ThreatQuotient

ThreatQ App of QRadar Implementation Guide

Version 1.2.3

Friday, May 1, 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.

Last Updated: Friday, May 1, 2020

Contents

Introduction

The ThreatQ App for QRadar is a bi-directional application that runs within the QRadar SIEM's application framework. The application allows for the ingestion of ThreatQ's indicators of compromise (IoCs), exports QRadar Offenses to ThreatQ as Events, parses QRadar Events and adds the Offense Source as Indicators in ThreatQ, and provides right-click actions that allow QRadar Analysts to interact with their ThreatQ instance from within the QRadar SIEM's UI.

> ⚠ If a previous version of the application has been installed, it's advised to purge all ThreatQ reference sets of data before installing and pulling indicators with the new version. This will ensure the differential has the most accurate information.

Prerequisites

The ThreatQ App for QRadar prerequisites are:

- Chrome browser
- ThreatQ version 4.30.0+ to take advantage of the client credentials CLI command.
- QRadar SIEM version greater than or equal to 7.3.1 patch 7.
- Authentication Token from a ThreatQ Authorized Service
- Configuration of the ThreatQ App for QRadar requires administrative privileges. All other actions are available to users.

Firefox and Internet explorer are not fully supported at this time. Please use the Chrome browser with v1.2.0+ of the app.

⚠ Upgrading does not overwrite the configurations. ThreatQuotient recommends that you uninstall the current version and then install the new version.

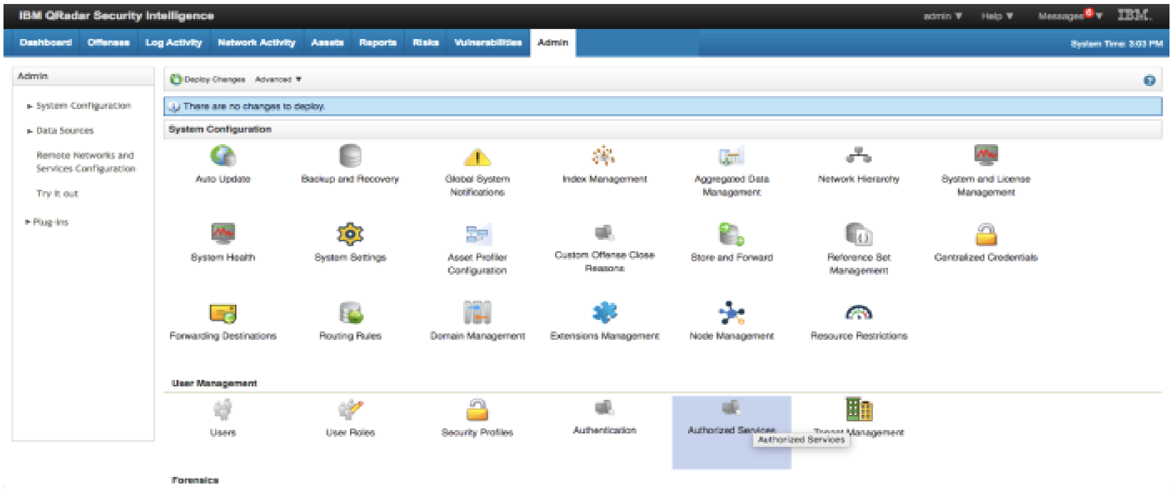
Installation

If a previous version of the application has been installed, it's advised to purge all ThreatQ reference sets of data before installing and pulling indicators with the new version. This will ensure the differential has the most accurate information.

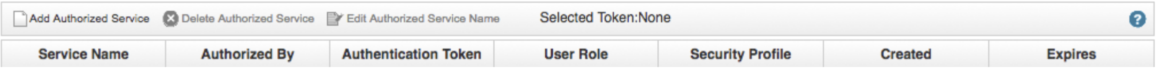
This application will be available for download from the IBM App Exchange.

The application package may be installed via the Extension Management component of the QRadar SIEM.

1. Click on the **Admin** tab In the QRadar SIEM.
2. Click on **Authorized Services** located under the User Management section.

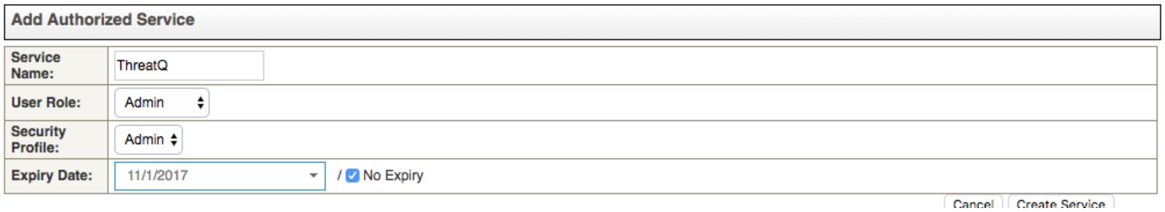


3. Click on **Add Authorized Service**.



Service Name	Authorized By	Authentication Token	User Role	Security Profile	Created	Expires

4. Complete the applicable fields and click **Create Service**.

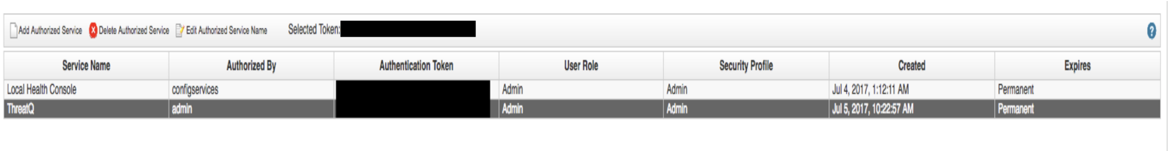


Add Authorized Service

Service Name:	ThreatQ
User Role:	Admin
Security Profile:	Admin
Expiry Date:	11/1/2017 / ☑ No Expiry

5. Copy the **Authentication Token** that was generated and store it for later use.

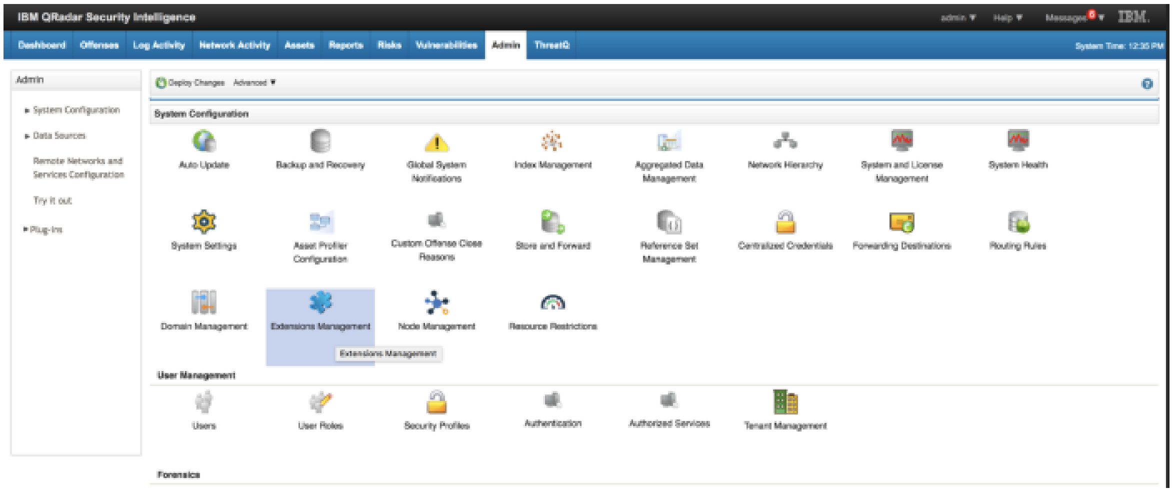
> Be sure to deploy the changes so that the app can use the token.



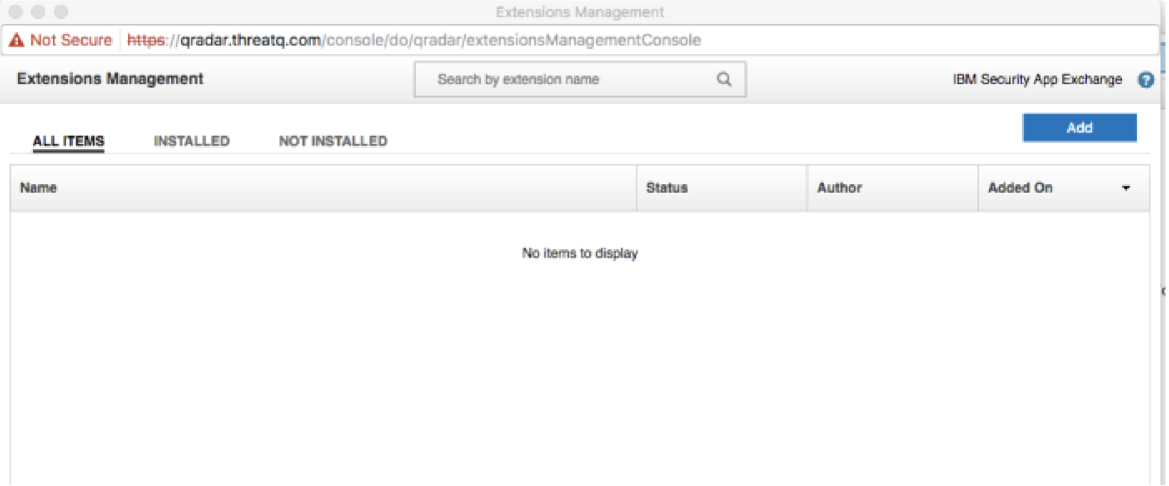
Selected Token: [redacted]

Service Name	Authorized By	Authentication Token	User Role	Security Profile	Created	Expires
Local Health Console	configservices	[redacted]	Admin	Admin	Jul 4, 2017, 1:12:11 AM	Permanent
ThreatQ	admin	[redacted]	Admin	Admin	Jul 5, 2017, 10:22:57 AM	Permanent

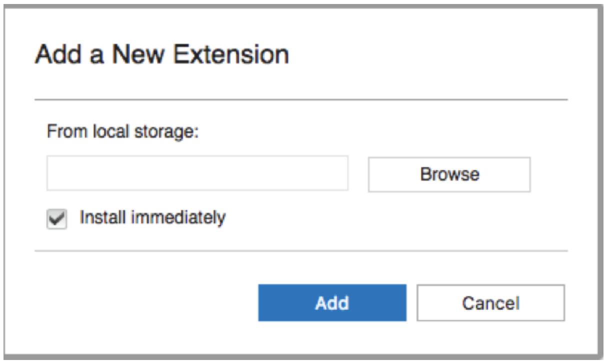
6. Open Extensions Management from within the QRadar Admin tab.



7. Click on **Add** to add a new extension.



8. Browse to the location where the ThreatQ App for QRadar was downloaded, select it, and ensure that the **Install immediately** checkbox is checked.

