

ThreatQuotient



ThreatQ Integration with Hadoop Guide

Version 3.3.0

Wednesday, December 16, 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.....	5
Introduction	6
Deployment Prerequisites	7
Networking.....	7
Hardware/Software/Virtual Appliance(s).....	8
NiFi	9
Import XML Template	9
Add the Template to the Canvas.....	9
JVM Heap Maximum.....	10
User Permissions.....	11
Required packages	12
Kafka	13
HBase	14
HDFS	15
User Permissions.....	15
Create a Folder to Store the Processed Logs from NiFi	15
ELK stack.....	16
NiFi process flow	16
List Files in a Directory (<i>only for NetFlow Streaming Correlation</i>).....	17
Convert NetFlow Files to NiFi Flowfiles (<i>Only for NetFlow Streaming Correlation</i>)..	18
Convert NetFlow Files to NiFi Flowfiles (<i>Only for NetFlow Historical Correlation</i>)...	19
Remove the Footer with Summary Statistics	20
Add a Schema to the Flowfile	20
Read Selected Columns from NetFlow flowfile	21
Split the Flowfile into Multiple Flowfiles	23

Match to HBase	24
Distribute HBase Lookup Load.....	24
Lookup Records in HBase (20x processors)	25
Merge Content	29
Update the Data from HBase	29
Save the Results to HDFS.....	30
Matched to HBase?.....	32
Convert File to JSON	32
Split JSON into Individual Files.....	33
Publish to Kafka Topic.....	34
Push to Elasticsearch Index (Optional).....	35
Adding NetFlow Log Data to Hive Table (Optional).....	37
Infer Avro Schema	37
Convert CSV to Avro format.....	38
Put in Hive Table	39
Change Log	41

Versioning

- Integration Version: 3.3.0
- ThreatQ Version: 4.41.0 or greater

Introduction

This document describes the implementation of a NiFi process flow used to read and parse NetFlow log files as well as match the destination IP address for each record against a threat intelligence table in HBase.

The resulting modified dataset is then written to HDFS, and optionally to a Hive table. Additionally, the NetFlow records that matched to HBase are published to a Kafka topic, and read by an Elasticsearch instance downstream.

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/configuration-best-practices.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)

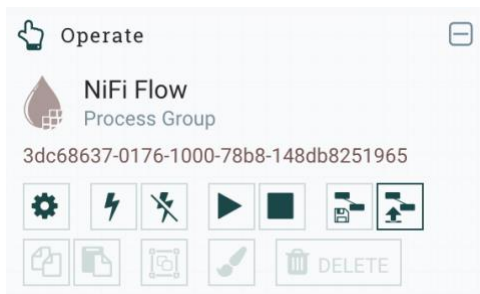
- All ThreatQuotient equipment/virtual appliances are provisioned, online, and in-service
- All third-party products and/or services are installed, configured, and operating normally
- If ThreatQ is already installed, ThreatQuotient engineers will require:
 - The username/password for command line root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

NiFi

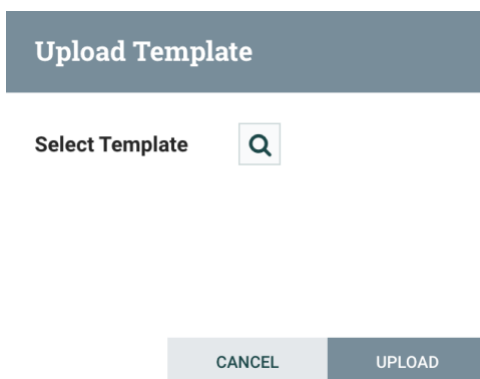
This section discusses the prerequisites for NiFi.

Import XML Template

Navigate to the NiFi UI. On the Instruments menu, click the right-most button, *Upload Template*.

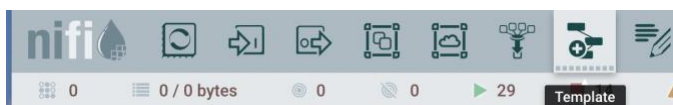


Next, click the magnifying glass to the right of *Select Template*. This opens the window that allows you to navigate to the XML template to import.



Add the Template to the Canvas

To add the template to your canvas, go to the NiFi UI and drag a *Template* from the instruments onto the canvas.



This opens a modal window with a dropdown from which you can choose the template that was just imported. Select the template *threatq-netflow-correlation-<version>* which parses NetFlow files and matches their content against threat intelligence in HBase.

Add Template

Choose Template:

hadoop-netflow-correlation-v3.2.0 ▼

threatq-hbase-integration-v2.2.0

hadoop-netflow-correlation-v3.2.0

CANCEL

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, navigate to Ambari, click the NiFi application, and then click *Configs* for NiFi. Search for “*Max memory allocation*”, as shown in the snapshot below. Change the value to *8192m* and save it. After the changes are made, Ambari prompts you to restart all the NiFi services. Click restart and wait for the application to restart.

