

Hortonworks Cybersecurity Package

User Guide

(December 12, 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
2. Introduction to Metron Dashboard	2
2.1. Functionality of Metron Dashboard	2
2.2. Metron Default Dashboard	4
2.2.1. Events	4
2.2.2. Enrichment	4
2.2.3. YAF	5
2.2.4. Snort	5
2.2.5. Web Request Header	6
2.2.6. DNS	6
3. Customizing Your Metron Dashboard	8
3.1. Launching the Metron Dashboard	8
3.2. Changing the Metron Dashboard Background Color	9
3.3. Adding a New Data Source	9
3.3.1. Prerequisites	9
3.3.2. Configuring a New Data Source Index	9
3.3.3. Reviewing the New Data Source Data	10
3.4. Querying, Filtering, and Visualizing Data	11
3.5. Customizing Your Dashboard	12
4. Sharing the Metron Dashboard	13
5. Triaging Alerts	14
5.1. Launching the Alerts User Interface	14
5.2. Viewing Alerts	14
5.2.1. Using the Alerts Table	15
5.2.2. Searching Alerts	20
5.2.3. Filtering Alerts	21
5.2.4. Managing Alert Status	23
5.2.5. Grouping Alerts	25
5.2.6. Creating a Meta Alert	27
5.3. Saving Your Searches	29
5.4. Viewing Your Recent and Saved Searches	30
6. PCAP	32
6.1. Capturing pcap Data	32
6.2. Processing pcap Data	32
6.3. Viewing pcap Data	34
6.4. Filtering pcap Data	34
6.4.1. Using CLI to Query pcap Data	34
6.4.2. Using REST API to Query pcap Data	37
6.5. Porting pcap Data to Another Application	38

List of Figures

2.1. Dashboard-Snort Panel	2
2.2. Events	4
2.3. Enrichment	5
2.4. YAF	5
2.5. Dashboard-Snort Panel	6
2.6. Dashboard-Bro Panel	6
2.7. Dashboard-DNS Panel	7
3.1. Ambari Task List	8
3.2. Configure an Index Pattern	10
3.3. Discover Tab with Squid Elements	11
5.1. Alerts Configure Table	17
5.2. Alerts Settings Panel	18
5.3. Alerts Information Panel	19
5.4. Searches Field	20
5.5. Time Selector Dialog Box	21
5.6. Alerts Information Panel	24
5.7. Alerts Information Panel	29
5.8. Searches Panel	30

List of Tables

3.1. Querying, Filtering, and Visualizing Data	11
4.1. Sharing the Metron Dashboard	13
5.1. Alerts UI Tools and Purposes	14

1. Overview

This guide is intended for use by Security Operations Center (SOC) analysts and investigators.

This guide describes two user interfaces and a tool included with HCP that are designed for the SOC analysts and investigators:

- **Metron Dashboard**

A Kibana-based dashboard designed to identify, investigate, and analyze cybersecurity data. 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.

Refer to the following chapters:

- [Introduction to the Metron Dashboard](#)
- [Customizing Your Metron Dashboard](#)

- **Alerts User Interface**

This GUI is a standalone user interface that connects to Elasticsearch to show the alerts but also store all other data in the browser cache.

Refer to the following chapter:

- [Triaging Alerts](#)

- **pcap**

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.

Refer to the following chapter:

- [PCAP](#)

2. Introduction to Metron Dashboard

This chapter describes the Metron dashboard which is a component of Hortonworks Cybersecurity Package (HCP) powered by Apache Metron. This next chapter 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:

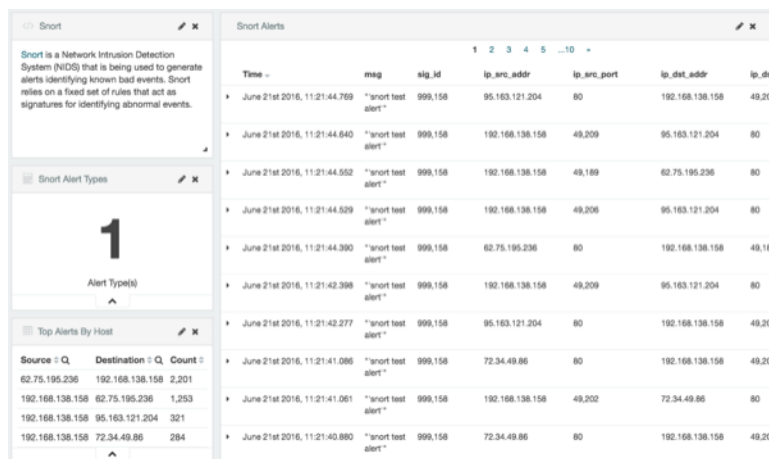
- [Functionality of Metron Dashboard \[2\]](#)
- [Metron Default Dashboard \[4\]](#)

2.1. 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 2.1. 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.

2.2. 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 \[4\]](#)
- [Enrichment \[4\]](#)
- [YAF \[5\]](#)
- [Snort \[5\]](#)
- [Web Request Header \[6\]](#)
- [DNS \[6\]](#)

2.2.1. 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 2.2. Events

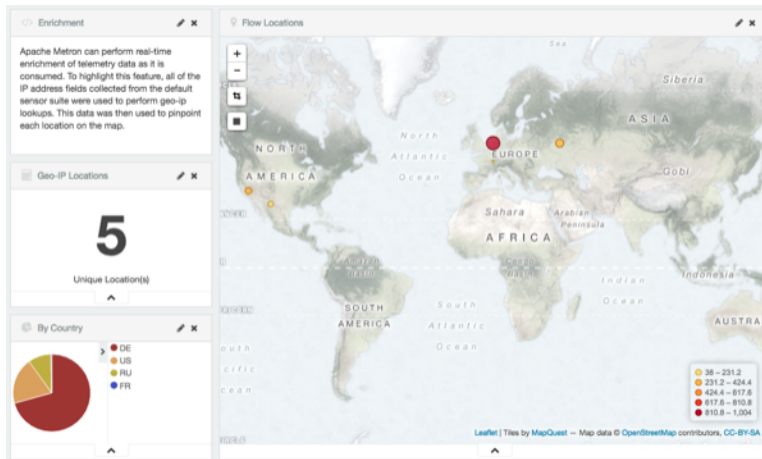


2.2.2. Enrichment

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.

