

The Configure Streams UI provides a workflow wizard that walks you through the process of creating a new stream. When creating a new metadata stream, in addition to selecting a protocol and fields, you can apply aggregation and create filters for the specific stream.

**Create new metadata stream**

1. In the Splunk App for Stream (splunk_app_stream) main menu, click Configure > Configure Streams. The Configure Streams UI opens.
3. Under Basic Info, select a Protocol for your new stream. For example, http.
4. Enter a Name and optionally enter a Description. Click Next.
5. (optional) Under Aggregation, click Yes, Every. Then enter a time in seconds, for example, 60 seconds. This is the interval over which data aggregation occurs. For more information, see Use Aggregation on this page.
6. Click Next.
7. Under Fields, select the specific protocol attributes (fields) that you want to capture. For example, dest_ip, src_ip, bytes_in, bytes_out, and so on. If you have enabled aggregation for the stream, you can also optionally change the selected Aggregation type (Key or Aggregated) for any field. Click Next.
8. (optional) Under Filters, click Create New Filter. The Create New Filter modal appears. Select the Field for which you want to create the filter, for example, http_method. Next, select the Comparison type. Then, enter the Value. Click Create. Your new filter allows data to pass through based on the condition defined in the filter. For more information on filters, see Create new filter on this page.
9. Click Next.
10. Under Settings, select the Index to which you want to send captured data. Then choose the Status for the new stream: Enabled, Disabled, or Estimate. Click Next.
11. Select the checkbox for the specific forwarder groups to which you want to add this stream. Click Create Stream. This creates the new stream.
12. Click Done. This sends your new stream configuration to the streamfwd binary on Splunk_TA_stream where data capture occurs.

Your new stream appears in the Configure Streams UI.

After you create a new metadata stream, you can modify the stream to apply additional stream capture rules, including aggregation methods and filters. You can also define content extraction rules to capture a subset of data from a protocol string. See Use Content Extraction on this page.

**Set stream mode**

To enable data capture for a particular stream, you must set the stream to the Enabled mode. You can set the stream mode in the Configure Stream UI at any time. Choose from the following three modes:

- **Enabled**: Enabled mode starts stream data capture and indexing, and generates index volume stats by default.
- **Estimate**: Estimate mode generates data index volume stats only for any stream, without sending data to indexers.
- **Disabled**: Disabled mode stops stream data capture and indexing, and stops the collection of index volume stats.

On the Configure Streams UI main page, select a mode for any stream:
How Estimate mode works

When you set a metadata stream to Estimate mode, the app generates index volume stats for the stream, without sending the actual data to your indexers. These index volume stats populate the Stream Estimate dashboard.

Use the Estimate mode and Stream Estimate dashboard to determine the amount of data that a particular stream will ingest. This can help you calculate your indexer requirements, fine-tune your stream capture configurations, and conserve indexer space.

For more information, see Stream Estimate in this manual.

Splunk Stream also collects index volume stats for all streams in the Enabled mode. All pre-built streams are in Estimate mode by default.

Select protocol fields

You can select the specific fields that you want to capture for any stream.

1. On the Configure Streams main page, select the name of the network protocol that you want to capture.
2. The Protocol Events page for the selected protocol opens.
3. Select the Enable checkbox for the specific field(s) you want to add.
4. Click Save.

![Protocol Events screenshot](image-url)
Create new filters

You can add filters to specific fields in a stream that allow data to pass through based on conditions that you define. You define conditions by setting up a comparison, such as "Less than," "Equals," "Contains," and so on, between a field name and a specific value. Filter values can be either strings or numeric values.

For example, a filter might specify the condition that the "http_method" field contains the value "GET." If all (or any) http_method events match this condition, the stream event data passes through.

1. In the Configure Streams UI, click on the name of the stream.
2. Click the Filters tab.
3. Click Create New Filter. The Create New Filter modal opens
4. In the Field drop-down menu, select the name of the field to which you want to apply the filter. For example, "bytes_in."
5. In the Comparison drop-down menu, select the type of comparison you want to use for your filter. For example, "Less than."
6. Enter the Value that defines the condition of the comparison. For example, "25000."
7. Click Save.
8. For Match Filters, select All or Any. This applies to events that return multiple values for a field. If match is set to "All", then all values of the field must satisfy the condition for the filter to engage. If the match is set to "Any", then any value of the field that satisfies the condition causes the filter to engage.
9. Click OK.

The app adds the filter to the streamfwd.conf configuration file, and the filter now appears in the list of filters for that stream.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Comparison</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>http_method</td>
<td>Contains</td>
<td>GET</td>
</tr>
<tr>
<td>bytes_in</td>
<td>Less than</td>
<td>25000</td>
</tr>
</tbody>
</table>

Apply aggregation

You can apply aggregation methods for statistical analysis of captured events over specific time intervals. The app groups events into aggregation buckets, with one bucket allocated for each unique collection of Key fields. At the end of the time interval, the app emits an object that represents each bucket.
For example, you might enable aggregation over a 60 second interval, assign Key to the src_ip field, and apply the sum aggregate function to the bytes_in field. In this case, the app creates a bucket for each unique src_ip value it sees, and sums the number of bytes_in over a 60 second interval for each bucket.

**Aggregation types**

You must set each field that you enable for aggregation to one of the following two aggregation types:

- **Key**: Fields that have aggregation type "Key" are used for grouping data into buckets.
- **Aggregated**: Fields that have aggregation type "Aggregated" can have one or more aggregate functions applied to them. For a complete list of aggregate functions, see Stream aggregation methods in this manual.