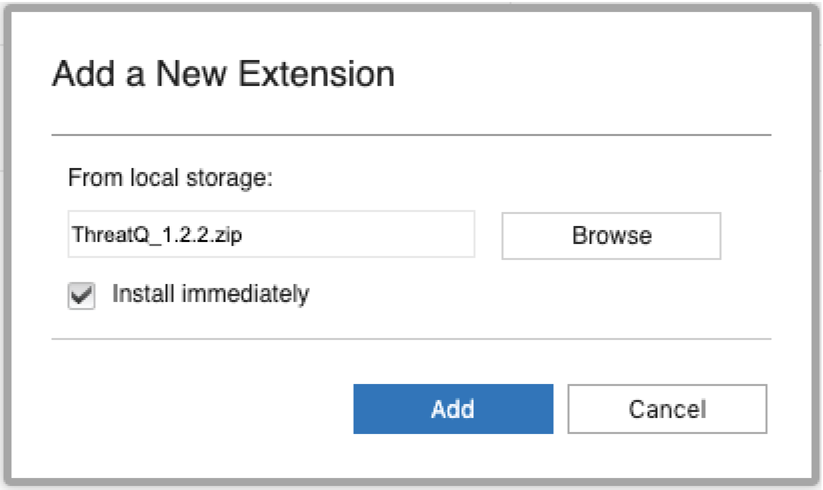


Add a New Extension

From local storage:

☑ Install immediately

9. Click **Add** on the Add a New Extension window.

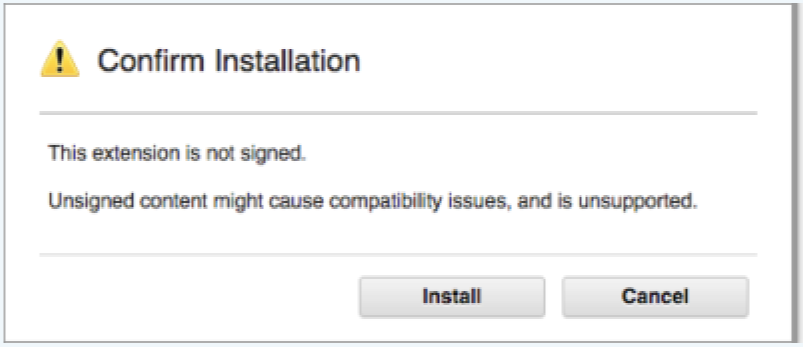


Add a New Extension

From local storage: ThreatQ_1.2.2.zip

☑ Install immediately

If this is a pre-release extension (it has not been submitted to IBM for validation) you will see a warning indicating that the extension has not been signed. This is expected and will be removed once the App goes through IBM validation and is available with the IBM App Exchange. Proceed to step 10.

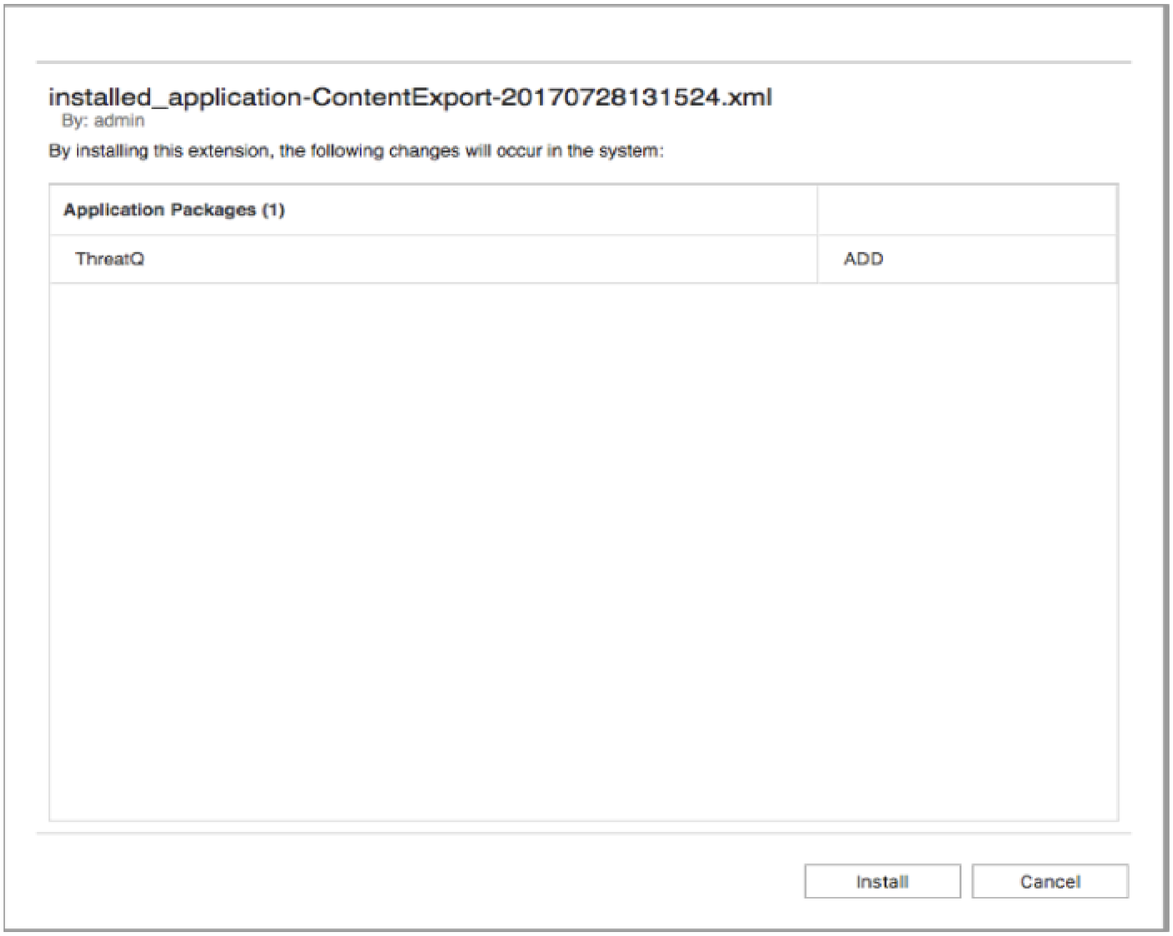


Confirm Installation

This extension is not signed.

Unsigned content might cause compatibility issues, and is unsupported.

10. Click **Install**.



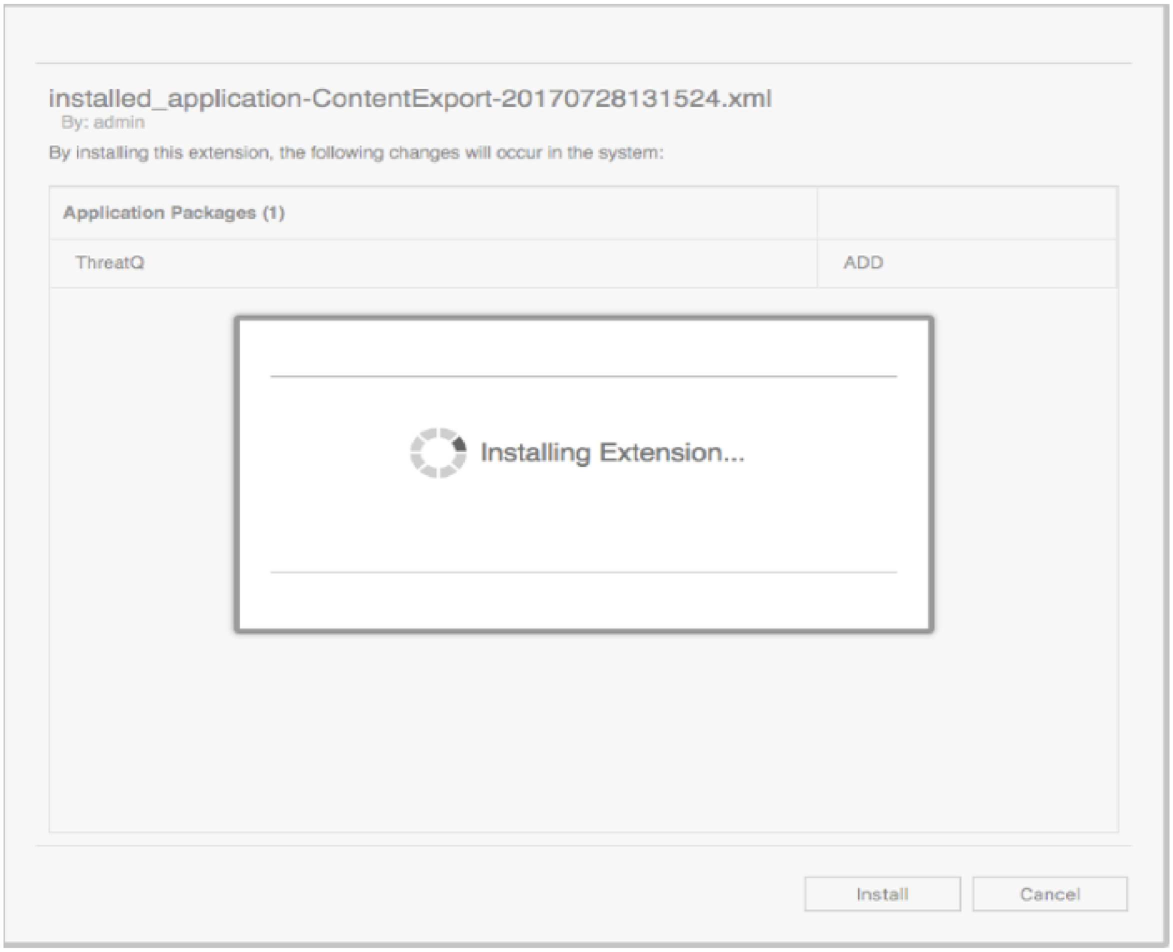
installed_application-ContentExport-20170728131524.xml

By: admin

By installing this extension, the following changes will occur in the system:

Application Packages (1)	
ThreatQ	ADD

An installation window will pop up with a loading wheel while installing the extension.

