ThreatQuotient

ThreatQ Splunk Implementation Guide

Version 2.0.0

Thursday, February 20, 2020

ThreatQuotient

11400 Commerce Park Dr., Suite 200

Reston, VA 20191

Support

Email: support@threatq.com

Web: support.threatq.com

Phone: 703.574.9893

Warning and Disclaimer

ThreatQuotient, Inc. provides this document "as is", without representation or warranty of any kind, express or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of such information contained herein. ThreatQuotient, Inc. does not assume responsibility for the use or inability to use the software product as a result of providing this information.

Contents

Versioning

App Version

- TA-threatquotient-add-on: 2.0.0

- ThreatQAppforSplunk: 2.0.0

Supported Splunk Version

These apps have been tested with Splunk versions Splunk 7.2.x, Splunk 7.3.x, Splunk 8.X (Python 2), Splunk 8.X (Python 3).

Supported ThreatQuotient Version

These apps require ThreatQ version 4.16.0 or higher. They will NOT work with a ThreatQ version prior to 4.16.0.

Features

The ThreatQuotient App for Splunk provides the following capabilities:

Distributed Deployment

The solution is packaged as two separate Splunk packages:

- **ThreatQuotient Add-on for Splunk:** Deployed on Splunk heavy forwarder and search head.

- **ThreatQuotient App for Splunk:** Deployed on Splunk search head.

Support for Splunk's Common Information Model (CIM) and Enterprise Security (ES)

- **CIM Support**: For users who map third party data (firewall events, logs for example) to Splunk's data models in CIM. This App provides optimized performance by leveraging those data models.

- **ES Support:** Indicator data exported from ThreatQ is mapped to lookup tables native to Splunk ES. Threat Intelligence support for Enterprise Security is provided using its REST APIs.

Export Indicators from ThreatQ using Score and Status Filters

- **Score Filter:** You can choose to export indicators with scores greater than or equal to the value configured in the score filter.

- **Status Filter:** You can choose to export indicators with statuses matching the ones configured in the status filter.

Detect Sightings and Return to ThreatQ

- **Detect Sightings:** Indicators from ThreatQ are matched against raw events in Splunk looking for evidence of sightings.

- **Report Sightings:** Sightings are reported back to ThreatQ as events that contain the most up to date information.

Contextualize ThreatQ Data

All data exported from ThreatQ is highly contextualized for Splunk. Context provided for exported indicators includes:

- Indicator sources

- Indicator adversaries

- Indicator attributes

- Indicator status, score and type

Workflow Actions in Splunk to Interact with ThreatQ Data

> Workflow actions are only available for fields that are configured to be extracted. Additional fields can be configured for extraction by clicking **Event Actions -> Extract Fields**

This App provides the following workflow actions to an analyst to interact with ThreatQ:

- Add Indicator to ThreatQ

  - The user provides indicator type, status and source

- Whitelist an indicator in ThreatQ

- Look up an indicator in ThreatQ

  - Additional context is fetched if this indicator exists in ThreatQ

- Mark an indicator **False Positive** in ThreatQ

- Mark an indicator **True Positive** in ThreatQ

Dashboard for Visualization

The dashboard provides a rich set of real time updated widgets and tables to summarize information, including (but not limited to):

- Total exported indicators and sightings filtered by time range, type and score

- Top 10 indicators with sightings

- Top 10 sources and adversaries (due to the context available from ThreatQ) with sightings

- Static tables summarizing indicators and sightings filtered by time range, type and score

Installation

From the main Splunk interface:

1. Click on the **Down** arrow on the Apps menu located in the main navigation bar.

2. Select the **Find More Apps** option.
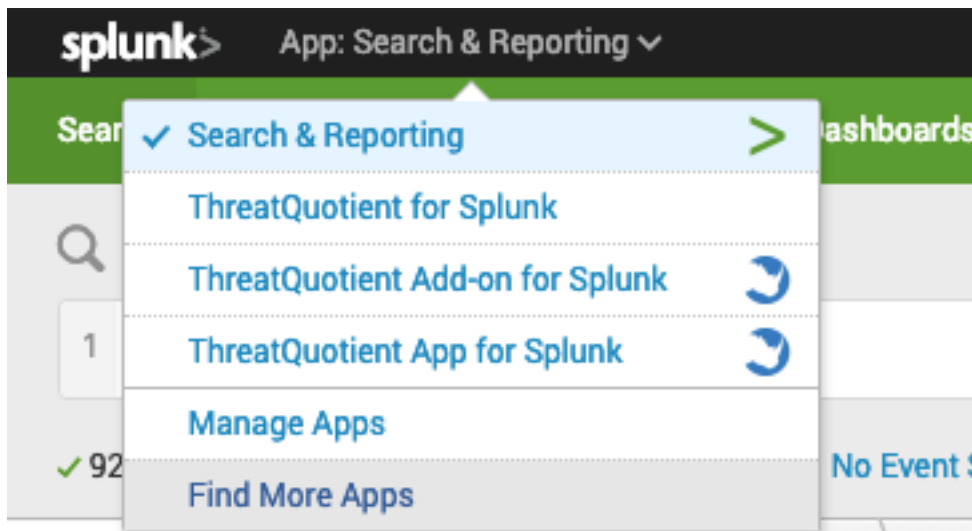
Figure 1: Installation from the Splunk App

3. Search for "ThreatQuotient" and follow the onscreen prompts to install the ThreatQuotient App and ThreatQuotient Add-on.

App Usage

This App can be used in one of three possible modes. Follow the flow diagram below to determine which mode to use.
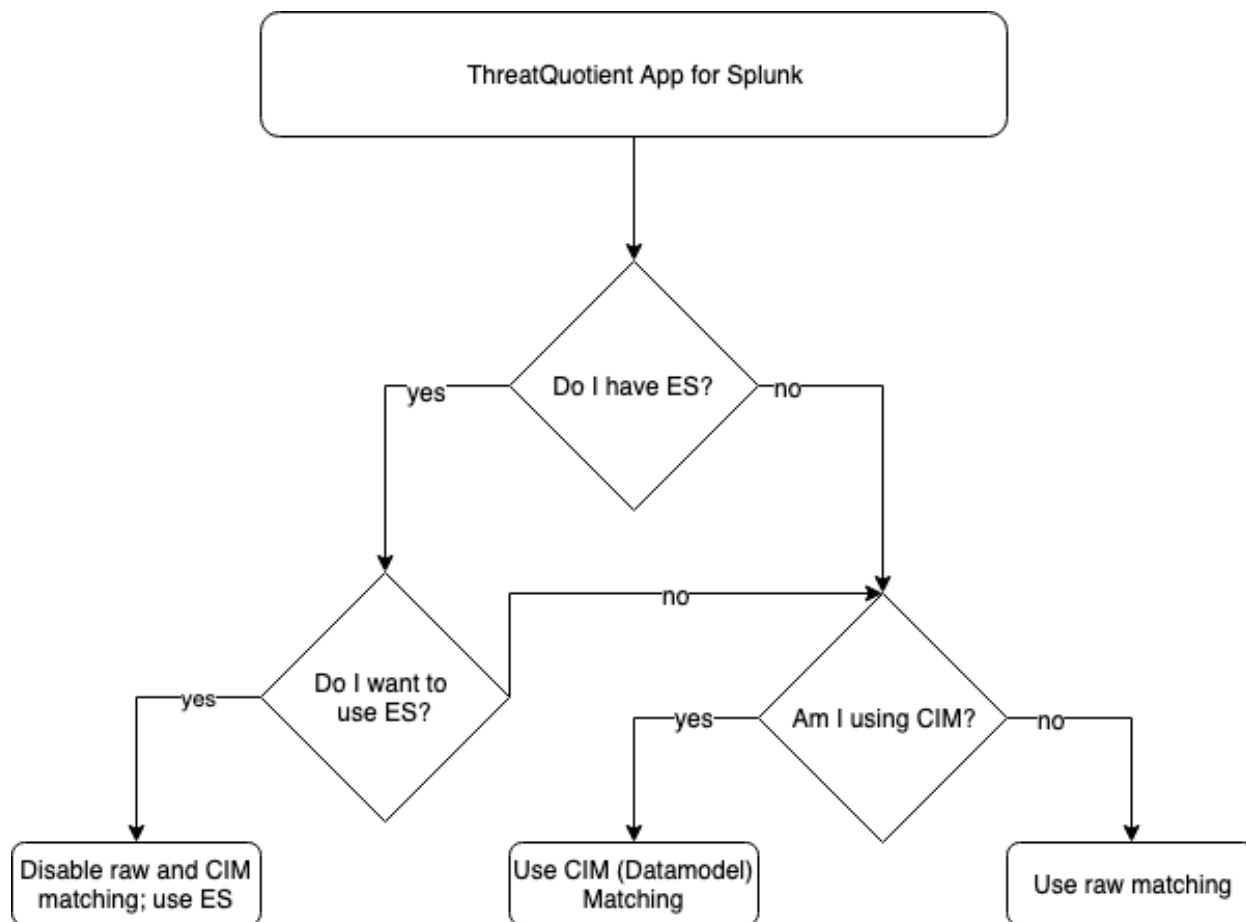
Figure 2: Splunk App Usage Flowchart

- **Raw Matching Mode:** This mode is applicable if you do not have Splunk Enterprise Security (ES) and do not map your traffic to Splunk's CIM. In this mode, the App treats all events as raw binary data and looks for evidence of sightings inside said data using optimized regexes. For the expected performance data, see the tables in Performance.