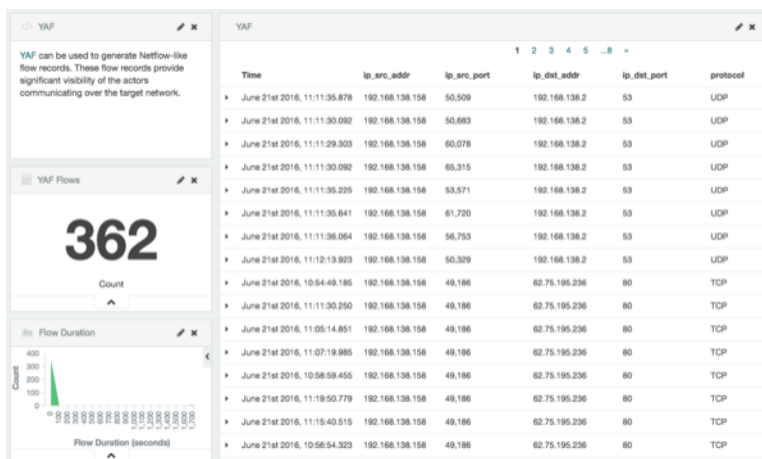
Figure 2.3. Enrichment



2.2.3. 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 2.4. YAF

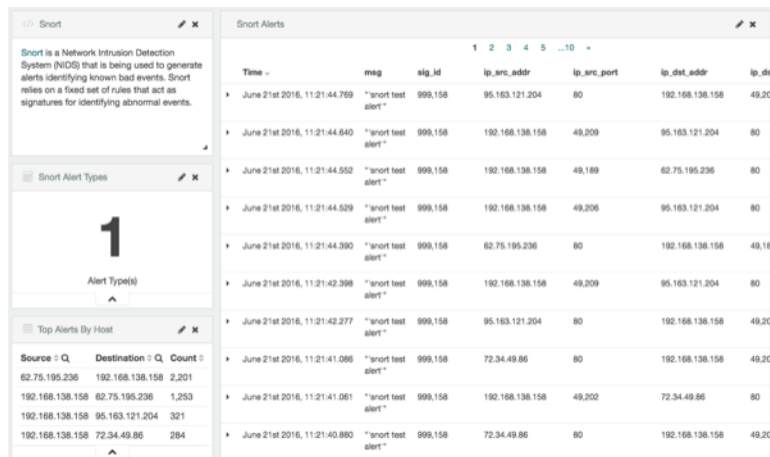


2.2.4. Snort

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.

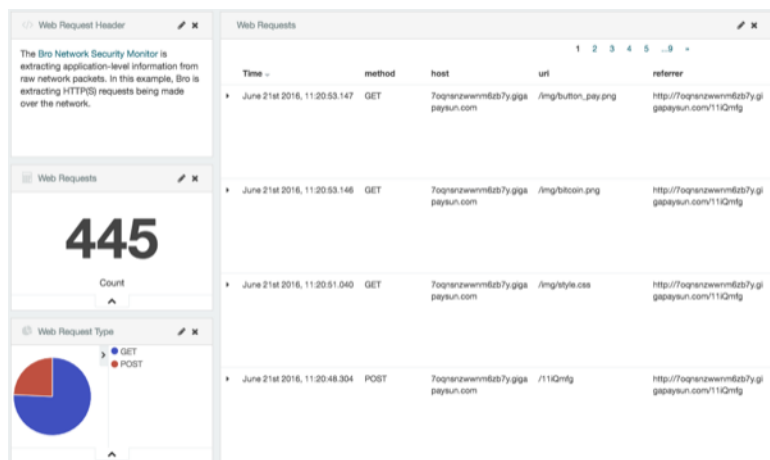
Figure 2.5. Dashboard-Snort Panel



2.2.5. 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.

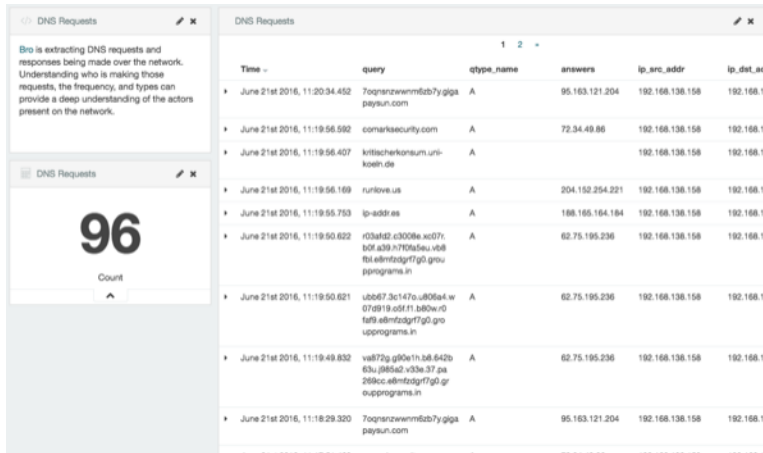
Figure 2.6. Dashboard-Bro Panel



2.2.6. 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 2.7. Dashboard-DNS Panel



3. Customizing Your Metron Dashboard

This chapter discusses customizing the Metron dashboard to display information, alerts, and the context you need to identify and analyze cybersecurity issues. For more information, see the [Kibana User Guide](#).

This chapter provides the following information:

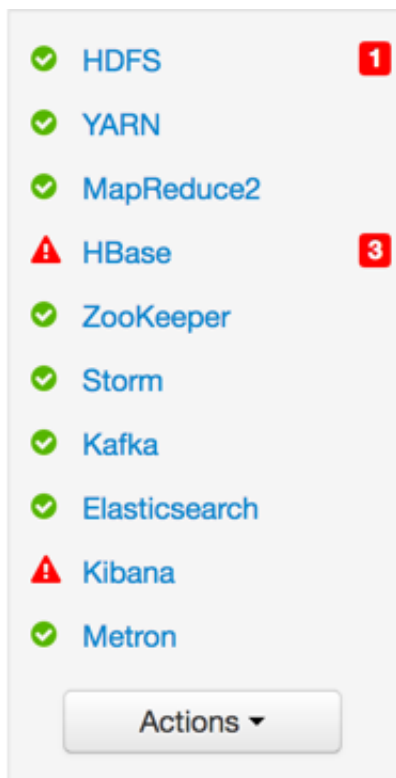
- [Launching the Metron Dashboard \[8\]](#)
- [Changing the Metron Dashboard Background Color \[9\]](#)
- [Adding a New Data Source \[9\]](#)
- [Querying, Filtering, and Visualizing Data \[11\]](#)
- [Customizing Your Dashboard \[12\]](#)

3.1. Launching the Metron Dashboard

You can launch the Metron Dashboard two ways:

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

Figure 3.1. Ambari Task List




- Enter the following text in a browser:

```
$KIBANA_HOST:9995
```

3.2. 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.

3.3. 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 \[9\]](#)
- [Configuring a New Data Source Index \[9\]](#)
- [Reviewing the New Data Source Data \[10\]](#)

3.3.1. 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](#).

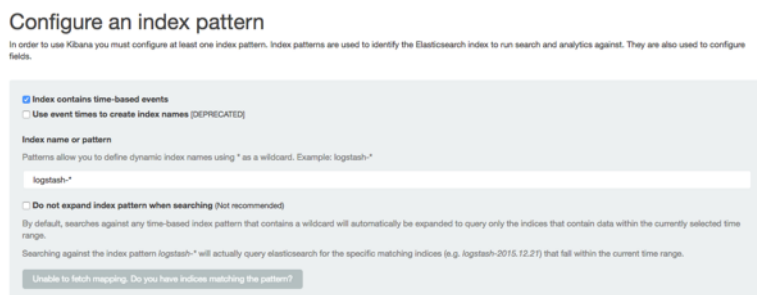
3.3.2. Configuring a New Data Source Index

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 3.2. 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.

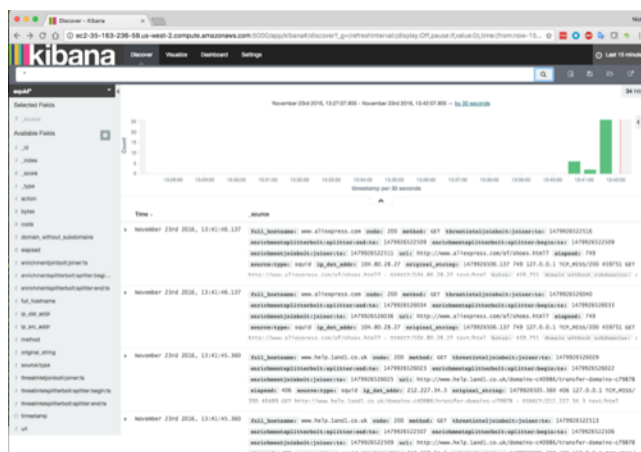
If you would like this new index pattern to be the default, click the Green Star icon (★).