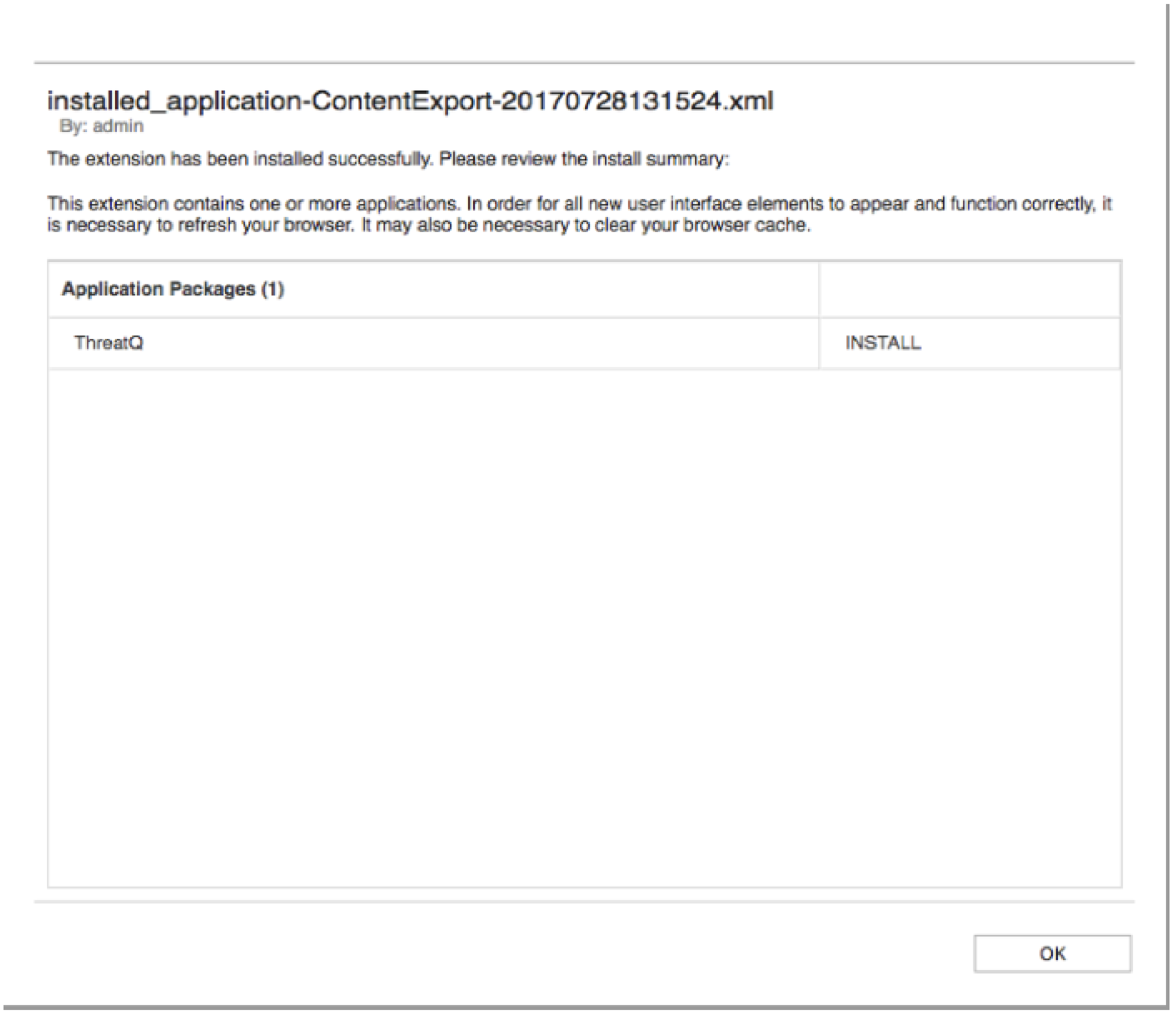


installed_application-ContentExport-20170728131524.xml
By: admin
By installing this extension, the following changes will occur in the system:

Application Packages (1)	
ThreatQ	ADD

Installing Extension...

11. Click **OK**.



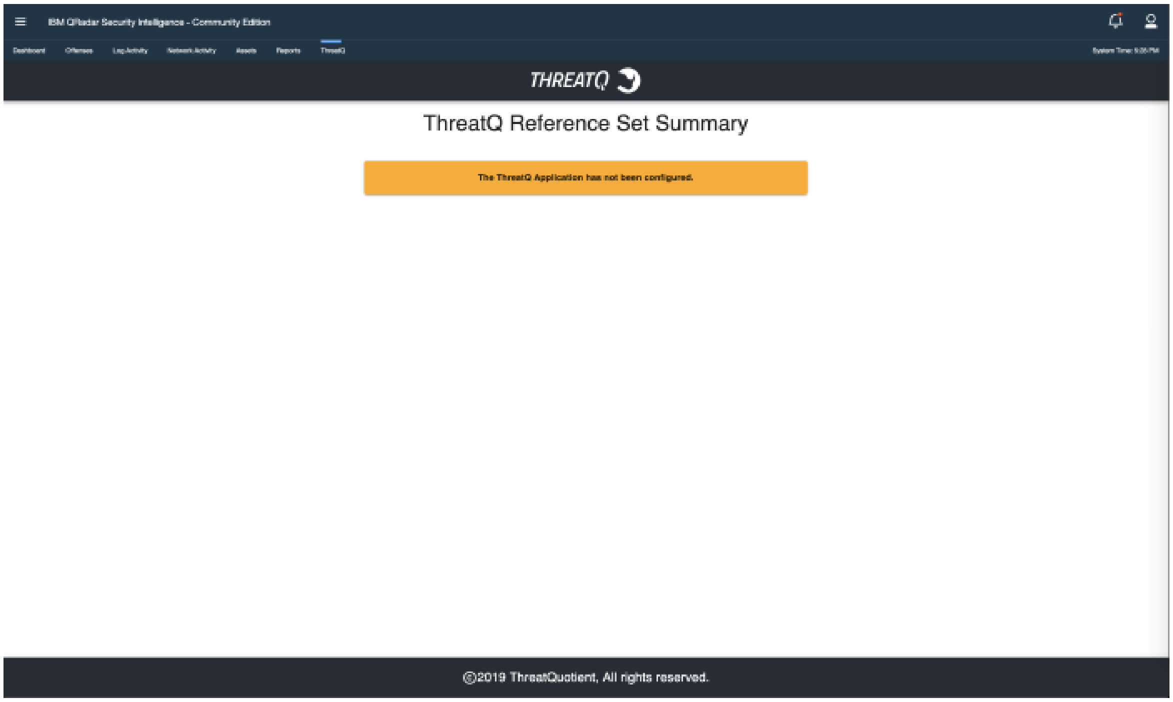
installed_application-ContentExport-20170728131524.xml
By: admin

The extension has been installed successfully. Please review the install summary:

This extension contains one or more applications. In order for all new user interface elements to appear and function correctly, it is necessary to refresh your browser. It may also be necessary to clear your browser cache.

Application Packages (1)	
ThreatQ	INSTALL

Upon successful installation, a new tab labeled ThreatQ will be displayed in the QRadar SIEM UI. The tab will initially display a note indicating that the configurations need to be set in the ThreatQ Configuration page within the Admin tab.



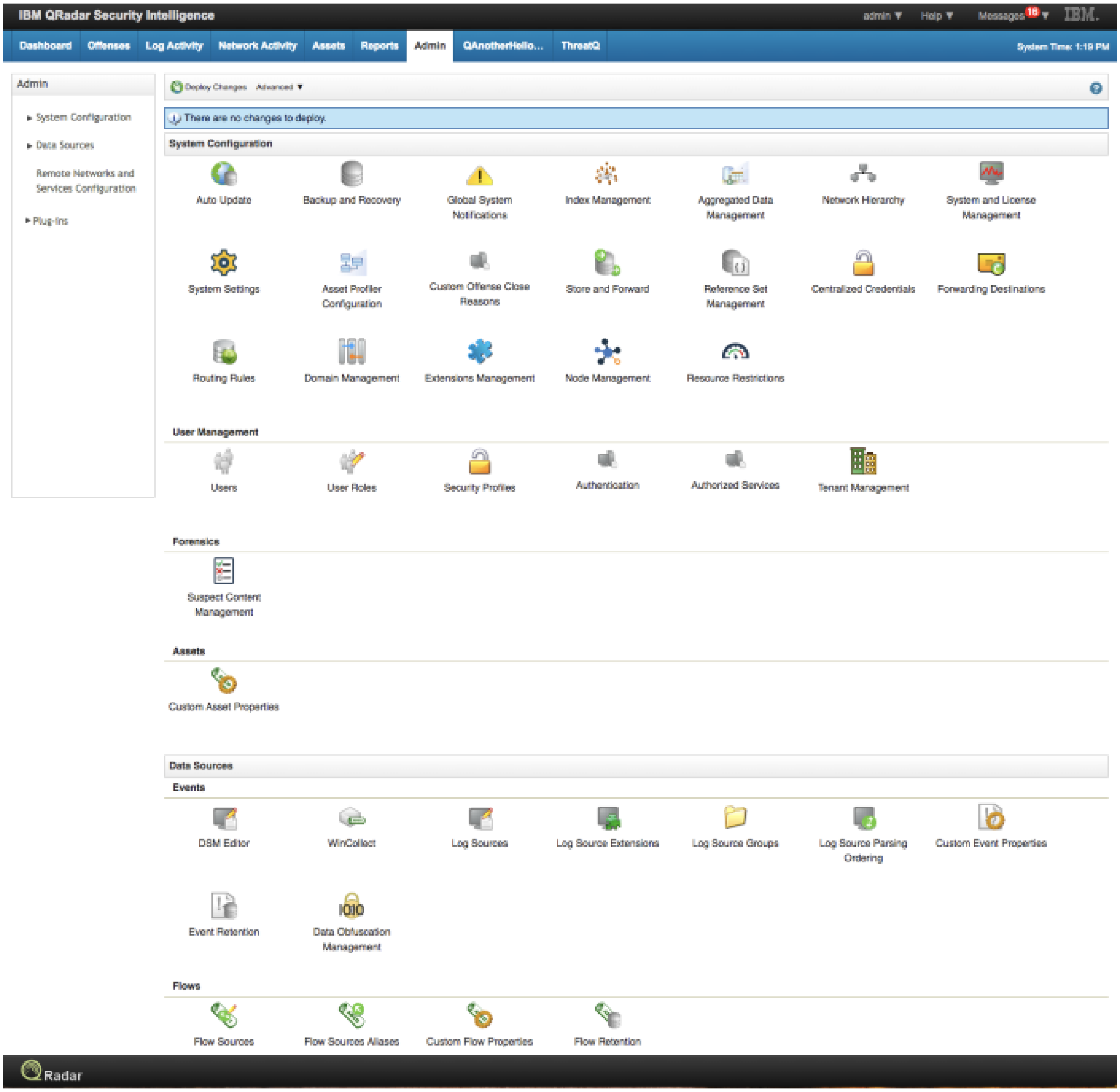
ThreatQ Reference Set Summary

The ThreatQ Application has not been configured.

©2019 ThreatQuotient, All rights reserved.

ThreatQ Configuration

Once installed, configure the following settings within the ThreatQ Configuration page located under Admin/Plug-ins in QRadar.



There are no changes to deploy.

Generating ThreatQ Oauth Credentials

You will need to generate oauth2 client credentials in order to successfully have the app authenticate with ThreatQ. This is performed using the command line of the ThreatQ appliance.

1. SSH into the console for the ThreatQ Appliance.
2. Execute the Oauth2Client command:

   ```
   sudo /var/www/api/artisan threatq:oauth2-client
   --name=Qradar
   ```

   This name can be updated to match your needs.

3. Copy the **client_id** and **client_secret** for use in the configuration of the ThreatQ-QRadar application.

   You should then see a similiar output to below.