- **CIM Matching Mode (also sometimes referred to as Datamodel Match Mode):** This mode should be used by users who do not wish to use Enterprise Security (ES), but do map their traffic using Splunk's CIM. In this mode, the app uses the mapping table described in CIM Support to find evidence of sightings and report matches back. This

form of matching is more optimized since the algorithm can now reference well known fields in standard data models instead of looking for matches in the whole binary data.

- **Enterprise Security:** This mode is applicable to the users who want to use Enterprise Security for their end to end workflow, and simply want to get the threat data in ES. In this mode, you do not use any capability of the ThreatQuotient App, and instead rely on Enterprise Security to find and report on evidence of sightings.

Deployment

Two Splunk packages need to be deployed for the App to work.

- **TA-threatquotient-add-on:** This package needs to be deployed both on the Splunk **heavy forwarder** and Splunk **search head**.
  - On the heavy forwarder, the add-on App extracts indicators from the ThreatQ appliance and forwards them to the configured splunk index.
  - On the search head, the App provides support for ThreatQuotient workflow actions in Splunk.
- **ThreatQAppforSplunk:** This package needs to be deployed only on the Splunk **search head**.

There are two ways in which both Apps can be deployed in Splunk:

1. **Standalone Mode:** In this mode, both Apps are deployed and configured on the same machine.

2. **Distributed Mode:** In this mode, deployment is done as described in the picture below.
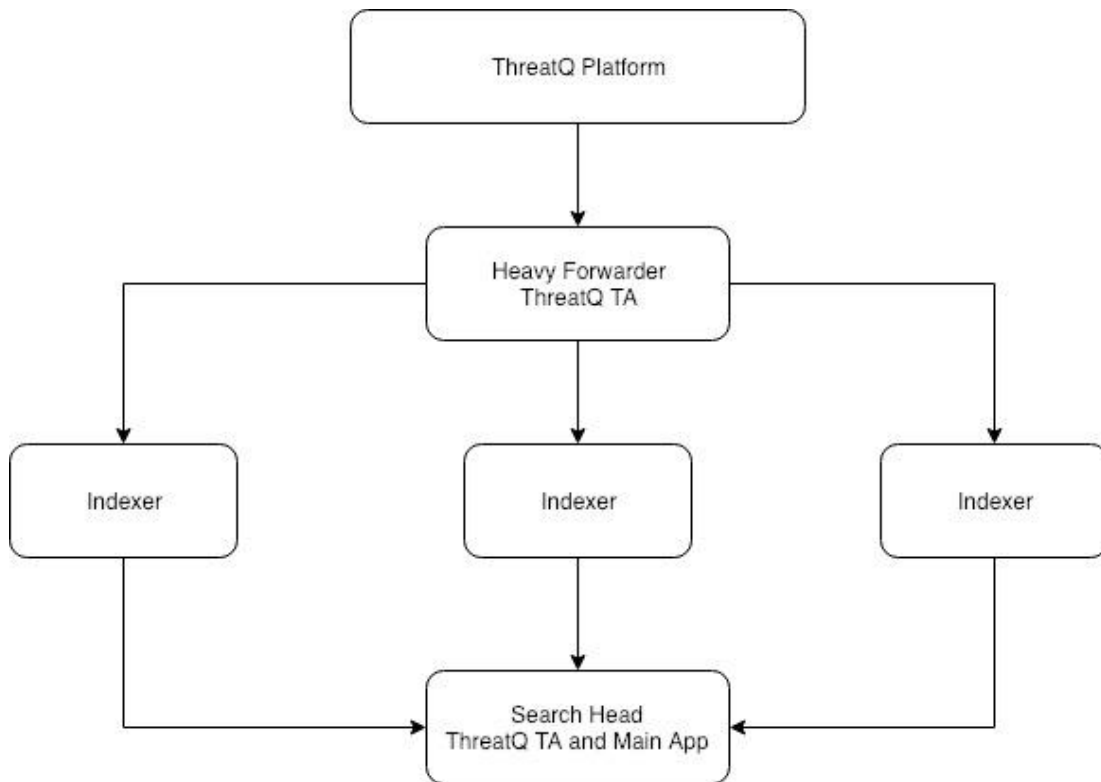
Figure 3: Deployment of Splunk App in Distributed Environment

For a distributed environment with a **cluster of search heads**, you will need to configure the ThreatQuotient Add-on App on the master node, and use the Splunk App deployer to propagate that configuration to all nodes. For the heavy forwarder, it is **not recommended** that you deploy the Add-on app on a cluster, since the data extraction takes place with a custom script, and works the best with a single node.

The table below summarizes the deployment in the distributed Splunk environment:

	Heavy Forwarder	Indexer	Search Head
ThreatQuotient Add-on	Yes • Requires configuration with ThreatQuotient credentials • Requires creating the data collection job	No	Yes • Requires configuration with ThreatQuotient credentials • **Must not be** configured with data collection job
ThreatQuotient App	No	No	Yes • No configuration is required

Table 1: Deployment Matrix for Distributed Environment

> **Advanced Configuration**
> If you desire to configure multiple heavy forwarders for a single ThreatQuotient App - this is not typical since the indicators exported from ThreatQ do not exceed a few thousand at most - you would have to make multiple copies of the default ThreatQ Splunk Export, and use a different Export ID on each heavy forwarder. This way, the ThreatQ server can keep track of incremental indicator changes as seen by each distinct Export.
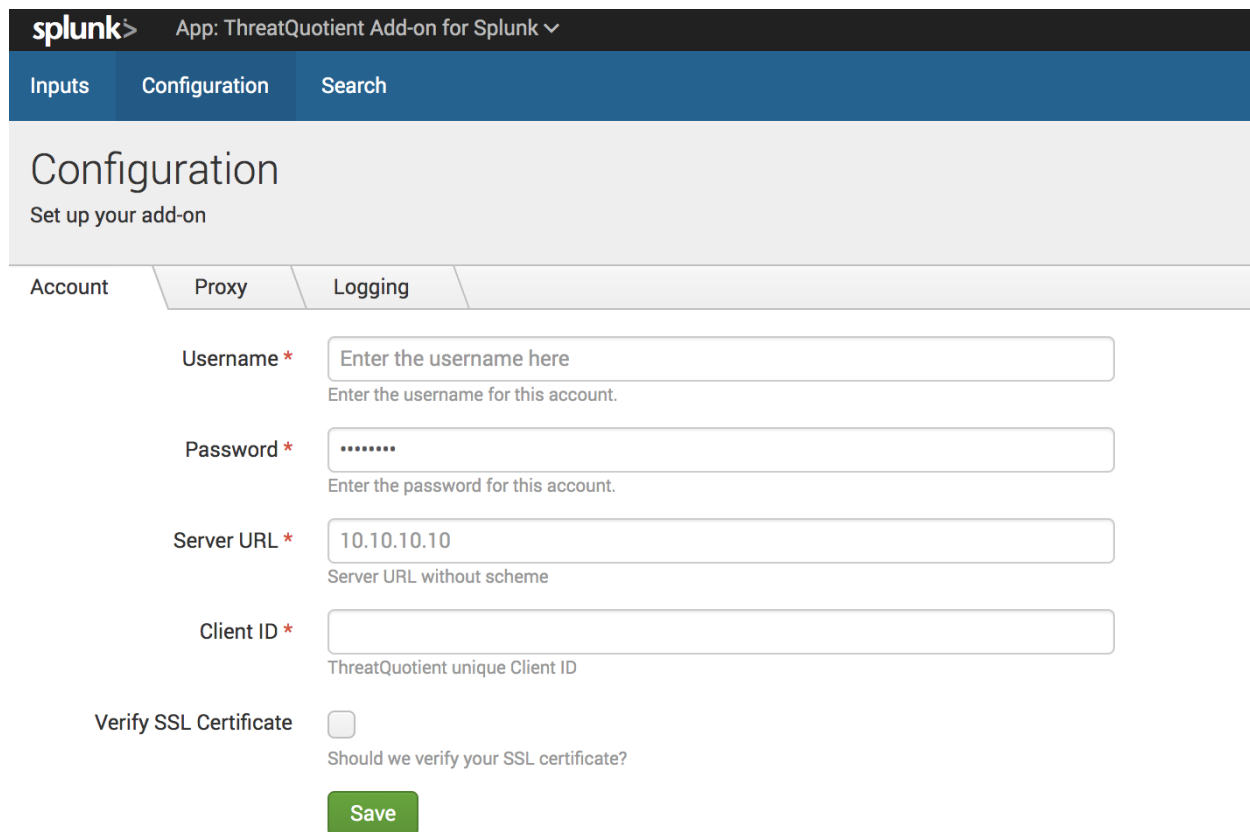
Configuration

ThreatQuotient Add-on

The **ThreatQuotient add-on** is responsible for the following:

Authentication with ThreatQ

On the **Configuration** tab, fields are presented to configure the ThreatQ account authentication as shown below.