**Set up aggregation**

1. In the Configure Streams UI, click on the name of your stream. The configuration page for that particular stream opens.
2. Under Aggregation, click Yes, every, then enter a time in secs. This enables Aggregation for the particular stream and determines the time interval over which data aggregation occurs.
3. (optional) Under Top Fields, click Yes, only index top, then enter a number. From the dropdown, you can select count or any aggregated field which is configured, e.g. \( \text{sum}(\text{bytes_in}) \), as the basis for sorting.
4. Select the Enable checkbox for each field you want to aggregate.
5. Select an Aggregation Type for each enabled field.
   - Select Key to use the field for specifying aggregation buckets. A separate bucket will be generated for each distinct value of the Key field over the selected time interval.
   - Select Aggregated to enable aggregation for that particular field. The number of currently selected aggregate functions appears in parentheses. For example, (1). The sum aggregate function is selected by default.
6. Click on the number in parentheses (x) to select aggregate functions for a particular field. A modal showing a list of available aggregate functions for that field appears.
7. Select one or more aggregate functions. Click Save.
8. Click Save.
About multiple Key fields

If an aggregate event includes multiple Key fields, Stream looks for unique combinations of values of those fields and creates a separate bucket for each combination.

For example, if you assign Key to the field src_ip and you apply the sum aggregate function to the field bytes_in, then for each time interval, the app creates a bucket for each unique src_ip value that occurs, and sums the number of bytes_in over that interval for each bucket. If you also assign Key to the dest_ip field, the app creates a bucket for each unique pair of src_ip and dest_ip.

For more information, see Stream aggregation methods in this manual.

Use content extraction

You can use content extraction rules to capture and index a specific subset of data from a protocol string field. You can also use content extraction to generate MD 5 hashes or hexadecimal numbers of non-numeric fields.
**Content extraction types**

Splunk Stream supports the following content extraction types:

- Regex
- MD5 hash
- Hexadecimal

**Create content extraction rules**

Content extraction rules use regular expressions to extract sections of data from a parent field. This lets you capture only the specific pieces of data that you require for analysis, without indexing extraneous data.

For example, some string fields, such as `src_content dest_content`, and `cookies`, contain long strings of data. In many cases, the entire string of data is not useful. Using content extraction rules, you can limit data capture to specific pieces of data, such as a name, ID, account number and so on.

You can also specify a capturing group match, which outputs either the first value that matches the regular expression, or the "list" of all values that match the regular expression. Each content extraction rule creates a new field that captures only that data specified by the rule. The original field is not modified.

To create a content extraction rule:

1. In the Configure Streams UI, click on the stream that contains the field from which you want to extract content. The events page opens for the particular stream.
2. Click the **Actions** menu for the particular field, then select **Extract New Field**.
   
   ![Extract New Field dialog](image)

   The Content Extraction dialog appears.

   **Note:** It is not necessary for the original field to be enabled. If you also want to index the original field, consider using a field transformation instead. For more information, see field transformations in the Knowledge Manager Manual.
3. Enter a name for the content extraction rule. For example, "readable_cookie."

4. Enter a description for the content extraction rule. For example, "Separate the cookies into name/value pairs in a readable format."

5. In the **Extraction Rule** field, enter the regular expression for the content that you want to extract. For example, we can use the following regular expression to extract a name and value pair from the cookie field:

```
(.*?);=(.*?);</code>
```

**Note:** Stream uses Boost Perl Regular Expression syntax.

6. Under **Match**, select:
   - Select **First** to return only the first value that matches the regex.
   - Select **All** to return the "list" of all values that match the regex.

7. In the **Extraction Format** plain text box, enter the format for the extraction. For example, enter $1, $2 to return the first and second values that match the regex.

This image shows the complete content extraction rule for our "readable_cookie" example:
8. Click **Save**.

The new field appears in the list of Fields on the events page for the particular stream.

Splunk search results for the "readable_cookie" field in our example should appear similar to this:

```
Cookie name is BARBAR-AGENT-FILES Cookie value is 245a43620
Cookie name is BARBAR-AGENT-FILES Cookie value is 30;
Cookie name is BARBAR-AGENT-FILES Cookie value is 31;
Cookie name is BARBAR-AGENT-FILES Cookie value is 32;
Cookie name is BARBAR-AGENT-FILES Cookie value is 33;
```

**Note:** The `<MaxEventQueueSize>` option in `streamfwd.conf` determines the maximum number of events that Splunk_TA_stream can queue for delivery to Splunk indexers. By default `<MaxEventQueueSize>` supports 10k events. To increase or decrease the maximum event queue size, modify the value of `<MaxEventQueueSize>` in `streamfwd.conf`. See Configure streamfwd.conf in the Splunk Stream Installation and Configuration Manual.

**Extract fields as MD5 hash or hexadecimal**

You can use content extraction to generate an MD5 hash or hexadecimal encoding of any non-numeric field for any protocol. MD5 hashes are useful for masking sensitive data in search results, such as user names, passwords, and other important account information. Hexadecimal encoding is useful for representing arbitrary binary data that can interfere with the Splunk search UI.

You can also use file hashing to detect if a specific file is being transmitted over your network, without having to store the entire contents of the file, which might be quite large. For example, you might store the hash of a file (such as a sensitive document or piece of malware), then compare that hash to the hash of email attachments you capture to see if it matches.

Each field that you extract as an MD5 hash or hexadecimal generates a new field. The original field does not need to be enabled. This lets you view a secure fingerprint of the field value in search results, without exposing the original field value.

An MD5 hash is 32 characters long by default. For additional security, you can truncate MD5 hash values to any number of characters, and specify an offset from the 1st character at left.

To extract a field as an MD5 hash:

1. Click the **Actions** menu for the particular field that you want to hash, then select **Extract New Field**.
2. Enter a name for the MD5 hash field that you want generate. For example, "src_content_MD5_hash."
3. Enter a description for the new MD5 hash field. For example, "MD5 hash of src_content_field."