Group: Default (5) Manage Config Groups Max memory allocation ▼

< >

V3 admin 15 days ago HDP-2.6 ✓

V2 admin 16 days ago HDP-2.6

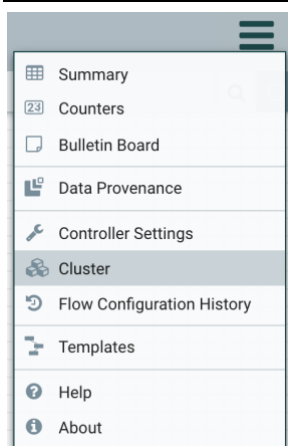
V1 admin 2 years ago HDP-2.6

⚙ V3 ✓ admin authored on Mon, Nov 23, 2020 08:43 Discard Save

▼ Advanced nifi-ambari-config

Max memory allocation 8192m + c

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



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.



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

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

If you need to change the user, navigate to the NiFi configuration in Ambari, and in the search, type *“run.as”*. This should bring up the content of the `bootstrap.conf` file. Make the required changes and restart all the services Ambari asks for. In the example below, NiFi runs as the *“nifi”* user.

Group
Default (5)
Manage Config Groups
run.as

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

V3 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

```

Required packages

Install nfdump on the instance that runs NiFi. This package is used for extracting the content of the NetFlow files.

```
sudo yum install -y epel-release
```

```
sudo yum install -y nfdump
```

After installation of the package, validate that nfdump works. The example below extracts the first 300 lines from a NetFlow file.

```
nfdump -r /path/to/netflow/logs/nfcapd.<timestamp> -c 300
```

Kafka

In Kafka, create a topic with the same name as in the PutKafka processor in NiFi. Refer to the section NiFi Flow Configuration for the topic in the PutKafka processor.

Below is a simple example that creates a topic called “incidents” and lists all Kafka topics. Use the parameters (partitions, replication factor, Zookeeper host/port) applicable to your environment. For a production environment, we recommend using a higher partitions and replication factor.

```
ssh <user>@<Kafka Host>

./bin/kafka-topics.sh --zookeeper <Zookeeper Host>:<Zookeeper
Port> --create --topic <Kafka Topic> --partitions 1 --
replication-factor 1

./bin/kafka-topics.sh --zookeeper <Zookeeper Host>:<Zookeeper
Port> --list
```

For more information refer to these documents:

- Kafka in a Cloudera Hadoop deployment:
https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.5/kafka-working-with-topics/content/creating_a_kafka_topic.html
- General Kafka documentation:
<https://kafka.apache.org/documentation/>

HBase

The NiFi flow described in this document ingests NetFlow log data, parses the individual records from that file, and searches for each destination IP address in the NetFlow records within an HBase table.

At a minimum, the NiFi flow expects that there is an HBase table already configured to ingest Threat Intelligence from ThreatQ. In the provided XML template, the HBase table has the following configuration:

- Name: *threatqdata*
- Column family: *msg*
- Column qualifier: *status*

If the HBase table in the customer's environment has different properties, the above values need to be modified in NiFi,

The proper configuration of HBase is key to optimizing the NiFi flow and reducing the time it takes to search an HBase table for the destination IP address from each NetFlow log record. HBase configuration and fine tuning is beyond the scope of this document. For further information, please consult your IT team, and if applicable, your Hadoop deployment vendor.

HDFS

User Permissions

The NiFi flow contains a processor that writes files to HDFS. For the processor to work properly, it is important that the user NiFi runs under has read and write permissions on HDFS. If the user does not have those permissions, the processor will fail with a Permission Denied error.

Create a Folder to Store the Processed Logs from NiFi

Create a folder in HDFS to store the NetFlow logs processed using the NiFi flow. The path to that folder is required to configure the HDFS processor as described in the [Save the Results to HDFS](#) section.

ELK stack

The NiFi flow uses Kafka to send all NetFlow records that match to the Threat Intelligence in HBase into an Elastic index. In order for this work, the integration requires that the ELK stack – Elasticsearch, Logstash and Kibana - is deployed and configured. Additionally, firewall rules should be in place to allow traffic to flow between the Kafka host(s) and Elastic.

The only configuration required is adding an ingestion source for Logstash. The following are the steps to add a new source:

```
ssh <user>@<ELK Host>
```

```
vi /etc/logstash/conf.d/kafka_input.conf
```

Enter the following configurations for Input, Output and Filter. This template contains the minimum required configurations to establish the connection between Elastic and Kafka.

```
input {
  kafka {
    bootstrap_servers => "<Kafka Broker 1>:<Port>,<Kafka Broker 2>:<Port>,<Kafka Broker 3>:<Port>"
    topics => ["<Kafka Topic>"]
    group_id => "<Kafka Topic Consumer Group ID>"
  }
}
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "<Elastic index for the data>"
    manage_template => false
    user => <username>
    password => "<password>"
  }
}
filter {
  json {
    source => "message"
  }
}
```