Figure 4: Configuration of Authentication Parameters

Upon clicking the **Save** button, you can see the status of the Authentication action. If the ThreatQuotient appliance is down, and/or the authentication parameters are invalid, an error message will be displayed. Unless the appliance is up and the authentication parameters are valid, this App will not work.

Authentication with the Use of Self Signed Certificates in ThreatQ

It is common for many ThreatQuotient users to leverage self signed certificates. If this is the case, you must perform the following additional configuration steps in the Splunk Add-On App.

In ${SPLUNK_HOME}/etc/apps/TA-threatquotient-add-on/default/ta_threatquotient_add_ on_settings.conf, make the following configuration change:

Splunk Search for Listing TQ Indicators

```
[additional_parameters]
verify_cert = false
```

Data Extraction from ThreatQ

On the **Inputs** tab, you can click **Create New Input** to add a data collection job as shown below.

Figure 5: Configuration of Input Data Extraction

ThreatQ instances starting with versions **4.16.0** are shipped with an **Export** that this App uses. Upon the first execution of this job, it results in the export of all indicators. Every subsequent run of this job only results in getting new indicators as well as previously exported indicators that have since changed. Various configuration parameters are described below.

- **Interval:** The frequency of this job. For a faster detection and response, this value can be reduced. Minimum allowed is 60 seconds.

- **Threshold Indicator Score:** Any indicator below this score is not indexed in Splunk. This threshold is very useful to reduce the data being indexed in the **ThreatQuotient App. Default: 8**

- **Indicator Status:** Similar to the score threshold, any indicator not matching the status configured here is not indexed in Splunk. Again, this technique is useful for reducing indexed data. **Default: Active, Whitelisted**

- **Export ID:** Defaulted to **splunk** if you are using the default Splunk export in ThreatQ. If you make a copy of the export, you must configure the ID of the export in this field as seen on the ThreatQ instance.

- **Export Token:** On the ThreatQ instance, find the export named as **Splunk Indicators Export** and click **Connection Settings**. The token is available in the following configuration screen. See the picture below for reference.

		Splunk Indicators Export		api/export/splunk	connection settings		duplicate
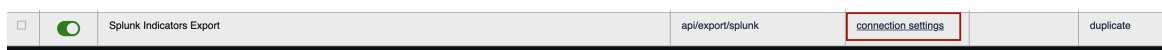
Figure 6: Splunk Export in ThreatQuotient

- **Export Hash:** Defaults to **1**. In the event you want to re-export all indicators from ThreatQ for any reason (such as installing a new Splunk instance), use this configuration. You can configure a different alphanumeric value of length up to 32 and cause exporting all indicators from ThreatQuotient again.

> **Limitations**
>
> - Reducing the set of indicators in Splunk comes at the expense of inability to detect change of scores and/or statuses in indicators. We recommend that users use the "Whitelisted" status in ThreatQ to mark indicators as false positives rather than reducing the indicator score or using custom statuses.
>   - It is possible to configure custom indicator statuses (other than Active and Whitelisted) and use those statuses in the workflow for interaction with the **ThreatQuotient Add-on**.
> - If you want to use advanced filters (such as adversaries, attributes or sources) to export only a subset of indicators from ThreatQuotient to Splunk, there are two ways to do it:
>   - Duplicate the default export, and configure advanced filters. On the Splunk Add-On App, configure scoring filter in such a way that all indicators are accepted (i.e. value of 0).

> - Configure a scoring policy to influence indicator scores on certain adversaries, sources or attributes only. On the Splunk Add-On App, configure the scoring filter to accept only certain scores (i.e. value >= 8 for example).

Exporting a Large Number of Indicators from ThreatQ

It is not recommended that you export an exceptionally large number of indicators from ThreatQ to Splunk. We recommend that at any one time (including the initial load up), users export no more than 500K indicators. If this limit is not observed, you may encounter problems including loading the data to Splunk, and assuming the data was loaded correctly anyway, with the performance of your Splunk deployment itself.

The default export shipped with the ThreatQ appliance does not apply any filters on the indicators to restrict the set of data being exported. However, you may make a copy of this export and specify any additional filters under Special Parameters. An example is shown in the picture below in which a user has configured a filter with score > 5.

Output Format

×

Type of information you would like to export?

Indicators ▼

Output type

text/plain ▼

Special Parameters *(optional)*

indicator.deleted=N&indicator.score>5

Provide URL Parameters to further refine information being exported: **See examples.**

Insert Variable ▼

Output Format Template

```
{* $indicator.id $indicator.value $indicator.score $indicator.type
$indicator.status $indicator.updated_at $indicator.adversaries
$indicator.attributes  $indicator.sources *}
[
{foreach $data as $indicator}
{$indicator|json_encode}{if !$indicator.last},{/if}
{/foreach}
]
```

Save Settings   Cancel
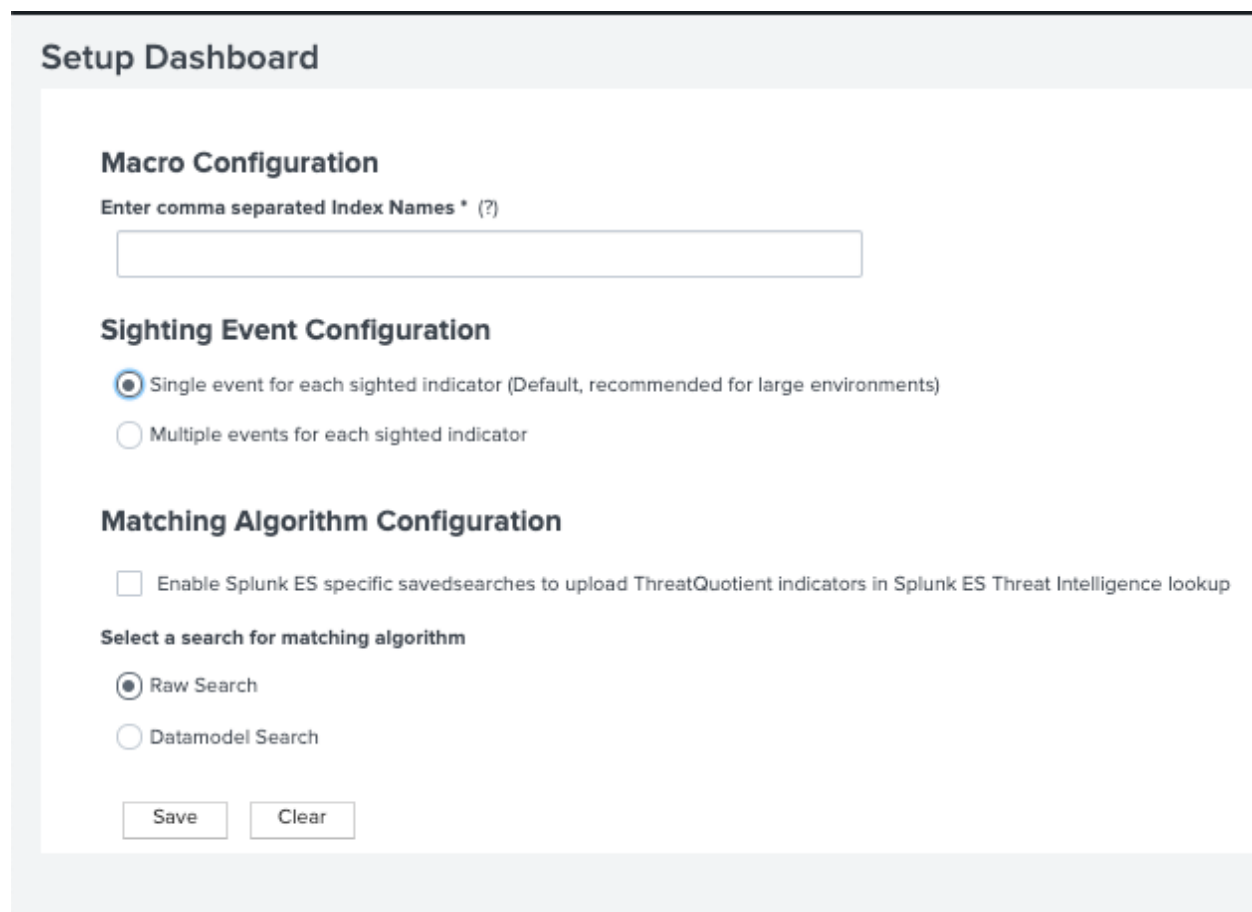
Figure 7: Example of Filters in Splunk Export

Data Loading in Splunk

As shown in Figure 7, the **Index** parameter allows you to map the data extracted from a job in a predetermined Splunk index. You can create multiple jobs and map them to different Splunk indexes as desired.

ThreatQuotient App

The Threatquotient App allows you to select one the three modes of operation described in [App Usage](#). The configuration is available on the Splunk's **setup** page. Navigate to all Apps, locate the **Threatquotient App for Splunk** and click on **setup**. The image below displays an example configuration.