4. In the Extraction Type menu, select MD5 hash.

5. In the Hash Length field, specify the number of characters to use for the hash. Leave blank for default 32 characters.

6. In the Hash Offset field, specify the number of characters to offset from 1st character at left. Leave blank for default of 0.

7. Click Save.

The new field appears in the list of Fields on the events page for the particular stream.

Splunk search results for the "src_content_MD5_hash" field in our example should appear similar to this:

MD5 hash content extraction is pre-configured for specific fields in HTTP and SMTP protocol streams.

**Use file extraction**

Splunk Stream 7.1.0 and later supports file extraction from metadata streams. File extraction lets you capture files from network traffic, such as emails, email attachments, images, pdfs, and so on. You can identify extracted files in Splunk search results, and use workflow actions to download those files to your local machine.
File extraction supports http and smtp protocols only.

**File extraction prerequisites**

Before you can use file extraction to capture files with metadata streams, you must map your Splunk Stream deployment to a remote file server. The app uses the file server to store files extracted by Stream forwarder based on the metadata stream definition. See Configure file extraction in the Splunk Stream *Installation and Configuration Manual*.

Splunk Stream lets you capture network event data for a variety of network protocols. Make sure to consider your privacy and security obligations when selecting and using a remote file server for Splunk Stream data.

File extraction is not supported on Splunk Cloud.

**Configure file extraction for a metadata stream**

1. Create or clone a new HTTP or SMTP metadata stream. See Configure metadata streams.
2. In the Fields tab, enable one or both of the following fields:
   - `file_extracted_req`
   - `file_extracted_resp`
   
   For example, if you only need http uploads, enable `file_extracted_req` field.
3. Click *Save*.
   
   The metadata stream now extracts files from request and response data.
   
   Extracted files appear in search results. See Search for extracted files.

**Search for extracted files**

To search for extracted files:

1. In the Splunk Search and Reporting app, in the Search bar, enter the following event type:

   `eventtype="stream_extractedfilesaved"`

2. In your search results, look for the `extracted_file []` multi-value field.
   
   All extracted files in the network event appear in this field.
**Download extracted files**

To download extracted files that you identify in search results:

1. Expand the Event tab.
2. Find the `extracted_file` field in the expanded tab.
3. Click **Actions > Downloaded Extracted File**.
   The extracted file downloads to your local machine.

**Configure packet streams**

Splunk Stream 7.1.0 and later support packet streams. Packet streams let you capture raw network packets based on targets that you define. You can run Splunk searches against full packet data, and use workflow actions to download pcap files containing that data to your local machine.

**How targeted packet capture works**

Packet streams use targeted packet capture to collect full network packets. Unlike metadata streams, which send all data that match the stream to indexers, packet streams capture only those packets that match pre-defined target fields.

When you create a new packet stream, Stream forwarder picks up the packet stream definition, then captures and stores targeted packets in pcap files on a remote file server. Stream forwarder also indexes metadata that identifies the pcap files in searches and workflow actions.

**Packet stream prerequisites**

Before you can collect data using packet streams, you must map your Splunk Stream deployment to a remote file server. The app uses the file server to store pcap files that Stream forwarder generates based on the packet stream definition. See Configure targeted packet capture in the Splunk Stream Installation and Configuration Manual.

Splunk Stream lets you capture network event data for a variety of network protocols. Make sure to consider your privacy and security obligations when selecting and using a remote file server for Splunk Stream data.

Targeted packet capture is not supported on Splunk Cloud
Create new packet stream

1. Click **New Stream > Packet Stream**.
   The packet stream workflow wizard appears.
2. Enter a **Name** and **Description** (optional) for the new packet stream. Click **Next**.
3. On the Targets page, click **Create New Target**.
4. Configure the new target:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Specify the protocol field that you want to target.</td>
</tr>
<tr>
<td>Comparison</td>
<td>Select a comparison type to filter target field data based on specific values.</td>
</tr>
<tr>
<td>Value</td>
<td>Enter a value to compare candidate values against.</td>
</tr>
<tr>
<td>Any/All</td>
<td>Select the condition that events with multiple values for the field must satisfy.</td>
</tr>
</tbody>
</table>

5. Click **Create**.
   Your new target appears in the targets list.
6. Click **Next**.
7. On the expiration page, click **Add condition**.
   Packet stream capture is ephemeral.
8. Specify the conditions for packet stream expiration. For example, Elapsed Time/1 hour. Click **Next**.
9. On the Fields page, enable the fields that you want to include in the packet stream. Click **Next**.
10. On the Settings page, configure the following:

    | Field | Description                                                                 |
    |-------|-----------------------------------------------------------------------------|
    | Index | Select the index to use for storage of metadata generated by the packet stream. |
    | Status| Choose if the packet stream is Enabled or Disabled upon creation.             |

11. Add additional targets. (optional)
12. Select the match condition (**Any/All**) for the list of targets. This condition applies to all targets in the list.
13. Click **Next**.
14. Select the forwarder groups to use for this stream. Click **Create Stream**.
   The app creates the new packet stream.
15. Click **Done**. This sends your new packet stream configuration to the `streamfwd` binary where data capture occurs. Your new packet stream now appears in the configure streams UI.

**Search packet stream data**

To run a search against captured packets:

1. In the Splunk Search and Reporting app, in the Search bar, enter the following event type:

   `eventtype="stream_pcapsaved"`

2. Optionally add additional event terms to restrict search results.

**Download pcap files**

To download pcap files associated with a search:

1. Expand the **Event** tab.
2. Click **Event Actions > Download capture file**.
The pcap file downloads to your local machine.