```
systemctl restart logstash
```

NiFi process flow

The following sections detail the configuration of each processor in the NiFi flow that parses NetFlow logs and matches them against threat intelligence in HBase. The flow has also been provided separately as an XML file to import into NiFi. The

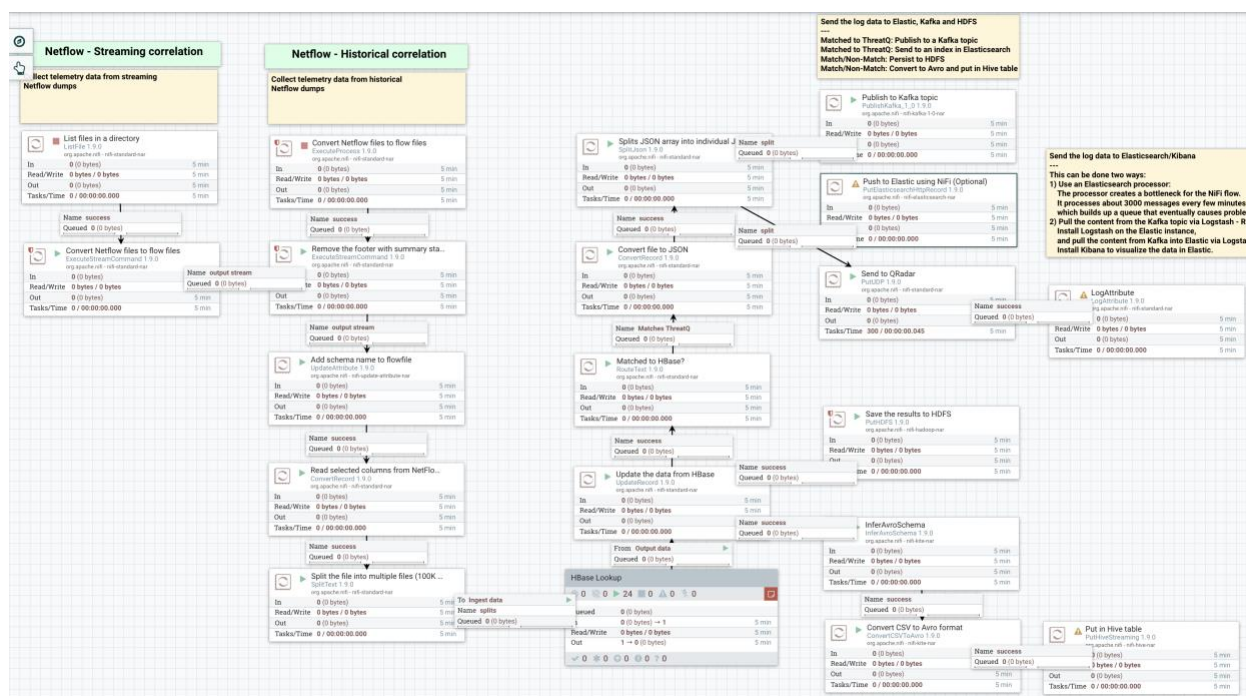
After the NiFi template has been imported, it is displayed on the canvas as a Processor Group. Double click it to open.

NetFlow logs - Streaming and Historical Correlation summary. The summary bar shows 0 errors, 0 warnings, 29 successful connections, 10 failed connections, 2 alerts, and 0 disabled connections. Below this, a table displays the following statistics:

Category	Value	Time
Queued	0 (0 bytes)	
In	0 (0 bytes) → 0	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 → 0 (0 bytes)	5 min

At the bottom, a status bar shows 1 checked, 0 failed, 0 pending, 1 warning, 0 error, and 0 unknown.

The process flow in the template will be similar to this diagram:



List Files in a Directory (*only for NetFlow Streaming Correlation*)

ThreatQ Integration with Hadoop Guide
Version 3.3.0

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field +

Property	Value
Input Directory	? /opt/nifi-data/data/netflow_bein/streaming
Listing Strategy	? Tracking Timestamps
Recurse Subdirectories	? false
Input Directory Location	? Local
File Filter	? [\\].*
Path Filter	? No value set
Include File Attributes	? true
Minimum File Age	? 2 sec
Maximum File Age	? No value set
Minimum File Size	? 0 B
Maximum File Size	? No value set
Ignore Hidden Files	? true
Target System Timestamp Precision	? Auto Detect

Change the following values:

Property	Description
Input Directory	Change the path for the <i>Input Directory</i> to your environment.
Recurse Subdirectories	If files are located in subfolders, set this to value to <i>true</i> .
Input Directory Location	If the files are not on the NiFi instance, provide hostname or IP address.

Convert NetFlow Files to NiFi Flowfiles

(Only for NetFlow Streaming Correlation)

Select the *ExecuteStreamCommand* processor, and configure it as shown below. This dumps the content of a NetFlow file to a NiFi flowfile.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Command Arguments	-o csv -r \${absolute.path}\${filename}		
Command Path	/usr/bin/nfdump		
Ignore STDIN	false		
Working Directory	No value set		
Argument Delimiter			
Output Destination Attribute	No value set		
Max Attribute Length	256		

Change the following values:

Property	Description
Command Path	If needed, change the path to the <i>nfdump</i> executable

Convert NetFlow Files to NiFi Flowfiles (Only for NetFlow Historical Correlation)

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

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Command	/usr/bin/nfdump		
Command Arguments	-o csv -R /opt/nifi-data/data/netflow_bein/streaming -c 20000		
Batch Duration	40 s		
Redirect Error Stream	false		
Working Directory	No value set		
Argument Delimiter			

Change the following values:

Property	Description
Command	If needed, change the path to the <i>nfdump</i> executable.
Command Arguments	Change the path to the NetFlow log files. Additionally, the argument "-c 20000" is used for

Property	Description
	testing the flow. Remove the argument with its value when the flow is ready to run in production.

Remove the Footer with Summary Statistics

Select the *ExecuteStreamCommand* processor, and configure it as shown below. This removes the last four lines of the output from *nfdump*.

