

Hortonworks Cybersecurity Package

User Guide

(April 13, 2017)

Hortonworks Cybersecurity Package: User Guide

Copyright © 2012-2017 Hortonworks, Inc. Some rights reserved.

Hortonworks Cybersecurity Package (HCP) is a modern data application based on Apache Metron, powered by Apache Hadoop, Apache Storm, and related technologies.

HCP provides a framework and tools to enable greater efficiency in Security Operation Centers (SOCs) along with better and faster threat detection in real-time at massive scale. It provides ingestion, parsing and normalization of fully enriched, contextualized data, threat intelligence feeds, triage and machine learning based detection. It also provides end user near real-time dashboards.

Based on a strong foundation in the Hortonworks Data Platform (HDP) and Hortonworks DataFlow (HDF) stacks, HCP provides an integrated advanced platform for security analytics.

Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [Contact Us](#) directly to discuss your specific needs.



Except where otherwise noted, this document is licensed under
Creative Commons Attribution ShareAlike 4.0 License.
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Table of Contents

1. Overview	1
1.1. Functionality of Metron Dashboard	1
1.2. Metron Default Dashboard	3
1.2.1. Events	3
1.2.2. Enrichment	4
1.2.3. YAF	4
1.2.4. Snort	5
1.2.5. Web Request Header	5
1.2.6. DNS	6
2. Customizing Your Metron Dashboard	7
2.1. Launching the Metron Dashboard	7
2.2. Changing the Metron Dashboard Background Color	8
2.3. Adding a New Data Source	8
2.3.1. Prerequisites	8
2.3.2. Configuring a New Data Source Index	8
2.3.3. Reviewing the New Data Source Data	9
2.4. Querying, Filtering, and Visualizing Data	10
2.4.1. Querying Your Data	10
2.4.2. Filtering Your Query Results	12
2.4.3. Visualizing Your Data	14
2.5. Customizing Your Dashboard	25
2.6. Using pcap to View Your Raw Data	28
2.6.1. Capturing pcap Data	28
2.6.2. Processing pcap Data	29
2.6.3. Viewing pcap Data	30
2.6.4. Filtering pcap Data	30
2.6.5. Porting pcap Data to Another Application	33
3. Sharing the Metron Dashboard	35
3.1. Exporting Search Information	35
3.2. Importing Search Information	35

List of Figures

1.1. Dashboard-Snort Panel	1
1.2. Events	3
1.3. Enrichment	4
1.4. YAF	4
1.5. Dashboard-Snort Panel	5
1.6. Dashboard-Bro Panel	5
1.7. Dashboard-DNS Panel	6
2.1. Ambari Task List	7
2.2. Configure an Index Pattern	9
2.3. Discover Tab with Squid Elements	10
2.4. Time Filter	12
2.5. Query Search Text Entry Box	14

1. Overview

This guide is intended for use by Security Operations Center (SOC) analysts and investigators. This guide describes the Metron dashboard which is a component of Hortonworks Cybersecurity Package (HCP) powered by Apache Metron. This guide provides instructions on setting up and modifying the dashboard. It also provides instructions on querying and viewing pcap information.

This chapter contains the following information:

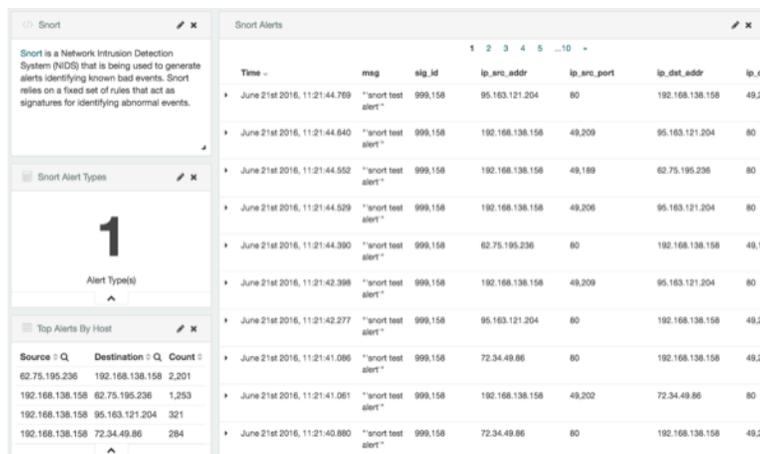
- [Functionality of Metron Dashboard \[1\]](#)
- [Metron Default Dashboard \[3\]](#)

1.1. Functionality of Metron Dashboard

The Metron dashboard is a Kibana-based dashboard designed to identify, investigate, and analyze cybersecurity data. HCP supports Kibana 4.x. Kibana is an open source analytics and visualization platform.

The Metron dashboard displays all of the data on a single dashboard enabling you to filter through the irrelevant data and display just the information, alerts, and context for which you are looking. The Metron dashboard has several advantages over conventional SIEM tools, including flexibility, and the single pane of glass approach that displays all of the data on the same screen, requiring no jumping from console to console to gather the information.

Figure 1.1. Dashboard-Snort Panel



HCP supports two types of messages: metadata and alerts. By convention there should be one panel per metadata telemetry and one panel that is a "catch all" panel for alerts. The Snort panels are a good example of these two panel types. However, the Snort alerts panel only lists alerts from Snort because the default Metron dashboard contains only one data source that produces alerts.

When HCP parses the telemetry data on ingest, it extracts and normalizes different parts of the message into a standard Metron JSON. Standardizing and normalizing field names and format allows HCP to search different telemetry messages with a single query.

The first telemetry type that HCP supports is metadata messages. Metadata messages are parsed enriched messages in the JSON format.

The second telemetry type that HCP supports is alerts telemetries. Alerts telemetries come from IDS sensors like Snort or mixed telemetries like application logs that contain some metadata and some alert messages. While it is possible to set up a new panel for each alert telemetry, it is more desirable to set up a single panel that contains all of the alerts. This guarantees that the query will pull in alerts from multiple telemetries (even mixed mode telemetries that have some metadata and some alerts associated with them). You can then set up a detailed table containing only the alerts. To set telemetry as alert you need to set `is_alert = true`. This is already set up for HCP under the "Alerts" table.

The fields displayed for each alerts table can be customized. Ideally you want the fields of most importance (as well as the standard fields that telemetries are correlated on) to be displayed.

The following table contains a description of each of the Kibana components in the Metron dashboard.

Area Chart Panel	You can use the area chart panel for stacked timelines for which you want to see the total.
Data Table Panel	Use the data table panel to provide a detail breakdown, in tabular format, of the results of a composed aggregation. You can generate a data table from many other charts by clicking the grey bar at the bottom of the chart.
Detailed Message Panel	A detailed message panel displays the raw data from your search query.
Document Table	When you submit a search query, the 500 most recent documents that match the query are listed in the Documents table which is displayed in the center of the Discover window.
Field List	A list of all of the fields associated with a selected index pattern. This list is displayed on the left side of the Discover window.
Line Chart Panel	Use the line chart when you want to display high density time series. This chart is useful for comparing one series with another.
Mark Down Widget Panel	You can use the mark down widget panel to provide explanations or instructions for the dashboard.
Metric Panel	You can use a metric panel to display a single large number such as the number of hits or the average of a numeric field.

Pie Chart Panel

A **pie chart** is a circular statistical graphic that is ideal for displaying the parts of some whole.

Tile Map Panel

The **tile map panel** type displays a map populated with your search results. This panel type requires an Elasticsearch `geo_point` field that is mapped as `type:geo_point` with latitude and longitude coordinates.

Vertical Bar Chart Panel

You can use the **vertical bar chart panel** to display histograms. Histogram panels represent ingest rates for each individual telemetry. By convention, you should set up one for each type.

1.2. Metron Default Dashboard

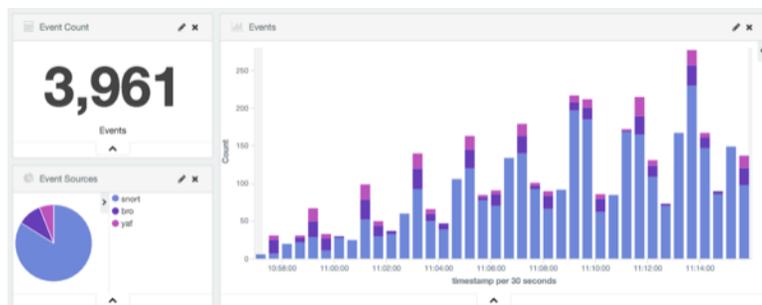
The default telemetry data sources installed with HCP help highlight the useful components available in Kibana 4. The default Metron dashboard serves as a starting point for you to build your own customized dashboards. During installation, HCP sets up several telemetry data sources bundled with the platform and creates panels to display the associated data.

- [Events \[3\]](#)
- [Enrichment \[4\]](#)
- [Overview \[1\]](#)
- [Snort \[5\]](#)
- [Web Request Header \[5\]](#)
- [DNS \[6\]](#)

1.2.1. Events

The first panel in the dashboard highlights the variety of events being consumed by HCP. It shows the total number of events received, the variety of those events, and a histogram showing when the events were received.