Figure 8: ThreatQuotient App for Splunk Configuration

The following rules apply for selection.

Field	Description
Macro Configuration	ThreatQ indicators will be matched against the events from the selected indexes.
Sighting Event Configuration	Configuration option for event creation in ThreatQ for sighted indicators.
Matching Algorithm Configuration	
Spunk ES Specific Saved Searches	You can select **Enable Splunk ES** in conjunction with either **Raw Search** or **Datamodel Search**.
Search for Matching Algorithm	At the initial setup, you do not have to select either **Raw Search** or **Datamodel Search** modes. This disables the matching algorithm completely, and gives you the opportunity to determine the right scale of data your installation can handle. See .[Performance](#). You can select either **Raw Search** or **Datamodel** Search if you do make a selection, but not both. You can select only up to five data models in the **Datamodel** Search mode

Sightings and Feedback to ThreatQ

One of the primary features of this solution is to identify sightings and report them back to ThreatQ.

Sighting in this context is defined as evidence that a **ThreatQ Indicator** was discovered in one or more of the events in Splunk collected via other sources. Recording these sightings and reporting them back to ThreatQ provides analysts with important context around indicators included in their threat intelligence holdings. This section describes various user configurations (in form of macros and saved searches) available to the user to achieve this, and concludes with a summary diagram that describes the whole process.

Separation of Data

ThreatQ indicator data is separated from the rest of the data in this App using a specific **sourcetype**. You can use the following Splunk search query to discover all indicators exported from ThreatQuotient.

Splunk Search for Listing TQ Indicators

```
sourcetype="threatq:indicators"
```

> Note that the same indicator can be exported multiple times if it experienced a change of status and/or score.

Macros

The following macros are used in most of the saved searches this App is configured with (available under **Settings > Advanced Search > Search Macros**).

Figure 9: Configurable Macros in ThreatQuotient App

The description of some of these search macros is below.

Saved Search Macro	Description
threatq_index	Configures the name of the Splunk index that all ThreatQ indicators are mapped to.
threatq_match_indices	Configures which Splunk indices are considered for matching. The users can apply more specific filters here.
threatq_match_sourcetypes	Configures which sourcetypes should be **excluded** from matching (the sourcetype **threatq:indicators** is automatically excluded).
threatq_score_filter	Configures a score filter for all indexed indicators from ThreatQ that should be considered for detecting sightings.
threatq_status_filter	In conjunction with the score filter, configures a status filter for all indexed indicators from ThreatQ to be considered for detecting sightings.

[Table 2: Configurable Macros]

Saved Searches

The Splunk App uses saved searches for discovering sightings and reporting them back to ThreatQ. The App is preconfigured with saved searches, which are periodic processes (registered to the crontab) designed to map indicators to specific Splunk indices and match these indicators to events. Saved search processes also move older indicators out of the main lookup tables and for ES customers, move indicators to specific ES lookup tables according to the mapping described in this document.

The table below describes some of the saved searches with which this App is preconfigured. This table displays two searches applicable only for Raw Matching Mode. Equivalent searches are available for each data model in the Datamodel Matching Mode.

> ThreatQuotient does not recommend setting the frequency to less than 30 minutes, the application default for threatq_match_ indicator saved searches, if using the configuration option for creating multiple events for each sighted indicator.

Saved Search	Description	Default Period
`threatq_consume_indicators_new`	Post matched indicators to the consume endpoint of ThreatQ and create atomic events. This search will only be enabled if using the "Create multiple events for each sighted indicator" configuration.	30 mins
`threatq_update_master_lookup`	Fetch ThreatQ indicators from the **default** index and link them against the master lookup table. This prepares the indicators for detecting evidence	15 mins

Saved Search	Description	Default Period
	of sightings.	
`threatq_match_indicators (Raw Matching Mode only)`	Finds evidence of sightings for all indicators in the **master lookup table**. If sightings are detected, indicators are moved to the **match lookup table**.	30 mins
`threatq_update_master_lookup`	Fetch ThreatQ indicators from the **default** index and link them against the **master lookup table**. This prepares the indicators for detecting evidence of sightings.	15 mins
`threatq_match_indicators`	Finds evidence of sightings for all indicators in the **master lookup table**. If sightings are detected, indicators are moved to the **match lookup table**.	30 mins
`threatq_cleanup_indicators_on_ indicators_change`	If indicator status changes from Active to Whitelisted (or any other status not considered for finding evidence of sightings), or if the indicator score drops below a threshold (making the indicator ineligible for finding evid- ence of sightings), removes those indicators from both **master lookup table** and **match lookup table**.	15 mins
`threatq_update_matched_indicators`	Finds evidence of sightings for all indicators in the **match lookup table**.	30 mins
`threatq_consume_indicators`	Creates events in ThreatQ for all newly detected sightings.	15 mins

Saved Search	Description	Default Period
threatq_update_retired_indicators	Clean up indicators that haven't been matched on in the last 90 days from both **master lookup table** and **match lookup table**.	1440 mins

[Table 3: Saved Searches for Discovering and Reporting of Sightings]

Editability rules
Because of the way sightings are found in Splunk using two saved searches (**threatq_match_indicators** and **threatq_update_ matched_indicators**), their frequency must be the same if edited. The default frequency for both saved searches is 30 mins.

Saved Searches Documentation

The following table documents the macros for saved searches as configured by default on the ThreatQuotient App.

Saved Search	Default Macro
threatq_consume_indic-ators_new	`| inputlookup threatq_matched_indicators | eval start_time=relative_time(now(), "-35m") | where match_time > start_time | sort 10000 -num(score), -num(match_count) | threatqconsumeindicatorsnew`
threatq_update_master_lookup	`` `threatq_index` sourcetype="threatq:indicators" `threatq_score_filter` `threatq_status_filter` | dedup value | eval ioc_id=id, ioc_value=value, sources='sources{}.value', adversaries='adversaries{}.value' | table value, updated_at, status, type, score, ioc_id, ioc_value, sources, adversaries | outputlookup key_field=value master_lookup | join ioc_value [| inputlookup threatq_matched_indicators | table ioc_value, match_time, first_seen, last_seen, match_count, sid] | eval value=ioc_value | table ioc_id, ioc_value, value, match_time, first_seen, last_seen, match_count, score, status, type, updated_at, sources, adversaries, sid | outputlookup key_field=value threatq_matched_indic-ators ``

Saved Search	Default Macro
`threatq_cleanup_indic-ators_on_indicators_change`	`| inputlookup master_lookup | search NOT [search` `threatq_index` `sourcetype="threatq:indicators" | dedup value | search [| inputlookup master_lookup | table ioc_value | rename ioc_value as value | format] NOT (`threatq_score_filter` `threatq_status_filter`) | table value | rename value as ioc_value | format] | outputlookup master_lookup | join ioc_value [| inputlookup threatq_matched_indicators | table ioc_value, match_time, first_seen, last_seen, match_count, sid] | out-putlookup threatq_matched_indicators`
`threatq_match_indicators` (only Raw Matching Mode)	`` `threatq_match_indices` `threatq_match_sourcetypes` `source-type!="threatq:indicators" | threatqmatchiocs`
`threatq_update_matched_indicators` (only Raw Match-ing Mode)	`` `threatq_match_indices` `threatq_match_sourcetypes` `source-type!="threatq:indicators" | threatqmatchiocs is_update=true`
`threatq_consume_indicators`	`| inputlookup threatq_matched_indicators | eval start_time=relative_time(now(), "-16m") | where last_seen > start_time | threatq-consumeindicators`
`threatq_update_retired_indicators`	`| inputlookup master_lookup | search NOT [| inputlookup master_lookup |`

Saved Search	Default Macro
	`search NOT [| inputlookup threatq_matched_indicators	search NOT [| inputlookup threatq_matched_indicators	eval threshold_time=now()-7776000, value=ioc_value	where last_seen < threshold_time	outputlookup key_field=value threatq_retired_matched_indicators	table ioc_value	format]	outputlookup threatq_matched_indicators	table ioc_value	format]	eval threshold_time=now()-7776000, updated_at_epoch=`threatq_parse_updated_at(updated_at)`, value=ioc_value	where updated_at_epoch < threshold_time	outputlookup key_field=value threatq_retired_indicators	table ioc_value	format]	outputlookup master_lookup`

[Table 4: Saved Search Macros]

As described above, two of the saved searches are applicable only for the Raw Matching Mode. If you select **Datamodel Matching Mode** from the configuration as described in the **Configuration** section, the above two saved searches for **Raw Matching Mode** will disable automatically, and the equivalent saved searches for the **Datamodel Matching Mode** will be enabled.

Reporting Sightings in ThreatQ

A sighting in Splunk is evidence that an indicator from ThreatQ was seen in one or more events in Splunk. This is important information for an analyst that can be reported back in form of an Event.

Single Event for Each Sighted Indicator

ThreatQ captures all sightings for an indicator in a single event. When more sightings are detected for the same indicator, certain attributes for that event are updated. This allows the analyst to gather context on sightings for that indicator.

Multiple Events for Each Sighted Indicator

If multiple sightings for the event are seen during the same time period, all sightings will be captured in a single event. However, if more sightings are seen in the future for the same indicator, a new event will be created in ThreatQ.