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field

Property	Value
Command Arguments	-n -4
Command Path	/usr/bin/head
Ignore STDIN	false
Working Directory	No value set
Argument Delimiter	
Output Destination Attribute	No value set
Max Attribute Length	256

Change the following values:

Property	Description
Command Path	If needed, change the path to the <i>head</i> executable

Add a Schema to the Flowfile

Select the *UpdateAttribute* processor, and configure it as shown below. This processor adds a schema name to the flowfile. Do not change the schema name.

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS






Required field

Property	Value
Delete Attributes Expression	No value set
Store State	Do not store state
Stateful Variables Initial Value	No value set
Cache Value Lookup Cache Size	100
schema.name	netflowAggregateSchema

Read Selected Columns from NetFlow flowfile

Select the *ConvertRecord* processor, and configure it as shown below. The processor reads the NetFlow flowfile using a *CSVReader* controller and writes out a new flowfile via the *CSVRecordSetWriter* controller. The new flowfile contains only a subset of the original fields. The fields that are includes are listed in the *AvroSchemaRegistry* controller below.

1. To enable the *CSVReader-Netflow schema* controller, click the arrow on the right. That will take you to another screen that lists all the controllers.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field +			
Property	Value		
Record Reader		CSVReader-Netflow schema	
Record Writer		CSVRecordSetWriter-Netflow schema-No header	
Include Zero Record FlowFiles		true	

IMPORTANT: To add fields to the NetFlow flowfile, update the *AvroSchemaRegistry* controller.

2. Click the gear for the *CSVReader-Netflow schema* controller.

NetFlow logs - Streaming and Historical Correlation Configuration

GENERAL

CONTROLLER SERVICES

<

3. Click on the arrow for the *AvroSchemaRegistry-Netflow schema* controller

SETTINGS
PROPERTIES
COMMENTS

Required field

Property	Value
Schema Access Strategy	Use 'Schema Name' Property
Schema Registry	AvroSchemaRegistry-Netflow schema →
Schema Name	\${schema.name}
Schema Version	No value set
Schema Branch	No value set
Schema Text	\${avro.schema}
CSV Parser	Apache Commons CSV
Date Format	No value set
Time Format	No value set
Timestamp Format	No value set
CSV Format	Custom Format
Value Separator	,
Treat First Line as Header	true
Ignore CSV Header Column Names	false

4. Click the gear next to *AvroSchemaRegistry-Netflow schema*

NetFlow logs - Streaming and Historical Correlation Configuration

GENERAL
CONTROLLER SERVICES

Name	Type	Bundle	State	Scope
AvroSchemaRegistry-Netflow ...	AvroSchemaRegistry 1.9.0	org.apache.nifi - nifi-registry-n...	Enabled	NetFlow logs - Streaming an...
CSVReader-Netflow schema	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...
CSVReader-Netflow schema r...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...
CSVReader-Netflow schema-...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...

5. The following controller setting opens and shows the Avro Schema. To see the subset of the fields that are read from the NetFlow file open the value for the *netflowAggregateSchema* property

SETTINGS
PROPERTIES
COMMENTS

Required field

Property	Value
Validate Field Names	true
netflowAggregateSchema	{ "name": "netflowAggregateSchema", "namesp...

Expand the window to display the definition for the Avro Schema which should look similar to the following snapshot.