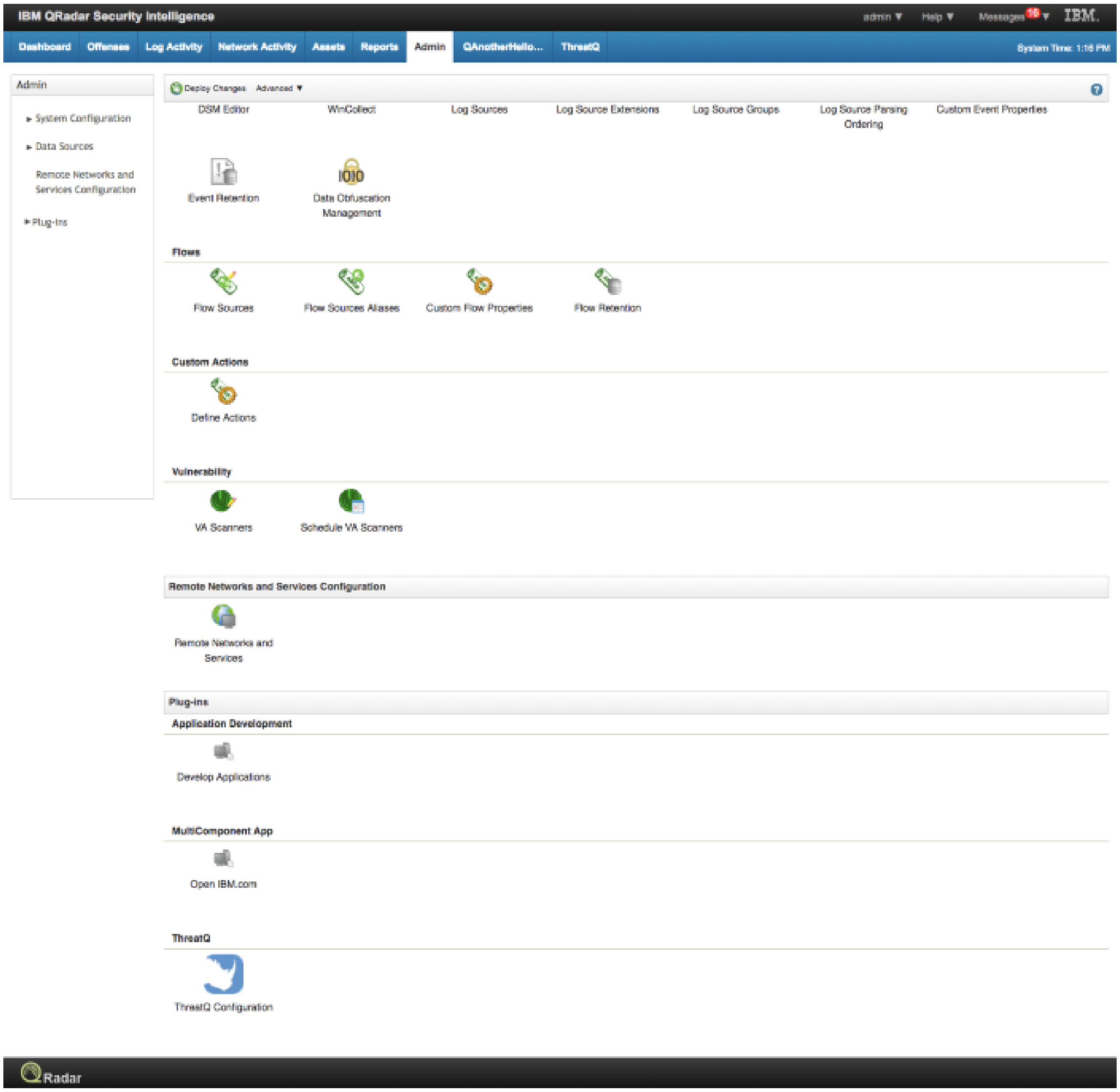
   ```
   # sudo /var/www/api/artisan threatq:oauth2-
   client --name=Qradar

   session_timeout_minutes: 1440

   name: Qradar

   type: private

   client_id: ywewmmyymmm4mde3y2uyzdc2ytk2mjdh

    client_secret:
   ```

```
MjY1OWUyM2RlZTQwZjdiODUxN2MzNGM5ZDZhMTA0MjE1M2V
kOTdlNjUxMTI0MGY0

updated_at: 2020-01-24 20:21:53

 created_at: 2020-01-24 20:21:53

 id: 130
```

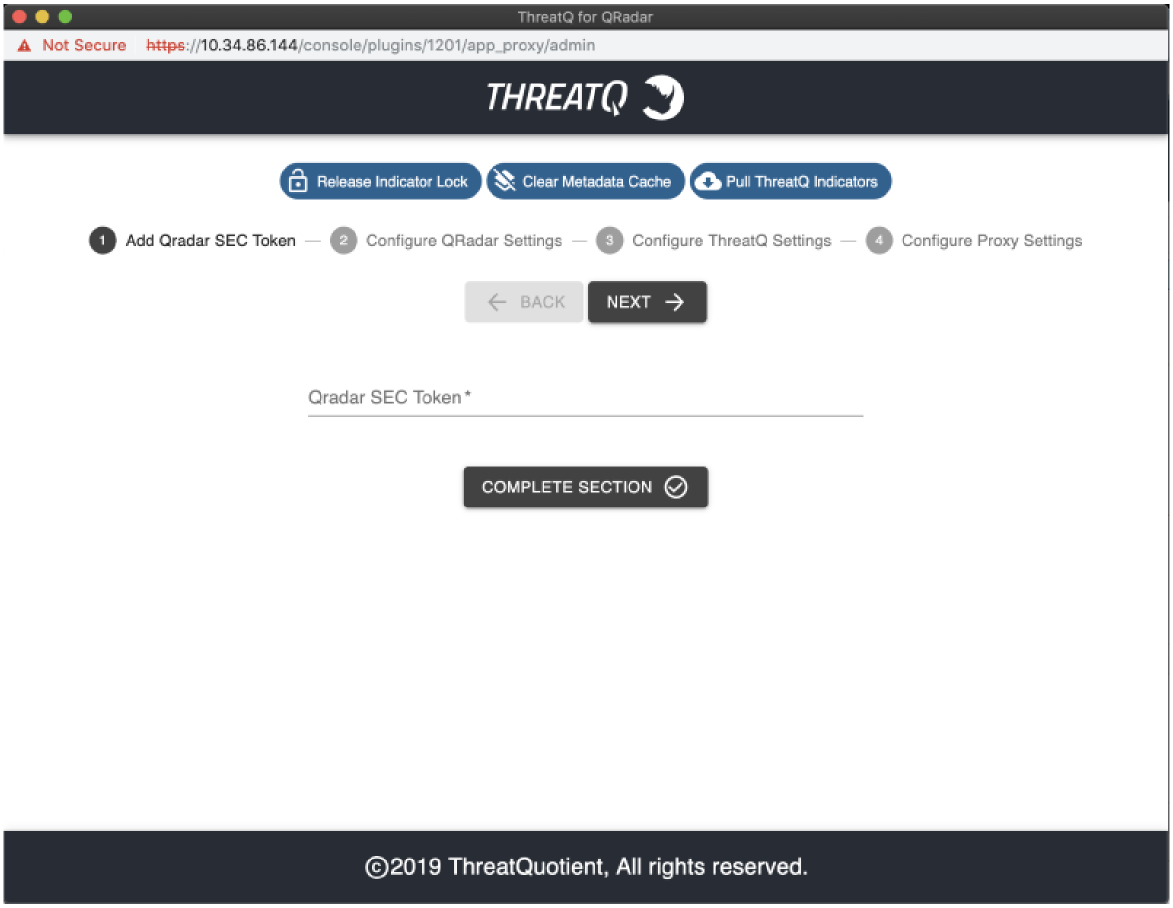
ThreatQ Configuration Page

1. Click on the **Admin** tab in the QRadar SIEM UI.

2. Then scroll down to the Plug-ins section and click on the **ThreatQ** icon under Plug-ins.



A popup window will display the ThreatQ Configuration page.



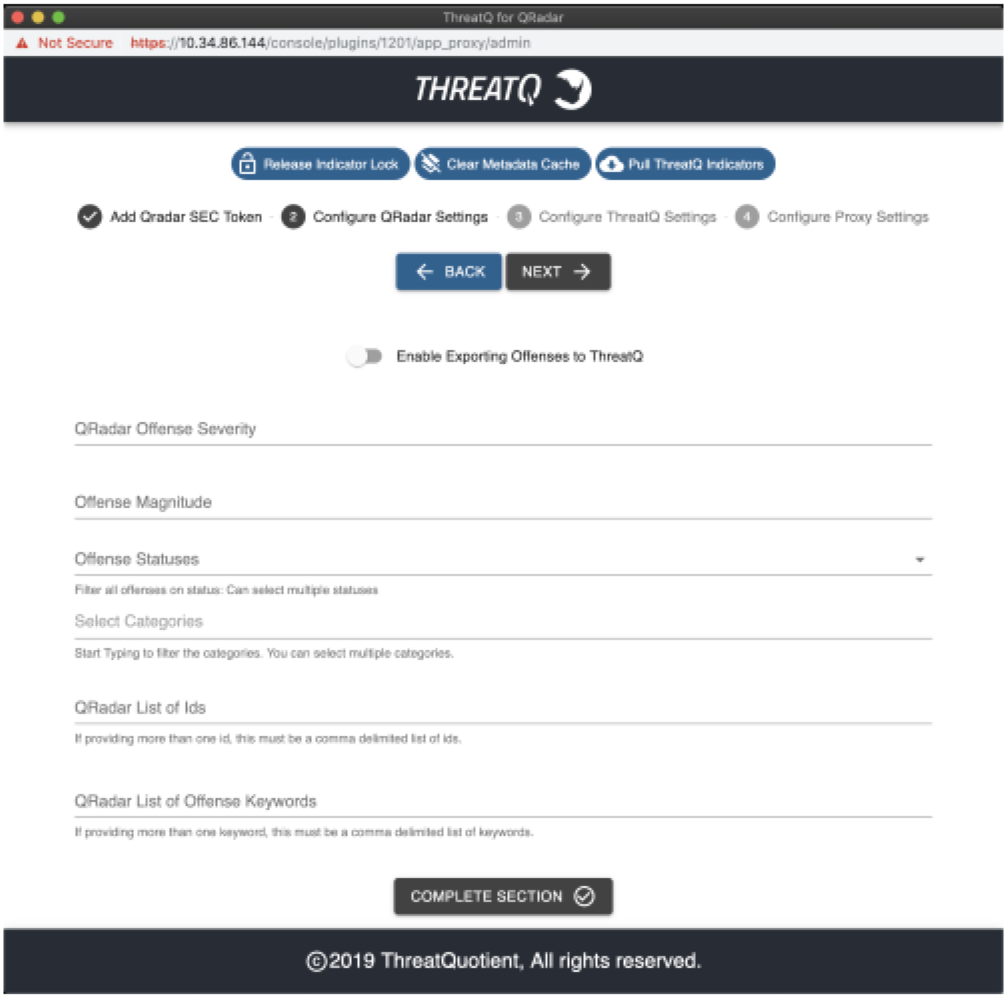
ThreatQ for QRadar
Not Secure https://10.34.86.144/console/plugins/1201/app_proxy/admin
THREATQ
Release Indicator Lock
Clear Metadata Cache
Pull ThreatQ Indicators
1 Add Qradar SEC Token
2 Configure QRadar Settings
3 Configure ThreatQ Settings
4 Configure Proxy Settings
BACK
NEXT
Qradar SEC Token *
COMPLETE SECTION
©2019 ThreatQuotient, All rights reserved.