Create custom (cloned) streams

You can clone any existing streams to create new custom streams. This lets you create variations on your streams and capture data with additional granularity.

When you clone a stream, the app produces an exact duplicate of the original stream, including all enabled fields and existing filters. You can then add additional capture rules, such as aggregation, filters, content extraction, and file extraction.

To create a custom (cloned) stream:

1. In the Configure Streams UI, click on the name of the stream you want to clone.
   The events page for the particular stream opens.
2. Click **Clone**.
   The Clone Stream dialog appears.
3. Enter a **Name** and **Description** for the new stream. Click **OK**.
   The new stream appears in the list of streams in the Configure Streams UI.
4. Click **Enabled** to enable capture for the cloned stream.

5. Click **Save**.

**About built-in metadata streams**

Splunk Stream includes several built-in metadata streams as examples. You can use these built-in streams as a starting point for creating new metadata streams. All built-in streams begin with "Splunk_" in their name. You can view and clone built-in streams in the Configure Streams UI.

The data that built-in metadata streams capture populates the apps Informational dashboards. For more information, see **Informational Dashboards** in this manual.
Support for ephemeral streams

Ephemeral streams give users of external Splunk apps the ability to schedule stream capture for a user-defined period of time. This is useful if you want to capture and analyze a limited number of network events pertaining to specific network activity or transactions over a specific time interval.

While you must configure and schedule ephemeral stream capture from within the context of the external Splunk app, you can view and perform certain actions on existing ephemeral streams (such as enable/disable) inside the Configure Stream UI.

How ephemeral streams work

Splunk apps that support ephemeral streams take advantage of the Stream REST API. Ephemeral streams are similar to normal streams, but ephemeral streams require two additional parameters, `createDate` and `expirationDate`, which specify the stream start and stop times, respectively, in epoch time. For example:

```
createDate: 1404259338
expirationDate: 1414259338
```

Splunk app for Stream automatically and permanently deletes the ephemeral stream when the server system time is greater than or equal to `expirationDate`.

To view ephemeral streams:

In the Configure Streams UI, click **Ephemeral Streams**. This opens the Stream Buckets page, which displays a list of your existing ephemeral streams.

Stream field details

This topic provides information about the specific protocol fields captured by Splunk Stream.

Latency information

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time_taken</td>
<td>The event duration in microseconds, i.e. time difference between timestamps of the last and first packets that comprise an event plus</td>
</tr>
</tbody>
</table>
Field | Description
--- | ---
client_rtt | The average round trip time in microseconds from the client to the point of capture — calculated based on a complex algorithm involving correlating data packet timestamps with corresponding acknowledgment packet timestamps.

Field | Description
--- | ---
server_rtt | The average round trip time in microseconds from the server to the point of capture — calculated based on a complex algorithm involving correlating data packet timestamps with corresponding acknowledgment packet timestamps.

Field | Description
--- | ---
request_time | The number of microseconds that it took the client to send the request, i.e. time difference between last and first request data packets (0 if request fits in a single packet).

Field | Description
--- | ---
response_time | Similar to request time, but for the server response data.

Field | Description
--- | ---
reply_time | The number of microseconds between the last request packet and the first response packet.

Field | Description
--- | ---
request_ack_time | The time difference between the last request packet and the ACK packet from the server acknowledging the last request packet.

Field | Description
--- | ---
response_ack_time | Similar to request_ack_time, but timing the acknowledgment of the last response packet.

### Stream aggregation methods

Splunk Stream lets you apply aggregation to network data at capture-time on the collection endpoint before data is sent to indexers. You can use aggregation to enhance your data with a variety of statistics that provide additional insight into activities on your network.
When you apply aggregation to a stream, only the aggregated data is sent to indexers. Using aggregation can thus help you decrease both storage requirements and license usage.

**Stream aggregate functions**

Splunk Stream supports a subset of the aggregate functions provided by the SPL (Splunk Processing Language) `stats` command to calculate statistics based on fields in your network event data. You can apply aggregate functions to your data when you configure a stream in the Configure Streams UI.

Splunk Stream supports these aggregate functions:

- sum
- sum squared
- max
- min
- mean
- median
- mode
- sample standard deviation
- population standard deviation
- sample variance
- population variance
- distinct count
- distinct values

For more information on aggregate functions, see Statistical and charting functions in the Splunk Enterprise Search Reference.

**How aggregates work**

You apply aggregate functions to stream events over a user-defined time interval. When Stream calculates the selected aggregates, it groups events into aggregation buckets, with one bucket allocated for each unique value of the "Key" field (or unique combination of values if there are multiple "Key" fields). At the end of the time interval, the app emits an object that represents each bucket.

For example, to gain more insight into the amount of inbound http traffic, you might select `src_ip` as a Key field, and apply aggregate functions such as `max`, `mean`, `std dev` (standard deviation), and so on to the `bytes_in` field of an http stream, over a 60 second time interval.
Stream calculates these aggregates for the bytes_in field for each unique value of src_ip that appears in the http stream, over the specified time interval. Search results for these aggregates might appear as follows:

```
6/27/16 5/01:11:13PM
5/01:11:134 PM
count: 73
endtime: 2016-06-28T00:01:11.134704Z
max(bytes_in): 161
mean(bytes_in): 161
count: 73
src_ip: 127.0.0.1
```

**Aggregated field syntax**

Aggregated fields in Splunk Stream version 6.6.0 and later have the following syntax:

```
function(field_name)
```

This is a change from version 6.5.x and earlier, where the aggregated field names matched the original field name (such as bytes_in) while actually containing the sum aggregate. To access the latest field aggregation capabilities in Splunk Stream, upgrade to Splunk Stream version 7.0.0, see Upgrade to Splunk Stream 7.0.0 in the Splunk Stream Installation and Configuration Manual.