```
1 {
2   "name": "netflowAggregateSchema",
3   "namespace": "nifi.examples",
4   "type": "record",
5   "fields": [
6     { "name": "ts", "type": "string" },
7     { "name": "td", "type": ["null", "double"], "default": null },
8     { "name": "pr", "type": ["null", "string"], "default": null },
9     { "name": "sa", "type": ["null", "string"], "default": null },
10    { "name": "sp", "type": ["null", "int"], "default": null },
11    { "name": "da", "type": ["null", "string"], "default": null },
12    { "name": "dp", "type": ["null", "int"], "default": null },
13    { "name": "flg", "type": ["null", "string"], "default": null },
14    { "name": "stos", "type": ["null", "int"], "default": null },
15    { "name": "ipkt", "type": ["null", "int"], "default": null },
16    { "name": "ibyt", "type": ["null", "long"], "default": null },
17    { "name": "opkt", "type": ["null", "int"], "default": null },
18    { "name": "obyt", "type": ["null", "long"], "default": null },
19    { "name": "in", "type": ["null", "int"], "default": null },
20    { "name": "out", "type": ["null", "int"], "default": null },
21    { "name": "ra", "type": ["null", "string"], "default": null },
22    { "name": "threatq_status", "type": "string", "default": "null" }
23  ]
24 }
```

IMPORTANT: Enable all controller services by clicking on the lightning bolt for each of the following:

- CSVReader-Netflow schema
- CSVRecordSetWriter-Netflow schema-No header
- AvroSchemaRegistry-Netflow schema

Split the Flowfile into Multiple Flowfiles

Select the *SplitText* processor, and configure it as shown below. This processor splits the original NetFlow flowfile into individual files, each with the number of lines defined by the *Line Split Count* property in the processor.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property		Value	
Line Split Count	?	100000	
Maximum Fragment Size	?	No value set	
Header Line Count	?	1	
Header Line Marker Characters	?	No value set	
Remove Trailing Newlines	?	true	

IMPORTANT: In the provided template, the *Line Split Count* is set to 100,000. This value needs to be changed to an optimized value for your HBase instance.
















Change the following values:

Property	Description
Line Split Count	Change the line split to a value that is optimized for your HBase instance

Match to HBase



After splitting the complete NetFlow flowfile into smaller files, each of them is matched against the threat intelligence data in HBase.

For that purpose, a processor group is added which, in the default template, has twenty separate *LookupRecord* processors that do the matching against HBase in parallel. Double click the processor group to see the processors.

HBase Lookup		
 0	 0	 24
 0	 0	 0
 0		
		
Queued	0 (0 bytes)	
In	0 (0 bytes) → 1	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	1 → 0 (0 bytes)	5 min
 0		
 0	 0	 0
 0	 0	 0

Distribute HBase Lookup Load

Select the *DistributeLoad* processor, and configure it as shown below. This processor sends each of the flowfiles from upstream to separate *LookupRecord* processors. For the proper configuration, attach as many *LookupRecord* processors as the *Number of Relationships* value.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property		Value	
Number of Relationships		 20	
Distribution Strategy		 next available	

Change the following values:

Property	Description
Number of Relationships	This value can be changed to use less, or more, <i>LookupRecord</i> processors downstream.

Lookup Records in HBase (20x processors)

Select the *LookupRecord* processor, and configure it as shown below. The number of processors should be equal to the value of the *Number of Relationships* property in the *DistributeLoad* processor. All of the *LookupRecord* processors should use the same configurations.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Record Reader	?	CSVReader-Netflow schema-Match to HBase	→
Record Writer	?	CSVRecordSetWriter-Netflow schema-No header	→
Lookup Service	?	HBase_1_1_2_RecordLookupService-Search threatqtest	→
Result RecordPath	?	/threatq_status	
Routing Strategy	?	Route to 'success'	
Record Result Contents	?	Insert Record Fields	
rowKey	?	/da	

Change the following values:

Property	Description
Record Reader	The record reader controller for the input flowfile. There is no need to change it, unless the input flowfile format is changed
Record Writer	The record writer controller for the output flowfile. There is no need to change it, unless the output flowfile format and/or the columns are changed
Lookup Service	HBase lookup service. Change it only if there is a new version
Result RecordPath	The name of the flowfile field that identifies if a NetFlow log record is successfully matched against threat intel in HBase. Do not change the name, unless the name is also changed in the Avro Schema
rowKey	The Property value is an HBase table property against which the NetFlow logs are matched. For

Property	Description
	<p>example, using <i>rowKey</i> means that the NetFlow logs should be matched against the row key in HBase.</p> <p>The value <i>/da</i> is the Destination IP Address from the NetFlow logs which will be matched against HBase.</p>

The following are the steps to configure the HBase lookup service controller in NiFi.

1. Click on the arrow to the right of HBase_1_1_2_RecordLookupService

NetFlow logs - Streaming and Historical Correlation Configuration

GENERAL		CONTROLLER SERVICES				
Name	Type	Bundle	State	Scope		
AvroSchemaRegistry-Netflow ...	AvroSchemaRegistry 1.9.0	org.apache.nifi - nifi-registry-n...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
CSVReader-Netflow schema	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
CSVReader-Netflow schema r...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