See the *Sighting Event Configuration* instructions under the [ThreatQuotient App](link) section for more details.

The following 4 attributes are recorded for the event.

Attribute	Description
First Seen	Timestamp when the first sighting for this indicator was recorded in Splunk. This attribute does not change.
Last Seen	Timestamp when the latest sighting for this indicator is recoded in Splunk. This attribute updates as newer sightings are detected.
Count	The total count of all sightings recorded for this indicator starting from the time **First Seen** until **Last Seen**.
Splunk URL	The URL that allows the analyst to view all sightings for this indic-

Attribute	Description
	ator in Splunk starting from **First Seen** until **Last Seen**.

The screen capture below shows an example event recorded in ThreatQuotient by the Splunk App.

Figure 10: Example Event in ThreatQ

The following contextual data are added to the indicator :

Attribute	Description
Splunk Sighting Timestamp	When the latest sighting for this indicator was recorded in Splunk.
Match Count	The total count of all sightings recorded for this indicator.
Source	**Splunk** will be added as the **Source** for this indicator.

Putting Everything Together

The following steps summarize how indicators are stored in Splunk and how sightings are reported back to ThreatQ.

1. The Input job configured on **ThreatQuotient Add-on** (on the heavy forwarder) pulls indicators from ThreatQ.

2. The heavy forwarder sends the indicators to the indexer which indexes the indicators to the **default** index (user can override).

3. The periodic saved search job **threatq_update_master_lookup** maps the newly indexed ThreatQ indicators to the **master lookup table**.

4. The periodic saved search job **threatq_match_indicators** finds evidence of sightings of all indicators in the **master lookup table** against all events in Splunk (as filtered via various configurable macros described above in this section).

   a. If evidence of sightings is found for a specific indicator, it is moved to the **match lookup table**.

5. Simultaneously, another periodic saved search job **threatq_update_matched_indicators** finds more sightings for all indicators from the **match lookup table** against all events in Splunk (as filtered by the same configurable macros).

6. A periodic saved search **threatq_consume_indicators** will create events in ThreatQ to represent evidence of sightings in Splunk.

7. The periodic saved search job **threatq_update_retired_indicators** takes all indicators that are not updated in the past 90 days out of both the **master lookup table** and **matched lookup table**.

The following diagram summarizes this process.
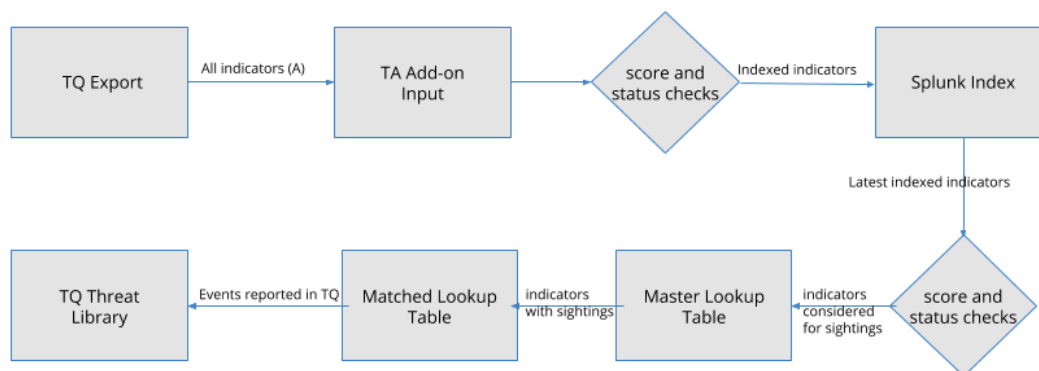
Figure 11: Detecting and Reporting Indicator Matches to ThreatQ

Workflow Actions

The **ThreatQuotient Add-on** provides five user workflow actions to the analysts for providing interactivity with the ThreatQuotient platform from Splunk. As shown on the diagram below, the actions can be invoked on any Splunk event by expanding the event view and clicking on the downarrow in the column below **Action**.

Figure 12: ThreatQuotient Workflow Actions

The actions are described below.

ThreatQ: Add Indicator

This workflow action adds the indicator to ThreatQ. You are presented with UI inputs that allow you to select indicator type, status and source. If the data and type do not match, an error is reported. Successful completion of this workflow action results in the indicator being successfully added to the ThreatQ Threat Library.

ThreatQ: Add to Whitelist

This workflow action sets the status of the indicator to Whitelisted in ThreatQ. If the indicator does not exist in ThreatQ, an error is reported.

ThreatQ: Lookup Indicator

This workflow action searches for an indicator in ThreatQ and pulls additional context for that indicator. If the indicator does not exist in ThreatQ, an error is reported.

ThreatQ: Mark as False Positive

This workflow action adds the attribute key-value **"False Positive: True"** to the indicator in ThreatQ. If the indicator does not exist in ThreatQ, an error is reported.

ThreatQ: Mark as True Positive

This workflow action adds the attribute key-value **"True Positive: True"** to the indicator in ThreatQ. If the indicator does not exist in ThreatQ, an error is reported.

CIM Support

The App runs in the Datamodel Search mode when you are taking advantage of Splunk's CIM and mapping your logs and events to various data models provided by Splunk. The following table summarizes how the matching algorithm will match specific data model fields to specific indicator types in ThreatQuotient.

CIM Data Models	Data Model Fields	ThreatQ Indicators Types Matched
Authentication	Authentication.src_user	Username
	Authentication.user	Username
Certificates	Certificates.All_Certificates.SSL.ssl_hash	SHA-1, SHA-256, SHA-384, SHA-512
	Certificates.All_Certificates.SSL.ssl_issuer_	Email Address

CIM Data Models	Data Model Fields	ThreatQ Indicators Types Matched
	email	
	Certificates.All_Certificates.SSL.ssl_subject_email	Email Address
	Certificates.All_Certificates.SSL.ssl_subject_common_name	String
	Certificates.All_Certificates.SSL.ssl_issuer_common_name	String
	Certificates.All_Certificates.SSL.ssl_subject_organization	String
	Certificates.All_Certificates.SSL.ssl_issuer_organization	String
	Certificates.All_Certificates.SSL.ssl_serial	String
	Certificates.All_Certificates.SSL.ssl_subject_unit	String
	Certificates.All_Certificates.SSL.ssl_issuer_unit	String
Endpoint	Endpoint.Services.service	Service Name
	Endpoint.Processes.process_name	Service Name
	Endpoint.Filesystem.file_name	Filename
	Endpoint.Filesystem.file_hash	SHA-1, SHA-256, SHA-384, SHA-

CIM Data Models	Data Model Fields	ThreatQ Indicators Types Matched
		512
Email	Email.All_Email.file_name	Filename
	Email.All_Email.file_hash	SHA-1, SHA-256, SHA-384, SHA-512
	Email.All_Email.subject	Email Subject
	Email.All_Email.src_user	Email Address
Intrusion_ Detection	Intrusion_Detection.IDS_Attacks.src	IP Address, IPv6 Address
	Intrusion_Detection.IDS_Attacks.signature	String
	Intrusion_Detection.IDS_Attacks.user	Username
Inventory	All_Inventory.User.user	Username
Malware	Malware.Malware_Attacks.file_name	Filename
	Malware.Malware_Attacks.file_hash	SHA-1, SHA-256, SHA-384, SHA-512

CIM Data Models	Data Model Fields	ThreatQ Indicators Types Matched
	Malware.Malware_Attacks.signature	String
	Malware.Malware_Attacks.sender	Email Address
	Malware.Malware_Attacks.src	IP Address, IPv6 Address
	Malware.Malware_Attacks.user	Username
Network_ Traffic	Network_Traffic.All_Traffic.src	IP Address, IPv6 Address
Network Res- olution (DNS)	Network_Resolution.DNS.query	String
	Network_Resolution.DNS.answer	String
Updates	Updates.Updates.file_name	Filename
	Updates.Updates.file_hash	SHA-1, SHA-256, SHA-384, SHA-512
Web	Web.Web.user	Username
	Web.Web.http_referrer	URL

CIM Data Models	Data Model Fields	ThreatQ Indicators Types Matched
	Web.Web.url	URL
	Web.Web.http_user_agent	User-agent
	Web.Web.src	IP Address, IPv6 Address
	Web.Web.dest	IP Address, IPv6 Address
Incident_Man-agement	Incident_Management.Notable_Events.src	IP Address, IPv6 Address
	Incident_Management.Suppressed_Notable_Events.src	IP Address, IPv6 Address
	Incident_Management.Notable_Event_Sup-pressions.Suppression_Audit.signature	String
	Incident_Management.Notable_Event_Sup-pressions.Suppression_Audit_Expired.sig-nature	String
	Incident_Management.Notable_Event_Sup-pressions.Suppression_Audit.user	Username

Table 5: ThreatQ indicator type to CIM field map for the matching algorithm

Enterprise Security Support

ThreatQ Indicators to Splunk Enterprise Security Lookup Tables

The App provides support to the Splunk Enterprise Security (ES) customers by making ThreatQ data more accessible using Splunk's native ES lookup tables. The following table provides how ThreatQ data is mapped to the Splunk ES lookup tables. This data is then available in various ES dashboards.