3.3.3. 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 3.3. Discover Tab with Squid Elements



3.4. 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.

For more information about exploring and analyzing your data, refer to the Kibana documentation:

Table 3.1. Querying, Filtering, and Visualizing Data

Task	Description	Where to Look
Querying your data	<p>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.</p> <p>HCP includes queries for the following telemetries:</p> <ul style="list-style-type: none"> • YAF • Bro • Alerts (populated by Snort) <p>You can also add custom queries for new telemetry types.</p>	Discovering Your Data
Filter your query results	You can use the Metron dashboard to filter your query results to further refine the	Discover

Task	Description	Where to Look
	<p>information. The Metron dashboard provides two types of filters:</p> <p>Time Filter Restricts the search results to a specific time period.</p> <p>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.</p>	
Visualizing your data	You can filter search results to display only those documents that contain a particular value in a field. You can also create negative filters than exclude documents that contain the specified field value.	Visualize

3.5. Customizing Your Dashboard

The visualizations in your Metron dashboard are stored in resizable containers that you can arrange on the dashboard. For more information about customizing your dashboard, see [Building a Dashboard](#).

4. Sharing the Metron Dashboard

You might want to share the queries and visualizations you've set up with other SOC personnel.

Table 4.1. Sharing the Metron Dashboard

Task	Description	Where to Look
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.	Sharing a Dashboard
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.	Loading a Dashboard

5. Triaging Alerts

Any event that triggers your threat intelligence thresholds will trigger an alert. These alerts are how you are notified that an event needs your attention. HCP provides a graphics user interface (GUI) to view these alerts. This GUI is a standalone user interface that connects to Elasticsearch to show the alerts but also stores all other data in the browser cache. This chapter covers launching and using the Alerts user interface to identify and track cybersecurity issues:

- [Launching the Alerts User Interface \[14\]](#)
- [Viewing Alerts \[14\]](#)
- [Saving Your Searches \[29\]](#)
- [Viewing Your Recent and Saved Searches \[30\]](#)

5.1. Launching the Alerts User Interface

The Alerts user interface is bundled with HCP and installed with the Ambari management pack.

Prerequisite:

- Elasticsearch must be up and running and should have alerts populated by HDP topologies.
- The Alerts UI defaults to port 4201. If you are already using port 4201 for another purpose, you must change the default port for the Alerts UI to another port number.

To display the Alerts user interface, complete the following steps:

1. Display the **Ambari** user interface.
2. In the Services pane, select **Metron**.
3. From the **Quick Links** menu, choose **Alerts UI**.



Note

There is no login module for the Alerts UI.

5.2. Viewing Alerts

The Alerts user interface defaults to displaying the Alerts table when first opened. You can modify the alerts displayed in the Alerts table to help identify issues.

Table 5.1. Alerts UI Tools and Purposes

Tools	Description	More Information
Alerts table	The Alerts table displays the alerts generated by the HCP framework.	Using the Alerts Table [15]

Tools	Description	More Information
	The Alerts UI polls for alerts and refreshes the Alerts table at an interval that you can configure.	
Searches field	You can search for alerts using the search bar above the Alerts table. The search tool follows the Lucene syntax which supports a rich query language.	Searching Alerts [20]
Filters	The Alerts UI currently provides five filters that you can apply to alerts. You can use these filters to refine the list of alerts and collect additional information on the alerts.	Filtering Alerts [21]
Alert status	You can change the status of or dismiss an alert.	Managing Alert Status [23]
Group By	You can group alerts so you can apply filters, status, etc. on multiple alerts at a time.	Grouping Alerts [25]
Meta Alerts	The meta alert feature enables you to create a system entity that contains a collection of filtered alerts.	Creating a Meta Alert

5.2.1. Using the Alerts Table

The Alerts table displays the alerts generated by the HCP framework. The Alerts UI polls for alerts and refreshes the Alerts table at an interval that you can configure (see [Customizing the Alerts Table](#)). This polling is paused whenever you open any configuration panels or use the **Searches** field.

By default, the alerts table shows the recent alerts at the top. For example, alerts are sorted descending on timestamp. For information on modifying these configurations, see [Customizing the Alerts Table](#).

The Alerts table also provides the threat intelligence score for each alert. Next to the score is a bar that indicates the severity of the score:

Red A score of 69 or higher

Orange A score between 39 and 69

Yellow A score below 39

Alerts (265379)

Filters: enrichment_country: 3, host: 10, ip_src_addr: 10, ip_src_type: 9, source_type: 2

Group By: 2 source_type, 10 ip_src_addr, 10 host, 3 enrichment_country, 9 ip_src_addr

Score	ID	timestamp	source_type	ip_src_addr	enrichment_country	ip_dst_addr	host	alert_status
-	829e39f6-6_a51eb0bdee	2017-08-31 11:47:55	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	ESCALATE
-	de534302-b_2f4fc0994d	2017-08-31 11:47:55	bro	192.168.66.1		192.168.66.121	node1	NEW
-	06af55c9-3_34dc1f9523	2017-08-31 11:47:55	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	DISMISS
-	13c7509a-0_a99936968b	2017-08-31 11:47:55	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	c4956536-0_2fcb707e67	2017-08-31 11:47:55	bro	192.168.66.1		192.168.66.121	node1	NEW
-	8169742b-e_a51f275699	2017-08-31 11:47:55	bro	192.168.138.158	FR	62.75.195.236	62.75.195.236	NEW
-	2270e69e-6_ac20f9146	2017-08-31 11:47:55	bro	192.168.138.158	US	72.34.49.86	comarksecurity.com	NEW
-	1e31b227-0_2132c0ea69	2017-08-31 11:47:55	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	b8a579fe-f_ae8b0b0256	2017-08-31 11:47:55	bro	192.168.66.1		224.0.0.251		NEW
-	395618f-c_d73c95a056	2017-08-31 11:47:55	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	b73646a8-6_219a89109d	2017-08-31 11:47:38	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	17c2b850-c_d60eaf70a8	2017-08-31 11:47:38	bro	192.168.66.1		224.0.0.251		NEW
-	76923098-4_9c39303d42	2017-08-31 11:47:38	bro	192.168.138.158	FR	62.75.195.236	62.75.195.236	NEW
-	9d96d71-1_999e074e6	2017-08-31 11:47:38	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	cb0cd34a-b_32baf74b9a	2017-08-31 11:47:38	bro	192.168.66.1		224.0.0.251		NEW
-	20390982-d_461d3149e	2017-08-31 11:47:38	bro	192.168.138.158		192.168.138.2		NEW
-	8592e071-4_f58e0caf19	2017-08-31 11:47:38	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	5d7a29b4-7_aead99226f	2017-08-31 11:47:38	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	c0a5cbda-8_11be5a243a	2017-08-31 11:47:38	bro	192.168.66.1		192.168.66.121	node1	NEW
-	b60c97c2-0_635cb58726	2017-08-31 11:47:38	bro	192.168.138.158	FR	62.75.195.236	62.75.195.236	NEW
-	8f05f0cd-5_bcb004462	2017-08-31 11:47:26	bro	192.168.138.158	FR	62.75.195.236	r03af02.c3...rogams.in	NEW
-	4e31b7fe-9_63d3f530d8	2017-08-31 11:47:26	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	320afe0d-9_cf860b7c14	2017-08-31 11:47:26	bro	192.168.66.1		224.0.0.251		NEW
-	767203af-c_acb8efc0e9	2017-08-31 11:47:26	bro	192.168.138.158	RJ	95.163.121.204	7oganzwzwn...paysun.com	NEW
-	6ea1b462-6_d4d5204765	2017-08-31 11:47:26	bro	192.168.66.1		192.168.66.121	node1	NEW

1 - 25 of 265379



Note

The Alerts table shows only the columns that can be displayed in its horizontal view port. Any additional columns are displayed using ellipses. These hidden columns are visible if the UI is opened in a display that has a wider view port.