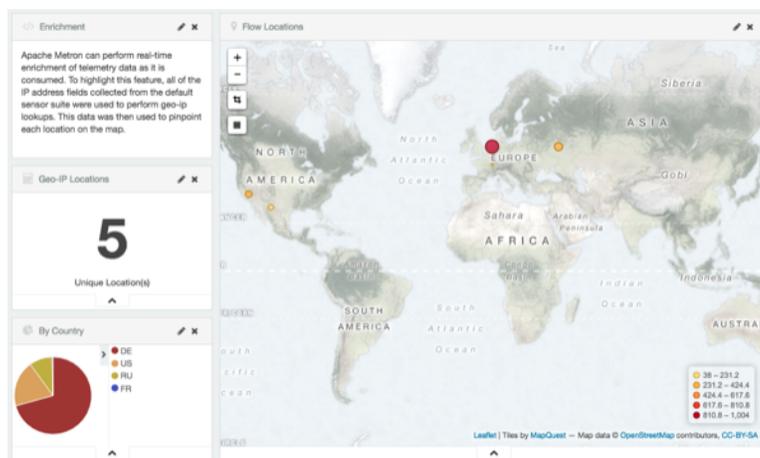
Figure 1.2. Events



1.2.2. Enrichment

The next set of dashboard panels shows how HCP can be used to perform real-time enrichment of telemetry data. All of the IPv4 data received by HCP was cross-referenced against a geo-ip database. These locations were then used to build this set of dashboard components.

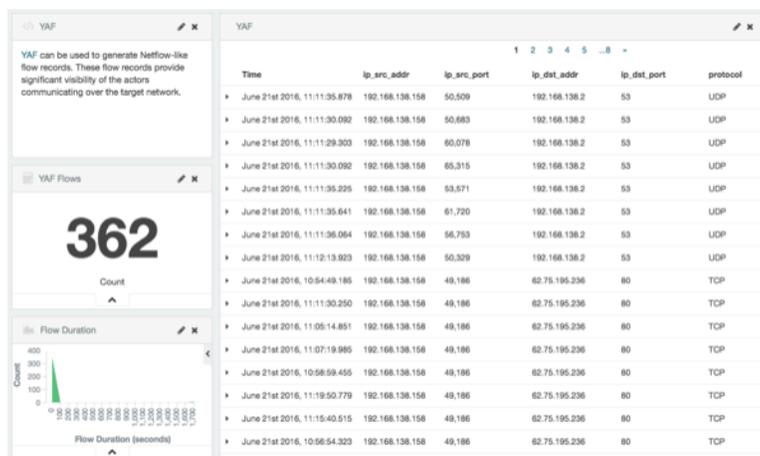
Figure 1.3. Enrichment



1.2.3. YAF

As part of the default sensor suite, **YAF** is used to generate flow records. These flow records provide significant visibility into which actors are communicating over the target network. A table panel displays the raw details of each flow record. A histogram of the duration of each flow illustrates that while most flows are relatively short-lived there are a few that are much longer in this example. Creating an index template that defined this field as numeric was required to generate the histogram.

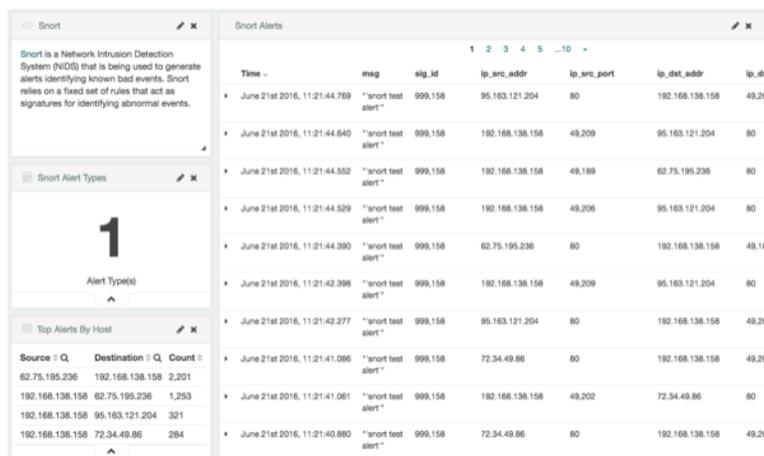
Figure 1.4. YAF



1.2.4. Snort

Snort is a Network Intrusion Detection System (NIDS) that is being used to generate alerts identifying known bad events. Snort relies on a fixed set of rules that act as signatures for identifying abnormal events. Along with displaying the relevant details of each alert, the panel shows that there is only a single unique alert type; a test rule that creates a Snort alert on every network packet. Another table was created to show source/destination pairs that generated the most Snort alerts.

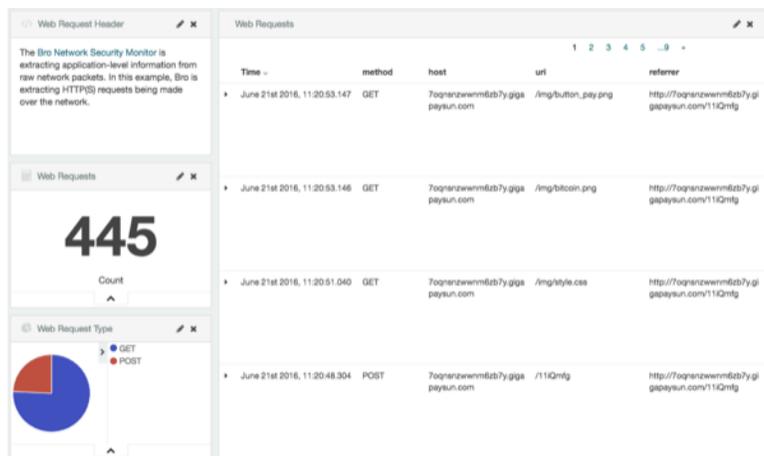
Figure 1.5. Dashboard-Snort Panel



1.2.5. Web Request Header

The Bro Network Security Monitor extracts application-level information from raw network packets. In this example, Bro is extracting HTTP and HTTPS requests being made over the network. The panels highlight the breakdown by request type, the total number of web requests, and raw details from each web request.

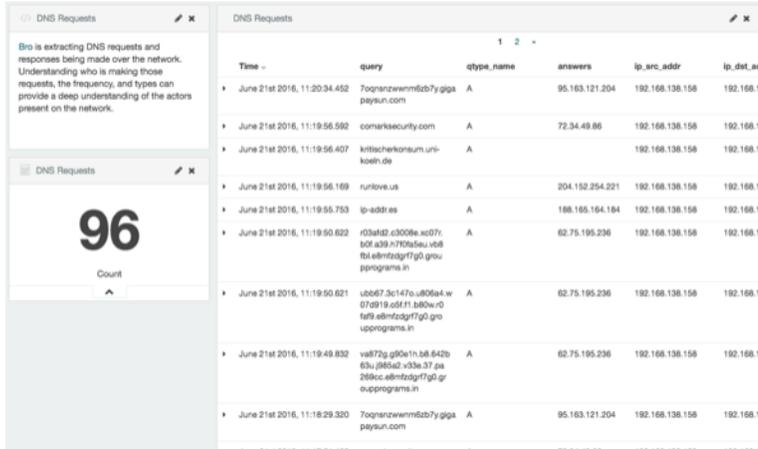
Figure 1.6. Dashboard-Bro Panel



1.2.6. DNS

Bro extracts DNS requests and responses being made over the network. Understanding who is making those requests, the frequency, and types can provide a deep understanding of the actors present on the network.

Figure 1.7. Dashboard-DNS Panel



2. Customizing Your Metron Dashboard

This chapter provides the steps to customize the Metron dashboard to display information, alerts, and the context you need to identify and analyze cybersecurity issues.

This chapter provides the following information:

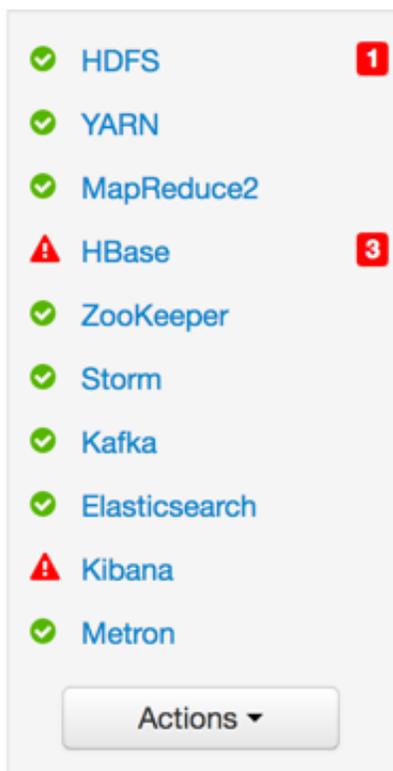
- [Launching the Metron Dashboard \[7\]](#)
- [Changing the Metron Dashboard Background Color \[8\]](#)
- [Adding a New Data Source \[8\]](#)
- [Querying, Filtering, and Visualizing Data \[10\]](#)
- [Customizing Your Dashboard \[25\]](#)
- [Using pcap to View Your Raw Data \[28\]](#)

2.1. Launching the Metron Dashboard

You can launch the Metron Dashboard two ways:

- From Ambari, click Kibana in the list of quick tasks.

Figure 2.1. Ambari Task List



- Enter the following text in a browser:

```
$KIBANA_HOST:9995
```

2.2. Changing the Metron Dashboard Background Color

You can choose to view the Metron dashboard with either a light or dark background. The dark background is sometimes preferred in a dimly lit security operations center. To switch the background color, perform the following steps:

1. Click the Gear icon () in the top right of the Metron dashboard.

You should see a check box next to **Use dark theme** near the top of the dashboard.

2. Select the check box to use the dark theme for the dashboard.

To return to the light theme, clear the check box.

