

ThreatQuotient



ThreatQ Integration with HBase Guide

Version 1.0.0

Monday, December 14, 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.

Copyright © 2020 ThreatQuotient, Inc.

All rights reserved. This document and the software product it describes are licensed for use under a software license agreement. Reproduction or printing of this document is permitted in accordance with the license agreement.

Contents

Warning and Disclaimer	2
Contents	3
Versioning.....	4
Introduction	5
Deployment Prerequisites	6
Networking.....	6
Hardware/Software/Virtual Appliance(s).....	7
ThreatQ.....	8
Install a Signed Certificate	8
Create a ThreatQ Export.....	9
NiFi	12
Import XML template	12
Add Template to the Canvas	12
JVM Heap maximum.....	13
User permissions.....	14
NiFi Flow Configuration.....	16
Fetch indicators from ThreatQ	16
Split the JSON array into separate documents	18
Parse the data into JSON	18
Convert attributes to JSON	18
Put JSON Documents into HBase Table	19
Option 1	20
Option 2	20
Change Log	22

Versioning

- Integration Version: 1.0.0
- ThreatQ Version: 4.30.0 or greater

Introduction

The ThreatQ Integration with HBase integration fetches indicators from a ThreatQ instance and ingests them into an HBase table.

Deployment Prerequisites

The following personnel and dependencies have been identified to ensure for a smooth deployment of the agreed-upon products and/or services.

Networking

- All required firewall rules are applied to allow for communications to, from, or between the applicable products, services, and/or API endpoints.
Specifically:
 - Ports are opened and firewall rules configured between ThreatQ and NiFi
 - Ports are opened and firewall rules configured for communication among all applications in the Hadoop cluster
 - At a minimum all ports listed in these documents should be opened in a Cloudera Hadoop deployment:
 - Hadoop Data Platform:
<https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/administration/content/configuring-ports.html>
 - Hadoop Data Flow:
<https://docs.cloudera.com/HDPDocuments/HDF3/HDF-3.5.1/nifi-configuration-best-practices/content/port-configuration.html>
- Network access control modifications, proxy and firewall configurations to allow for the necessary communications between internal and external tools and data feeds
- If applicable, the customer will inform ThreatQuotient of any custom network configurations that would require modification(s) to the standard ThreatQ configuration to include, but not limited to:
 - DNS resolution
 - Proxy configuration
 - Routing tables

Hardware/Software/Virtual Appliance(s)

- All ThreatQuotient equipment/virtual appliances are provisioned, online, and in service
- All third-party products and/or services are installed, configured, and operating normally
- If ThreatQ is already installed:
 - ThreatQuotient engineers will require the username/password for command line root access to the appliance via SSH port 22
 - ThreatQuotient engineers will require the username/password for the maintenance account in order to access the appliance via the UI

ThreatQ

Install a Signed Certificate

The ThreatQ virtual appliance is supplied with a self-signed certificate, which is not trusted by NiFi, and as such NiFi will not initiate traffic from ThreatQ. In order for the ingestion process to work, the customer will need to install a CA-signed certificate on ThreatQ.

Create a ThreatQ Export

For the detailed steps on how to create an export in ThreatQ, please visit this Helpcenter topic, and scroll to Adding an Export:

https://helpcenter.threatq.com/index.htm#t=ThreatQ_Platform%2FExports%2FManaging_Exports.htm

1. Use the following values to fill out the export template:

Field	Description
Type of information you would like to export?	Indicators
Output Type	Text/plain
Special Parameters	<code>indicator.status=Send to HBase&indicator.deleted=N&indicator.type=IP Address</code>

Output Format Template:

```
{ldelim}

  "data":

    [

      {foreach $data as $indicator}

        {ldelim}

          "indicator": "{$indicator.value}",

          "type": "{$indicator.type}",

          "status": "{$indicator.status}",

          "score": {$indicator.score},

          "created_at": "{$indicator.created_at}",

          "updated_at": "{$indicator.updated_at}"

        {rdelim}

        {if !$indicator.last},{/if}

      {/foreach}

    ]

  {rdelim}
```

- Click on **Save Settings**, and when completed, it should look similar to the snapshot below.

Output Format

Type of information you would like to export?