5.2.1.1. Customizing the Alerts Table

You can configure both the table columns and the table row settings in the Alerts table.



Note

The Alerts UI polling is paused whenever you open a configuration panel, such as Configure Table or Settings.

Configure Table Columns

Use this feature to customize the type of information you display in the Alerts table. You can modify the information that shows in each column, the title of the column, and the order in which the columns are displayed.

To modify the column information in the Alerts table, complete the following steps:

1. Click  (gear icon).


The Alerts UI displays the Configure Table that lists all the columns available across all the valid search indexes.

Figure 5.1. Alerts Configure Table

Field	Short Name	Type
<input checked="" type="checkbox"/> Score		STRING
<input type="checkbox"/> AA	<input type="text" value="rename"/>	BOOLEAN
<input type="checkbox"/> adapter:geoadapter:begin:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> adapter:geoadapter:end:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> adapter:hostfromjsonlistadapter:begin:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> adapter:hostfromjsonlistadapter:end:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> adapter:threatinteladapter:begin:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> adapter:threatinteladapter:end:ts	<input type="text" value="rename"/>	STRING
<input checked="" type="checkbox"/> id	<input type="text" value="rename"/>	STRING
<input checked="" type="checkbox"/> timestamp	<input type="text" value="rename"/>	DATE
<input checked="" type="checkbox"/> source.type	<input type="text" value="rename"/>	STRING
<input checked="" type="checkbox"/> ip_src_addr	<input type="text" value="rename"/>	IP
<input checked="" type="checkbox"/> enrichments:geo:ip_dst_addr:country	<input type="text" value="rename"/>	STRING
<input checked="" type="checkbox"/> ip_dst_addr	<input type="text" value="rename"/>	IP
<input checked="" type="checkbox"/> host	<input type="text" value="rename"/>	STRING
<input checked="" type="checkbox"/> alert_status	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> answers	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> bro_timestamp	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> comments	<input type="text" value="rename"/>	OTHER
<input type="checkbox"/> dgmlen	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> enrichment:joinbolt:join:ts	<input type="text" value="rename"/>	STRING
<input type="checkbox"/> enrichments:geo:ip_dst_addr:city	<input type="text" value="rename"/>	STRING

2. Select the fields you want to display and unselect the fields you do not want to display.
3. You can rename the column titles by entering a new name in the **Short Name** column.
For example, 'enrichments:geo:ip_dst_addr:country' can be renamed to 'Dst Country'.

This is just for display convenience and the changes are not propagated to any system in HCP.

4. You can also configure the order in which the selected columns will appear in the table by using the arrow icons.
5. Click **Save** to save your changes and dismiss the **Configure Table** panel.
6. You can pause the Alerts UI polling by clicking the  (pause button).

Configure Table Row Settings

Use this feature to modify the appearance of the Alerts table and the refresh rate. To modify the Alerts table row settings, complete the following steps:



Note

Settings are not saved in a saved search. These settings are global.


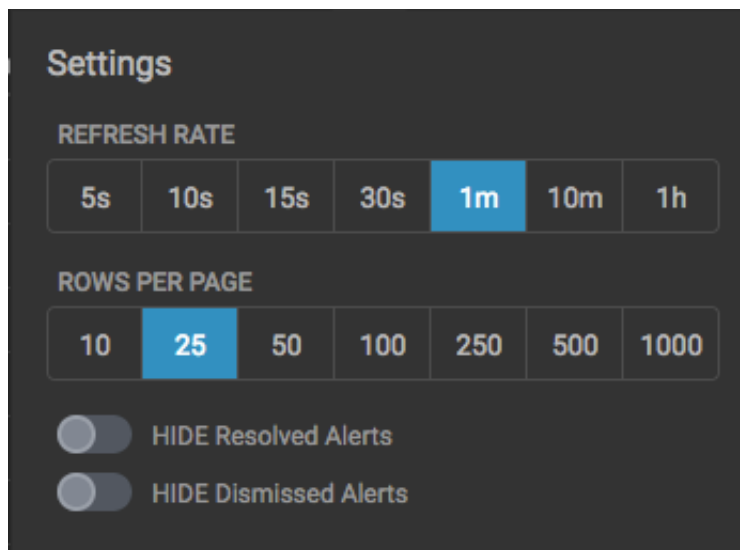
1. Click the  (slides icon) at the top of the table to display the Settings dialog box.

Figure 5.2. Alerts Settings Panel



2. To modify the rate at which the Alerts table is refreshed with new alert information, choose a value under **Refresh Rate**.
3. To modify the number of rows displayed in the Alerts table, choose a value under **Rows Per Page**.



Note

The number of rows that are visible in the Alerts table is restricted by the size of your browser window.

4. To hide resolved alerts or dismissed alerts, click the slide button next to the appropriate action.

HIDE Resolved Alerts and HIDE Dismissed Alerts are non-functional features in this release.

5.2.1.2. Displaying Additional Alerts Information

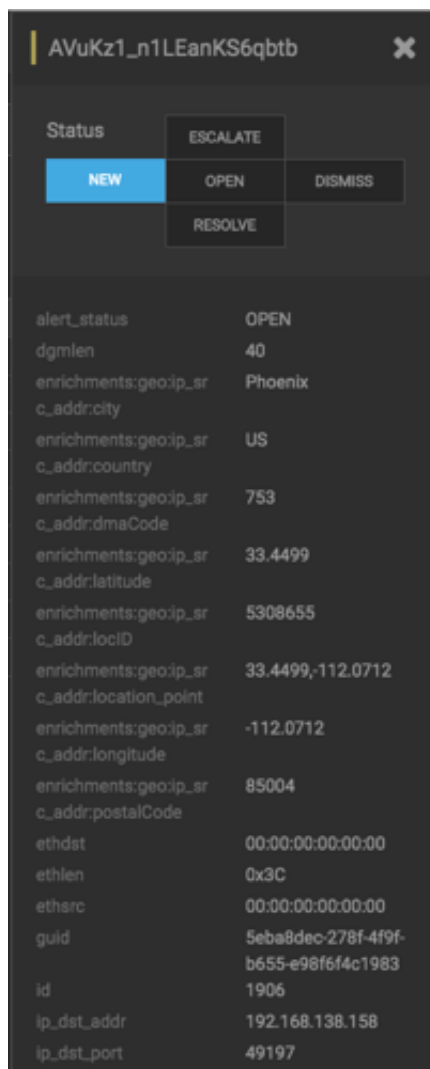
In addition to displaying alert information in the Alerts table, you can display all the information about the alert in Elasticsearch in a separate panel.

To display this additional information about the alert, complete the following steps:

1. Select an alert by clicking on empty space in the alert row.

The Alerts UI displays a panel listing all available data in Elasticsearch about the alert.