2.3. Adding a New Data Source

After a new data telemetry source has been added to HCP, you will need to also add it to the Metron dashboard before you can create queries and filters for it and add telemetry panels displaying its data. Complete the steps in the following sections to add a new telemetry source to the Metron dashboard:

- [Prerequisites \[8\]](#)
- [Configuring a New Data Source Index \[8\]](#)
- [Reviewing the New Data Source Data \[9\]](#)

2.3.1. Prerequisites

Before you can add a new data telemetry source to the Metron dashboard, you must complete the following steps:

- The data telemetry source must be added to HCP.

For information on how to add a new data telemetry source, see [Adding a New Telemetry Data Source](#).

- An index template must be created for the data telemetry source.

For information on how to create an index template, see [Creating an Index Template](#).

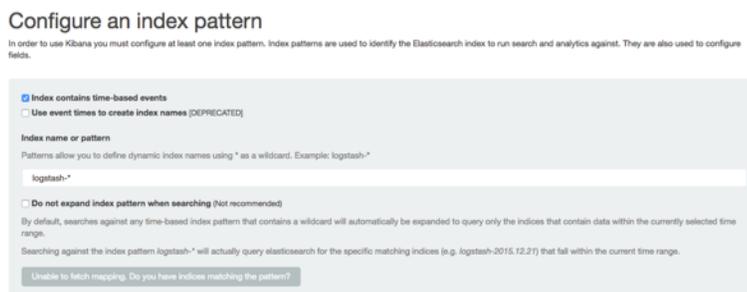
2.3.2. Configuring a New Data Source Index

Now that you have an index for the new data source with all of the right data types, you need to tell the Metron dashboard about this index.

1. Click the **Settings** tab on the Metron dashboard.
2. Make sure you have the **Indices** tab selected, then click **+Add New**.

Kibana displays the **Configure an index pattern** window. Use the index pattern window to identify your telemetry source.

Figure 2.2. Configure an Index Pattern



3. In the **Index name or pattern** field, enter the name of the index pattern of your data telemetry source.

In most cases the name of the index pattern will match the sensor name. For example, the 'bro' sensor has an index pattern of 'bro-*'.

4. If your data telemetry source **does not** contain time-based events, clear the **Index contains time-based events** check box.

If your data telemetry source does contain time-based events, leave the check box as is. Most of your data telemetry sources will contain time-based events.

5. Click **Create** to add the index pattern for your new data telemetry source.

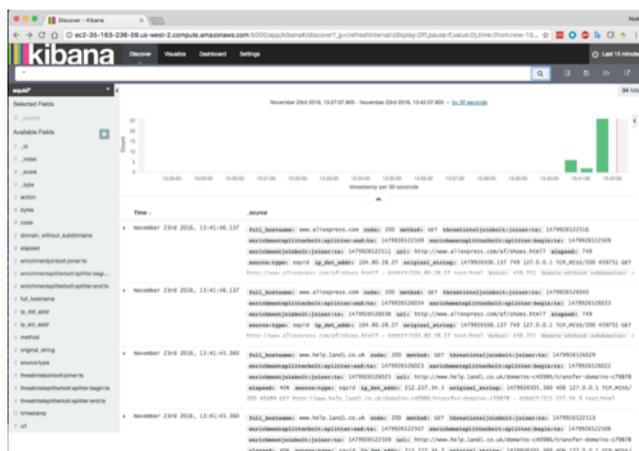
If you would like this new index pattern to be the default, click the Green Star icon (★).

2.3.3. Reviewing the New Data Source Data

Now that the Metron dashboard is aware of the new data source index, you can look at the data.

1. Click on the **Discover** tab and then choose the newly created data source index pattern.
2. Click any of the fields in the left column to see a representation of the variety of data for that specific field.
3. Click the Right Facing Arrow icon next to a specific record in the center of the window (the **Document** table) to expand the record and display the available data.

Figure 2.3. Discover Tab with Squid Elements



2.4. Querying, Filtering, and Visualizing Data

You can interactively explore your data source data using the Metron dashboard. When HCP parses a telemetry, it extracts and normalizes different parts of the message into a standard Metron JSON object. Standardizing and normalizing field names and formats allows HCP to search different telemetry messages with a single query. You have access to every document in every index that matches your selected index patterns. The Metron dashboard enables you to submit search queries on the data source data, filter the search results, and view the results in a number of visualizations.

In HCP, if telemetry indexing is enabled, a rotating index for every telemetry is created. By convention this index will have a name [telemetry_name]_[timestamp]. Telemetry documents indexed into this index will by convention be called [telemetry_name]_doc. Queries reference the document type of the indexed telemetries.

This section details how to explore and analyze your data using the following methods:

- [Querying Your Data \[10\]](#)
- [Filtering Your Query Results \[12\]](#)
- [Visualizing Your Data \[14\]](#)

2.4.1. Querying Your Data

You can search and refine the data you receive from your data source by creating a query from the **Discover** page. You should create and save a query for each data source not provided by HCP.

When you submit a search query, the histogram, documents table, and fields list are updated to reflect the search results. The total number of hits is shown in the upper right corner of the histogram. The documents table shows the first five hundred hits.

HCP includes queries for the following telemetries:

- YAF

- Bro
- Alerts (populated by Snort)

You can also add custom queries for new telemetry types.

To create a custom query, complete the following steps:

1. Click the **Discover** tab to display the **Discover** window.
2. In the search field at the top of the window, enter a query looking for a new type of telemetry document.

For example, adding Snort would look something like this:



You have several options when you create a query:

- To perform a free text search, enter a text string.
For example, if you're searching web server logs, you could enter safari to search all fields for the term safari.
- To search for a value in a specific field, you prefix the value with the name of the field.
For example, you could enter status:200 to limit the results to entries that contain the value 200 in the status field.
- To search for a range of values, you can use the bracketed range syntax, [START_VALUE TO END_VALUE].
For example, to find entries that have 4xx status codes, you could enter status:[400 TO 499].
- To search for a range of IPv4 values, you can use a similar bracketed range syntax, [IP_START_VALUE TO IP_END_VALUE].
For example, to find all class A private network addresses, you could enter ip_address:[10.0.0.0 TO 10.255.255.255].
- To search for a range of values, you can use the bracketed range syntax, [START_VALUE TO END_VALUE].
For example, to find entries that have 4xx status codes, you could enter status:[400 TO 499].

3. Click Enter or click the Search () button to submit your search query.

4. Click the Save Search button () in the Discover toolbar to save the search.

Saving a search saves both the query string and the currently selected index pattern.

5. Enter a name for the search and click **Save**.

2.4.2. Filtering Your Query Results

You can use the Metron dashboard to filter your query results to further refine the information. The Metron dashboard provides two types of filters:

Time Filter	Restricts the search results to a specific time period.
Filter by Field	Filters to display only those documents that contain a particular value in a field. You can filter either from the Fields list or the Documents table.

2.4.2.1. Setting a Time Filter

You can set a time filter if your index contains time-based events and a time-field is configured for the selected index pattern.

The time filter is in the upper right corner of the **Discover** window. By default, the time filter is set to 15 minutes. You can use the Time Picker to change the time filter or select a specific time interval or time range in the histogram at the top of the page.

Figure 2.4. Time Filter



To set up a time filter, complete the following steps:

1. Click the **Time Filter** displayed in the upper right corner of the Dashboard menu bar.



2. To set a **Quick** filter, click one of the shortcut links.
3. To specify a relative time filter, click **Relative** and enter the start time.

You can specify the relative start time as any number of seconds, minutes, hours, days, months, or years ago.

4. To specify an absolute time filter, click **Absolute** and enter the start date in the **From** field and the end date in the **To** field.

Click  (caret icon) at the bottom of the Time Picker to dismiss it.

You can also set a time filter from a histogram. To perform this task, complete the following steps:

1. Click the bar on the histogram that represents the time interval you want to zoom in on.
2. Click and drag to view a specific timespan.

You must start the selection with the cursor over the background of the chart. The cursor changes to a plus sign when you hover over a valid start point.

The histogram lists the time range you're currently exploring, as well as the intervals that range is currently using.

3. To change the intervals, click the link and select an interval from the drop-down list.

The default behavior automatically sets an interval based on the time range.

You can use the browser back button to undo your changes.

2.4.2.2. Filtering By Field

You can filter search results to display only those documents that contain a particular value in a field. You can also create negative filters that exclude documents that contain the specified field value.

You can add filters from the Fields list or from the Documents table. When you add a filter, it is displayed in the filter bar below the search query. From the filter bar, you can enable or disable a filter, invert the filter (change it from a positive filter to a negative filter and vice-versa), toggle the filter on or off, or remove it entirely.

To filter the results of a query, complete the following steps:

1. Click the **Discover** tab to display the **Discover** window.
2. Choose the index pattern for which you want to create a query.

The Metron Dashboard displays the fields associated with the index pattern in the **Field list** and also in the **Document table**. The following steps assume you are using the **Fields list**.

3. In the **Fields list**, click the name of the field on which you want to filter.

The Metron dashboard displays the top five values for that field.

4. Add a filter by clicking one of the Filter buttons (positive or negative magnifying glass icon) next to the value you want to filter in or out.

To filter out documents that don't contain the value in the field, click the Positive

Magnifying Glass icon ().

To filter out documents that do contain the value in the field, click the Negative

Magnifying Glass icon (.

5. Click the Save Search () button in the Discover toolbar to save the search.
Saving a search saves both the query string and the currently selected index pattern.
6. Enter a name for the search and click **Save**.