3. Complete the SEC Token Section:

Field	Description
QRadar SEC Token	The API token for QRadar SIEM.

4. Click **Next** or **Complete Section**.

5. Complete the QRadar Settings Section:



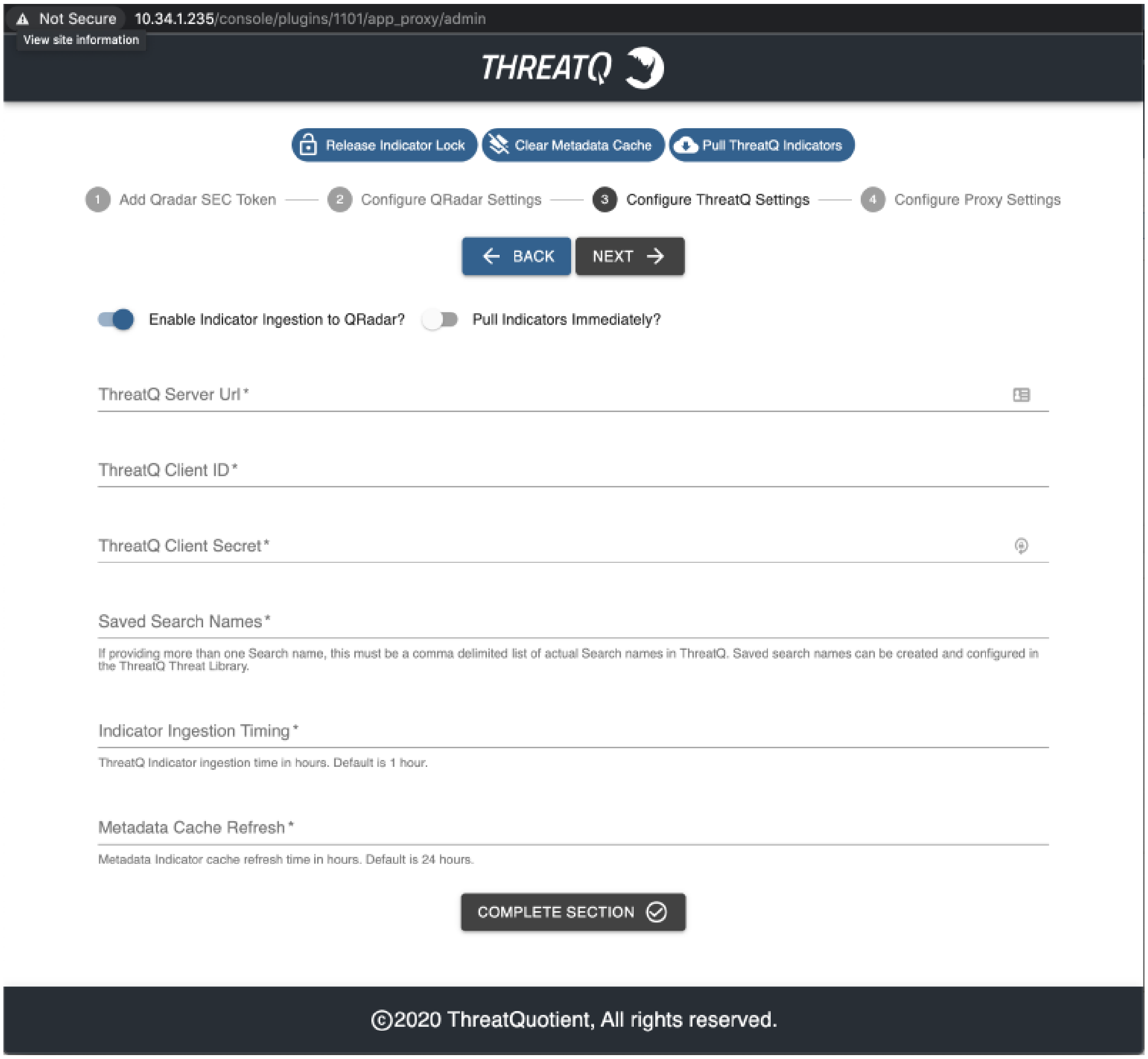
ThreatQ for QRadar
Not Secure https://10.34.86.144/console/plugins/1201/app_proxy/admin
THREATQ
Release Indicator Lock
Clear Metadata Cache
Pull ThreatQ Indicators
Add Qradar SEC Token
2 Configure QRadar Settings
3 Configure ThreatQ Settings
4 Configure Proxy Settings
BACK
NEXT
Enable Exporting Offenses to ThreatQ
QRadar Offense Severity
Offense Magnitude
Offense Statuses
Filter all offenses on status: Can select multiple statuses
Select Categories
Start Typing to filter the categories. You can select multiple categories.
QRadar List of Ids
If providing more than one id, this must be a comma delimited list of ids.
QRadar List of Offense Keywords
If providing more than one keyword, this must be a comma delimited list of keywords.
COMPLETE SECTION
©2019 ThreatQuotient, All rights reserved.

Field	Description
Enable Exporting Offenses to ThreatQ (toggle switch)	If desired, toggling this switch on will enable QRadar to export Offenses to ThreatQ based on the filtering fields defined below.
QRadar Offense Severity	Offenses with severity greater than or equal to this value will be exported to ThreatQ.
Offense Magnitude	Offenses with magnitude greater than or equal to this value will be exported to ThreatQ.

Field	Description
Offense Statuses	Offenses with one of these statuses will be exported to ThreatQ.
Categories	Offenses with one of these categories will be exported to ThreatQ.
QRadar list of IDs	The list of ID's that of offenses that should be included in the Export to ThreatQ.
QRadar List of Offense Keywords	Comma-delimited list of strings that may be found in an Offense Description. Used to provide additional filtering of exporting Offenses to ThreatQ.

6. Click **Next** or **Complete Section**.

7. Complete the ThreatQ Settings section:



Enable Indicator Ingestion to QRadar?

Pull Indicators Immediately?

ThreatQ Server Url *

ThreatQ Client ID *

ThreatQ Client Secret *

Saved Search Names *

If providing more than one Search name, this must be a comma delimited list of actual Search names in ThreatQ. Saved search names can be created and configured in the ThreatQ Threat Library.

Indicator Ingestion Timing *

ThreatQ Indicator ingestion time in hours. Default is 1 hour.

Metadata Cache Refresh *

Metadata Indicator cache refresh time in hours. Default is 24 hours.

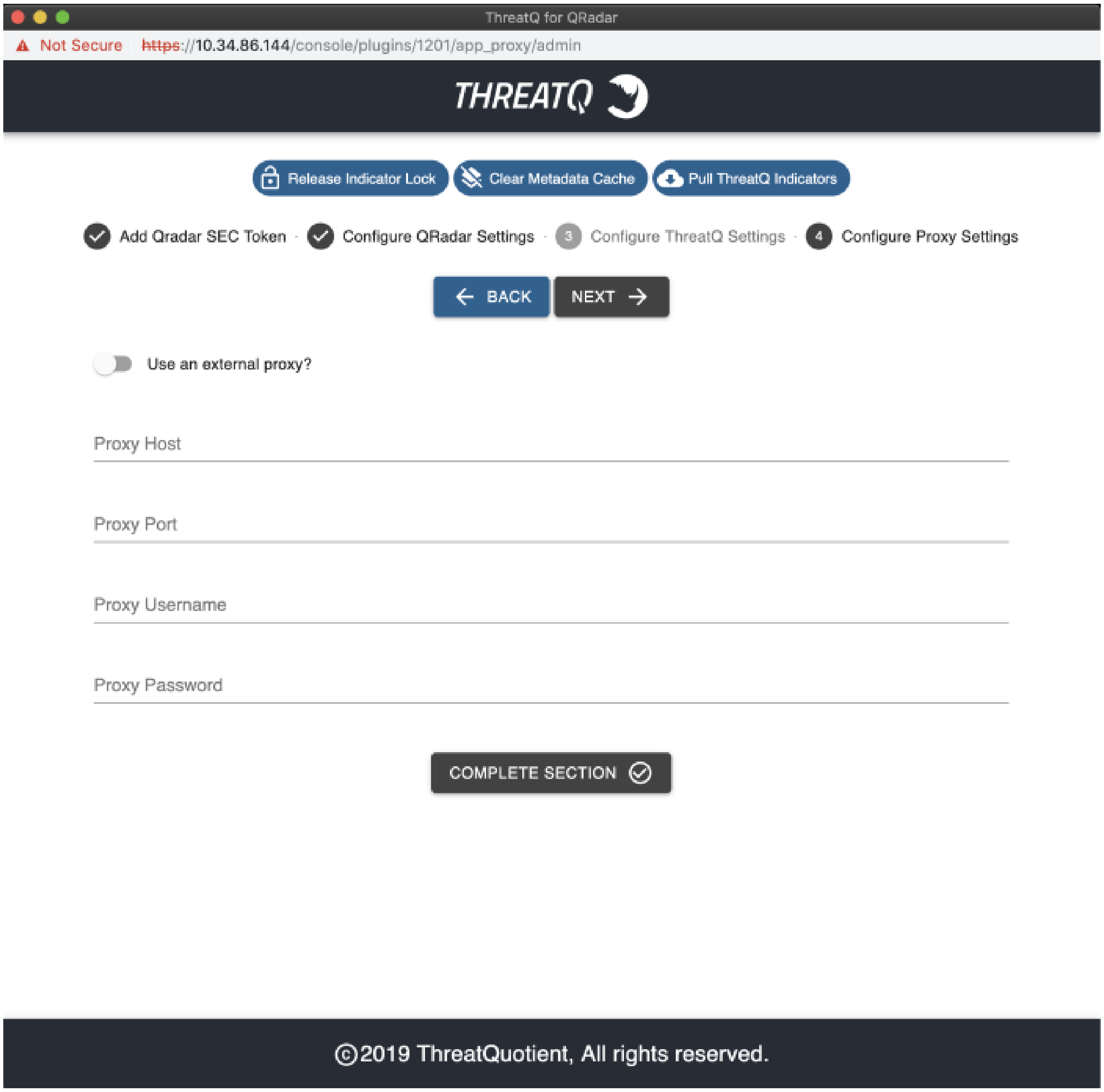
©2020 ThreatQuotient, All rights reserved.