CSVReader-Netflow schema-...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
CSVRecordSetWriter-Netflow ...	CSVRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
CSVRecordSetWriter-Netflow ...	CSVRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1...	org.apache.nifi - nifi-hbase_1_...	Enabling	NiFi Flow	➡️	
HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1...	org.apache.nifi - nifi-hbase_1_...	Enabling	NetFlow logs - Streaming an...	⚙️	➡️
HBase_1_1_2_RecordLookupS...	HBase_1_1_2_RecordLookup...	org.apache.nifi - nifi-hbase_1_...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
JsonRecordSetWriter-Netflow ...	JsonRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️
JsonTreeReader-Netflow sche...	JsonTreeReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	➡️

2. Click on the gear for the HBase_1_1_2_RecordLookupService

SETTINGS

PROPERTIES

COMMENTS

Required field

Property		Value	
HBase Client Service		HBase_1_1_2_ClientService	→
Table Name		threatqdata	
Authorizations		No value set	
Columns		msg:status	
Character Set		UTF-8	

Change the following values:

Property	Description
HBase Client Service	This is the HBase lookup service controller. Do not change it, unless you have a different version.
Table Name	The name of the HBase table that contains the threat intelligence data
Columns	A key:value pair of the <column family>:<column> in HBase. Use the <column family> and column configured in your HBase table

3. Click on the arrow for the HBase_1_1_2_ClientService

NetFlow logs - Streaming and Historical Correlation Configuration

GENERAL		CONTROLLER SERVICES				
Name	Type	Bundle	State	Scope		
AvroSchemaRegistry-Netflow ...	AvroSchemaRegistry 1.9.0	org.apache.nifi - nifi-registry-n...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
CSVReader-Netflow schema	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
CSVReader-Netflow schema r...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
CSVReader-Netflow schema-...	CSVReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
CSVRecordSetWriter-Netflow ...	CSVRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
CSVRecordSetWriter-Netflow ...	CSVRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1...	org.apache.nifi - nifi-hbase_1...	Enabling	NiFi Flow	→	
HBase_1_1_2_ClientService	HBase_1_1_2_ClientService 1...	org.apache.nifi - nifi-hbase_1...	Enabling	NetFlow logs - Streaming an...	⚙️	🔗
HBase_1_1_2_RecordLookupS...	HBase_1_1_2_RecordLookup...	org.apache.nifi - nifi-hbase_1...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
JsonRecordSetWriter-Netflow ...	JsonRecordSetWriter 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗
JsonTreeReader-Netflow sche...	JsonTreeReader 1.9.0	org.apache.nifi - nifi-record-se...	Enabled	NetFlow logs - Streaming an...	⚙️	🔗

4. Click on the gear for the HBase_1_1_2_ClientService

SETTINGS

PROPERTIES

COMMENTS

Required field

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

Change the following values:

Property	Description
Hadoop Configuration Files	Provide the path to the following Hadoop configuration files: hbase-site.xml, hdfs-site.xml, hive-site.xml, core-site.xml If needed, copy these files to the NiFi host from the respective hosts in the Hadoop cluster
Kerberos Credentials Service	If using Kerberos, enter the credentials service
Kerberos Principal	If using Kerberos, enter the principal
Kerberos Keytab	If using Kerberos, enter the keytab
ZooKeeper Quorum	Comma-separated list of Zookeeper hosts. Change the provided list in the template to your environment's values
Zookeeper Client Port	The port Zookeeper listens on (default is 2181)

IMPORTANT: Enable all the controller services by clicking on the lightning bolt for each of the following:

- HBase_1_1_2_RecordLookupService-Search threatqtest
- HBase_1_1_2_ClientService
- CSVReader-Netflow schema-Match to HBase

- CSVRecordSetWriter-Netflow schema-No header

Merge Content

Select the *MergeContent* processor, and configure it as shown below. No need to change any configurations, unless required by your specific environment architecture.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property		Value	
Merge Strategy	?	Bin-Packing Algorithm	
Merge Format	?	Binary Concatenation	
Attribute Strategy	?	Keep Only Common Attributes	
Correlation Attribute Name	?	No value set	
Metadata Strategy	?	Do Not Merge Uncommon Metadata	
Minimum Number of Entries	?	1	
Maximum Number of Entries	?	1000000	
Minimum Group Size	?	0 B	
Maximum Group Size	?	No value set	
Max Bin Age	?	No value set	
Maximum number of Bins	?	5	
Delimiter Strategy	?	Filename	
Header	?	No value set	
Footer	?	No value set	

Update the Data from HBase

Select the *UpdateRecord* processor, and configure it as shown below. This processor modifies the *threatq_status* column from the matching against HBase. In the provided template, that column has the value of "Send to HBase" if the NetFlow record is found in HBase, *null* otherwise.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			+
Property		Value	
Record Reader	?	CSVReader-Netflow schema-Match to HBase	→
Record Writer	?	CSVRecordSetWriter-Netflow schema	→
Replacement Value Strategy	?	Record Path Value	
/threatq_status	?	substringBefore(substringAfter(/threatq_status;=),;))	🗑

Change the following values:

Property	Description
Record Reader	The record reader controller for the input flowfile. There is no need to change it, unless the input flowfile format is changed.
Record Writer	The record writer controller for the output flowfile. There is no need to change it, unless the output flowfile format and/or the columns are changed.
/threatq_status	The name of the key in the Avro Schema that contains the result from the match against HBase. If a NetFlow record is matched against the threat intel in HBase, the <i>threatq_status</i> field will be non-null (e.g. Send to HBase). Otherwise, the value is <i>null</i> . There is no need to change it, unless the name of that field has been changed in the Avro Schema.