Figure 5.3. Alerts Information Panel



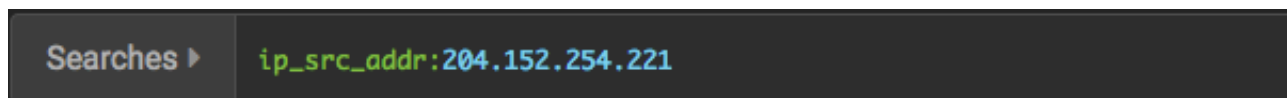
2. The Status states at the top of the panel display the current status of the alert.

5.2.2. Searching Alerts

You can search for alerts using the search bar above the Alerts table. The search tool follows the Lucene syntax which supports a rich query language. For more information, see [Apache Lucene - Query Parser Syntax](#).

1. To search on an item that is displayed in the Alerts table, simply click on the item and it will display in the **Searches** field.

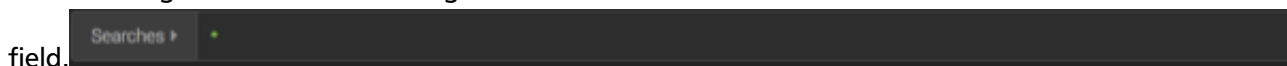
Figure 5.4. Searches Field



2. You can also directly type in the **Searches** field to enter search criteria.

For example, you can enter `source:type:snort`.

3. To remove an item in the **Searches** field, mouse over the information in the **Searches** field until an **x** appears at the end of the text. Click on the **x** to remove the search filter and the operator following or preceding it.
4. To clear the entire **Searches** field, click the **x** at the end of the field.
5. You can specify the time range of your search by using the time range selector on the far right of the **Searches**

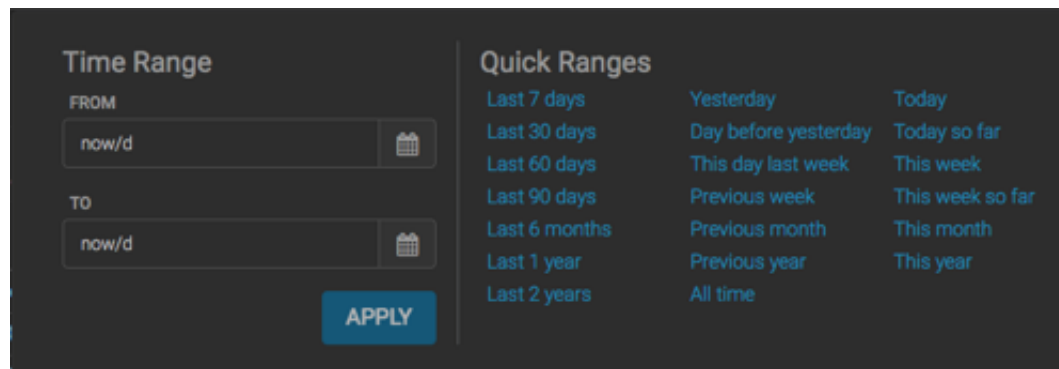


Note

The time-range selector is not available if you put a timestamp in the **Searches** field.

The time-range button defaults to **All time** which displays all alerts corresponding to the Searches parameters. To customize the time range, click the time-range drop-down menu and select one of the following:

- | | |
|--------------|--|
| Time Range | Enables you to choose the start and end dates and times for your search. |
| Quick Ranges | Provides a list of pre-specified time ranges that you can choose. |

Figure 5.5. Time Selector Dialog Box

After you make your choice, the time-selector label will reflect your selection.



5.2.3. Filtering Alerts

The Alerts UI currently provides five filters that you can apply to alerts. You can use these filters to refine the list of alerts and collect additional information on the alerts. These filters are listed in the **Filters** panel on the left of the **Alerts** window:

- source.type
- ip_src_addr
- ip_dist_addr
- host
- enrichments:geo_dst_addr:country

To apply filters to alerts, complete the following steps:

1. Click one of the filters in the **Filters** panel on the left of the window.

The Filter expands to list all of the facet values contained in the filter. For example, in the following figure, the **enrichments:geo_dst_addr:country** filter contain the countries Russia, France, and USA.

The screenshot shows the Metron Alerts interface. At the top, there's a search bar and a status bar with 'ESCALATE' and 'NEW', 'OPEN', 'DISMISS', 'RESOLVE' buttons. The main area is titled 'Alerts (147925)'. On the left, there are filters for 'enrichm_country' (RU, US) and 'host' (10). The main table is grouped by 'source_type' (2), 'ip_dst_addr' (10), and 'host' (10). The table columns are: Score, id, timestamp, source_type, ip_src_addr, enrichm_country, and ip_dst_addr. The table shows a list of alerts with their respective scores, IDs, timestamps, source types, source IP addresses, enriched countries, and destination IP addresses.

Score	id	timestamp	source_type	ip_src_addr	enrichm_country	ip_dst_addr
-	06af35c9-3...34dc1f9523	2017-08-31 11:47:55	bro	192.168.138.158	RU	95.163.121.20
-	829ed3f6-6...a31eb0bdee	2017-08-31 11:47:55	bro	192.168.138.158	RU	95.163.121.20
-	de53d302-b...2fafcc094d	2017-08-31 11:47:55	bro	192.168.66.1		192.168.66.12
10	f8f44a88-8...c938c011be6	2017-08-30 08:07:59	snort	192.168.66.1		192.168.66.12
-	10a83fa3-8...41995e6201	2017-08-30 12:43:58	bro	192.168.138.158	FR	62.75.195.236
-	da000ec0-4...ef0ec68715	2017-08-30 12:43:58	bro	192.168.138.158	FR	62.75.195.236
-	f63236fe-3...2208f9994	2017-08-30 12:43:58	bro	192.168.66.1		192.168.66.12
-	9af44ca9-c...a991447655	2017-08-30 12:43:58	bro	192.168.138.158	US	204.152.254.2
-	0bf980d6-7...fb113e8bce	2017-08-30 12:43:58	bro	192.168.138.158	FR	62.75.195.236
-	576f8cb-a...7420f2bac2	2017-08-30 12:43:58	bro	192.168.66.1		192.168.66.12
-	3964e90f-b...aeb9439c39	2017-08-30 12:43:58	bro	192.168.138.158	RU	95.163.121.20
-	90403c32-0...2ac148dbd8	2017-08-30 12:43:58	bro	192.168.66.1		224.0.0.251
-	88950cee-5...359faa0a78	2017-08-30 12:43:58	bro	192.168.138.158	RU	95.163.121.20
-	60bf112-c...c2619a8a36	2017-08-30 12:43:58	bro	192.168.138.158		192.168.138.2
-	327e885e-1...3c21a1a799	2017-08-30 12:44:06	bro	192.168.138.158	RU	95.163.121.20
-	9ebf85f9-5...5ec9069f53	2017-08-30 12:44:06	bro	192.168.138.158	FR	62.75.195.236
-	ca0d424a-6...40148306a	2017-08-30 12:44:06	bro	192.168.138.158	FR	62.75.195.236
-	778b607e-2...f1365e19e0	2017-08-30 12:44:06	bro	192.168.138.158	US	72.34.49.86
-	3b22480a-a...b8a14515cb	2017-08-30 12:44:06	bro	192.168.138.158		192.168.138.2
-	e56c2754-b...737a2ecb5e	2017-08-30 12:44:06	bro	192.168.66.1		224.0.0.251



Note


The UI displays the number of alerts corresponding to each facet next to the facet.

- You can continue to apply filters to the alerts displayed in the **Alerts** window to further refine the alerts list.

As you select filters and facets, they are displayed in the **Searches** field.