2.4.2.3. Managing Your Filters

When you create a filter anywhere in the Metron dashboard, the filter conditions display in a green oval under the search text entry box.

Figure 2.5. Query Search Text Entry Box



You can perform several actions on the filter. Hover over the green oval to display the actions you can perform on the filter. If you click on the **Actions** menu in the filter bar, it will open a menu that allows you manipulate **all** of the filters.

- | | |
|---|---|
| <p>Enable</p>  | <p>Toggles between disabling and enabling the filter without removing it.</p> |
| <p>Pin</p>  | <p>Pinned filters persist across Kibana tabs.</p> |
| <p>Toggle</p>  | <p>Toggles between inclusion and exclusion. If a filter is set to inclusion, only elements that match the filter are displayed. If a filter is set to exclusion, the dashboard will display elements that don't match the filter. Exclusion filters display in red.</p> |
| <p>Remove</p>  | <p>Removes the filter entirely.</p> |
| <p>Custom</p>  | <p>Enables you to customize the JSON representation of the filter and specify an alias to use for the filter name. For example, you can use JSON filter representation to implement predicate logic, with <code>should</code> for OR, <code>must</code> for AND, and <code>must_not</code> for NOT.</p> |
| <p>Global</p> | <p>Applies any of the filter actions to all the filters currently in place.</p> |

2.4.3. Visualizing Your Data

You can use visualizations to display and compare your query results in a number of formats. Telemetries provided by HCP are set up on the dashboard by default. However, you can add visualizations for the data sources that you add. HCP supports the following types of panels:

- [Adding a Detail Panel \[15\]](#)
- [Adding a Mark Down Widget Panel \[16\]](#)
- [Adding a Metric Panel \[16\]](#)

- [Adding a Pie Chart Panel \[16\]](#)
- [Adding a Tile Map Panel \[18\]](#)
- [Adding a Vertical Bar Chart Panel \[23\]](#)
- [Setting up Details Message Panels for Telemetry Data \[24\]](#)

By convention there should be one panel per metadata telemetry and one panel that is a “catch all” panel for alerts.

2.4.3.1. Adding a Detail Panel

The detail panel contains the raw data from a search. This is the only panel that contains raw data.

To add a detail panel, complete the following steps:

1. Click the **Discover** tab.
2. Select the `$DATASOURCE*` index.
3. Search for docs in this index with type of `$DATASOURCE_doc`.
 - a. Type the following in the search field: `_type: $DATASOURCE_doc`.
 - b. Click the **Search** icon.
4. Select the subset of the fields that you want to display in the detail panel.

In the left panel under **Available Fields**, click each of the following fields, then click **Add**.

- `full_hostname`
 - `ip_src_addr`
 - `ip_dst_addr`
 - `original_string`
 - `method`
 - `type`
5. Click the Save Search () button in the Discover toolbar to save the search.
 6. Enter a name for the search (`$DataSource Event Details`) and click **Save**.
 7. Select the **Dashboard** tab then click the Plus (+) button.
 8. Click the **Searches** tab and select `$DataSource Event Details`.

The visualization will be added to the bottom of the dashboard.
 9. Click the Save Dashboard () button in the Discover toolbar to save the dashboard.

2.4.3.2. Adding a Mark Down Widget Panel

The Markdown widget is a text entry field that accepts GitHub-flavored Markdown text. Kibana renders the text you enter in this field and displays the results on the dashboard.

For instructions on how to write and format on GitHub, see <https://help.github.com/categories/writing-on-github/>. Click **Apply** to display the rendered text in the Preview pane or **Discard** to revert to a previous version.

2.4.3.3. Adding a Metric Panel

Metric is the most simple panel that you can add to the Metron dashboard. This panel displays a single number for an aggregation that you select.

To add an event count panel for a new data source, complete the following steps:

1. Select the **Visualize** tab and then click **Metric**.
2. Click **From a new search** and then choose a data source index.
The dashboard displays the Visualization Editor.
3. In the search box, enter `is_alert = true` and then execute the search.

4. Click the Save Visualization () button to save the visualization.
5. Name the visualization **Threat Intel Hits** and click Save.
6. Select the **Dashboard** tab and then click the Plus () button.
7. Select the **Visualization** tab and select "\$DATASOURCE Event Count".
8. The visualization will be added to the bottom of the dashboard .
9. Click the Save Dashboard () button in the dashboard toolbar to save the dashboard.

2.4.3.4. Adding a Pie Chart Panel

You can use the pie chart panel to illustrate how different values or fields represent a portion of the whole. For example, a pie chart can illustrate the percentage HTTP and HTTPS requests represented out of all of the requests.

The slice size of a pie chart is determined by the metrics aggregation. The following aggregations are available for this axis:

Count	Returns a raw count of the elements in the selected index pattern.
Sum	Returns the total sum of a numeric field. Select a field from the drop-down list.
Unique Count	Returns the number of unique values in a field. Select a field from the drop-down list.

To create a pie chart panel, complete the following steps:

1. Enter a string in the **Custom Label** field to change the display label.
2. Specify if you are splitting slices within a single chart or splitting into multiple charts.
A multiple chart split must run before any other aggregations.
3. Specify if the splits will be displayed in a row or a column by clicking the **Rows | Columns** selector.
4. Choose the bucket aggregations for your pie chart.

You can specify any of the following bucket aggregations for your pie chart:

Data Histogram	A data histogram is built from a numeric field and organized by date. You can specify a time frame for the intervals in seconds, minutes, hours, days, weeks, months, or years. You can also specify a custom interval frame by selecting Custom as the interval and specifying a number and a time unit in the text field. Custom interval time units are s for seconds, m for minutes, h for hours, d for days, w for weeks, and y for years. Different units support different levels of precision, down to one second.
Histogram	A standard histogram is built from a numeric field. Specify an integer interval for this field. Select the Show empty buckets check box to include empty intervals in the histogram.
Range	With a range aggregation, you can specify ranges of values for a numeric field. Click Add Range to add a set of range endpoints. Click the red (x) symbol to remove a range.
Date Range	A date range aggregation reports values that are within a range of dates that you specify. You can specify the ranges for the dates using date math expressions. Click Add Range to add a set of range endpoints. Click the red (/) symbol to remove a range.
IPv4 Range	The IPv4 range aggregation enables you to specify ranges of IPv4 addresses. Click Add Range to add a set of range endpoints. Click the red (/) symbol to remove a range.
Terms	A terms aggregation enables you to specify the top or bottom n elements of a given field to display, ordered by count or a custom metric.
Filters	You can specify a set of filters for the data. You can specify a filter as a query string or in JSON format, just as in the Discover search bar. Click Add Filter to add another filter. Click the  Label () button to open the label field, where you can type in a name to display on the visualization.
Significant Terms	You can specify a set of filters for the data. You can specify a filter as a query string or in JSON format, just as in the Discover

search bar. Click **Add Filter** to add another filter. Click the

Label () button to open the label field, where you can type in a name to display on the visualization.

After defining an initial bucket aggregation, you can define sub-aggregations to refine the visualization.

5. To define a sub-aggregation, click + **Add Sub Aggregation**, then choose **Split Slices** to select a sub-aggregation from the list of types.

You can customize the colors of your visualization by clicking the color dot next to each label to display the color picker.

6. When multiple aggregations are defined on a chart's axis, you can use the up or down arrows to the right of the aggregation's type to change the aggregation's priority.
7. You can click the **Advanced** link to display more customization options for your metrics or bucket aggregation:

Exclude Pattern	Specifies a pattern in this field to exclude from the results.
Exclude Pattern Flags	A standard set of Java flags for the exclusion pattern.
Include Pattern	Specifies a pattern in this field to include in the results.
Include Pattern Flags	A standard set of Java flags for the inclusion pattern.
JSON Input	A text field where you can add specific JSON-formatted properties to merge with the aggregation definition, as in the following example:

```
{ "script" : "doc['grade'].value * 1.2" }
```

The availability of these options varies depending on the aggregation you choose.

8. Select the **Options** tab to change the following aspects of the table:

Donut	Displays the chart as a sliced ring instead of a sliced pie.
Show Tooltip	Check this box to enable the display of tooltips.
Show Legend	Check this box to enable the display of a legend next to the chart.

9. After changing options, click the green **Apply** changes button to update your visualization, or the grey **Discard** changes button to keep your visualization in its current state.

2.4.3.5. Adding a Tile Map Panel

A tile map displays a geographic area overlaid with circles keyed to the data determined by the buckets you specify.



Note

By default, Kibana uses the Elastic Tile Service to display map tiles. To use other tile service providers, refer to the Kibana 4 documentation.

You can select any of the following aggregations as the metrics aggregation:

Count	Returns a raw count of the elements in the selected index pattern. The count aggregation is the default metrics aggregation for a tile map.
Average	Returns the average of a numeric field. Select a field from the drop-down list.
Sum	Returns the total sum of a numeric field. Select a field from the drop-down list.
Min	Returns the minimum value of a numeric field. Select a field from the drop-down list.
Max	Returns the maximum value of a numeric field. Select a field from the drop-down list.
Unique Count	Returns the number of unique values in a field. Select a field from the drop-down list.

To create a tile maps panel, complete the following steps:

1. Enter a string in the **Custom Label** field to change the display label.
2. Specify if you are splitting the chart or displaying the buckets as Geo Coordinates on a single chart.

A multiple chart split must run before any other aggregations.