To use the latest agg

To upgrade aggregated streams from earlier versions of the app to the new syntax in 6.6.0, Splunk Stream provides a migration script that runs automatically when you upgrade to version 6.6.0. For more information,

**About the count field**

Each aggregated event has a single count field that reflects the total number of raw events aggregated. For example, a search result that displays `count: 73` contains 73 total aggregated events, as shown:

```
6/27/16 5/01:11:134 PM
count: 73
endtime: 2016-06-28T00:01:11.134704Z
max(bytes_in): 161
mean(bytes_in): 161
count: 73
```
About the values aggregate

The `values` aggregate function produces a list (JSON array) of distinct values of the target field, even if the list contains a single entry. The values in the array are sorted in alphabetical order for text fields and in ascending order for numeric fields.

For example, you might apply the `values` aggregate to the `time_taken` field in an http stream to get a list of values for the number of microseconds it took to complete each flow event over the selected time interval. Search results for the `values(time_taken)` aggregate appear as follows:

```json
6/27/10
7:31:39 PM

{...
  "count": 8
  "ontime": "2016-06-28T02:31:39.030878Z"
  "mean(time_taken)": "256688"
  "src_ip": "209.53.7.304"
  "time_taken": "2016-06-28T02:31:39.030878Z"
  "values(time_taken)": [1.01
    0
    200365
    200685
    205565
    300425
    803983
    8422581
  ]
}
```

About the sum of squares aggregate

In version 6.5.x and earlier, any field X which was being aggregated had a corresponding field `psrsvd_ss_X` which contained the sum of squares of X. This field did not appear in the stream configuration, but was automatically generated. As of version 6.6.0, the corresponding field is called `sumsq(X)`, and can be selected for generation in the same way as any other aggregation method. (See Configure Streams UI.)
Global IP Filters

Use Global IP filters

You can use whitelist and blacklist filter rules to allow or ignore network data capture based on IP address.

Define a whitelist to allow data capture from IP addresses on that list only. Define a blacklist to ignore data capture from IP addresses on the list, and allow data capture from all other IPs.

Whitelist and blacklist IP filters follow these rules:

<table>
<thead>
<tr>
<th>Whitelist</th>
<th>Blacklist</th>
<th>Filter results</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>Captures all IPs</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Captures all IPs except blacklist items</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Captures only whitelist IPs</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Captures all IPs in whitelist OR IPs not in blacklist</td>
</tr>
</tbody>
</table>

Each filter entry may be a specific IP (v4 or v6) address, or a range of addresses using the following forms:

- 192.168.2.* (IPv4 octets may use * to indicate wildcard)
- 10.20.30.0/24 (IPv4 CIDR notation)
- 2001:0db8:85a3:0042:1000:8a2e:0370:7300/120 (IPv6 CIDR notation)

For more information, see Whitelist or blacklist specific incoming data.
Distributed Forwarder Management

Distributed Forwarder Management lets you create groups of stream forwarders with different stream capture configurations. You can use this feature to apply a specific configuration to multiple stream forwarders that have identical roles, for example specific server types (such as Linux or Windows OS) or network connection points.

Distributed Forwarder Management is useful in large scale enterprise deployments that might include hundreds or thousands of stream forwarders.

Create a stream forwarder group

To create a stream forwarder group you must specify a regex rule that matches host servers, define HTTP event collector endpoints (URLs), then select stream protocol capture for the group. You can group stream forwarders using the default Forwarder ID (which is the hostname), or specify custom Forwarder IDs to create new logical groups.

1. Click Distributed Forwarder Management. This opens the Distributed Forwarder Management page, which displays your existing stream forwarder groups. If you have not yet defined a stream forwarder group, this page displays the default group, which is configured to capture all stream protocols.

2. Click Create New Group.

3. Enter a name and description for the group.

4. Click Yes if you want the group to Include Ephemeral Streams (in addition to selected permanent streams).

5. Click On to enable HTTP Event Collector AutoConfig. This provides automated configuration of indexer endpoint URLs for any deployment architecture. Click Next.

1. When HTTP Event Collector Autoconfig is on, splunk_app_stream automatically sends all configuration data to Stream forwarders including endpoint URLs for all indexers (8 max). There is no need to configure data forwarding to indexers from streamfwd.

2. When HTTP Event Collector Autoconfig is off, you must manually enter the endpoint URL of each indexer for each forwarder group definition. The matching forwarder group receives the configuration
from splunk_app_stream then sends the captured data to the specified indexers.

6. Enter a regex rule. For example: sr-centos*.*
   A list of Forwarder IDs that match the regex rule appears in the dialog.
7. Click Next. The Select/De-select Stream in Forwarder Group dialog appears.
8. Select the protocols that you want this group of stream forwarders to capture. Click Finish.
   Your new stream forwarder group appears on the Distributed Forwarder Management page.

Note: You can change the default Forwarder ID (which is the hostname) and specify a new Forwarder ID. This lets you organize your stream forwarders into new logical groupings based on the Forwarder ID.

Enable HTTP event collector to send data to indexers

1. Go to Configuration > Distributed Forwarder Management.
2. Click Install Stream Forwarders.
3. If the Install Stream Forwarders modal shows that HTTP Event Collector streamfwd token configuration is disabled. Click View Configuration.
   The HTTP Event Collector page opens.
4. Click Global Settings. Click Enabled. This enables the streamfwd
5. Make sure HTTP Port Number is set to 8088.
6. Click Save.

For information on how to configure HTTP Event Collector in an indexer cluster environment, see Propagate HTTP Event Cluster configuration to indexer cluster in the Splunk Stream Installation and Configuration Manual.
Distributed Forwarder Management Examples

Manage by hostname

If the existing hosts on which you install stream forwarder (Splunk_TA_stream) use a naming convention, you can create a regex rule that matches the naming convention to define a stream forwarder group.