ThreatQ type	Threat intelligence type
CIDR Block	local_ip_intel
Email Address	local_email_intel
Email Subject	local_email_intel
File Name	local_file_intel
FQDN	local_domain_intel
Fuzzy Hash	local_file_intel
GOST Hash	local_file_intel
IP Address	local_ip_intel
MD5	local_file_intel
Registry Key	local_registry_intel
Service Name	local_service_intel
SHA-1	local_file_intel

ThreatQ type	Threat intelligence type
SHA-256	local_file_intel
SHA-384	local_file_intel
SHA-512	local_file_intel
x509 Serial	local_certificate_intel
x509 Subject	local_certificate_intel
URL	local_http_intel
URL Path	local_http_intel
Username	local_user_intel

Table 6: ThreatQ indicator type mapping
to Enterprise Security lookup tables

To view the events and indicators, navigate to **Enterprise Security > Security Intelligence > Threat Intelligence**.

- **Threat Activity:** Shows the list of events which are compatible with CIM apps.

- **Threat Artifacts:** Shows the list of indicators fetched from the ThreatQ.

Using Threat Intelligence Data in Splunk Enterprise Security

Splunk's Enterprise Security App provides the means of using your threat intelligence data to match against events mapped to standard Splunk models. Refer to the Splunk's documentation on **Enterprise Security Workflow for Threat Intelligence** as described here: http://dev.splunk.com/view/enterprise-security/SP-CAAAFBC.

ThreatQuotient provides mapping of the threat intelligence data to the standard lookup tables in Splunk Enterprise Security via the saved searches described above. Using the

default Threat Generation Searches in the Enterprise Security, the ES app will find matches and report those matches in the `threat_activity` index as described in the link above.

Threat Intelligence data will be added to Enterprise Security using their REST APIs with a threat_key of `threatq_indicator`. The score for ThreatQ Indicators will be mapped to the **Weight** attribute in ES. Any updates to the score will be automatically reflected in ES using the periodic saved searches.

The indicator will be updated in ES and put into a disabled state (will no longer be used in further correlation) if the score or status of a ThreatQ indicator changes to a value that is no longer within the parameters configured in the macro settings for ThreatQ Splunk App.

> ⚠️ If you are using ES and are upgrading from an older version of ThreatQ Splunk App, please run the "threatq_cleanup_es_lookups" saved search once to remove the old data. All the threat intelligence data is auto-matically added upon upgrade using the Enterprise Security's REST APIs

> 📝 When using the Enterprise Security App, you will not have additional con-text (sources and adversaries), workflow actions ,and reporting sightings back to ThreatQuotient available to you.

Saved Searches for Enterprise Security

In addition to the core saved searches, the following saved searches apply for Enterprise Security (ES) customers. The saved searches listed run once a day and map ThreatQ indic-ators by type to Splunk ES lookup tables as described in the **Mapping Table** section of the document.

> 📝 By default, the **scheduling** of all saved searches for porting Threat Intel-ligence data from ThreatQ to lookup tables in the ES are **disabled**. This is because not all users have Enterprise Security App installed. If you have this App installed and want to port the Threat Intelligence data over, you will need to enable the scheduling of these saved searches.

ES Saved Search	Description
`threatq_update_threat_intelligence_lookup_email_address`	Map ThreatQ **type 4** indicators to **local_email_intel**
`threatq_update_threat_intelligence_lookup_email_subject`	Map ThreatQ **type 6** indicators to **local_email_intel**
`threatq_update_threat_intelligence_lookup_file_name`	Map ThreatQ **type 9** indicators to **local_file_intel**
`threatq_update_threat_intelligence_lookup_fqdn`	Map ThreatQ **type 10** indicators to **local_domain_intel**
`threatq_update_threat_intelligence_lookup_hash`	Map ThreatQ **type [11,12,15,20,21,22,23]** indicators to **local_file_intel**
`threatq_update_threat_intelligence_lookup_ip`	Map ThreatQ **type 14** indicators to **local_ip_intel**
`threatq_update_threat_intelligence_lookup_registry`	Map ThreatQ **type 18** indicators to **local_registry_intel**
`threatq_update_threat_intelligence_lookup_service`	Map ThreatQ **type 19** indicators to **local_service_intel**
`threatq_update_threat_intelligence_lookup_certificate_serial`	Map ThreatQ **type 25** indicators to **local_certificate_intel**
`threatq_update_threat_intelligence_lookup_certificate_subject`	Map ThreatQ **type 26** indicators to **local_certificate_intel**
`threatq_update_threat_intelligence_lookup_url`	Map ThreatQ **type 27** indicators to **local_http_intel**

ES Saved Search	Description
`threatq_update_threat_intel-ligence_lookup_user`	Map ThreatQ **type 30** indicators to **local_user_intel**

[Table 7: Saved Searches for Mapping ThreatQ Indicator data to Splunk's CIM]

Performance

The primary objective of this App is to find evidence of sightings and report those sightings back to ThreatQuotient. The sightings are discovered using the **matching algorithm** that works either in the Raw Matching or **Datamodel Matching** Mode. Broadly speaking, the matching algorithm will take the set of indicators from ThreatQuotient, a set of events from Splunk and find which indicators (and how many times) appear in the events. The matching algorithm by default runs every 30 minutes in a saved search, so it is important that it completes in under 30 minutes on average just to keep up with incoming load.

The tables below demonstrate the performance of matching algorithm for both modes. These tables are meant to be used as guidelines so you can configure your App to run for an optimal performance.

Total Indicators from TQ	Total Raw Events in Splunk	Total Indicators Matched	Time to Complete (s) Machine Specs:(16 Core, 32GB RAM)
100,000	500,000	0	885.36
100,000	500,000	10,000	899.75
100,000	1,000,000	20,000	1,932.04
500,000	1,000,000	0	1,926.62

Total Indicators from TQ	Total Raw Events in Splunk	Total Indicators Matched	Time to Complete (s) Machine Specs:(16 Core, 32GB RAM)
500,000	**1,000,000**	**10000**	**2,020.56**
1,000,000	1,000,000	0	2,174.18
1,000,000	1,000,000	25,000	2,294.39
1,000,000	5,000,000	0	11,354.64
10,000	50,000,000	0	35,233.185 (9 hr 47 min)

Table 8: Raw Matching Performance Table

Experiments were conducted on a machine with 16 cores and 32 GB RAM. The parameters are total number of indicators from TQ, total events from Splunk and number of indicators matched. Events were generated from various standard templates covering a wide range of firewall and web proxy logs. The bolded rows show the upper limit of performance in that the time to complete is slightly over 30 mins. We discovered that the **upper limit** was reached at around 1 million Splunk events and was largely invariant of number of indicators from ThreatQ (due to how this algorithm is implemented). As more matches are found, it takes more time to write them in the lookup tables, thus slightly increasing the runtime.

Total Indic-ators from TQ	Total Raw Events in Splunk	Total Indic-ators Matched	Time to Complete (s) Machine Specs:(16 Core, 32GB RAM)
100,000	500,000	0	29.282
100,000	500,000	10,000	36.92

Total Indicators from TQ	Total Raw Events in Splunk	Total Indicators Matched	Time to Complete (s) Machine Specs:(16 Core, 32GB RAM)
100,000	1,000,000	20,000	77.649
500,000	1,000,000	0	99.473
500,000	1,000,000	10000	130.991
1,000,000	1,000,000	0	166.517
1,000,000	1,000,000	25,000	261.362
1,000,000	5,000,000	0	420.111
100,000	5,000,000	10,000	619.047
1,000,000	10,000,000	10,000	1316.541
1,000,000	**15,000,000**	**10,000**	**1866.059**
1,000,000	50,000,000	25,000	6,554.610

Table 9: Datamodel Matching Performance Table

Similar experiments were done for a Datamodel Matching case. From the table above, we determined that at around 15 million mark for Splunk events, the algorithm runtime started exceeding 30 minutes. **Thus, for a single saved search, this represents the upper limit of how much data this algorithm can handle every 30 minutes**.

We advise that you experiment on your system to ensure that your system does not have data loaded at higher rates than is implied by the above tables. If the machine specs are different, it is advisable to first simply run the match queries in the Splunk's search bar and get a sense of how long it takes a typical query to finish. Once you find the right amount of data your installation can handle, you are advised to instrument the App in a way that it will only perform matching on the said amount of data.

Scaling the App

The tables displayed in [Performance](#) offer a guideline of how many Splunk events the App can handle with default configuration. As found from the internal testing, table 8 demonstrates the upper limit for raw search is about **1 million events/30 minutes**, and the same is **15 million events/30 minutes** for the datamodel search (on a dedicated box with 16 cores and 32 GB RAM). However, this is valid only for one saved search running on one node.

The best way to scale the App is to run **multiple saved searches for matching**. This is easy to do in datamodel search mode. If your data is mapped to multiple Spunk data models from Table 5, each data model is handled by a separate saved search. In such an instance, you would need to deploy your search head in a cluster, and ensure that these saved searches are distributed in that cluster. You can run up to five of them, thus potentially scaling your App to handle 5 times the traffic.