Field	Description
Enable Indicator Ingestion to QRadar	Toggling this switch on will enable importing indicators from ThreatQ to QRadar.
Pull Indicators Immediately	Toggling this switch on will have indicators pulled from ThreatQ to QRadar as soon as the form is saved.
ThreatQ Server	The URL for the applicable ThreatQ server.

Field	Description
URL	
ThreatQ Client ID	The client id generated from the oauth2-client command on the ThreatQ appliance.
ThreatQ Client Secret	The client secret generated from the oauth2-client command on the ThreatQ appliance.
Saved Search Names	Comma-delimited list of the names of saved searches used for import indicators from ThreatQ to QRadar.
Indicator Ingestion Timing	The time (in hours) in which the indicators from ThreatQ to QRadar will be updated. Default is 1 hour.
Metadata Cache Refresh	The time (in hours) in which indicators will remain in the metadata cache before they will start updating information. Default is 24 hours.

8. Click **Next** or **Complete Section**.

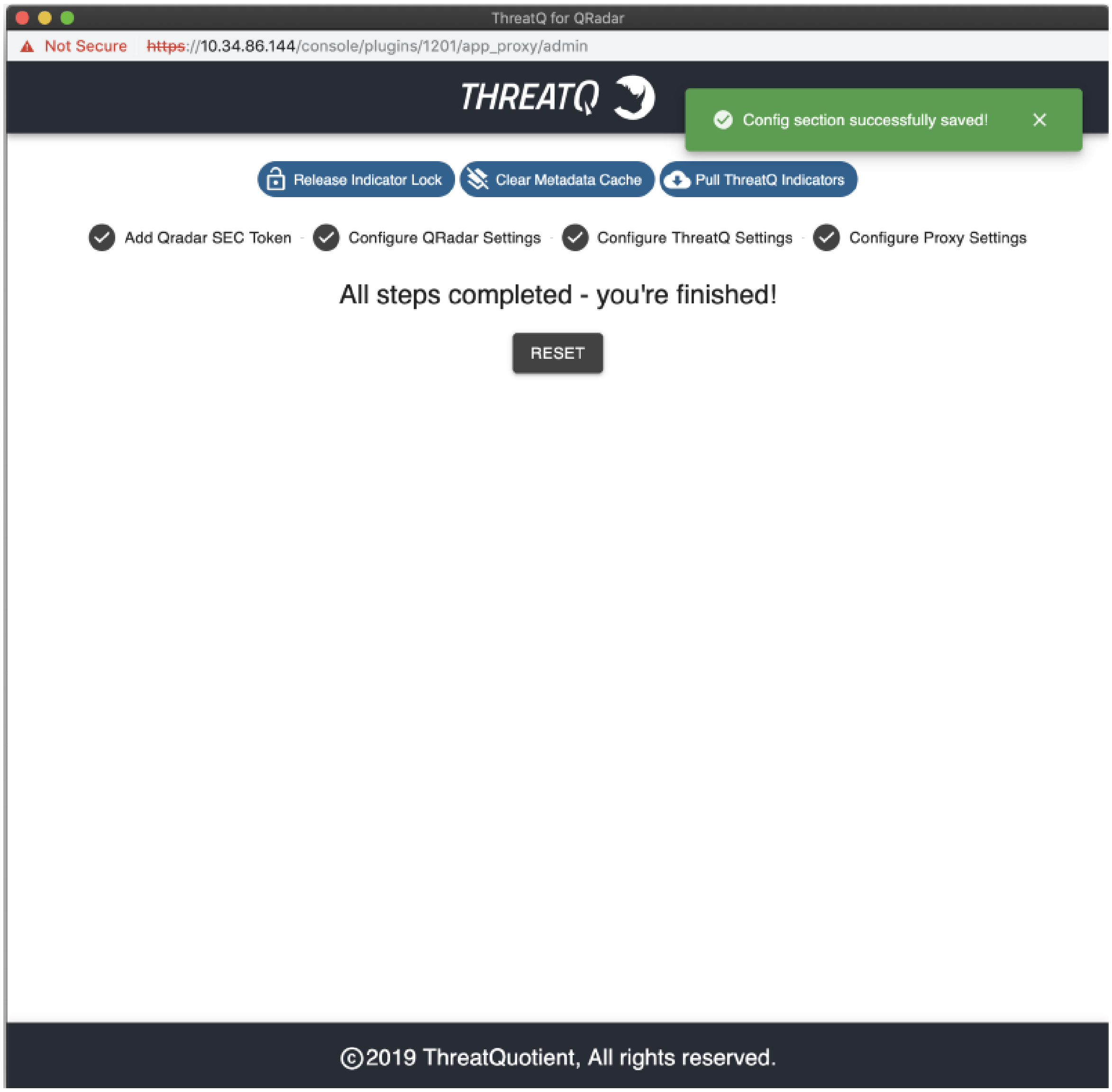
9. Complete the Proxy Configuration section:



Field	Description
Proxy Host	The host url/ip of the proxy server
Proxy Port	The port being used for the proxy server
Proxy Username	The username used for the proxy server
Proxy Password	The password used for the proxy server

10. Click **Next** or **Complete Section**.

    You will receive a message that all steps have been completed.


ThreatQ for QRadar
Not Secure https://10.34.86.144/console/plugins/1201/app_proxy/admin
THREATQ
Release Indicator Lock
Clear Metadata Cache
Pull ThreatQ Indicators
Add Qradar SEC Token
Configure QRadar Settings
Configure ThreatQ Settings
Configure Proxy Settings
RESET
©2019 ThreatQuotient, All rights reserved.

Config section successfully saved!

All steps completed - you're finished!

ThreatQ Configuration Page Buttons

The following buttons are available on the ThreatQ Configuration page:

Button	Action
Release Indicator	This button will release the Indicator lock.

Button	Action
Lock	
Clear Metadata Cache	This button will clear the indicator information stored in the metadata cache.
Pull ThreatQ Indicators	This button will trigger the app to download ThreatQ indicators.

Importing Indicators

IoCs will be imported into QRadar based on the configurations from the Threat Library Search. The parameters for the search, or multiple searches, must be configured in ThreatQ.

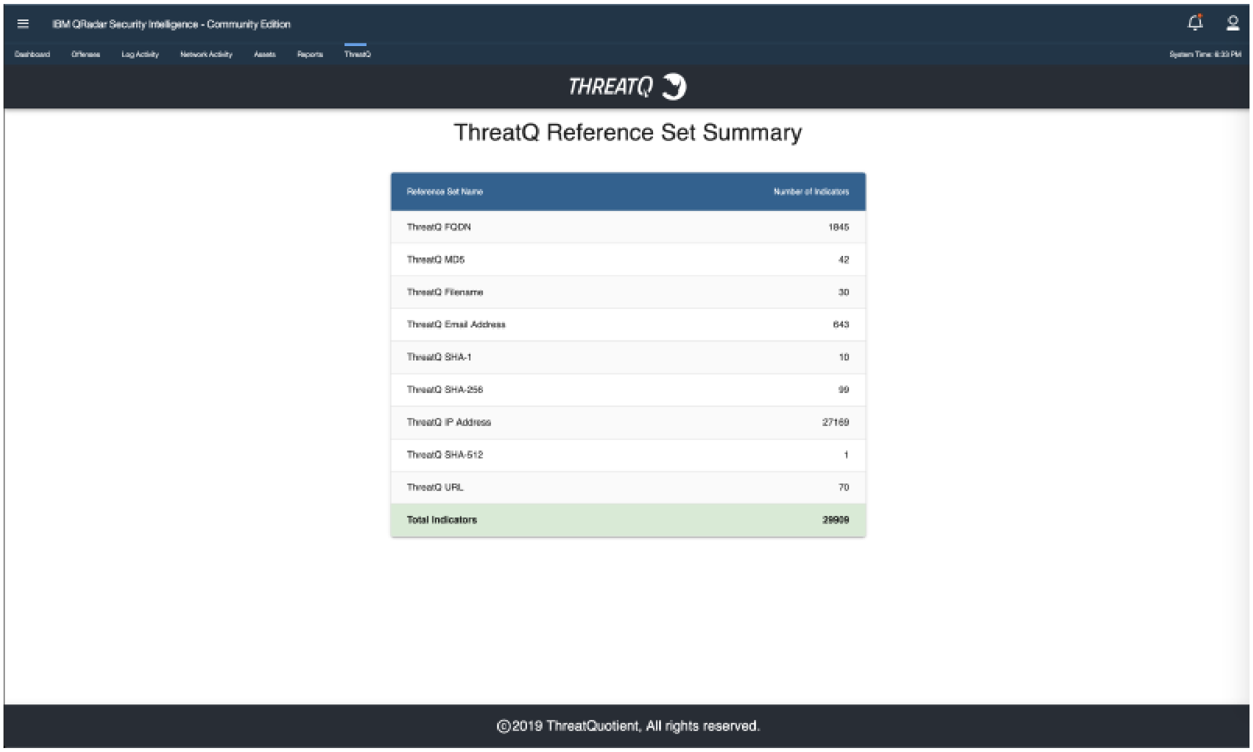
The current parameters for filtering indicators in the Threat Library search are:

- Indicator Type
- Indicator Status
- Indicator Score
- Date Created
- Last Modified
- Import ID
- Relationship
- Source, Tag
- With Attribute
- Without Attribute.

The steps to set up the saved search are provided in the Threat Library Search section.

Once the configurations have been set, Reference Sets will be dynamically created based on the **Indicator Type** found in the search results from ThreatQ. If this is the first run, the ThreatQ App will create Reference Sets for all **Indicator Types** found in the results from the search query. If there are updates in subsequent searches, the App will *purge* the contents of the Reference Set, but the integrity of the Reference Set will remain intact so as to maintain any custom rules that may have been linked to the applicable Reference Set. If there are no updates for a particular set, the Reference Set is left in its current state. ThreatQ Reference Sets may then be used to create custom rules to alert or notify Analysts to a possible threat to their organization.

The ThreatQ tab in the QRadar SIEM UI will then display a table of elements containing the ThreatQ Reference Sets.



ThreatQ Reference Set Summary

Reference Set Name	Number of Indicators
ThreatQ FQDN	1845
ThreatQ MD5	42
ThreatQ Filename	30
ThreatQ Email Address	643
ThreatQ SHA-1	10
ThreatQ SHA-256	99
ThreatQ IP Address	27169
ThreatQ SHA-512	1
ThreatQ URL	70
Total Indicators	**29909**

©2019 ThreatQuotient, All rights reserved.