Tile maps use the Geohash aggregation as their initial aggregation.

3. Select a field, typically coordinates, from the drop-down list.

The Precision slider determines the granularity of the results displayed on the map. See the Kibana documentation for the geohash grid aggregation for details on the area specified by each precision level.

4. Once you've specified a buckets aggregation, define sub-aggregations to refine the visualization.

Tile maps only support sub-aggregations as split charts. Click + **Add Sub Aggregation**, then **Split Chart** to select a sub-aggregation from the list of types:

Data Histogram	A data histogram is built from a numeric field and organized by date. You can specify a time frame for the intervals in seconds, minutes, hours, days, weeks, months, or years. You can also specify a custom interval frame by selecting Custom as the interval and specifying a number and a time unit in the text field. Custom interval time units are s for seconds, m for
----------------	--

minutes, h for hours, d for days, w for weeks, and y for years. Different units support different levels of precision, down to one second.

Histogram	A standard histogram is built from a numeric field. Specify an integer interval for this field. Select the Show empty buckets check box to include empty intervals in the histogram.
Range	With a range aggregation, you can specify ranges of values for a numeric field. Click Add Range to add a set of range endpoints. Click the red (x) symbol to remove a range. After changing options, click the green Apply changes button to update your visualization, or the grey Discard changes button to keep your visualization in its current state.
Date Range	A date range aggregation reports values that are within a range of dates that you specify. You can specify the ranges for the dates using date math expressions. Click Add Range to add a set of range endpoints. Click the red (/) symbol to remove a range.
IPv4 Range	The IPv4 range aggregation enables you to specify ranges of IPv4 addresses. Click Add Range to add a set of range endpoints. Click the red (/) symbol to remove a range.
Terms	A terms aggregation enables you to specify the top or bottom n elements of a given field to display, ordered by count or a custom metric.
Filters	You can specify a set of filters for the data. You can specify a filter as a query string or in JSON format, just as in the Discover search bar. Click Add Filter to add another filter. Click the  button to open the label field, where you can type in a name to display on the visualization.
Significant Terms	You can specify a set of filters for the data. You can specify a filter as a query string or in JSON format, just as in the Discover search bar. Click Add Filter to add another filter. Click the  button to open the label field, where you can type in a name to display on the visualization.
Geohash	The geohash aggregation displays points based on the geohash coordinates.



Note

By default, the Change precision on map zoom box is checked. Clear the box to disable this behavior.

5. You can click the [Advanced](#) link to display more customization options for your metrics or bucket aggregation:

Exclude Pattern	Specifies a pattern in this field to exclude from the results.
Exclude Pattern Flags	A standard set of Java flags for the exclusion pattern.
Include Pattern	Specifies a pattern in this field to include in the results.
Include Pattern Flags	A standard set of Java flags for the inclusion pattern.
JSON Input	<p>A text field where you can add specific JSON-formatted properties to merge with the aggregation definition, as in the following example:</p> <pre style="background-color: #f0f0f0; padding: 5px;">{ "script" : "doc['grade'].value * 1.2" }</pre> <p>The availability of these options varies depending on the aggregation you choose.</p>

6. Select the **Options** tab to change the following aspects of the chart:

Map type	Select one of the following options from the drop-down list.	
	Scaled Circle Markers	Scale the size of the markers based on the metric aggregation's value.
	Shaded Circle Markers	Displays the markers with different shades based on the metric aggregation's value.
	Shaded Geohash Grid	Displays the rectangular cells of the geohash grid instead of circular markers, with different shades based on the metric aggregation's value.
	Heatmap	<p>A heat map applies blurring to the circle markers and applies shading based on the amount of overlap. Heatmaps have the following options:</p> <ul style="list-style-type: none"> • Radius-Sets the size of the individual heatmap dots.

- **Blur**-Sets the amount of blurring for the heatmap dots.
- **Maximum zoom**- Tilemaps in Kibana support 18 zoom levels. This slider defines the maximum zoom level at which the heatmap dots appear at full intensity.
- **Minimum opacity**- Sets the opacity cutoff for the dots.
- **Show Tooltip**- Check this box to have a tooltip with the values for a given dot when the cursor is on that dot.

Desaturate map tiles

Check this box to have a tooltip with the values for a given dot when the cursor is on that dot.

WMS compliant map server

Check this box to enable the use of a third-party mapping service that complies with the Web Map Service (WMS) standard. Specify the following elements:

WMS url The URL for the WMS map service.

WMS layers A comma-separated list of the layers to use in this visualization. Each map server provides its own list of layers.

WMS version The WMS version used by this map service.

WMS format The image format used by this map service. The two most common formats are image/png and image/jpeg.

WMS attribution	An optional, user-defined string that identifies the map source. Maps display the attribution string in the lower right corner.
WMS styles	A comma-separated list of the styles to use in this visualization. Each map server provides its own styling options.

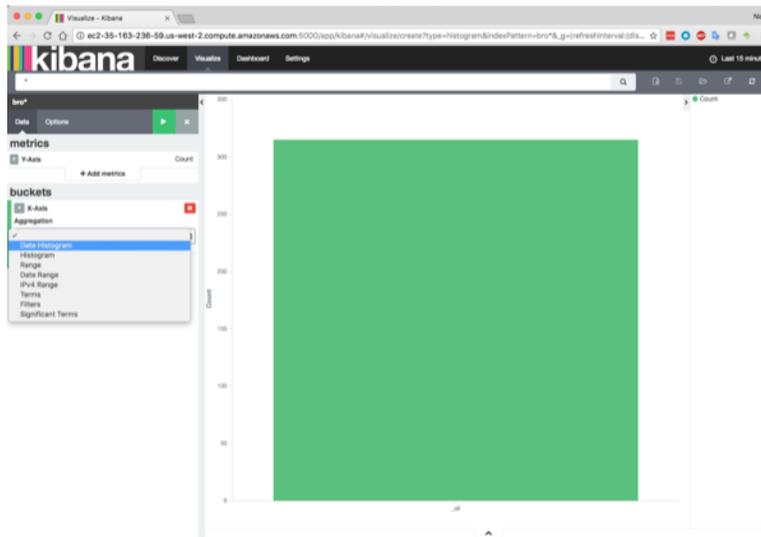
7. After changing options, click the green **Apply** changes button to update your visualization, or the grey **Discard** changes button to keep your visualization in its current state.

2.4.3.6. Adding a Vertical Bar Chart Panel

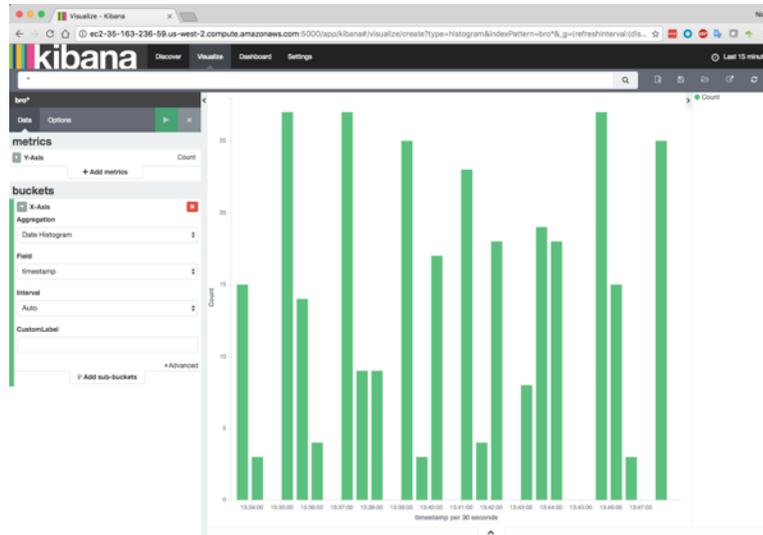
You can use the vertical bar chart panel type to display histograms. Histogram panels represent ingest rates for each individual telemetry type. By convention you should set up one for each data source type.

To set up a histogram panel, complete the following steps:

1. Create a query for the new telemetry type in the Discover window.
2. Select the **Visualize** tab then select **Vertical bar chart**.
3. Select **From a new search** for search source, and then select your new data source index pattern.
4. Click X-Axis and then, under Aggregation, click Data Histogram.



5. Click the green Apply arrow to populate the Visualization window.



6. Click the Save Visualization () button in the Visualization toolbar, enter a name for the new data source visualization, then click **Save**.

7. Select the **Dashboard** tab, and then click the Plus () button.

8. Select the Visualization tab and then search for the name of the new data source visualization.

The visualization will be added to the bottom of the dashboard.

9. Click the Save Dashboard () button in the dashboard toolbar to save the dashboard.

Alternatively, you can clone one of the existing histogram panels, rename it to reflect the new name of the panel, and point it to the x_doc query associated with the new telemetry.

You will see the panel now being populated with indexed documents of type x_doc.

2.4.3.7. Setting up Details Message Panels for Telemetry Data

HCP supports two types of messages: metadata and alerts. By convention there should be one panel per metadata telemetry and one panel that is a "catch all" panel for alerts. This section provides instructions for setting up both of these panel types.

- [Setting up a Metadata Panel \[24\]](#)
- [Setting up an Alerts Telemetry Panel \[25\]](#)

2.4.3.7.1. Setting up a Metadata Panel

To setup your own details panel for a metadata telemetry:

1. Make sure you have a query set up to look for x_doc.
2. Have an ingest histogram setup.

3. Clone one of the existing details panels, rename it, and point it to the `x_doc` pinned query.

You will see the panel now being populated with indexed documents of type `x_doc`.