For the raw matching mode, the App by default will only be able to run one saved search. In order to extend it to multiple searches, you will have to **break apart this one saved search into multiple**, and then, distribute these saved searches in the Splunk cluster of search heads. You can do this by running a separate saved search for:

- Splunk index for events
- ThreatQuotient indicator types.

For using a fixed Splunk index for the saved search, you can modify the default saved searches for matching as shown below.

Splunk Search for Listing TQ Indicators

```
index=<my_index> `threatq_match_sourcetypes` source-
type!="threatq:indicators" | threatqmatchiocs indicator_types-
='IP Address, FQDN'(threatq_match_indicators saved search)
index=<my_index> `threatq_match_sourcetypes` source-
type!="threatq:indicators" | threatqmatchiocs is_update=true
(threatq_update_matched_indicators saved search)
```

Figure 13: Example Saved Searches Extension for Scaling the App (Part I)

Compare the above saved searches with the defaults as shown in Table 4. The macro **threatq_match_indices** is replaced by passing an actual index to the saved search. Now, you can make multiple copies of the default saved search, run them on the same schedule, and have each saved search get events from a different Splunk index.

To use the similar technique for ThreatQuotient indicator types, you can pass an additional argument to the **threatqmatchiocs** module as shown below. This allows you to make the saved search use only a specific indicator type. Again, as before, you can then make multiple copies of the saved searches and have each one handle only specific ThreatQuotient indicator types. You are free to pass a single indicator type, or a comma separated list as shown below.

Splunk Search for Listing TQ Indicators

```
index=<my_index> `threatq_match_sourcetypes` source-
type!="threatq:indicators" | threatqmatchiocs indicator_types-
='IP Address, FQDN'(threatq_match_indicators saved search)
index=<my_index> `threatq_match_sourcetypes` source-
type!="threatq:indicators" | threatqmatchiocs is_update=true
```

```
indicator_types='IP Address, FQDN'(threatq_update_matched_
indicators saved search)
```

Figure 14: Example Saved Searches Extension for Scaling the App (Part II)

Finally, both these techniques for scaling the App are equally applicable for the **datamodel mode** as well.

Dashboards

Preconfigured dashboards are packaged with **ThreatQuotient App** to allow the analyst versatile visual representation of all indicator data from ThreatQ and the corresponding sightings. These dashboards are only a suggestion and can be modified via Splunk's standard dashboard editing capability to meet your needs.

Cumulative Counts

The top section of the dashboard shows total count for all ThreatQ indicators in the **master lookup table** (on the left) and the **match lookup table** (in the right) (all time and the last 24 hours). It is important to note that the data displayed as Sightings are not the total sightings; rather it is the total number of indicators for which evidence of sightings has been found. Example screen capture below.
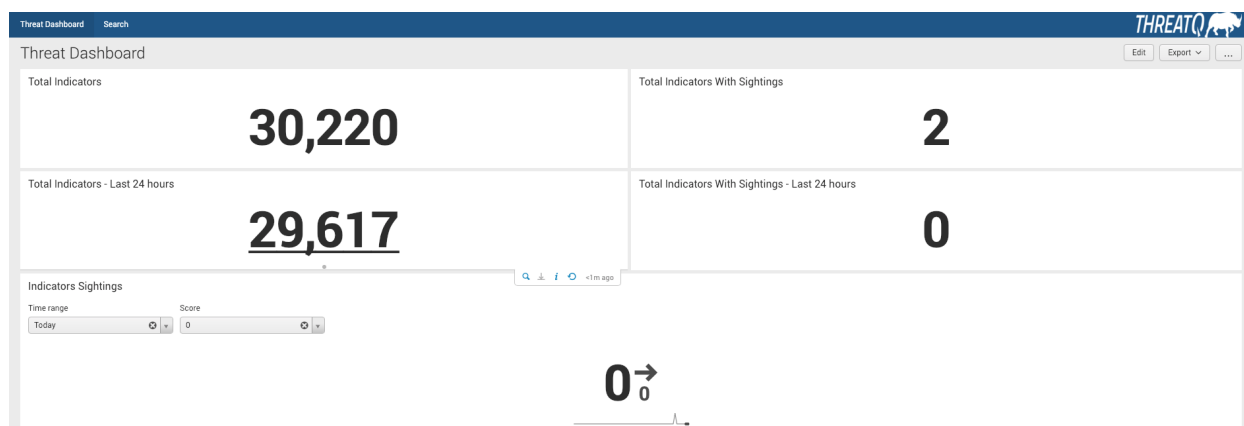
Figure 15: Dashboard: Cumulative Counts

Score Breakdown

The next section shows the distribution of indicator scores for indicators in master and match lookup tables as bar charts. Example screenshot below. These charts do not have a time filter. The counts for individual score breakdown represent the cumulative indicator count. As an example, notice that there are two indicators with sightings each with score 9 (which matches up with the cumulative sightings count of 2 in the chart above).
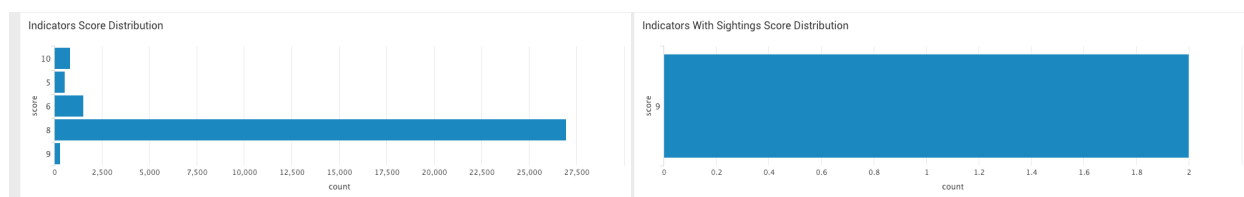

Figure 16: Dashboard: Score Breakdown

Type Breakdown

This section shows the distribution of indicator types for indicators in master and match lookup tables as pie charts. As the score distributions above, these are cumulative distributions. Example screenshot below. Hovering over each portion of the pie chart will display the indicator count for that specific portion.
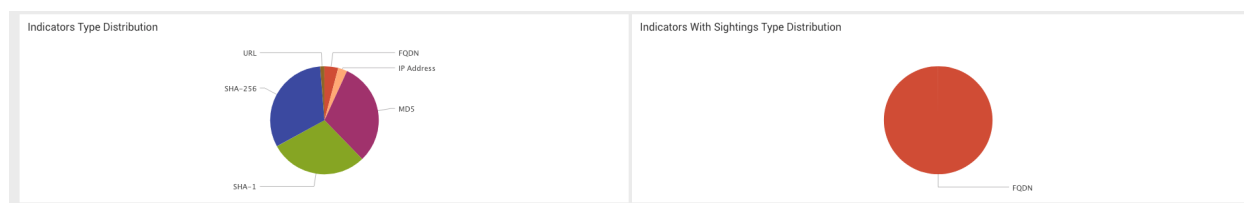
Figure 17: Dashboard: Type Breakdown

Source Breakdown

This section shows the breakdown of indicators and sighted indicators by sources. Example screenshot below. One thing to note here is that all indicators must have at least one source, but some indicators may have more than one. For this reason, the cumulative counts in the

charts below may exceed the total number of indicators and sighted indicators in the lookup tables.
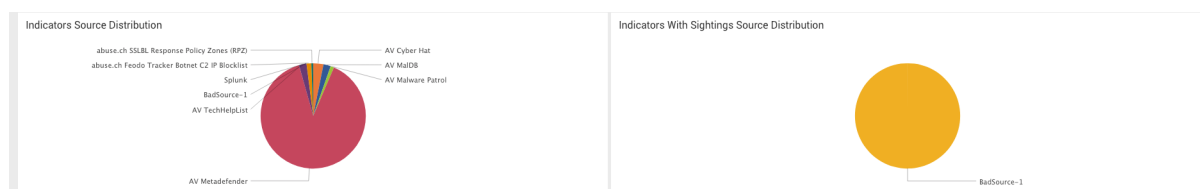

Figure 18: Dashboard: Source Breakdown

Adversaries Breakdown

This section shows the breakdown of indicators and sighted indicators by adversaries. Example screenshot below. One thing to note here is that not all indicators have adversaries; although some indicators may have more than one. Depending upon how many indicators have adversaries, the total cumulative counts in the charts below may be less or more than the total indicators and sighted indicators in the lookup tables. For the example dataset below, there is only one adversary assigned to a few indicators, and those same indicators are sighted.
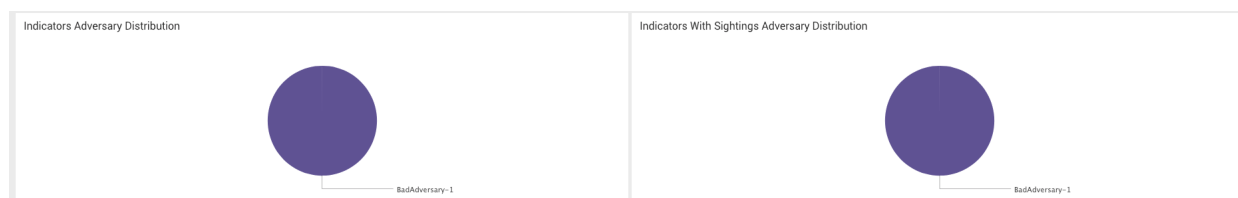
Figure 19: Dashboard: Adversary Breakdown

Static Table View

This section shows all indicators and sightings in static tables - time filters are provided and defaulted to the last 24 hours. Score and type filters are also available for both. This inform-

ation gives a threat analyst a single place to view all sightings in Splunk. In the screenshot below, notice there are two indicators sighted, each with 2 sightings.