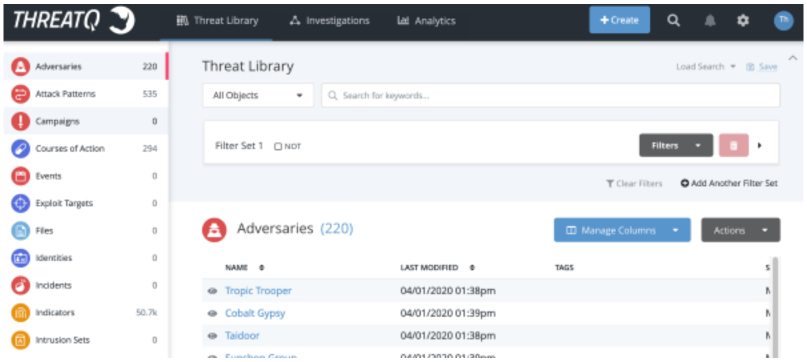
There are 25 possible Reference Sets based on ThreatQ Indicator Types:

- ThreatQ CIDR Block
- ThreatQ CVE
- ThreatQ Email Address
- ThreatQ Email Attachment
- ThreatQ Email Subject
- ThreatQ File Path
- ThreatQ Filename
- ThreatQ FQDN
- ThreatQ Fuzzy Hash
- ThreatQ GOST Hash
- ThreatQ IP Address
- ThreatQ MD5
- ThreatQ Mutex
- ThreatQ Password
- ThreatQ Registry Key
- ThreatQ SHA-1
- ThreatQ SHA-256
- ThreatQ SHA-384
- ThreatQ SHA-512
- ThreatQ String
- ThreatQ URL
- ThreatQ URL Path
- ThreatQ User-agent
- ThreatQ Username
- ThreatQ X-Mailer

Threat Library Search

In you will need to configure a Threat Library search in ThreatQ to use for importing indicators from ThreatQ to QRadar.

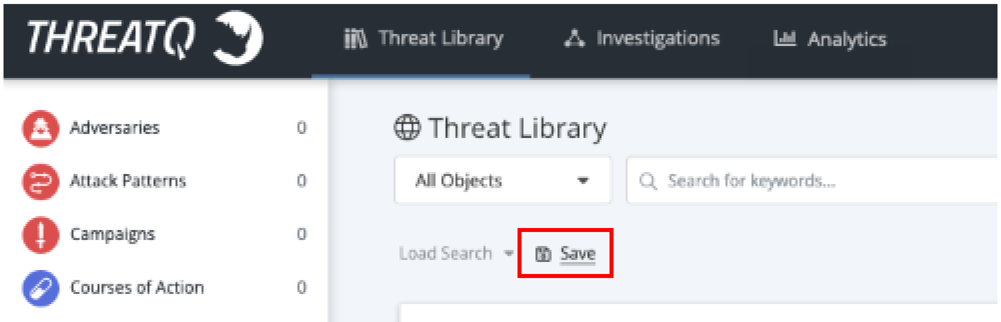
1. Navigate to the **Threat Library** in ThreatQ.
2. Pick your parameters from the supplied filters.



Adversaries	220
Attack Patterns	535
Campaigns	0
Courses of Action	294
Events	0
Exploit Targets	0
Files	0
Identities	0
Incidents	0
Indicators	50.7k
Intrusion Sets	0

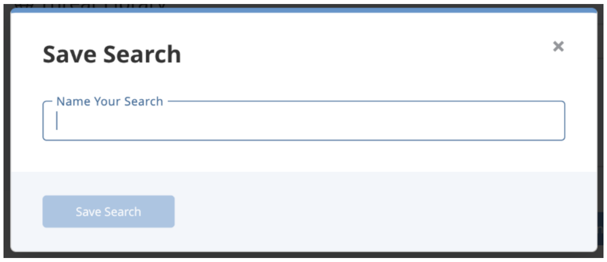
NAME	LAST MODIFIED	TAGS	S
Tropic Trooper	04/01/2020 01:38pm		N
Cobalt Gypsy	04/01/2020 01:39pm		N
Taidoor	04/01/2020 01:38pm		N

3. Click on Save.



Adversaries	0
Attack Patterns	0
Campaigns	0
Courses of Action	0

4. The Save Search dialog box opens.



5. Enter the name for your search and click **Save Search**.

Exporting Offenses/Events

QRadar Analysts may benefit from exporting Offenses and related Events directly to their ThreatQ instance. Once enabled by checking the **Enable Exporting Offenses to ThreatQ** switch, the ThreatQ App provides a number of configurations to filter and parse Offenses to ensure that only the certain Offenses and related Events are exported to ThreatQ.

REST filters

The ThreatQ App for QRadar currently provides three filters which are applied to the REST call to get a list of Offenses for exporting to ThreatQ.

The only pre-configured filter is, **start_time>[last_hour_in_milliseconds]**, which is hard-coded within the App to search for only Offenses that have a start time greater than the value generated by the App. If this is the first time the ThreatQ App has been configured and run, the initial value for the start_time>[last_hour_in_milliseconds] filter is set to search for Offenses that have a start time within the last hour. That value is converted to milliseconds and applied to the start_time> filter in the REST call.

Example: start_time>1509546854000 will search for Offenses that have a start_time greater than Wednesday, November 1, 2017 2:34:14 PM.

After the first run, the ThreatQ App stores the last time it checked for offenses as a value for the key **last_checked_offenses** in the **threatq_app_config.ini** file. The App regularly updates this value and uses this key/value pair for all subsequent calls to get Offenses.

The other filters are:

Filter	Description
QRadar Offense Status	The value from this field is applied to the **status in ([value])** filter and appended to the REST call.
QRadar Offense Severity	The value from this field is applied to **the severity>=[value]** filter and appended to the REST call.

Filter	Description
QRadar Offense Magnitude	The value from this field is applied to the **magnitude>=[value]** filter and appended to the REST call.
QRadar Category	The value from this field is applied to the categories contains **"[value]"** filter and appended to the REST call, for each selected category.

Offense Description Parsing

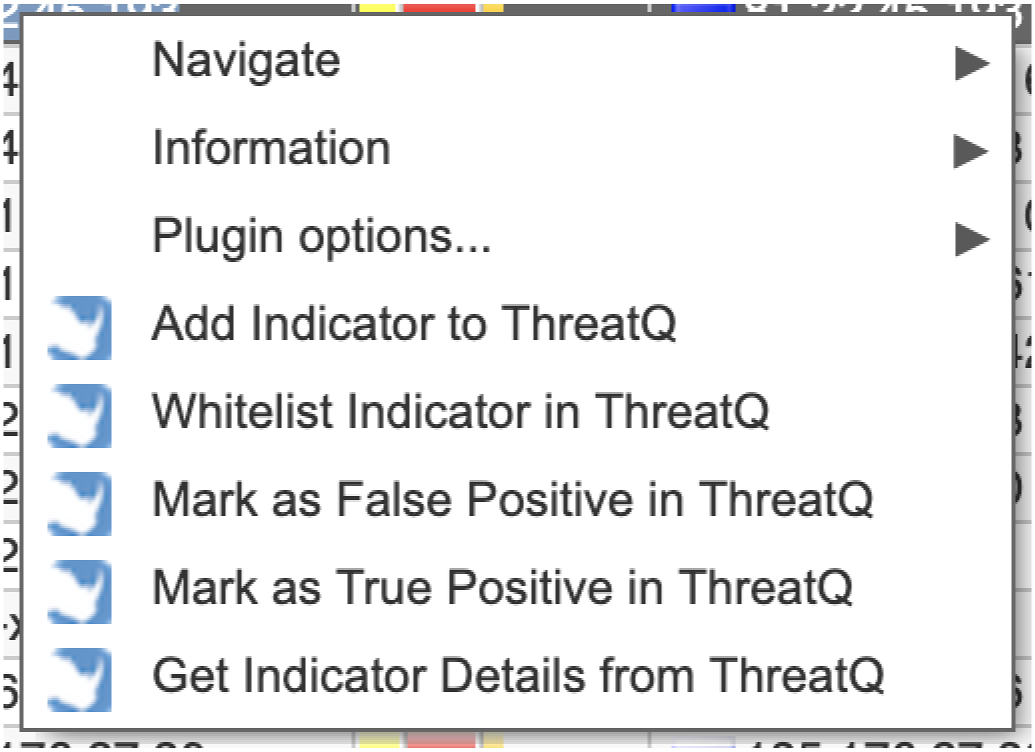
The last configuration related to the exporting of Offenses is the **QRadar List of Offense Keywords**. Also found within the ThreatQ Configuration page in the QRadar UI, this field is an optional field which takes a one or more comma-delimited strings that may be used to search within the text of a QRadar Offense **Description** field. If this field is set, only Offenses with one or more of the keywords in its **Description** and matching the filters described above will be exported to ThreatQ.

Configuration Store

The ThreatQ Configuration page in the QRadar SIEM's UI will collect and store the necessary fields in the **store/threatq_app_config.ini** within the App' docker container.

Right-Click Actions

ThreatQ for QRadar provides four right-click actions within various pages of the QRadar SIEM's UI.

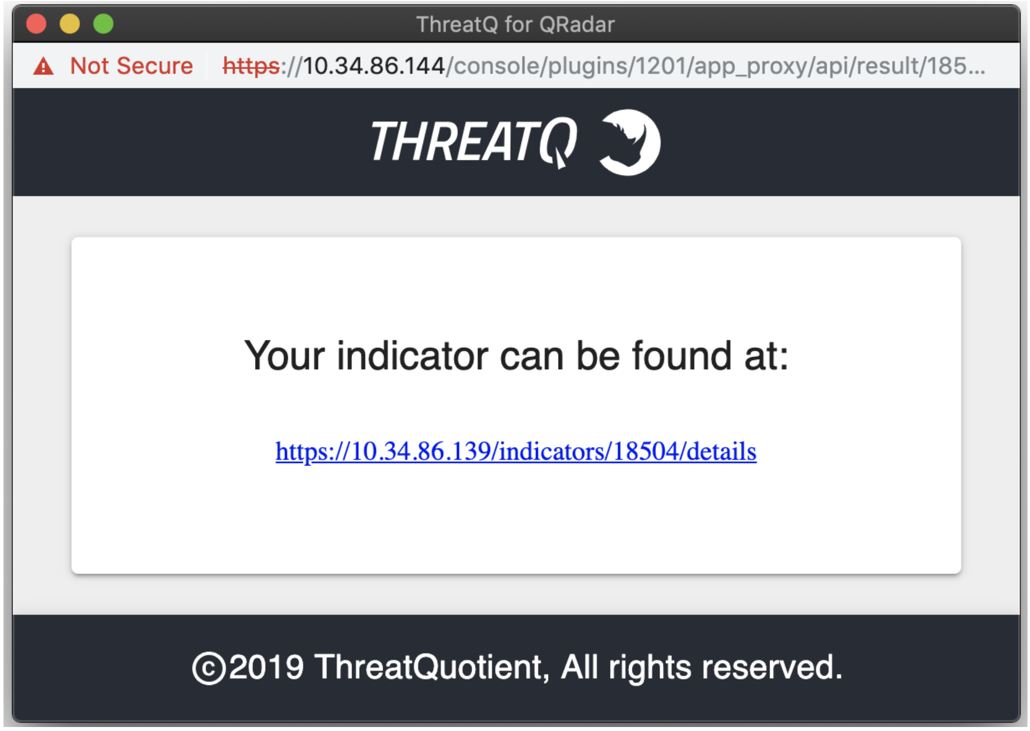

Navigate ▶
Information ▶
Plugin options... ▶
Add Indicator to ThreatQ
Whitelist Indicator in ThreatQ
Mark as False Positive in ThreatQ
Mark as True Positive in ThreatQ
Get Indicator Details from ThreatQ

Actions include:

Option	Action
Add Indicator to ThreatQ	Adds an IP Address to ThreatQ, if it does not already exist, and/or returns a URL to the ThreatQ Indicator Details page.
Whitelist Indicator in ThreatQ	Changes the status of an indicator that already exists in ThreatQ to **Whitelisted**.
Mark as False Positive in ThreatQ	Changes the status of an indicator that already exists in ThreatQ to **Review** and adds a **False Positive** attribute with a value of **Yes**.
Mark as True Positive in ThreatQ	Changes the status of an indicator that already exists in ThreatQ to **Review** and adds a **True Positive** attribute with a value of **Yes**.
Get Indicator	Returns the URL for the ThreatQ Indicator Details page.