IMPORTANT: Enable all the controller services by clicking on the lightning bolt for each of the following:

- CSVReader-Netflow schema-Match to HBase
- CSVRecordSetWriter-Netflow schema
- AvroSchemaRegistry-Netflow schema

Save the Results to HDFS

Select the *PutHDFS* processor, and configure it as shown below. The NetFlow log that was read at the beginning of the NiFi flow is written to HDFS, but with the following changes:

- Only the columns listed in the Avro Schema are preserved in the final file (see the [Read Selected Columns from NetFlow Flowfile](#) section).
- The *threatq_status* column is added which has the value of “Send to HBase” if the NetFlow log record is found in HBase, *null* otherwise

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			+
Property	Value		
Hadoop Configuration Resources	?	/opt/configs/hdfs-config/hdfs-site.xml,/opt/configs/hdf...	
Kerberos Credentials Service	?	No value set	
Kerberos Principal	?	No value set	
Kerberos Keytab	?	No value set	
Kerberos Relogin Period	?	4 hours	
Additional Classpath Resources	?	No value set	
Directory	?	/user/nifi/telemetry/\${now():format("yyyy-MM-dd")}	
Conflict Resolution Strategy	?	append	
Block Size	?	No value set	
IO Buffer Size	?	No value set	
Replication	?	No value set	
Permissions umask	?	No value set	
Remote Owner	?	No value set	
Remote Group	?	No value set	

Change the following values:

Property	Description
Hadoop Configuration Files	<p>Provide the path to the following Hadoop configuration files:</p> <ul style="list-style-type: none"> hbase-site.xml hdfs-site.xml hive-site.xml core-site.xml <p>If needed, copy these files to the NiFi host from the respective hosts in the Hadoop cluster.</p>
Kerberos Credentials Service	If using Kerberos, enter the credentials service.
Kerberos Principal	If using Kerberos, enter the principal.
Kerberos Keytab	If using Kerberos, enter the keytab.
Directory	The path to the directory where the files will be written in HDFS. The user that NiFi runs as, should have read/write access to that folder.

Property	Description
Conflict Resolution Strategy	The default value is to <i>append</i> the content from a single flowfile. Different flowfiles will be written into separate files on HDFS.

Matched to HBase?

Select the *RouteText* processor, and configure it as shown below. This processor will route all the NetFlow log records that match to HBase further into the NiFi flow. Those that do not match are dropped at this point.

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value
Routing Strategy	Route to each matching Property Name
Matching Strategy	Contains
Character Set	UTF-8
Ignore Leading/Trailing Whitespace	true
Ignore Case	false
Grouping Regular Expression	No value set
Matches ThreatQ	Send to HBase 🗑

Change the following values:

Property	Description
Matches ThreatQ	The value assigned to the threatq_status field, if the NetFlow log record is found in HBase. In the provided template that value is <i>"Send to HBase"</i> . If there is no match, the value is <i>null</i> .

Convert File to JSON

Select the *ConvertRecord* processor, and configure it as shown below. This converts the CSV schema to JSON.

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value
Record Reader	? CSVReader-Netflow schema read match from HBase →
Record Writer	? JsonRecordSetWriter-Netflow schema →
Include Zero Record FlowFiles	? true

Change the following values:

Property	Description
Record Reader	The record reader controller for the input flowfile. There is no need to change it, unless the input flowfile format is changed.
Record Writer	The record writer controller for the output flowfile. There is no need to change it, unless the output flowfile format and/or the columns are changed.

IMPORTANT: Enable all the controller services by clicking on the lightning bolt for each of the following:

- CSVReader-Netflow schema read match from HBase
- JsonRecordSetWriter-Netflow schema
- AvroSchemaRegistry-Netflow schema

Split JSON into Individual Files

Select the *SplitJson* processor, and configure it as shown below. This processor splits a large JSON object into its separate elements. The output is JSON.

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value
JsonPath Expression	? \$.*
Null Value Representation	? empty string

Publish to Kafka Topic

Select the *PublishKafka_<version>* processor, and configure it as shown below. This is a Kafka Producer that publishes messages to a specific Kafka topic.

IMPORTANT: The *PublishKafka* processor in the template uses the *Kafka v1.0 Producer API*. If your *Kafka API* is a different version, change the *PublishKafka* processor to the appropriate version. The available *Kafka Producer* versions in *NiFi v1.9.0* are *v0.10*, *v0.11*, *v1.0* and *v2.0*. To change it, add a new processor and select the desired processor version from the list.

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value
Kafka Brokers	<Kafka Broker>:<Port>
Security Protocol	PLAINTEXT
Kerberos Service Name	No value set
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
SSL Context Service	No value set
Topic Name	incidents1
Delivery Guarantee	Guarantee Replicated Delivery
Use Transactions	true
Attributes to Send as Headers (Regex)	No value set
Message Header Encoding	UTF-8
Kafka Key	No value set
Key Attribute Encoding	UTF-8 Encoded
Key Attribute Encoding	UTF-8 Encoded
Message Demarcator	No value set
Max Request Size	1 MB
Acknowledgment Wait Time	5 secs
Max Metadata Wait Time	5 sec
Partitioner class	DefaultPartitioner
Compression Type	none

Change the following values:

Property	Description
Kafka Brokers	Comma-separated list of Kafka brokers in the following format: <Kafka Broker>:<Port>