Figure 20: Dashboard: Static Indicators and Sighted Indicators Tables

Top 10 By Sightings

The final section displays top 10 indicators by sightings, top 10 sources by sightings and top 10 adversaries by sightings in form of a static table, bar chart and bar chart respectively. This information gives an analyst a quick view of the indicators sources and adversaries with the most matches within Splunk.

Figure 21: Dashboard: Top 10 Indicators by Sightings

Sources

Example screenshot below. Notice the source **BadSource-1** appears as the top source with sightings corresponding to the sighted indicators as displayed in the static table above. Also notice that the sightings count is 4, which corresponds to 2 sightings each for the sighted indicators.

Figure 22: Dashboard: Top 10 Sources by Sightings

Adversaries

Example screenshot below. Notice the source **BadAdversary-1** appears as the top adversary with sightings corresponding to the sighted indicators as displayed in the static table above. Also notice that the sightings count is 4, which corresponds to 2 sightings each for the sighted indicators.
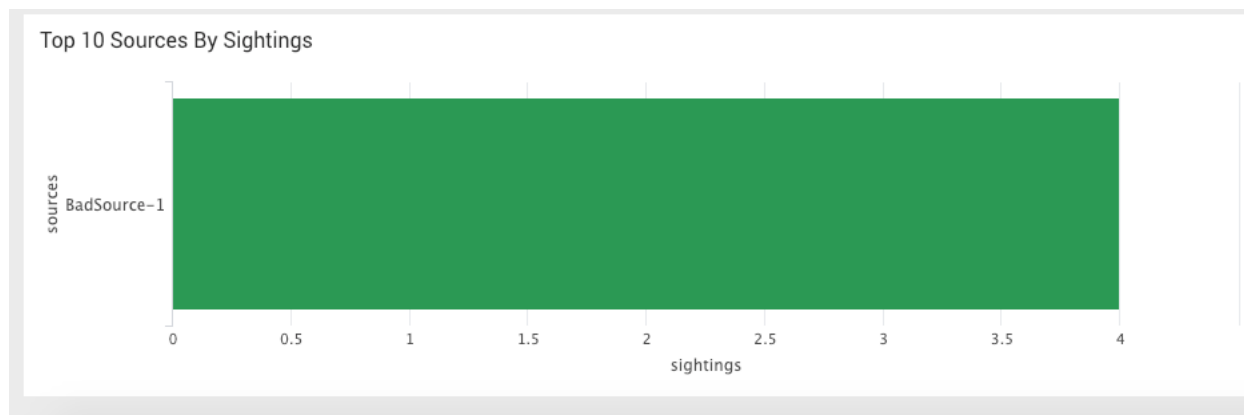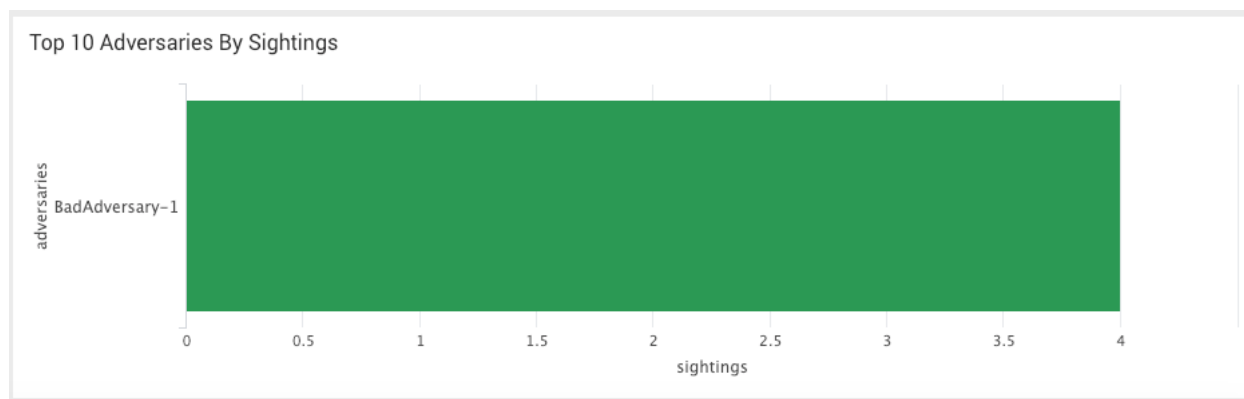

Figure 23: Dashboard: Top 10 Adversaries by Sightings

Troubleshooting

- To troubleshoot **ThreatQuotient Add-on** please check the log file below:

```
$SPLUNK_HOME/var/log/Splunk/ta_threatquotient_
add_on_threatq_indicators.log
```

- To find all unique indicators indexed in Splunk by the Add-On (Splunk App allows you to select a specific time range):

```
sourcetype="threatq:indicators" | dedup value
```

- To check the data collected by data collection use query like:

```
"index=your_index_name sourcetype=threatq_indic-
ators"
```

- Make sure all the saved searches are enabled.
- Make sure the macro is updated as per the settings.
- To troubleshoot any behavior with the master table lookup tables (which is used in the dashboards), the following query is useful:

```
index=_internal sourcetype="scheduler" saved-
search_name=threatq_update_master_lookup status-
s=success
```

The log file can be found at the following location:

```
/opt/splunk/var/log/splunk/scheduler.log
```

- If the user changes macros for global score and status thresholds, the audit logs can be accessed using the following two saved searches:

Splunk Search for Listing TQ Indicators

```
index=_internal threatq_score_filter source-
type="splunkd_ui_access"
index=_internal threatq_score_filter source-
type="splunkd_access"
```

- Logs for the saved search to update the master lookup table can be accessed using the following query (the same query can be used to check the run statuses of any saved search; just replace with the appropriate saved search name):

```
index=_internal sourcetype=scheduler saved-
search_id="nobody;threatqappforsplunk;threatq_
update_master_lookup
```

Change Log

Version 2.0.0

- Python 3 Support - ThreatQuotient App for Splunk is now compatible with Python 3. Supported versions include:
  - Splunk 7.2.x
  - Splunk 7.3.x
  - Splunk 8.X (Python 2)
  - Splunk 8.X (Python 3)

TA-threatquotient-add-on: Version 2.0.0

- Python 3 Support - ThreatQuotient Add-on for Splunk is now compatible with Python 3. Supported versions include:
  - Splunk 7.2.x
  - Splunk 7.3.x
  - Splunk 8.X (Python 2)
  - Splunk 8.X (Python 3)
- Resolved an issue where creating an indicator in Splunk would occasionally result in the creation of an indicator with an incorrect type within the ThreatQ platform.

Version 1.3.0

- Threat Intelligence support for Enterprise Security is now provided using its REST APIs

Version 1.2.0

- Added the following contextual data to Indicators:

  - Splunk Sighting Timestamp - Last seen value

  - Match Count

  - The Source for sighted indicators is now reported as Splunk in ThreatQ.

- Added Macro Configuration option to App Setup page. Users now have the ability to select indices, the location they want to search.

> 📝 If you have the macro configuration for threatq_match_indices set to
> *****, you will need to update the app configuration upon upgrade to
> 1.2.0 and add the required indexes where matching should take
> place with ThreatQ indicators. This step is mandatory for the app to
> continue to perform matching against the required indexes.

- Added Sighting Event Configuration option to the App Setup page. Users now have the ability to configure how a the app create events for a sighted indicator.

- Added a new Saved Search - threatq_consume_indicators_new

TA-threatquotient-add-on: Version 1.1.2

- Certificate-based errors will no longer appear in the Splunk log. They will now be added as a warning in the ThreatQ application log.

TA-threatquotient-add-on: Version 1.1.1

We have fixed an issue where Splunk credential parsing was generating a 500 error and leaving the configuration page in an unusable state.

Version 1.1.0

- The ThreatQuotient Splunk integration now includes support for the Common Information Model (CIM). For users who map third party data (firewall events, logs, for example) to Splunk's data models in CIM, this App provides optimized performance by leveraging those data models. As such, we now support the CIM Data

Model Search.

- We have enhanced Enterprise Security (ES) support to provide single-click enablement within the ThreatQ App for Splunk application settings.

We have fixed issues where:

- Users could not re-enable and use searches without crashing Splunk ES search head.

- threatq_match_indicators searches failed to complete. All saved search queries for matching can now accept an optional argument called indicator_types that allows users to match only specific indicator types from ThreatQ.

Version 1.0.1:

During authentication, users can now specify whether to verify or disable the SSL certificate.