For example, in the following figure, we've applied the `source.type` filter with the `bro` facet and then the `ip_dst_addr` filter with the IP address `95.163.121.204`.

The screenshot shows the Metron Alerts interface. At the top, there's a search bar with the query: `source: type: bro AND ip_dst_addr: 95.163.121.204`. Below the search bar, there are tabs for 'Alerts (68863)'. On the left, there are filter panels for 'enrichen...country', 'host', 'ip_dst_addr', 'ip_src_addr', and 'source: type'. The main area displays a table of alerts with the following columns: Score #, Id #, timestamp #, source: type #, ip_src_addr #, enrichen...country #, ip_dst_addr #, host #, and alert_status #. The table contains 25 rows of alert data, all with a score of 1 and a status of 'NEW'. The bottom of the screen shows a pagination control: '< 1 - 25 of 68863 >'. In the top right corner, it says 'Logged in as admin - Logout'.

3. To clear filters that have been populated to the **Searches** field, click  (delete icon) at the end of the **Searches** field.

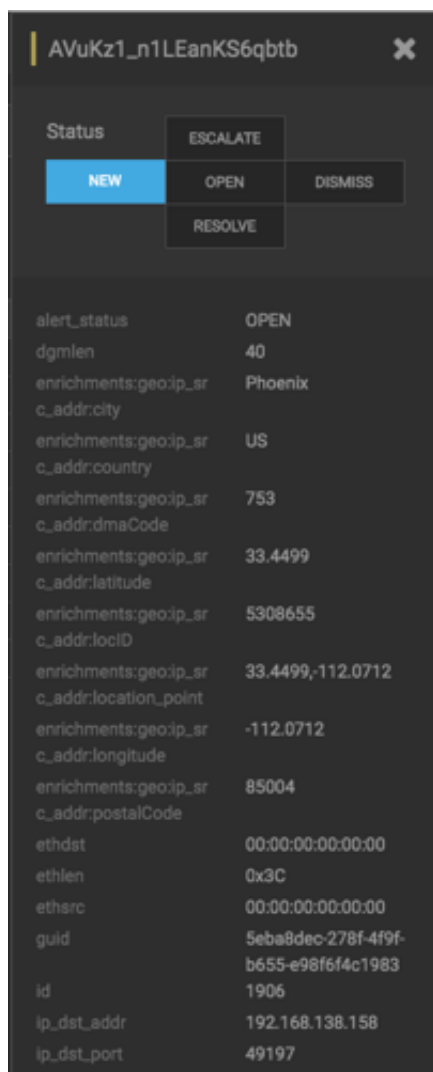
5.2.4. Managing Alert Status

You can manage one or more alerts at a time using the **ACTIONS** menu. To change the status of or dismiss an alert, complete the following steps:

1. Select an alert by clicking on empty space in the alert row.

The Alerts UI displays a panel listing the status of the alert all available data in Elasticsearch about the alert.

Figure 5.6. Alerts Information Panel




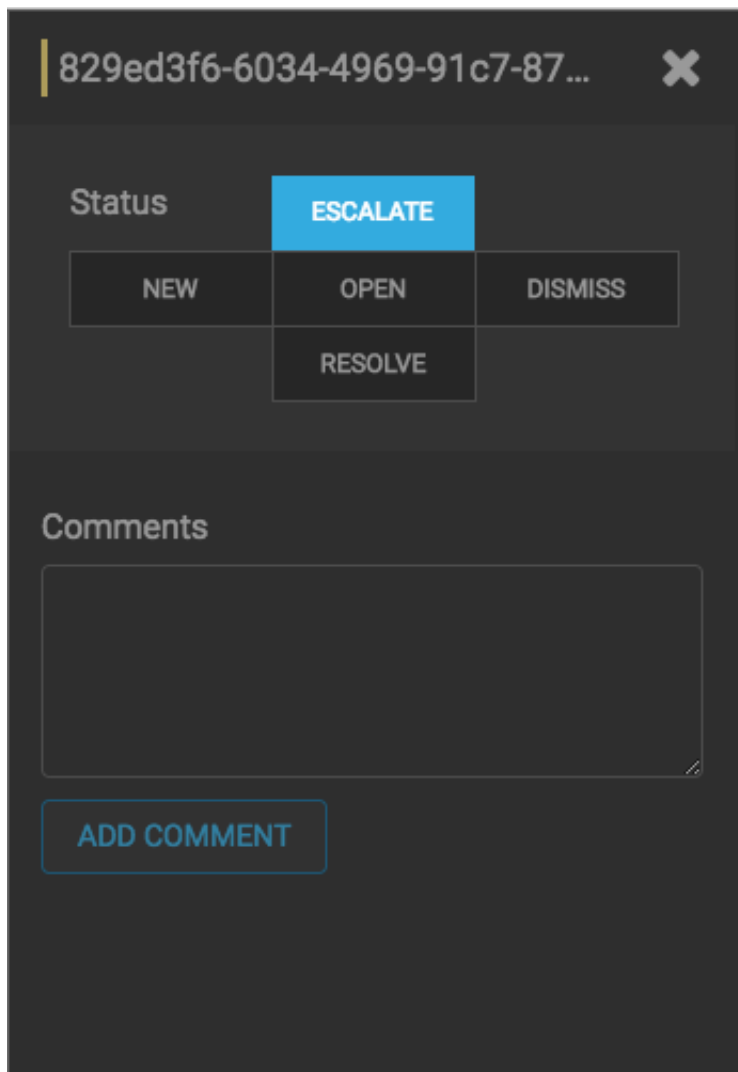
The current alert status is highlighted.




Note

To manage more than one alert at a time, click the check boxes at the end of alert rows, then select the action you want to perform from the ACTIONS menu.

2. Click the new status you want to apply to the alert, then dismiss the panel.
3. You can also add a comment to this action by clicking  (Comment button), entering your comment in the **Comments** field, and clicking **ADD COMMENT**.



The Alerts UI indicates that an alert has one or more comments by displaying  (comment icon) next to the alert status in the **Alerts** window.



Note

You cannot add a comment to an alert contained in a meta alert. You can only add comments to the meta alert.

4. To delete a comment, click the comment to delete, then click the trash can icon.

Click **OK** in the **Confirmation** dialog box.

5.2.5. Grouping Alerts

You can group alerts so you can apply filters, status, etc. to multiple alerts at a time.

The Alerts UI currently provides five group types you can use:

- source.type
- ip_dst_addr
- host
- enrichments:geo_dst_addr:country
- ip_src_addr

To apply a group to your alerts, complete the following steps:

1. Click one of the groups listed by **Group By**.

The **Alerts** table view changes to a tree view listing the values of the groups.

In the following example, the group is `source.type` and the values are Snort and Bro.