Option	Action
Details from ThreatQ	

Each right-click action will present a popup window with a URL to the applicable Indicator Details page in ThreatQ



ThreatQ for QRadar
Not Secure
~~https~~://10.34.86.144/console/plugins/1201/app_proxy/api/result/185...
THREATQ

Your indicator can be found at:

https://10.34.86.139/indicators/18504/details

©2019 ThreatQuotient, All rights reserved.

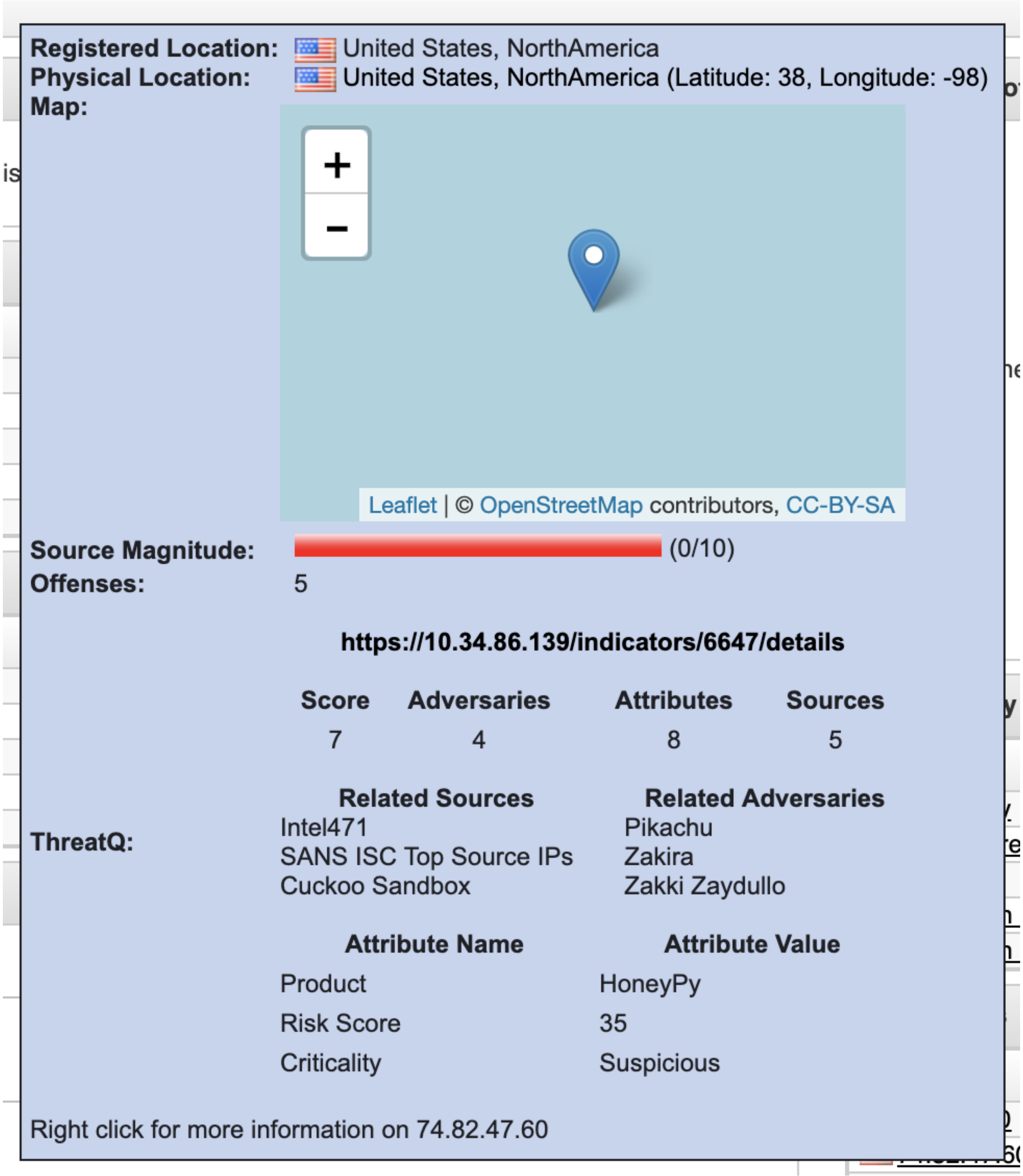
.

Metadata Provider

The ThreatQ app for QRadar now contains a metadata provider which will provide contextual information on hover of an indicator. Currently this works for **IP Address**, **Ariel:URL**, and **Ariel:Hostname** indicator types.

If contextual information is found in ThreatQ on the indicator, the following information will be displayed:

- ThreatQ Url for the indicator
- ThreatQ Score
- Total number of Adversaries
- Total number of Attributes
- Total number of Sources
- The first three Adversaries
- The first three Attributes
- The first three Sources



Registered Location: United States, NorthAmerica

Physical Location: United States, NorthAmerica (Latitude: 38, Longitude: -98)

Map:

Source Magnitude: (0/10)

Offenses: 5

ThreatQ:

https://10.34.86.139/indicators/6647/details

Score	Adversaries	Attributes	Sources
7	4	8	5

Related Sources	Related Adversaries
Intel471	Pikachu
SANS ISC Top Source IPs	Zakira
Cuckoo Sandbox	Zakki Zaydullo

Attribute Name	Attribute Value
Product	HoneyPy
Risk Score	35
Criticality	Suspicious

Right click for more information on 74.82.47.60

Execution

Upon installation and configuration. This application will automatically execute and continue to run on the QRadar SIEM.

There are two threads running: one to get indicators and one to get offenses. The thread to check indicators will run every hour. The thread to get offenses will check every 2 minutes until it finds and processes offenses.

Troubleshooting

The first step to troubleshooting a QRadar App is to retrieve the App ID for the affected application.

There are two methods to retrieve the App ID:

- Command Line Retrieval
- Interactive API for Developers

Command Line Retrieval

The App ID may be retrieved by SSH'ing into the console and running the following command:

Version	Command
QRadar SIEM's >= 7.3.0 <= 7.3.1 console	`shell $/opt/qradar/support/qapp_utils_730.py ps`
QRadar SIEM's >= 7.3.2 console	`shell $/opt/qradar/support/recon ps`

The output will display similar to:

```
shell Collecting app data........ Complete!
shell ID NAME PORT CONTAINER IMAGE STATUS
```

Interactive API for Developers

The second method of retrieving the App ID is from the **Interactive API for Developers** under the **Help** menu in the QRadar SIEM's UI.

1. Browse to the api_doc endpoint in QRadar by clicking Help/Interactive API for Developers
2. Open folder representing the API version, **8.0** (The API version, in this case 8.0, may be different depending on the version of the QRadar SIEM),
3. Locate the **/guiappframework** folder, click on it to expand the contents,
4. Locate the **/applications** endpoint and click on it to select it
5. Scroll to bottom of the page and click on **Try It Now**

In the Response Body of the request, there should be a list of all of the Applications that are installed. Find the appropriate application and copy down the value from the **application_id** key.

The full output from this command may also provide useful information in troubleshooting application issues and should be copied for review.

The output from the recon command provides some additional information with regards to how to access the application's docker container, as well as, to the state of the application. The value from the Container field in this output can be used to execute the following docker command in order to review any application specific logs:

```
shell $docker exec -it <container_id> bash
```

Once in the App's docker environment, the source code for the App is located in the app/ directory. All App specific logs are located in store/log/ directory.

⚠ For any application issues, ThreatQ TIS will be requesting the following:

1. The standard output from this command:

   ```
   shell $/opt/qradar/support/qapp_utils_730.py ps
   ```

   or

   ```
   shell $/opt/qradar/support/recon ps
   ```

2. Collect and send the following log files from the Application's docker container to ThreatQ:

   ```
   shell store/log/app.log store/log/startup.log
   store/log/supervisord.log
   ```

These steps were based on the guidance from IBM's App Troubleshooting page. Additional information can be found here: IBM App Troubleshooting.

Change Log

Version	Details
v1.2.3	• Fixed a bug where indicators with special characters failed to sync.
v1.2.2	• Fixed a bug with the pull indicators immediately option. • Updated indicator pull from ThreatQ to be more memory efficient. • Updated indicator pull from ThreatQ with a search differential.
v1.2.1	• Changed authentication from username/password auth to client credentials. • Removed the source name from the QRadar app, since this is now configurable via the client credentials command. • Added extra logging to provide better error reporting.
v1.2.0	• Redesigned User interface to improve user interactivity and looks • Added a new option to the right click context menu for 'Mark True Positive' • New filters for QRadar Offenses based on Status, Magnitude, Category, and Offense ID • Events created in ThreatQ by QRadar now contain a link back to the QRadar Offense. • Ability to import a larger initial offense set. Ability to change the source of indicators and events added to ThreatQ. • Http/Https are appended to URL type indicators if a scheme is not present.

Version	Details
	• Source name for QRadar app can now be modified. • Other bug fixes and performance enhancements
v1.1.0	• Metadata provider for IP Address Indicators and Ariel:URL/Ariel:Hostname • Shows a link to the ThreatQ indicator if found • Shows the indicator's Score • Shows the number of Adversaries, Attributes, and Sources. • Shows the first 3 Adversaries, Attributes, and Sources. • Ability to clear the Metadata cache of all stored indicators. • Ability to change the timing in which the Metadata cache will start to refresh indicator data. • Ingestion of indicators from ThreatQ to QRadar can be enabled/disabled. • The timing for indicator ingestion from ThreatQ can be changed. • A ThreatQ Threat Library Search can now be leveraged to import indicators into QRadar • Proxy settings for communicating from QRadar to ThreatQ have been added. • Other bug fixes and performance enhancements.
v1.0.0	• Right Click Context Menu which provides the following actions:

Version	Details
	• Add indicators to ThreatQ • Add Indicator to ThreatQ Whitelist • Mark Indicator as False Positive in ThreatQ • Check ThreatQ for Indicator Details • Dashboard that shows how many indicators of what types are in QRadar from ThreatQ