Indicators

Output type

text/plain

Special Parameters (optional)

indicator.status=Send to HBase&indicator.deleted=N&indicator.type=IP Address

Provide URL Parameters to further refine information being exported: [See examples.](#)

Output Format Template

Insert Variable

```
{ldelim}
"data":
[
{foreach $data as $indicator}
{ldelim}
"indicator": "{$indicator.value}",
"type": "{$indicator.type}",
"status": "{$indicator.status}",
"score": {$indicator.score},
"created_at": "{$indicator.created_at}",
"updated_at": "{$indicator.updated_at}"
{rdelim}
{if !$indicator.last},{/if}
{/foreach}
```

Save Settings

Cancel

- Click on the generated Export URL. This will execute the export process in the backend and will list all the indicators that match the export's special parameters.
- Copy the URL and paste it in the URL value in the `GetHTTP` processor in `NiFi`.

ThreatQ Integration with HBase Guide
Version 1.0.0

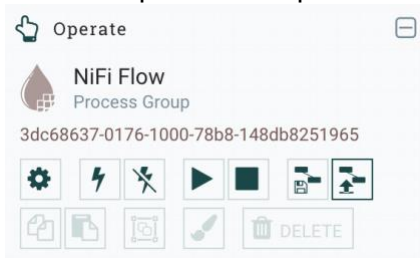
11

NiFi

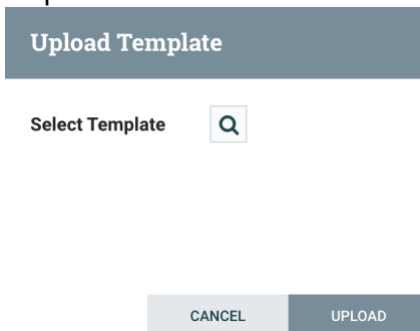
This section discusses the prerequisites for NiFi.

Import XML template

1. Navigate to the NiFi UI. On the instruments menu, click on the right-most button Upload Template.



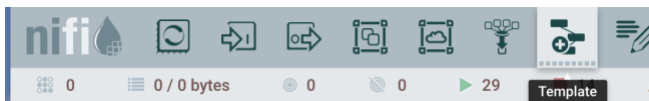
2. Next, click on the magnifying glass to the right of Select Template. This will open the window that will allow you to navigate to the XML template to import.



Add Template to the Canvas

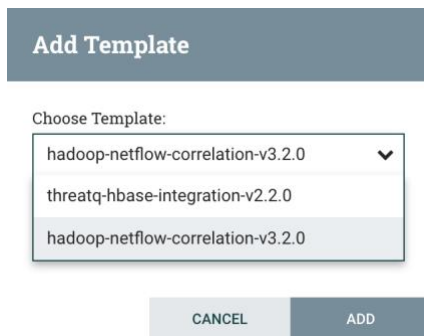
To add the template to your canvas:

1. go to the NiFi UI, and drag a Template from the instruments onto the canvas.



This will open a modal window with a dropdown from which you can choose the template that was just imported.

2. Select the template threatq-hbase-integration-<version> which will be used for ingesting threat intel from ThreatQ into HBase.



A dialog box titled "Add Template" with a "Choose Template:" label. Below the label is a dropdown menu showing three options: "hadoop-netflow-correlation-v3.2.0", "threatq-hbase-integration-v2.2.0", and "hadoop-netflow-correlation-v3.2.0". At the bottom are two buttons: "CANCEL" and "ADD".

JVM Heap maximum

The default memory allocation for NiFi is 512MB, which needs to be increased to at least 4GB, but the recommended is 8GB.

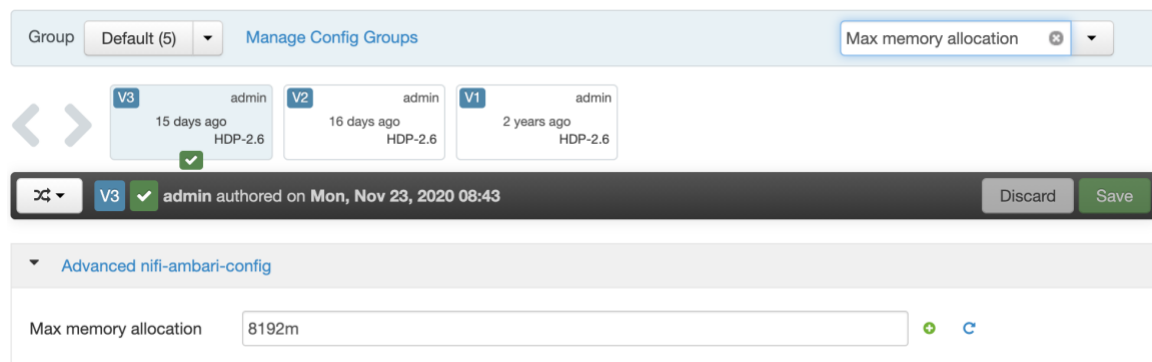
To increase it:

1. Navigate to Ambari
2. Click on the NiFi application, and then click on Configs for NiFi.
3. Search for "Max memory allocation", as shown in the snapshot below.
4. Change the value to 8192m and save it.