Note

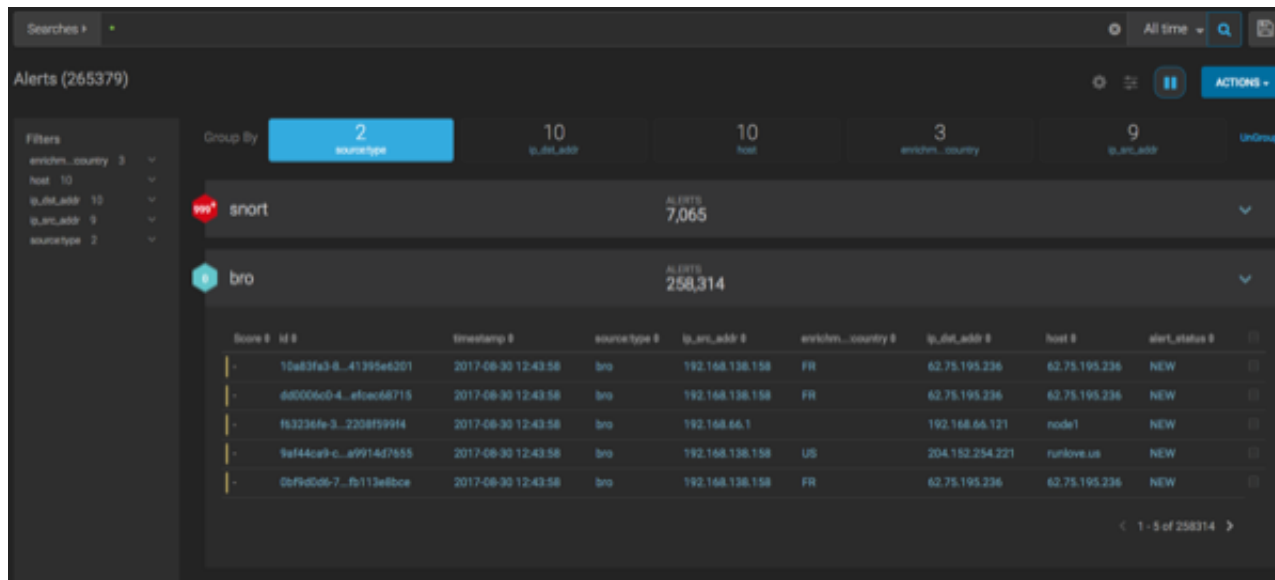
The icon to the left of the value provides the cumulative severity score for all the alerts in the value. If the score exceeds 999, then the value displays as 999+.

The screenshot shows the Apache Metron Alerts interface. At the top, it says "Alerts (265379)". Below this, there is a "Group By" section with a dropdown menu set to "source.type". The interface displays a tree view of alerts grouped by source.type. The groups are "snort" and "bro". The "snort" group has a cumulative severity score of 7,065, and the "bro" group has a cumulative severity score of 258,314. The interface also shows filters on the left and search options at the top.

Group By	Count
source.type	2
ip_dst_addr	10
host	10
enrichments:geo_dst_addr:country	3
ip_src_addr	9

Group	Alerts
snort	7,065
bro	258,314

2. Click one of the values to list the alerts for that value.



3. You can click an alert to add it to the Searches field.



Note

Searches will search through all the groups, not just the group containing the alert.

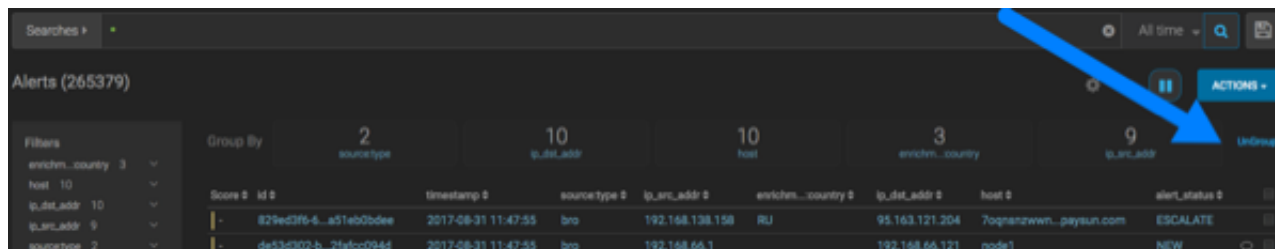
4. All features that are available for the Alerts table are available for the tree view.

For example, if you apply an action, such as Escalate, to an alert, it will apply to all alerts within the group. Similarly, if you search for a parameter, it will search all alerts within the group.

5. You can continue to refine your alerts by applying additional groups.

You can change the order in which the groups are applied to the alerts by clicking and dragging the groups on the **Groups By** line.

6. To ungroup your alerts and return to the Alerts window, click Ungroup which is located on the far right of the list of groups.



5.2.6. Creating a Meta Alert

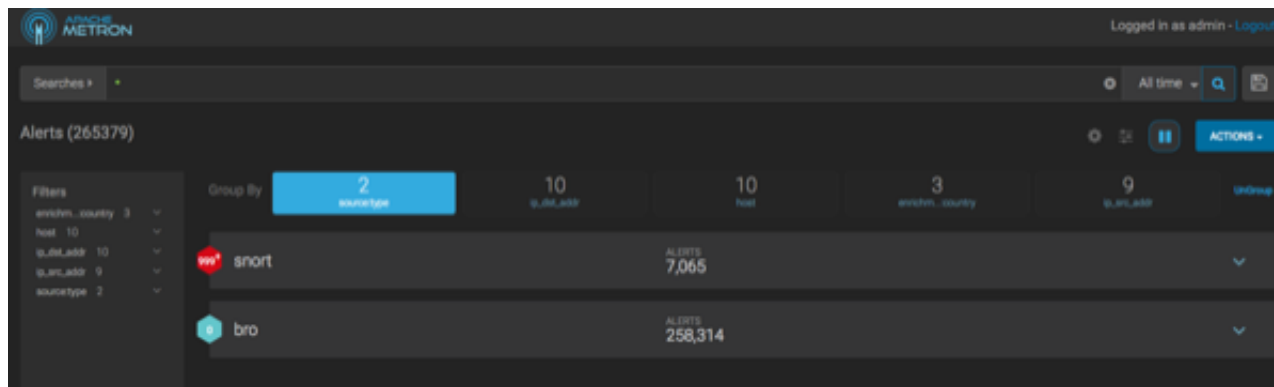
The meta alert feature enables you to create a save a group of filtered alerts. Like the group feature, you can group filtered alerts that pertain to an incident. However, with

meta alert, you can save your grouping, creating a system entity, to view it later. Also, when you filter alerts, if a relevant alert is contained in a meta alert, the entire meta alert will be included in the filter results.


To create a meta alert, complete the following steps:

1. Click one of the groups listed by **Group By**.

The **Alerts** table view changes to a tree view listing the values of the groups.



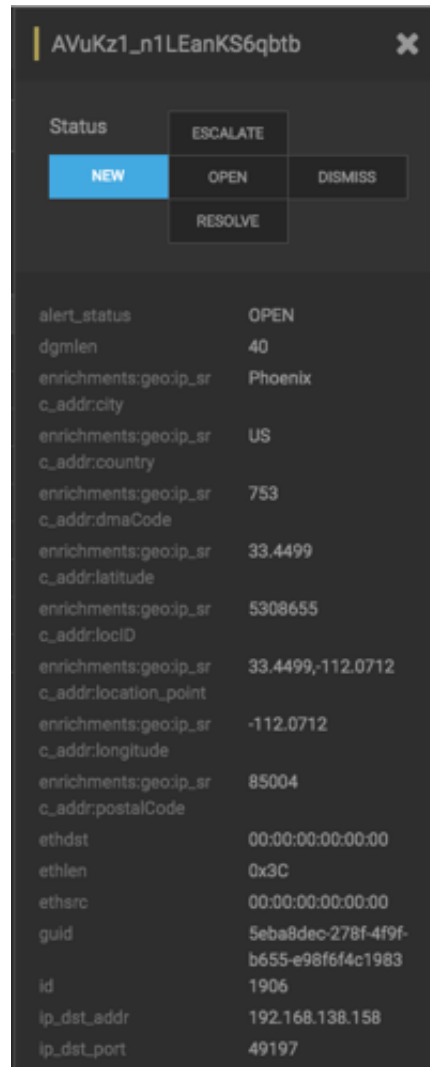
2. Use the **Search** and **GroupBy** options to create one or more groups containing alerts on which you want to focus.