For example, if your hosts use the naming convention my.server.01, my.server.02, my.server.03, and so on, you could use the regex *.server.* to define your stream forwarder group.

Manage by Forwarder ID

Each stream forwarder (Splunk_TA_stream) instance has its own Forwarder ID. You can change the Forwarder ID from its default value (which is the hostname) and specify a new Forwarder ID. This lets you create your own naming conventions and organize your stream forwarders into new logical groupings based on Forwarder ID.

You can specify the Forwarder ID using Splunk Web or from the command line using configuration files.

Specify a Forwarder ID using Splunk Web

1. In Splunk Web, go to Settings > Data Input > Wire Data.
2. Click on the name of the input for the specific stream forwarder.
3. In the Stream Forward Identifier field, enter a string for the Forwarder ID.
4. Click Save.
   The string that you specify becomes the new Forwarder ID of the stream forwarder.

Note: When you specify the Forwarder ID using Splunk Web, you perform the configuration on the search head that hosts Splunk Stream. The new Forwarder ID is propagated to the corresponding stream forwarder.

Specify a forwarder ID using configuration files

1. Go to $SPLUNK_HOME/etc/apps/Splunk_TA_stream/local/inputs.conf
2. In the [streamfwd://streamfwd] stanza, enter a value for stream_forwarder_id:

   [streamfwd://streamfwd]
splunk_stream_app_location =
stream_forwarder_id = <Forwarder_ID>
disabled = 0

Note: When you specify the Forwarder ID using configuration files, you must perform the configuration on the forwarder(s) that host Splunk_TA_stream.
Stream Estimate

Use Stream Estimate to preview index volume

The Stream Estimate dashboard shows index volume stats for all streams in Estimate mode. Streams in Estimate mode generate data index volume stats, without sending the actual data to your indexers.

You can use the Stream Estimate dashboard to preview the amount of data that you might need to index for any stream. This information can help you calculate your indexer requirements and configure your streams so that you capture only the data you require for analysis.

The Stream Estimate dashboard lets you monitor these data index volume stats:

- Total Events
- Total Incomming Traffic (MB)
- Total Outgoing Traffic (MB)
- Total Traffic (MB)
- Splunk Index Volume (MB)

In the Splunk App for Stream main menu, click **Stream Estimate**.
### Estimated Traffic and Volume Stats by Stream

<table>
<thead>
<tr>
<th>Stream Id</th>
<th>Total Events</th>
<th>Total Incoming Traffic (MB)</th>
<th>Total Outgoing Traffic (MB)</th>
<th>Total Traffic (MB)</th>
<th>Splunk Index Volume (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>snmp</td>
<td>2446</td>
<td>0.165</td>
<td>0.116</td>
<td>0.284</td>
<td>2.353</td>
</tr>
<tr>
<td>tcp</td>
<td>1151</td>
<td>2.448</td>
<td>2.688</td>
<td>5.126</td>
<td>1.151</td>
</tr>
<tr>
<td>sip</td>
<td>218</td>
<td>0.040</td>
<td>0.060</td>
<td>0.100</td>
<td>0.282</td>
</tr>
<tr>
<td>udp</td>
<td>194</td>
<td>2.730</td>
<td>4.080</td>
<td>6.810</td>
<td>0.111</td>
</tr>
<tr>
<td>pop3</td>
<td>135</td>
<td>0.000</td>
<td>0.007</td>
<td>0.007</td>
<td>0.095</td>
</tr>
<tr>
<td>diameter</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>imap</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>irc</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>ldap</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>imap</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>smtp</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Note: Streams in Enabled mode generate data index volume stats based on the actual amount of data sent to your indexers. You can view index volume stats for all enabled streams in the Stream Data Volumes dashboard. For more information, see Admin Dashboards in this manual.

For more information on the Estimate mode, see Configure Streams in this manual.
Dashboards

Stream Informational Dashboards

Splunk App for Stream (splunk_app_stream) provides a set of built-in informational dashboards, which give you a quick overview of activities taking place across your network. Informational dashboards include:

- Analytics Overview
- App Analytics
- Flow Visualization
- HTTP Overview
- HTTP Activity
- Database Activity
- DNS Overview
- DNS Activity
- SSL Activity

Informational dashboards are populated by a set of built-in streams that come with the app. You can clone built-in streams and use them as a starting point to create your own new streams in the Configure Streams UI. For more information, see Configure Streams in this manual.

**Impact of new aggregation methods on dashboards**

As of version 6.6.0, the Database Activity dashboard and built-in Splunk database streams (such as Splunk_Mysql, Splunk_Postgres, and so on) have been updated to use the `max(time_taken)` aggregate function to generate max query time statistics. As a result, the Database Activity dashboard in version 6.6.0 and later is not compatible with data generated by earlier versions of streamfwd.