After the changes are made, Ambari will prompt you to restart all the NiFi services.

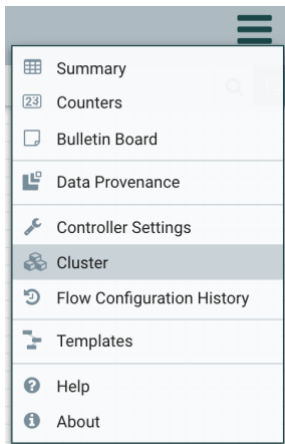
5. Click on restart and wait for the application to restart.

After the restart is complete, validate the amount of resources used by NiFi.



A screenshot of the Ambari web interface showing the configuration for NiFi. The top bar shows "Group: Default (5)" and "Manage Config Groups". A search bar on the right contains "Max memory allocation". Below this, there are three version cards: V3 (15 days ago, HDP-2.6), V2 (16 days ago, HDP-2.6), and V1 (2 years ago, HDP-2.6). A banner at the bottom indicates "V3 admin authored on Mon, Nov 23, 2020 08:43" with "Discard" and "Save" buttons. The main configuration area is titled "Advanced nifi-ambari-config" and shows the "Max memory allocation" field set to "8192m".

In Ambari-managed Hadoop clusters, this can be done by navigating to the **NiFi UI**, click on the **hamburger menu** in the upper right corner, and then click on the **Cluster** settings.



On the NiFi resources, navigate to the **JVM** tab, which shows the **Java** heap usage.



NiFi Cluster

Displaying 1 of 1

Filter by address

Node Address	Heap Max	Heap Total	Heap Used	Heap Utilization	Non-Heap Total	Non-Heap Used	GC	Uptime
hdp2.threatq.lan:9090	8 GB	3.23 GB	2.84 GB	35.0%	332.52 MB	315.26 MB		181:21:25.996

This is also a good way to determine the optimal memory needed for NiFi. Run the NiFi flow multiple times with different loads, and make sure that the Heap Utilization metric on the JVM tab stays below 70%. That leaves a buffer to handle occasional flows with peak memory demand.

User permissions

NiFi runs as the user specified in the `bootstrap.conf` file, the content of which is accessible via Ambari. This user should have the proper permissions to:

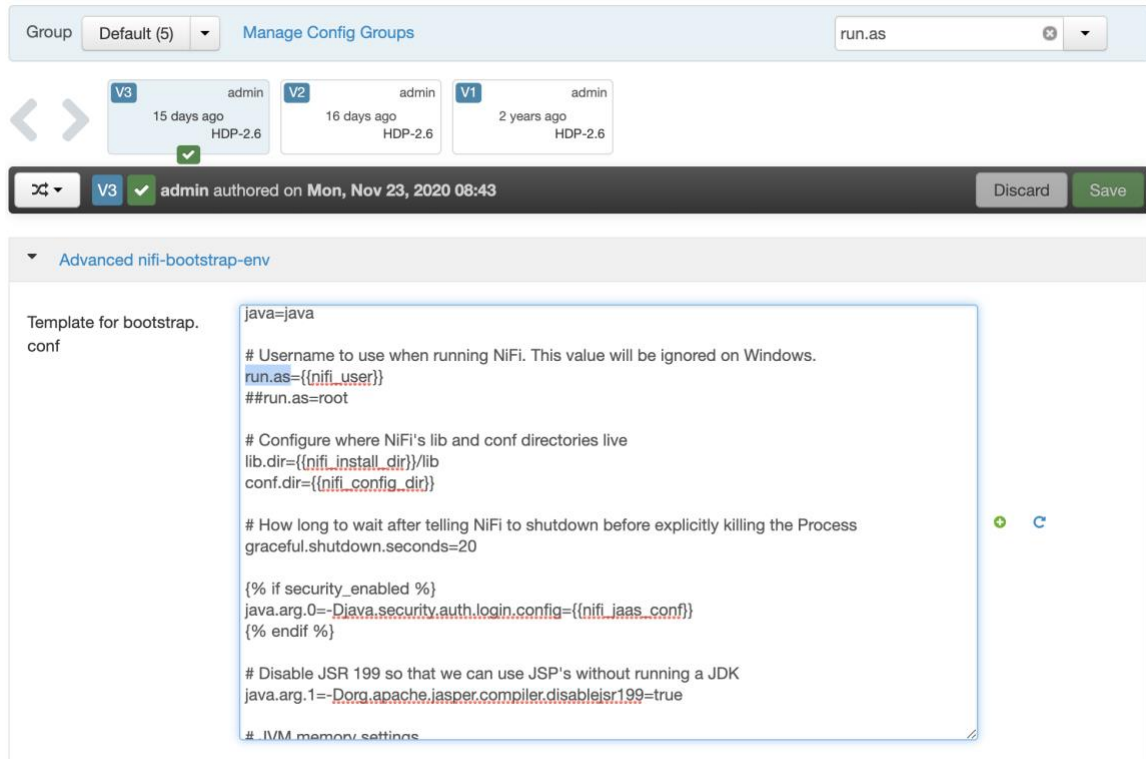
- Parse the NetFlow files with `nfdump`
- Write to HDFS
- Publish to Kafka topics
- Perform lookups against HBase