2.4.3.7.2. Setting up an Alerts Telemetry Panel

Alerts telemetry come from IDS sensors like Snort or mixed telemetries like application logs that contain some metadata and some alert messages. While it is possible to set up a new panel for each alert telemetry, it is more desirable to set up a single panel that contains all of the alerts. To do so you need a specific query. Each telemetry message in HCP that contains an alert is tagged with `is_alert=true` field.

1. Determine the type of telemetry panel you want to set up.

While it is possible to set up a new panel for each alert telemetry, it is more desirable to set up a single panel that contains all of the alerts.

2. Create a query.

A query to look for all alerts would look something like: `is_alert=true`.

This query will pull in alerts from multiple telemetries (even mixed mode telemetries that have some metadata and some alerts associated with them).

See the “Alerts” table for a detailed table containing only alerts.

3. Customize the fields displayed for each alerts table by checking or clearing the field name boxes.

Ideally, you want the fields of most importance as well as the standard fields that telemetries are correlated on to be displayed.

2.5. Customizing Your Dashboard

The visualizations in your Metron dashboard are stored in resizeable containers that you can arrange on the dashboard. This section discusses customizing the containers.

Adding visualizations

To add a visualization to the dashboard, click the New

Visualization () button on the Dashboard toolbar. Select a saved visualization from the list. You can filter the list of visualizations by typing a filter string into the **Visualization Filter** field.

Saving the dashboard

To save the dashboard, click the Save Dashboard () button on the Dashboard toolbar, enter a name for the dashboard in the **Save As** field, and click the **Save** button.

By default, dashboards store the time period specified in the time filter when you save a dashboard. To disable

this behavior, clear the **Store time with dashboard box** before clicking the **Save** button.

Loading a saved dashboard

Click the Load Saved Dashboard () button to display a list of existing dashboards. The saved dashboard selector includes a text field to filter by dashboard name and a link to the Object Editor for managing your saved dashboards.

Moving containers

Click and hold a container's header to move the container around the dashboard. Other containers will shift as needed to make room for the moving container. Release the mouse button to confirm the container's new location.

Resizing containers

Move the cursor to the bottom right corner of the container until the cursor changes to point at the corner. After the cursor changes, click and drag the corner of the container to change the container's size. Release the mouse button to confirm the new container size.

Removing containers

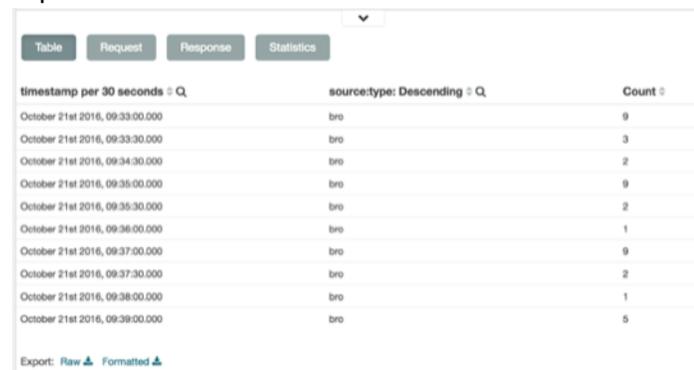
Click the x icon at the top right corner of a container to remove that container from the dashboard. Removing a container from a dashboard does not delete the saved visualization in that container.

Viewing detailed information

To display the raw data behind the visualization, click the caret at the bottom of the container. The dashboard replaces the visualization format with the raw data on which the visualization is based. You can view the data in the following formats by clicking the buttons at the top of the component:

Table

A representation of the underlying data, presented as a paginated data grid. You can sort the items in the table by clicking on the table headers at the top of each column.

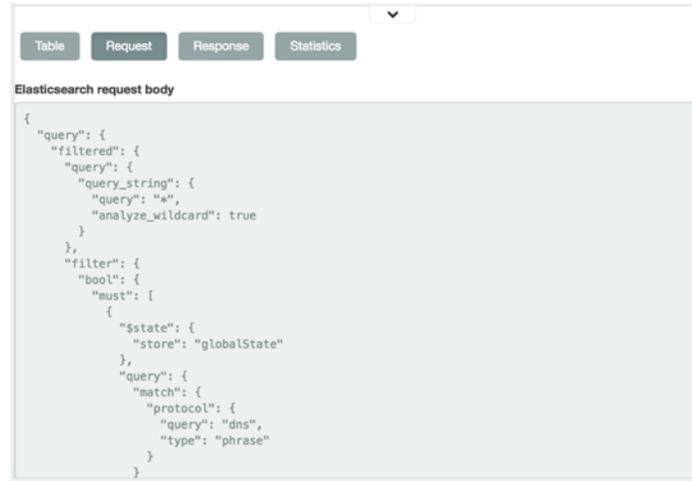


timestamp per 30 seconds	source:type	Count
October 21st 2016, 09:33:00.000	bro	9
October 21st 2016, 09:33:30.000	bro	3
October 21st 2016, 09:34:00.000	bro	2
October 21st 2016, 09:35:00.000	bro	9
October 21st 2016, 09:35:30.000	bro	2
October 21st 2016, 09:36:00.000	bro	1
October 21st 2016, 09:37:00.000	bro	9
October 21st 2016, 09:37:30.000	bro	2
October 21st 2016, 09:38:00.000	bro	1
October 21st 2016, 09:39:00.000	bro	5

Export: [Raw](#) [Formatted](#)

Request

The raw request used to query the server, presented in JSON format.

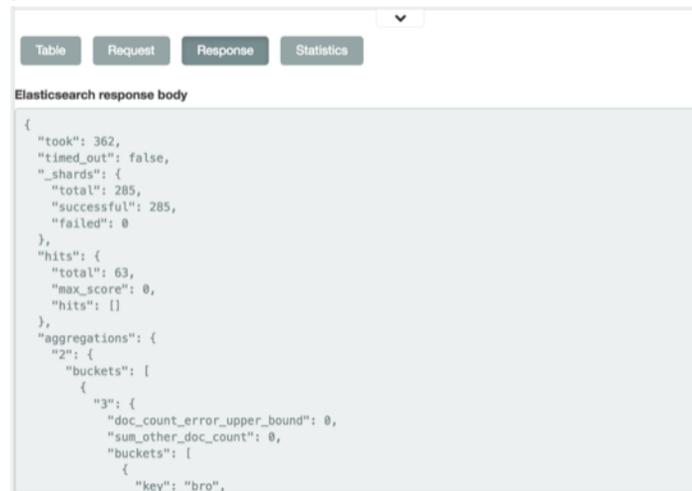


The screenshot shows a web interface with tabs for 'Table', 'Request', 'Response', and 'Statistics'. The 'Request' tab is selected. Below the tabs, the text 'Elasticsearch request body' is displayed above a JSON code block. The JSON represents a filtered query with a filter on the 'state' field and a match on the 'protocol' field.

```
{
  "query": {
    "filtered": {
      "query": {
        "query_string": {
          "query": "a",
          "analyze_wildcard": true
        }
      },
      "filter": {
        "bool": {
          "must": [
            {
              "$state": {
                "store": "globalState"
              }
            },
            {
              "query": {
                "match": {
                  "protocol": {
                    "query": "dns",
                    "type": "phrase"
                  }
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

Response

The raw response from the server, presented in JSON format.



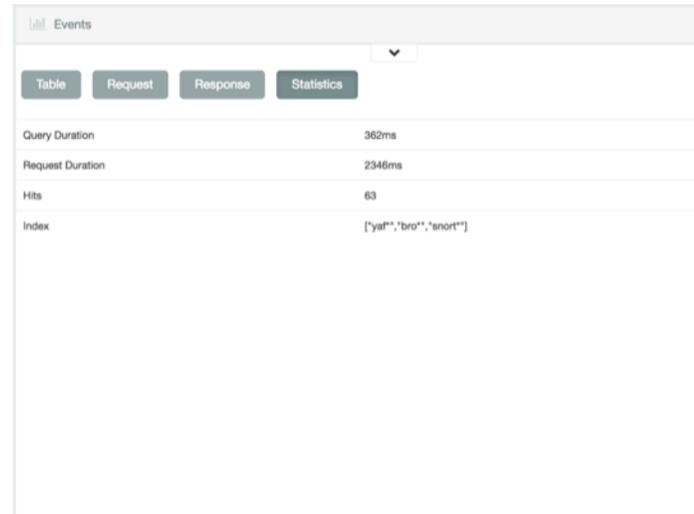
The screenshot shows the same web interface as above, but with the 'Response' tab selected. The text 'Elasticsearch response body' is displayed above a JSON code block. The JSON represents the server's response, including timing information, shard statistics, and aggregation results for the 'state' field.