Property	Description
	The default value in the template uses port 6667, which should be changed if your Kafka broker uses a different port
Kerberos Service Name	If using Kerberos, enter the service name.
Kerberos Credentials Service	If using Kerberos, enter the credentials service.
Kerberos Principal	If using Kerberos, enter the principal.
Kerberos Keytab	If using Kerberos, enter the keytab.
Topic Name	The name of the Kafka topic to which NiFi should publish the data. The topic should be created prior to configuring the processor.

Push to Elasticsearch Index (Optional)

IMPORTANT: This is an optional processor. The recommended process is for Elasticsearch to read the content from a Kafka topic, via Logstash, as described in the “ELK Stack” section above. The `PutElasticsearchHttpRecord` processor uses the Elasticsearch Bulk API which loads all content in memory and during peak times the processor could create a bottleneck for the whole NiFi flow, and slow down processors located upstream (e.g. `LookupRecord` against HBase) .

Select the `PutElasticsearchHttpRecord` processor, and configure it as shown below. This processor ingests the NetFlow records that were found in HBase to an Elastic index.

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field

Property	Value
Elasticsearch URL	http://<Elastic Host>:9200
SSL Context Service	No value set
Username	threatq
Password	Sensitive value set
Connection Timeout	5 secs
Response Timeout	15 secs
Proxy Configuration Service	No value set
Proxy Host	No value set
Proxy Port	No value set
Proxy Username	No value set
Proxy Password	No value set
Record Reader	JsonTreeReader-Netflow schema
Identifier Record Path	No value set
Index	telemetrydata
Index	telemetrydata
Type	logs
Character Set	UTF-8
Index Operation	index
Suppress Null Values	Never Suppress
Date Format	No value set
Time Format	No value set
Timestamp Format	No value set

Change the following values:

Property	Description
Elasticsearch URL	The Elasticsearch URL followed by the port. The default port is 9200.
SSL Context Service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. Only applies if the Elasticsearch cluster is secured with TLS/SSL.
Username	Username for authenticating with Elasticsearch.
Password	Password for authenticating with Elasticsearch.

Property	Description
Record Reader	The record reader controller for the input flowfile. There is no need to change it, unless the input flowfile format is changed.
Index	Name of the Elasticsearch index to insert to.
Type	The type of this document used by Elasticsearch for indexing and searching.
Index Operation	The type of the operation to use. Default is <i>index</i> .

IMPORTANT: Enable all the controller services by clicking on the lightning bolt for each of the following:

1. JsonTreeReader-Netflow schema
2. AvroSchemaRegistry-Netflow schema

Adding NetFlow Log Data to Hive Table (Optional)

The following processors are used to ingest the NetFlow flow files into Hive table. This is optional and should only be used if the customer needs to have all the data available for analysis.

Infer Avro Schema

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

SETTINGS
SCHEDULING
PROPERTIES
COMMENTS

Required field

Property	Value
Schema Output Destination	flowfile-attribute
Input Content Type	csv
CSV Header Definition	No value set
Get CSV Header Definition From Data	true
CSV Header Line Skip Count	0
CSV delimiter	,
CSV Escape String	\
CSV Quote String	'
Pretty Avro Output	true
Avro Record Name	telemetry
Number Of Records To Analyze	10000
Charset	UTF-8

Change the following values:

Property	Description
Input Content Type	The value in the provided template is csv. Do not change it, unless changes to the schema have been done upstream in the flow.
Avro Record Name	Value to be placed in the Avro record schema "name" field.

Convert CSV to Avro format

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

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Hadoop configuration Resources		No value set	
Record schema		\$(inferred.avro.schema)	
CSV charset		utf8	
CSV delimiter		,	
CSV quote character		"	
CSV escape character		\	
Use CSV header line		true	
Lines to skip		0	
Compression type		SNAPPY	

Put in Hive Table

Select the *PutHiveStreaming* processor, and configure it as shown below. This processor ingests the flow content to the designated Hive table.

IMPORTANT: The Hive table with its schema should be created prior to enabling this processor.

SETTINGS	SCHEDULING	PROPERTIES	COMMENTS
Required field			
Property	Value		
Hive Metastore URI		thrift://10.13.0.93:9083	
Hive Configuration Resources		/opt/configs/hdfs-config/hdfs-site.xml/opt/con...	
Database Name		default	
Table Name		telemetry	
Partition Columns		No value set	
Auto-Create Partitions		true	
Max Open Connections		8	
Heartbeat Interval		60	
Transactions per Batch		100	
Records per Transaction		10000	
Call Timeout		0	
Rollback On Failure		false	
Kerberos Credentials Service		No value set	
Kerberos Principal		No value set	

Change the following values:

Property	Description
Hive Metastore URI	The Hive store URI. The default port is 9083.

Property	Description
	<p>The value should be in the format:</p> <p><i>thrift://<Hive Host>:<Port></i></p>
Hive Configuration Resources	<p>Provide the path to the following Hive configuration files:</p> <ul style="list-style-type: none"> • hbase-site.xml • hdfs-site.xml • hive-site.xml • core-site.xml <p>If needed, copy these files to the NiFi host from the respective hosts in the Hadoop cluster.</p>
Database Name	Name of the Hive database created in Hive.
Table Name	Name of the Hive table to which data should be ingested.

Change Log

Version	Details
1.0.0	Initial Release
3.0.0	Adds NetFlow logs parsing ability
3.3.0	Updates the connectors to Elastic, HBase and HDFS