Perform the following steps if you need to change the user:

1. Navigate to the NiFi configuration in Ambari
2. Enter **run.as** in the search.

This should bring up the content of the `bootstrap.conf` file.

3. Make the required changes and restart all the services Ambari lists. In the example below, NiFi runs as the “nifi” user.



Version	User	Time	HDP Version
V3	admin	15 days ago	HDP-2.6
V2	admin	16 days ago	HDP-2.6
V1	admin	2 years ago	HDP-2.6

admin authored on Mon, Nov 23, 2020 08:43

Discard Save

Advanced nifi-bootstrap-env

Template for bootstrap.conf

```
java=java

# Username to use when running NiFi. This value will be ignored on Windows.
run.as={{nifi_user}}
##run.as=root

# Configure where NiFi's lib and conf directories live
lib.dir={{nifi_install_dir}}/lib
conf.dir={{nifi_config_dir}}

# How long to wait after telling NiFi to shutdown before explicitly killing the Process
graceful.shutdown.seconds=20

{% if security_enabled %}
java.arg.0=-Djava.security.auth.login.config={{nifi_jaas_conf}}
{% endif %}

# Disable JSR 199 so that we can use JSP's without running a JDK
java.arg.1=-Dorg.apache.jasper.compiler.disablejsr199=true

# JVM memory settings
```

NiFi Flow Configuration

The following sections details the configuration of each processors in the NiFi flow that ingests indicators from ThreatQ, parses and writes them to an HBase table. The flow has also been provided separately as an XML file to import into NiFi.

Fetch indicators from ThreatQ

1. Select the **GetHTTP** processor, and configure it as shown below.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field +			
Property	Value		
URL		https://tis-research-xl.threatq.com/api/export/bc974d8f23...	
Filename		threatq_iocs.log	
SSL Context Service		StandardSSLContextService-ThreatQ →	
Username		No value set	
Password		No value set	
Connection Timeout		30 sec	
Data Timeout		30 sec	
User Agent		No value set	
Accept Content-Type		No value set	
Follow Redirects		false	
Redirect Cookie Policy		default	
Proxy Configuration Service		No value set	
Proxy Host		No value set	
Proxy Port		No value set	

IMPORTANT: ThreatQ needs to have a CA-signed certificate in order for this to work, because NiFi checks the certificate.

2. Change the following value:

Vaule	Description
URL	Change the value to the URL your ThreatQ instance has generated.

3. Select the **StandardSSLContextService** for the SSL Context Service property.

Stream ThreatQ to HBase directly Configuration

GENERAL

CONTROLLER SERVICES

	Name ^	Type	Bundle	State	Scope	
	HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1.9.0	org.apache.nifi - nifi-hbase_1_1_2-client-ser...	Disabled	NiFi Flow	
	StandardSSLContextService-ThreatQ	StandardSSLContextService 1.9.0	org.apache.nifi - nifi-ssl-context-service-nar	Invalid	Stream ThreatQ to HBase directly	

4. Click on the **arrow** to the right, which will bring you to the NiFi Flow Configuration screen.
5. Click on the wheel for **the StandardSSLContextService** on the NiFi Flow Configuration screen.
6. After the controller details open, configure it as shown below, and enable it by clicking on the lightning bolt.

Note: For Truststore Password use `changeit`.