```
{
  "took": 362,
  "timed_out": false,
  "_shards": {
    "total": 285,
    "successful": 285,
    "failed": 0
  },
  "hits": {
    "total": 63,
    "max_score": 0,
    "hits": []
  },
  "aggregations": {
    "2": {
      "buckets": [
        {
          "3": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
              {
                "key": "bro",

```

Statistics

A summary of the statistics related to the request and the response, presented as a data grid. The data grid includes the query duration, the request duration, the total number of records found on the server, and the index pattern used to make the query.



The screenshot shows the 'Events' dashboard in Metron. It features a navigation bar with 'Table', 'Request', 'Response', and 'Statistics' tabs. Below the navigation bar, a table displays the following statistics:

Query Duration	362ms
Request Duration	2346ms
Hits	63
Index	['yaf*', 'bro*', 'snort*']

2.6. Using pcap to View Your Raw Data

The pcap data source can rapidly ingest raw data directly into HDFS from Kafka. As a result, you can store all of the raw packet capture data in HDFS and review or query it at a later date. The pcap data is not displayed in the Metron dashboard, but you can query, view, or retrieve the data in order to port it to another application like Wireshark. The following sections provide instructions on retrieving and filtering the pcap data using the utilities described in the following sections:

- [Capturing pcap Data \[28\]](#)
- [Processing pcap Data \[29\]](#)
- [Viewing pcap Data \[30\]](#)
- [Filtering pcap Data \[30\]](#)
- [Porting pcap Data to Another Application \[33\]](#)

2.6.1. Capturing pcap Data

In your production environment there is likely to be one or more hosts configured with one or more span ports that receives raw packet data from a packet aggregation device. You can use one of HCP's packet capture programs to capture the pcap data; pycapa and DPDK. These programs are responsible for capturing the raw packet data off the wire and sending that data to Kafka where it can be ingested by HCP.

The following example uses Pycapa.

```
service pycapa start
```

If everything worked correctly, the raw packet data can be consumed from a Kafka topic called pcap. The data is binary.

```
$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh -z
zookeeper1:2181 --topic pcap
E)###>K#####P#"ssLQlJ
P##0
E( #@##x#####>K###"PQlJ
ssLPF#
```

2.6.2. Processing pcap Data

After you capture some pcap data, the next step is to have HCP process the pcap data and store it in HDFS. Start the PCAP topology to begin this process. A Storm topology called 'pcap' is launched that consumes the raw pcap data from the Kafka topic and writes this data into sequence files in HDFS.

```
$ $METRON_HOME/bin/start_pcap_topology.sh
Running: /usr/jdk64/jdk1.8.0_77/bin/java -server -Ddaemon.name= -Dstorm.
options= -Dstorm.home=/usr/hdp/2.5.0.0-1245/storm -Dstorm.log.dir=/var/log/
storm -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib -Dstorm.conf.
file= -cp /usr/hdp/2.5.0.0-1245/storm/lib/log4j-core-2.1.jar:/usr/hdp/2.5.
0.0-1245/storm/lib/storm-core-1.0.1.2.5.0.0-1245.jar:/usr/hdp/2.5.0.0-1245/
storm/lib/minlog-1.3.0.jar:/usr/hdp/2.5.0.0-1245/storm/lib/objenesis-2.1.
jar:/usr/hdp/2.5.0.0-1245/storm/lib/ring-cors-0.1.5.jar:/usr/hdp/2.5.0.0-1245/
storm/lib/storm-rename-hack-1.0.1.2.5.0.0-1245.jar:/usr/hdp/2.5.0.0-1245/
storm/lib/disruptor-3.3.2.jar:/usr/hdp/2.5.0.0-1245/storm/lib/kryo-3.0.3.
jar:/usr/hdp/2.5.0.0-1245/storm/lib/log4j-over-slf4j-1.6.6.jar:/usr/hdp/2.5.0.
0-1245/storm/lib/reflectasm-1.10.1.jar:/usr/hdp/2.5.0.0-1245/storm/lib/log4j-
slf4j-impl-2.1.jar:/usr/hdp/2.5.0.0-1245/storm/lib/log4j-api-2.1.jar:/usr/
hdp/2.5.0.0-1245/storm/lib/clojure-1.7.0.jar:/usr/hdp/2.5.0.0-1245/storm/lib/
zookeeper.jar:/usr/hdp/2.5.0.0-1245/storm/lib/servlet-api-2.5.jar:/usr/hdp/
2.5.0.0-1245/storm/lib/slf4j-api-1.7.7.jar:/usr/hdp/2.5.0.0-1245/storm/lib/
asm-5.0.3.jar org.apache.storm.daemon.ClientJarTransformerRunner org.apache.
storm.hack.StormShadeTransformer /usr/metron/0.3.0/lib/metron-pcap-backend-0.
3.0.jar /tmp/d5f844e8b1a611e6a6d10a0a570e5f4d.jar
Running: /usr/jdk64/jdk1.8.0_77/bin/java -client -Ddaemon.name= -Dstorm.
options= -Dstorm.home=/usr/hdp/2.5.0.0-1245/storm -Dstorm.log.dir=/var/
log/storm -Djava.library.path=/usr/local/lib:/opt/local/lib:/usr/lib:/usr/
hdp/current/storm-client/lib -Dstorm.conf.file= -cp /usr/hdp/2.5.0.0-1245/
storm/lib/log4j-core-2.1.jar:/usr/hdp/2.5.0.0-1245/storm/lib/storm-core-1.
0.1.2.5.0.0-1245.jar:/usr/hdp/2.5.0.0-1245/storm/lib/minlog-1.3.0.jar:/
usr/hdp/2.5.0.0-1245/storm/lib/objenesis-2.1.jar:/usr/hdp/2.5.0.0-1245/
storm/lib/ring-cors-0.1.5.jar:/usr/hdp/2.5.0.0-1245/storm/lib/storm-rename-
hack-1.0.1.2.5.0.0-1245.jar:/usr/hdp/2.5.0.0-1245/storm/lib/disruptor-3.
3.2.jar:/usr/hdp/2.5.0.0-1245/storm/lib/kryo-3.0.3.jar:/usr/hdp/2.5.0.
0-1245/storm/lib/log4j-over-slf4j-1.6.6.jar:/usr/hdp/2.5.0.0-1245/storm/
lib/reflectasm-1.10.1.jar:/usr/hdp/2.5.0.0-1245/storm/lib/log4j-slf4j-
impl-2.1.jar:/usr/hdp/2.5.0.0-1245/storm/lib/log4j-api-2.1.jar:/usr/hdp/
2.5.0.0-1245/storm/lib/clojure-1.7.0.jar:/usr/hdp/2.5.0.0-1245/storm/lib/
zookeeper.jar:/usr/hdp/2.5.0.0-1245/storm/lib/servlet-api-2.5.jar:/usr/hdp/
2.5.0.0-1245/storm/lib/slf4j-api-1.7.7.jar:/usr/hdp/2.5.0.0-1245/storm/lib/
asm-5.0.3.jar:/tmp/d5f844e8b1a611e6a6d10a0a570e5f4d.jar:/usr/hdp/current/
storm-supervisor/conf:/usr/hdp/2.5.0.0-1245/storm/bin -Dstorm.jar=/tmp/
d5f844e8b1a611e6a6d10a0a570e5f4d.jar org.apache.storm.flux.Flux --remote /
usr/metron/0.3.0/flux/pcap/remote.yaml --filter /usr/metron/0.3.0/config/pcap.
properties
#####      ###      #####      ##
#####      ###      #####
#####      ###      ###      #####
```

```
#####   ##          ##          ##          #####
###          #####
###          #####          ##          ##
+-          Apache Storm          -+
+- data Flow User eXperience -+
Version: 1.0.1
Parsing file: /usr/metron/0.3.0/flux/pcap/remote.yaml
636 [main] INFO   o.a.s.f.p.FluxParser - loading YAML from input stream...
638 [main] INFO   o.a.s.f.p.FluxParser - Performing property substitution.
639 [main] INFO   o.a.s.f.p.FluxParser - Not performing environment variable
substitution.
907 [main] WARN   o.a.s.f.FluxBuilder - Found multiple invocable methods for
class org.apache.metron.spout.pcap.SpoutConfig, method from, given arguments
[END]. Using the last one found.
976 [main] INFO   o.a.s.f.FluxBuilder - Detected DSL topology...
----- TOPOLOGY DETAILS -----
Topology Name: pcap
----- SPOUTS -----
kafkaSpout [1] (org.apache.metron.spout.pcap.KafkaToHDFSspout)
----- BOLTS -----
----- STREAMS -----
-----
1157 [main] INFO   o.a.s.f.Flux - Running remotely...
1157 [main] INFO   o.a.s.f.Flux - Deploying topology in an ACTIVE state...
1194 [main] INFO   o.a.s.StormSubmitter - Generated ZooKeeper secret payload
for MD5-digest: -8340121339010421700:-4824301672672404920
1268 [main] INFO   o.a.s.s.a.AuthUtils - Got AutoCreds []
1343 [main] INFO   o.a.s.StormSubmitter - Uploading topology jar /tmp/
d5f844e8b1a611e6a6d10a0a570e5f4d.jar to assigned location: /data1/hadoop/
storm/nimbus/inbox/stormjar-49aedc3d-a259-409d-a96b-4b615ce07076.jar
1810 [main] INFO   o.a.s.StormSubmitter - Successfully uploaded topology jar
to assigned location: /data1/hadoop/storm/nimbus/inbox/stormjar-49aedc3d-
a259-409d-a96b-4b615ce07076.jar
1820 [main] INFO   o.a.s.StormSubmitter - Submitting topology pcap in
distributed mode with conf {"topology.workers":1,"storm.zookeeper.
topology.auth.scheme":"digest", "storm.zookeeper.topology.auth.
payload":"-8340121339010421700:-4824301672672404920"}
2004 [main] INFO   o.a.s.StormSubmitter - Finished submitting topology: pcap
```

2.6.3. Viewing pcap Data

To view the pcap data, use the pcap inspector utility, `$METRON_HOME/bin/pcap_inspector.sh`. This utility enables you to retrieve and view portions of the sequence files which store the pcap data in HDFS.