All other dashboards have been updated for the new aggregation methods introduced in version 6.6.0 and are compatible with data generated by both old and new versions of streamfwd. For more information, see Stream aggregation methods in this manual.
Analytics Overview

Top Applications by Volume (Bytes)

Flow Visualization

Web Traffic Overview

Client Errors Breakdown

Server Errors Breakdown
App Analytics

Top Applications by Volume (Bytes)

<table>
<thead>
<tr>
<th>Application</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssl</td>
<td>1.87 GB</td>
</tr>
<tr>
<td>top</td>
<td>858.88 MB</td>
</tr>
<tr>
<td>http</td>
<td>678.09 MB</td>
</tr>
<tr>
<td>dns</td>
<td>202.57 MB</td>
</tr>
<tr>
<td>unknown</td>
<td>83.25 MB</td>
</tr>
<tr>
<td>amazon_aws</td>
<td>36.06 MB</td>
</tr>
<tr>
<td>salesforce</td>
<td>9.13 MB</td>
</tr>
<tr>
<td>smtp</td>
<td>8.80 MB</td>
</tr>
<tr>
<td>sash</td>
<td>7.47 MB</td>
</tr>
<tr>
<td>udp</td>
<td>6.88 MB</td>
</tr>
<tr>
<td>dns</td>
<td>4.78 MB</td>
</tr>
<tr>
<td>dhcp</td>
<td>4.43 MB</td>
</tr>
<tr>
<td>dhcp6</td>
<td>2.49 MB</td>
</tr>
<tr>
<td>sctp</td>
<td>1.81 MB</td>
</tr>
<tr>
<td>mtpc</td>
<td>1.05 MB</td>
</tr>
<tr>
<td>postgres</td>
<td>922 KB</td>
</tr>
<tr>
<td>soap</td>
<td>621 KB</td>
</tr>
</tbody>
</table>

Rare Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>apprenux</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>bitorrent</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>ftps</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>google</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>ics</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>mozilla</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>opnet</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>teletels</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>windows_update</td>
<td>2</td>
<td>0.10%</td>
</tr>
<tr>
<td>apple_location</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>google</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>google_safebrowsing</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>htt2</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>portmap</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>linux</td>
<td>4</td>
<td>0.21%</td>
</tr>
<tr>
<td>thunder</td>
<td>4</td>
<td>0.21%</td>
</tr>
</tbody>
</table>
Flow Visualization

Flow Visualization

Time | Metric
--- | ---
1 hour window | Flows

Format Timeline

2:50 PM
Wed Feb 8
2017

Zoom Out
Zoom to Selection
Deselect

1,800
1,000

2:50 PM
2:50 PM
2:50 PM
2:50 PM
2:50 PM

3:00 PM
3:00 PM
3:00 PM
3:00 PM
3:00 PM

3:10 PM
3:10 PM
3:10 PM
3:10 PM
3:10 PM

3:20 PM
3:20 PM
3:20 PM
3:20 PM
3:20 PM

3:30 PM
3:30 PM
3:30 PM
3:30 PM
3:30 PM

Source IP

Destination IP

10.141 → 72.33
810 flows
10.143 → 72.33
13 flows
HTTP Overview
HTTP Activity

HTTP Requests by URI

<table>
<thead>
<tr>
<th>URI</th>
<th>Counts</th>
<th>Errors</th>
<th>Avg. Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>/cems/META-INF/services/javax.xml.parsers.SAXParserFactory</td>
<td>224</td>
<td>121</td>
<td>0.445330</td>
</tr>
<tr>
<td>/cems/sgpplts/serviceRouter</td>
<td>112</td>
<td>0</td>
<td>5.144256</td>
</tr>
</tbody>
</table>

HTTP Activity by Domain

<table>
<thead>
<tr>
<th>Domain</th>
<th>Count</th>
<th>Bytes In</th>
<th>Bytes Out</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>293.577.104</td>
<td>336</td>
<td>127467</td>
<td>1007997</td>
<td>1.539667</td>
</tr>
</tbody>
</table>

EndPoint (Client) Activity

<table>
<thead>
<tr>
<th>Client</th>
<th>Requests</th>
<th>Bytes In</th>
<th>Bytes Out</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>293.577.204</td>
<td>439</td>
<td>127467</td>
<td>1007997</td>
<td>1.539667</td>
</tr>
</tbody>
</table>

Database Activity

Hourly Average Response Time (seconds)

Hourly Maximum Response Time (seconds)

Top 10 Longest Running Queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT DISTINCT BAT_ID ORDER BY LIN_ID, NO</td>
<td>0.5</td>
</tr>
<tr>
<td>SELECT 8 EVENT, NAME, ... ORDER BY BID, NO</td>
<td>0.4</td>
</tr>
<tr>
<td>SELECT COUNT(bid), ... ORDER BYid, NO</td>
<td>0.3</td>
</tr>
<tr>
<td>SELECT DISTINCT 8 ORDER BY NAME, NO</td>
<td>0.2</td>
</tr>
<tr>
<td>SELECT EVENT, NAME, ... ORDER BYPART, NO</td>
<td>0.1</td>
</tr>
<tr>
<td>SELECT BAT, NAME, ... ORDER BY PART, NO</td>
<td>0.1</td>
</tr>
<tr>
<td>SELECT LISTS, BID, ... ORDER BY OPERATORS IS NOT NULL</td>
<td>0.0</td>
</tr>
<tr>
<td>SELECT AUDITED_BY ... ORDER BY 1</td>
<td>0.0</td>
</tr>
<tr>
<td>SELECT ADD_CHAIN ... ORDER BY 2</td>
<td>0.0</td>
</tr>
</tbody>
</table>

36
DNS Overview

DNS Overview

Servers
Last 60 minutes
all

DNS Queries

DNS Response Time

Error Counts

_time

_time

_time
DNS Activity

Queries

<table>
<thead>
<tr>
<th>Query</th>
<th>Counts</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>greyturtle02.sv.splunk.com</td>
<td>2578</td>
<td>0</td>
</tr>
<tr>
<td>gnsx-1.centos-1.sv.splunk.com</td>
<td>5478</td>
<td>0</td>
</tr>
<tr>
<td>_info-vkdnspdoman</td>
<td>2537</td>
<td>2027</td>
</tr>
<tr>
<td>splinux04.sv.splunk.com</td>
<td>2539</td>
<td>0</td>
</tr>
<tr>
<td>123.3.140.10.in-addr.arpa</td>
<td>2172</td>
<td>0</td>
</tr>
<tr>
<td>123.21.106.10.in-addr.arpa</td>
<td>2120</td>
<td>0</td>
</tr>
<tr>
<td>gwilton-centos7x64001.sv.splunk.com</td>
<td>1984</td>
<td>0</td>
</tr>
<tr>
<td>ea-systest-08.sv.splunk.com</td>
<td>1846</td>
<td>0</td>
</tr>
<tr>
<td>gnsx-centos-88.sv.splunk.com</td>
<td>1776</td>
<td>0</td>
</tr>
<tr>
<td>soln-perf07.sv.splunk.com</td>
<td>1518</td>
<td>0</td>
</tr>
</tbody>
</table>