Value	Description
Truststore Filename	Change the path to the Truststore in your environment.
Truststore Password	The default password is changeit. If you have changed it, use the new one.

SETTINGS

PROPERTIES

COMMENTS

Required field +

Property	Value
Keystore Filename	? No value set
Keystore Password	? No value set
Key Password	? No value set
Keystore Type	? No value set
Truststore Filename	? /opt/jdk1.8.0_261/jre/lib/security/cacerts
Truststore Password	? Sensitive value set
Truststore Type	? JKS
TLS Protocol	? TLS

Split the JSON array into separate documents

Select the **SplitJson** processor, and configure it as shown below. Below is the minimum list of properties that needs to be configured.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			+
Property		Value	
JsonPath Expression	?	\$.data.*	
Null Value Representation	?	empty string	

Parse the data into JSON

Select the **EvaluateJsonPath** processor, and configure it as shown below. Below is the minimum list of properties that needs to be configured.

There is no need to change any of the values in the template for the processor, unless more fields are added in the ThreatQ export.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			+
Property		Value	
Destination	?	flowfile-attribute	
Return Type	?	auto-detect	
Path Not Found Behavior	?	ignore	
Null Value Representation	?	empty string	
created_at	?	\$.created_at	🗑
id	?	\$.indicator	🗑
indicator	?	\$.indicator	🗑
score	?	\$.score	🗑
status	?	\$.status	🗑
type	?	\$.type	🗑
updated_at	?	\$.updated_at	🗑

Convert attributes to JSON

Select the **AttributesToJSON** processor, and configure it as shown below. Below is the minimum list of properties that needs to be configured.

There is no need to change any of the values in the template for the processor, unless more fields are added in the ThreatQ export. If more fields are added, add their names to the Attributes List in the processor.

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field +

Property	Value
Attributes List	id,indicator,type,status,score,created_at,updated_at
Attributes Regular Expression	No value set
Destination	flowfile-content
Include Core Attributes	false
Null Value	false

Put JSON Documents into HBase Table

1. Select the **PutHBaseJSON** processor and configure the minimum list of properties that needs to be configured.

Value	Description
Table Name	The name of the table to which the threat intel is stored in HBase.
Column Family	Column family in the threat intel HBase table.

Your screen should resemble the screenshot example below.

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field +

Property	Value
HBase Client Service	HBase_1_1_2_ClientService →
Table Name	threatqdata
Row Identifier	No value set
Row Identifier Field Name	id
Row Identifier Encoding Strategy	String
Column Family	msg
Timestamp	No value set
Batch Size	25
Complex Field Strategy	Text
Field Encoding Strategy	String

2. Click on the wheel for the **HBase_1_1_2_ClientService** on the NiFi Flow Configuration screen. After the controller details opens, configure it as shown below, and enable it by clicking on the lightning bolt.

NiFi Flow Configuration

GENERAL CONTROLLER SERVICES

Name	Type	Bundle	State	Scope
HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1.9.0	org.apache.nifi-nifi-hbase_1_1_2-client-ser...	Enabling	NiFi Flow

SETTINGS PROPERTIES COMMENTS

Required field

Property	Value
Hadoop Configuration Files	No value set
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
ZooKeeper Quorum	hdp1.threatq.ian,hdp2.threatq.ian,hdp3.threatq.ian,
ZooKeeper Client Port	2181
ZooKeeper ZNode Parent	/hbase-unsecure
HBase Client Retries	35
Phoenix Client JAR Location	No value set

This controller requires the HBase configuration details. There are two options for the configuration:

Option 1

Fill out the value for Hadoop Configuration Files with comma-separated list of Hadoop configuration file paths, such as hbase-site.xml, and core-site.xml for Kerberos, including full paths to the files. The configuration files should be located in a local path on the NiFi instance or mounted to a drive NiFi can read from.

Example: "/usr/local/hbase/conf/hbase-site.xml,/usr/local/hbase/conf/core-site.xml"

Option 2

Get the config details from the HBase config file hbase-site.xml which is usually located somewhere on the host that has HBase installed. Get the following details from that file and fill them out in the controller window:

Field	Description
Zookeeper Quorum	Comma-separated list of the Zookeeper hosts (from hbase-site.xml)
Zookeeper Client Port	The port Zookeeper is listening on (from hbase-site.xml)

Field	Description
Zookeeper ZNode Parent	Zookeeper ZNode Parent (from hbase-site.xml).
HBase Client Retries	HBase retries (from hbase-site.xml).

Change Log

Version	Details
1.0.0	Initial Release