```
usage: PcapInspector
-h,--help          Generate Help screen
-i,--input <SEQ_FILE>  Input sequence file on HDFS
-n,--num_packets <N>  Number of packets to dump
```

2.6.4. Filtering pcap Data

You can search or filter the pcap data using either a command line tool or a REST API. The following sections describe how to use each of these tools to query the pcap data:

- [Using CLI to Query pcap Data \[31\]](#)
- [Using REST API to Query pcap Data \[33\]](#)

2.6.4.1. Using CLI to Query pcap Data

You can search or filter the pcap data with one of the following command line tools:

- Fixed filter option
- Query filter option (Metron Stellar)

Both tools are executed by `${metron_home}/bin/pcap_query.sh [fixed|query]`



Note

Because the output from a query can be very large, we recommend appending `-rpf $Number_of_records` to the end of the filter command. This argument creates multiple files, populating them with the specified number of records and titling them with timestamps.

You can filter or query for the following fields in the pcap data:

- ip_scr_addr
- ip_dst_addr
- ip_src_port
- ip_dst_port
- protocol
- timestamp

Fixed filter options:

```
-bop,--base_output_path <arg> Query result output path. Default is '/tmp'.
-bp,--base_path <arg> Base PCAP data path. Default is '/apps/metron/pcap'.
-da,--ip_dst_addr <arg> Destination IP address.
-df,--date_format <arg> Date format to use for parsing start_time and end_time. Default is to use time in millis since the epoch.
-dp,--ip_dst_port <arg> Destination port.
-et,--end_time <arg> Packet end time range. Default is current system time.
-nr,--num_reducers <arg> The number of reducers to use. Default is 10.
-h,--help Display help.
-ir,--include_reverse Indicates if filter should check swapped src/dest addresses and IPs.
-p,--protocol <arg> IP Protocol.
-rpf Maximum number of records per file.
-sa,--ip_src_addr <arg> Source IP address.
-sp,--ip_src_port <arg> Source port.
-st,--start_time <arg> (required) Packet start time range.
```

Fixed filter examples:

```
$METRON_HOME/bin/pcap_query.sh fixed \
    -st "20160617" \
    -df "yyyyMMdd" \
    -sa 192.168.138.158 \
    -da 123.456.789.012 \
    -sp 49197 \
    -dp 80 \
    -p 6
    -rpf 500
```

Query filter options:

```
-bop,--base_output_path <arg> Query result output path. Default is
                              '/tmp'.
-bp,--base_path <arg>        Base PCAP data path. Default is
                              '/apps/metron/pcap'.
-df,--date_format <arg>     Date format to use for parsing start_time
                              and end_time. Default is to use time in
                              millis since the epoch.
-et,--end_time <arg>        Packet end time range. Default is current
                              system time.
-nr,--num_reducers <arg>    The number of reducers to use. Default
                              is 10.
-h,--help                    Display help.
-q,--query <arg>            Query string to use as a filter.
-rpf                           Maximum number of records per file.
-st,--start_time <arg>      (required) Packet start time range.
```

Query filter examples:

```
$METRON_HOME/bin/pcap_query.sh query \
    -st "20160617" \
    -df "yyyyMMdd" \
    --query "ip_src_addr == '192.168.138.
158' and ip_src_port == '49197' \
    and ip_dst_addr == '123.456.
789.012' and ip_dst_port == '80' \
    and protocol == '6'"
    -rpf 500
```

```
$METRON_HOME/bin/pcap_query.sh query \
    -st "146613600000" \
    --query "IN_SUBNET(ip_src_addr, '192.
168.0.0/24') and ip_src_port == '49197' \
    and ip_dst_addr == '123.456.
789.012' and ip_dst_port == '80' \
    and protocol == '6'"
    -rpf 500
```

```
# subnet function checks IP is in specified subnet
--query "IN_SUBNET(ip_src_addr, '192.168.0.0/24') \
    and ip_src_port == '49197' \
    and ip_dst_addr == '123.456.789.012' \
    and ip_dst_port == '80' \
    and protocol == '6'"
```

```
# range queries on ports
--query "ip_src_port <= 50000 and ip_dst_port >= 30000"
```

```
# range queries with conditionals and parens
--query "(ip_src_port < 50000 and ip_src_port > 40000) \
```

```
or (ip_src_port < 20000 and ip_src_port > 10000)"  
# in/not in list of values  
--query "ip_src_port < 10000 and ip_dst_port in ['54056', '54057',  
'8080']"
```

2.6.4.2. Using REST API to Query pcap Data

The REST API is an alternative to using the CLI and Stellar to query the pcap data. The purpose of this service is to provide a middle tier to negotiate retrieving packet capture data that flows into HCP. This packet data is in a form that libpcap-based tools can read. The REST API exposes the query functionality via http.

This fixed filter is less expressive than the CLI/Stellar filter option. The fixed filter gives you the following values that you can filter on:

- srclp
- srcPort
- dstlp
- dstPort
- startTime
- endTime

The fixed filter filters on explicit matches only so you cannot use any specialized functions or comparison operators. The query filter is far more expressive because it exposes Stellar, which enables the user to invoke specialized functions, use comparison operators, and perform boolean logic on the parameters.

You can start the fixed filter service either via the `init.d` script installed, `/etc/init.d/pcapservice` or directly via the `yarn jar` command:

```
yarn jar $METRON_HOME/lib/metron-api-$METRON_VERSION.jar org.apache.metron.  
pcapservice.rest.PcapService -port $SERVICE_PORT -query_hdfs_path $QUERY_PATH  
-pcap_hdfs_path $PCAP_PATH
```

where

- METRON_HOME is the location of the metron installation
- METRON_VERSION is the version of the metron installation
- SERVICE_PORT is the port to bind the REST service to
- QUERY_PATH is the temporary location to store query results. They are deleted after the service reads them.
- PCAP_PATH is the path to the packet data on HDFS

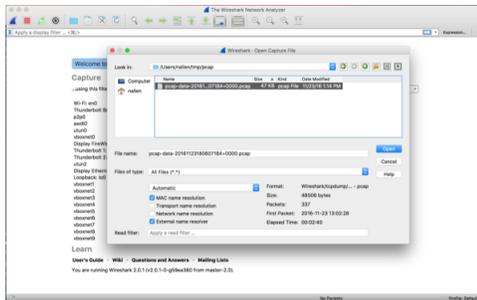
2.6.5. Porting pcap Data to Another Application

When you use the pcap query utility to extract pcap data, the utility creates a libpcap-compliant pcap file in the current working directory.

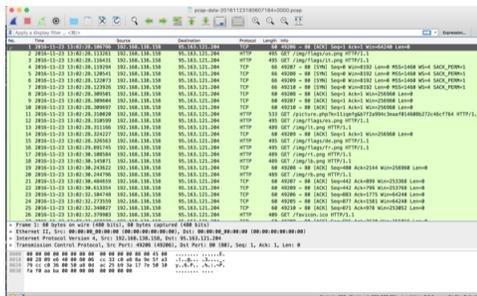
```
[root@ip-10-0-0-53 0.3.0]# ls -l
total 72
drwxr-xr-x. 2 livy games 4096 Nov 22 22:36 bin
drwxr-xr-x. 3 livy games 4096 Nov 23 17:10 config
drwxr-xr-x. 2 livy games 4096 Sep 29 17:44 ddl
drwxr-xr-x. 6 livy games 4096 Aug 22 14:54 flux
drwxr-xr-x. 2 root root 4096 Nov 23 17:07 lib
drwxr-xr-x. 2 livy games 4096 Nov 22 22:36 patterns
-rw-r--r--. 1 root root 48506 Nov 23 18:06 pcap-data-20161123180607184+0000.pcap

[root@ip-10-0-0-53 0.3.0]# file pcap-data-20161123180607184+0000.pcap
pcap-data-20161123180607184+0000.pcap: tcpdump capture file (little-endian) -
version 2.4 (Ethernet, capture length 65535)
```

You can open the libpcap-compliant pcap file with any third-party tool that supports the file type. For example, you can load Wireshark and choose **File > Open**. Wireshark will load the pcap file.



The content of the file will be similar to the following:



3. Sharing the Metron Dashboard

You might want to share the queries and visualizations you've set up with other SOC personnel. This chapter covers the following methods for sharing the Metron dashboard:

- [Exporting Search Information \[35\]](#)
- [Importing Search Information \[35\]](#)

3.1. Exporting Search Information

You can export the contents of a query or search. This option can be very useful after you've refined your search to display only the relevant information for a cybersecurity issue and you would like to send this information to another SOC team member.

To export a query, complete the following steps:

1. Click the **Settings** tab, then click the **Objects** tab.
2. Select the object that you want to view.

You can export the dashboard, searches, or visualization. To export the contents of a search, click **Searches**, then click the check box next to the search you want to export.

3. Click **Export** to select a location to which to write the exported JSON.



Note

Exported dashboards do not include their associated index patterns. Re-create the index patterns manually before importing saved dashboards to a Kibana instance running on another Elasticsearch cluster.

3.2. Importing Search Information

You can import the contents of a query or search. This option can be very useful if you need to view a colleague's refined search for a cybersecurity issue.

To import a query, complete the following steps:

1. Click the **Settings** tab, then click the **Objects** tab.
2. Click **Import** to navigate to the JSON file representing the set of objects to import.
3. Select the file you want to import and click **Open**.
4. If any objects in the set will overwrite objects already present in the Metron dashboard, confirm the overwrite.