Query Types

<table>
<thead>
<tr>
<th>Query Type</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>159,000</td>
</tr>
<tr>
<td>PTR</td>
<td>169,000</td>
</tr>
<tr>
<td>TXT</td>
<td>160,000</td>
</tr>
<tr>
<td>NVE</td>
<td>150,000</td>
</tr>
<tr>
<td>MX</td>
<td>140,000</td>
</tr>
<tr>
<td>AAAA</td>
<td>130,000</td>
</tr>
<tr>
<td>NS</td>
<td>120,000</td>
</tr>
<tr>
<td>SRV</td>
<td>110,000</td>
</tr>
</tbody>
</table>

Status

<table>
<thead>
<tr>
<th>Status</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>NXDomain</td>
<td>75,000</td>
</tr>
<tr>
<td>NotError</td>
<td>50,000</td>
</tr>
<tr>
<td>ServFail</td>
<td>25,000</td>
</tr>
</tbody>
</table>
Stream Admin dashboards

Splunk App for Stream (splunk_app_stream) provides a set of pre-built Admin dashboards, including

- Stream Data Volumes
- Network Metrics
- Stream Forwarder Status
- Stream Forwarder Metrics & Logs

Use Admin dashboards to identify spikes and trends in network activity that might indicate a network issue and to analyze customer behavior. Click in any dashboard graph to drill down to Splunk search results, and perform further analysis on network, streamfwd process, and log data.

Note: Data used to populate Stream Admin dashboards is collected by Stream Forwarders and stored in the _internal index. Hence, it is critical to have the _internal index forwarded from all Stream-enabled splunk instances (i.e. running on Universal and Heavy Forwarders) to the indexers.

Stream Data Volumes

The Stream Data Volumes dashboard shows index volume stats for all streams in the Enabled mode. The dashboard lets you monitor these data index volume stats:

- Total Events
- Total Incoming Traffic (MB)
- Total Outgoing Traffic (MB)
- Total Traffic (MB)
- Splunk Index Volume (MB)

In the Splunk App for Stream main menu, select Admin Dashboards > Stream Data Volumes.
### Stream Data Volumes

#### Splunk Index Volume by Stream (MB)

<table>
<thead>
<tr>
<th>Stream</th>
<th>Total Events</th>
<th>Total Incoming Traffic (MB)</th>
<th>Total Outgoing Traffic (MB)</th>
<th>Total Traffic (MB)</th>
<th>Splunk Index Volume (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql</td>
<td>11634</td>
<td>0.428</td>
<td>0.002</td>
<td>1.030</td>
<td>7.303</td>
</tr>
<tr>
<td>Splunk_MySql</td>
<td>11634</td>
<td>0.428</td>
<td>0.002</td>
<td>1.030</td>
<td>4.200</td>
</tr>
<tr>
<td>http</td>
<td>439</td>
<td>0.114</td>
<td>0.955</td>
<td>1.069</td>
<td>0.406</td>
</tr>
<tr>
<td>sr_http</td>
<td>224</td>
<td>0.073</td>
<td>0.684</td>
<td>0.757</td>
<td>0.221</td>
</tr>
<tr>
<td>sr_MySql_2</td>
<td>439</td>
<td>0.114</td>
<td>0.955</td>
<td>1.069</td>
<td>0.117</td>
</tr>
<tr>
<td>Splunk_HTTPURI</td>
<td>439</td>
<td>0.114</td>
<td>0.955</td>
<td>1.069</td>
<td>0.096</td>
</tr>
<tr>
<td>Splunk_DNSClientErrors</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Splunk_DNSClientQueryTypes</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Splunk_DNSExpiration</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Splunk_DNSRequestResponse</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Splunk_DNSServerErrors</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Splunk_DNSServerQuery</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Note: To view estimates of data index volume for streams in the Estimate mode, use the Stream Estimate dashboard. For more information, see Stream Estimate in this manual.

Network Metrics

The Network Metrics dashboard lets you monitor these network events:

- Bandwidth (Mbps)
- Active Network Flows
- Total Packets
- Dropped Packets

In the Splunk App for Stream main menu, select Admin Dashboards > Network Metrics.
Stream Forwarder Status

The Stream Forwarder Status dashboard displays a list your deployed stream forwarders, along with attributes, status, and configuration details for each stream forwarder. The Stream Forwarder Status dashboard is populated by a special stream of `sourcetype=stream:stats` that is not user configurable and does not appear in the Configure Streams UI.

In the Splunk App for Stream main menu, select Admin Dashboards > Stream Forwarder Status.
**Stream Forwarder Metrics & Logs**

Click on any stream forwarder in the Stream Forwarder ID list to open the Stream Forwarder Metrics & Logs dashboard for that particular stream forwarder. This dashboard provides additional detailed information on the status and behavior of individual stream forwarders, including log entries from `streamfwd.log` and the following `streamfwd` binary metrics:

- Total Events
- Event Queue Size
- Packet Queue Size
- SSL Session Keys
- TCP Reassembly Packet Count
- TCP Reassembly Payload Size
- Event Attributes