3. When you have selected a group of alerts that you want to focus on, click  (meta alert icon), then confirm that you wish to create a meta alert with the selected alerts.

The meta alert disappears from the tree view. You can still see the meta alert in the alerts table view.


4. You can rename your meta alert by completing the following steps:
 - a. Display the Alerts UI display panel by clicking on empty space in the meta alert row.

Figure 5.7. Alerts Information Panel



- b. Click the current meta alert name at the top of the panel and enter your new meta alert name.
- c. Dismiss the panel by clicking the X in the upper right corner of the panel.

5.3. Saving Your Searches

1. To save a search, click the  (save button) next to the **Searches** field.
2. When prompted, enter a name for the saved search parameters, then click **Save**.

This will save both the search parameters and the column configurations.

5.4. Viewing Your Recent and Saved Searches

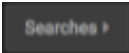
You can view both your recent searches and saved searches in the Alerts UI.



Note

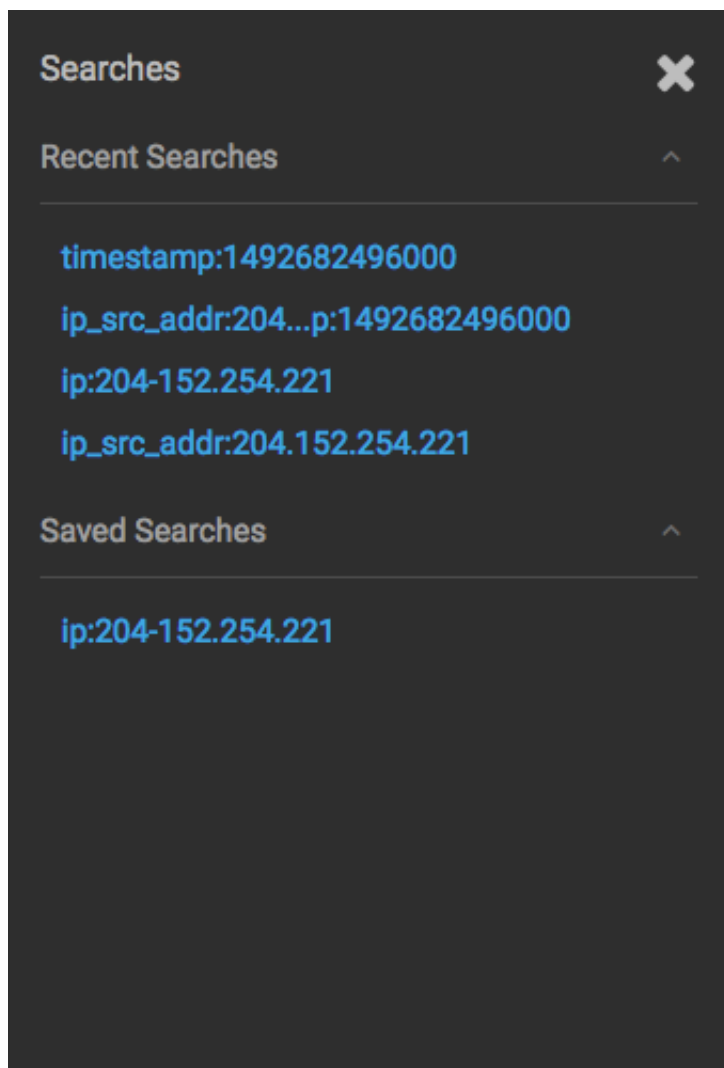
The Alerts UI polling is paused whenever you open the Searches panel.

To view these saved searches, complete the following steps:

1. Click the  button to the left of the **Searches** field.

The Alerts UI displays the Searches panel.

Figure 5.8. Searches Panel



The **Searches** panel lists two types of searches:

Recent Searches

This is a list of your most recent searches.

To display the saved search, simply click on the search name.

The Alerts UI saves a maximum of ten of your most recent searches.

Saved Searches

This is a list of your saved searches.

To display the saved search, simply click on the search name.

You can delete any of these saved searches by clicking the trash can icon that becomes visible when you mouse over each saved search.

6. PCAP

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 \[32\]](#)
- [Processing pcap Data \[32\]](#)
- [Viewing pcap Data \[34\]](#)
- [Filtering pcap Data \[34\]](#)
- [Porting pcap Data to Another Application \[38\]](#)

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#
```

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
```

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
```

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 \[34\]](#)
- [Using REST API to Query pcap Data \[37\]](#)

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)

Filtering can be done both by the packet header as well as via a binary regular expression which can be run on the packet payload itself. This filter can be specified via:

- The `-pf` or `--packet_filter` options for the fixed query filter
- The `BYTEARRAY_MATCHER(pattern, data)` Stellar function. The first argument is the regex pattern and the second argument is the data. The packet data will be exposed via the `packet` variable in Stellar.

Both the fixed filter option tool and query filter option tool 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.
-pf, --packet_filter <arg> Packet filter regex
-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
```

To search for every packet that has an `ip_dst_port` of 8080 and contains the text "persist", run:

```
$METRON_HOME/bin/pcap_query.sh fixed \  
  --ip_dst_port 8080 \  
  --packet_filter \  
  "\`persist\`" \  
  -st "20170425" \  
  -df "yyyyMMdd"
```

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.

The Query filter's `--query` argument specifies the Stellar expression to execute on each packet. To interact with the packet, a few variables are exposed:

- `packet` : The packet data (a `byte[]`)
- `ip_src_addr` : The source address for the packet (a `String`)
- `ip_src_port` : The source port for the packet (an `Integer`)
- `ip_dst_addr` : The destination address for the packet (a `String`)
- `ip_dst_port` : The destination port for the packet (an `Integer`)
- `BYTEARRAY_MATCHER` : The first argument is the regex pattern and the second argument is the data. The packet data will be exposed via the `packet` variable in Stellar.

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
```

To search for every packet that has an `ip_dst_port` of 8080 and contains the text "persist", run:

```
$METRON_HOME/bin/pcap_query.sh query \  
  --query "ip_dst_port == 8080 &&  
    BYTEARRAY_MATCHER('\`persist\`', packet)" \  
  -st "20170425" \  
  -df "yyyyMMdd"
```

You can also do proper binary regexes that look for packets containing the text "persist" and the 2 byte sequence 0x1F909 (in hex):

```
$METRON_HOME/bin/pcap_query.sh query \  
  --query "BYTEARRAY_MATCHER('1F90', packet) &&  
    BYTEARRAY_MATCHER('\`persist\`', packet)" \  
  -st "20170425" \  
  -df "yyyyMMdd"
```

Other examples:

```
$METRON_HOME/bin/pcap_query.sh query \  
  -st "1466136000000" \  
  --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']"
```

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:

- srcIp
- srcPort
- dstIp
- 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

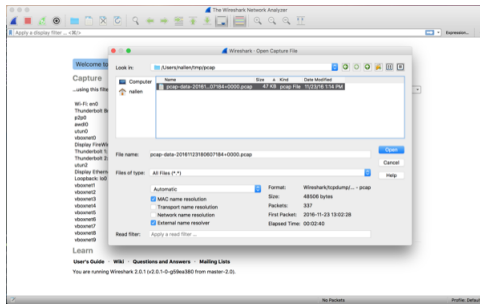
6.5. Porting pcap Data to Another Application

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:

