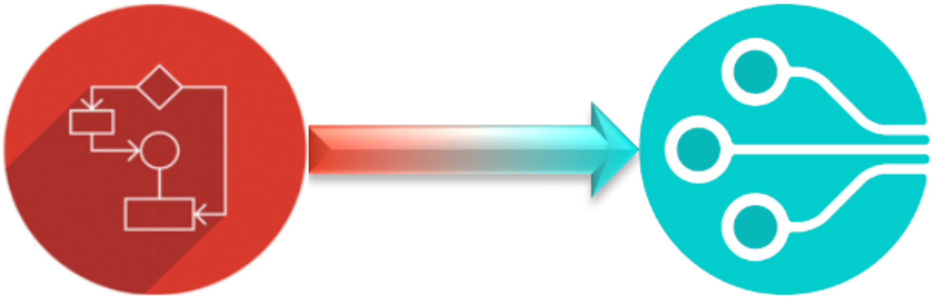
Collecting CrowdStrike SIEM Connector Data With Cribl Edge



Configuration Guide V1.4

Table of Contents

Overview

The Purpose of this Document

The purpose of this document is to provide current CrowdStrike and Cribl customers with a process of collecting CrowdStrike Event Streams data using the CrowdStrike SIEM Connector and Cribl Edge.

Minimum Requirements for this Process

1. A valid license for CrowdStrike Falcon that provides for access to the Event Streams Streaming API.
2. A valid license for Cribl Edge.
3. Access to or the ability to generate a valid set of CrowdStrike Oauth2 API credentials with the 'Event Streams' scope.
4. The ability to access, deploy and configure Cribl Edge.
5. The ability to deploy or admin level access to an existing CrowdStrike SIEM Connector

Test Environment for Current Documentation

Cribl Edge:

UI version: 4.1.3-15457782/2023-06-14T10:35:24.053Z
Backend version: 4.1.3-15457782/v4.1.3/2023-06-14T10:41:39.889Z

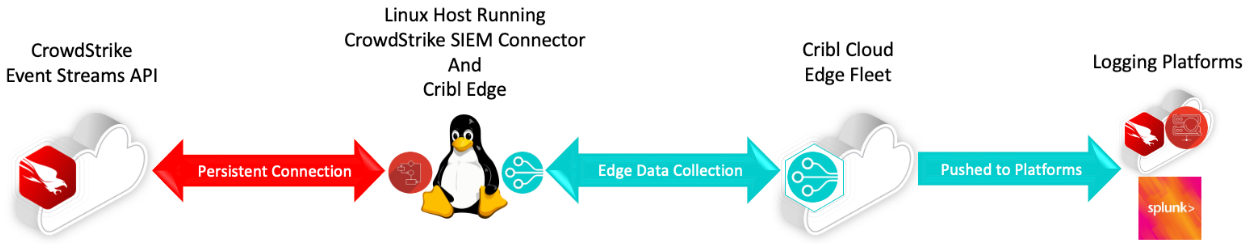
CrowdStrike SIEM Connector:

SIEM Connector v3 – CentOS 7

Release Notes

v1.3: Initial Document Release

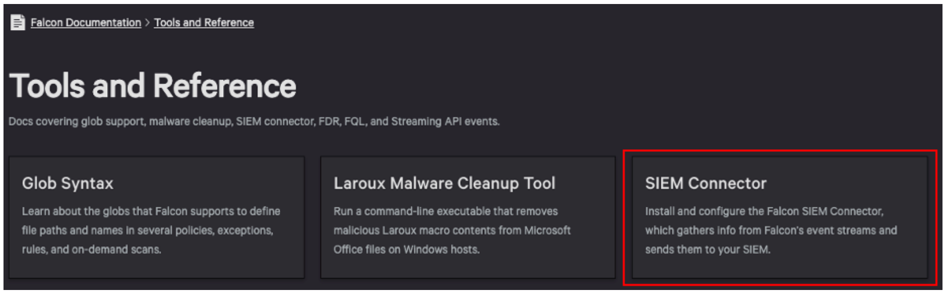
High Level Architecture



- A properly configured SIEM connector, running on a supported version of Linux, is used to create and maintain a persistent connection with the CrowdStrike Event Stream API.
- The SIEM Connector will process the CrowdStrike events and output them to a log file.
- The local Cribl Edge deployment will collect the event data from the monitored file and push it to the Cribl Cloud Edge Fleet.
- The Cribl Edge Fleet will process the event data and push the results to the configured platforms.

CrowdStrike Configurations

The CrowdStrike SIEM connector should be deployed or have been deployed following the documentation published in the Falcon UI.



Falcon Documentation > Tools and Reference

Tools and Reference

Docs covering glob support, malware cleanup, SIEM connector, FDR, FQL, and Streaming API events.

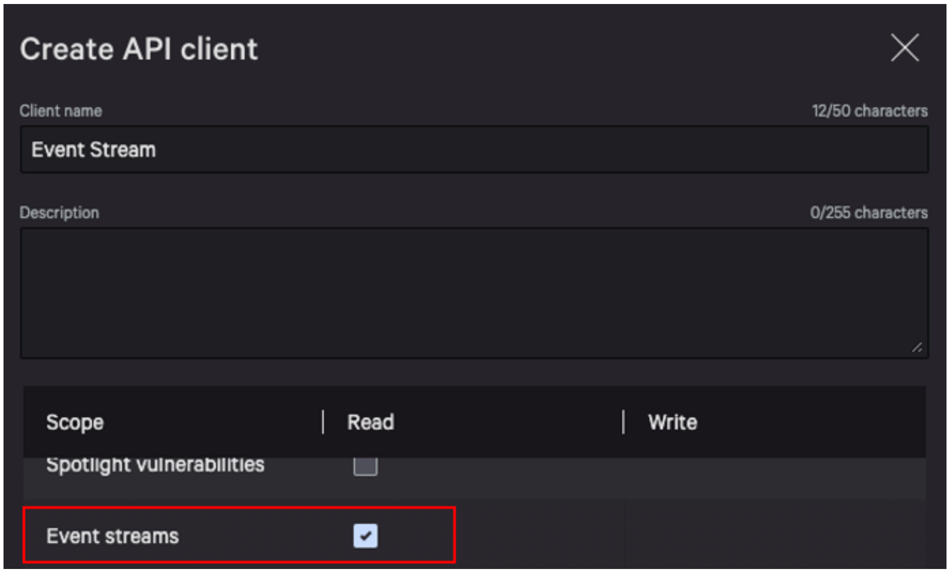
Glob Syntax
Learn about the globs that Falcon supports to define file paths and names in several policies, exceptions, rules, and on-demand scans.

Laroux Malware Cleanup Tool
Run a command-line executable that removes malicious Laroux macro contents from Microsoft Office files on Windows hosts.

SIEM Connector
Install and configure the Falcon SIEM Connector, which gathers info from Falcon's event streams and sends them to your SIEM.

API Client Credentials

If the SIEM connector has been collecting data previously this step can most likely be skipped. If this is an initial SIEM connector deployment ensure that the API client has been properly scoped with the 'Event streams' scope.



Create API client

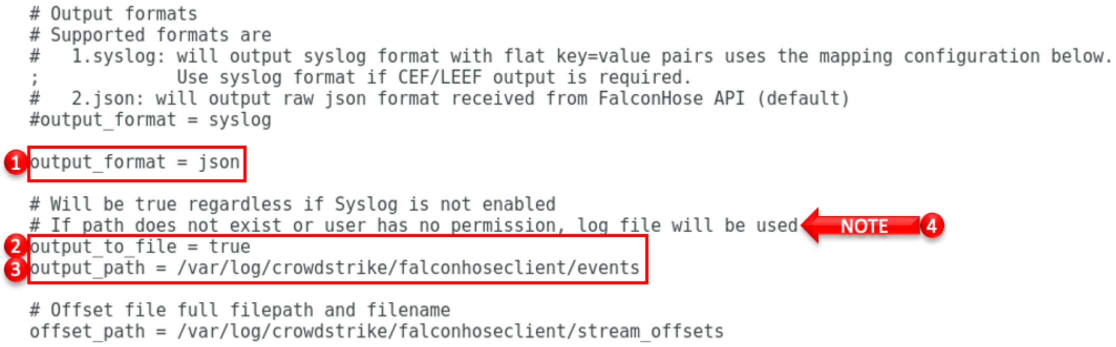
Client name 12/50 characters
Event Stream

Description 0/255 characters

Scope	Read	Write
Spotlight vulnerabilities	□	
Event streams	☑	

SIEM Connector ‘cs.falconhoseclient.cfg’ File

The CrowdStrike SIEM connector should be deployed following the documentation published in the Falcon UI. Once completed the following configurations should be made/validated:

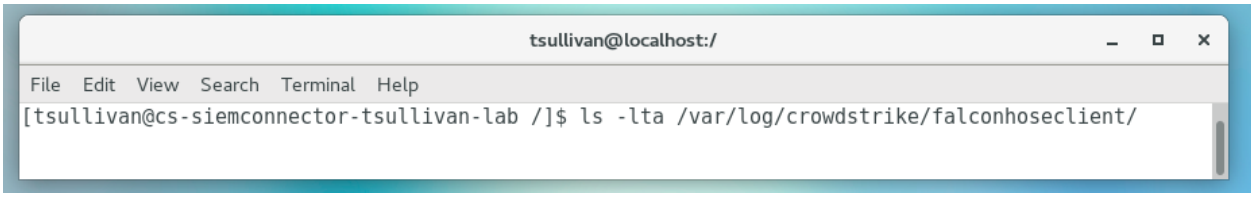


```
# Output formats
# Supported formats are
#   1.syslog: will output syslog format with flat key=value pairs uses the mapping configuration below.
;             Use syslog format if CEF/LEEF output is required.
#   2.json: will output raw json format received from FalconHose API (default)
#output_format = syslog

output_format = json

# Will be true regardless if Syslog is not enabled
# If path does not exist or user has no permission, log file will be used
output_to_file = true
output_path = /var/log/crowdstrike/falconhoseclient/events

# Offset file full filepath and filename
offset_path = /var/log/crowdstrike/falconhoseclient/stream_offsets
```

1. Ensure that the output_format is set to JSON.
2. Ensure that the output_to_file is set to true.
3. Ensure that the output_path is configured to a location that Cribl Edge will be able to properly collect from. The filename does not have to be ‘events’ but the filename used in this file must match the filename being monitored in Cribl.
4. Take notice of the warning for outputting to a path that does not exist or that the user doesn’t have permission to as this will impact the output and potentially the ability to properly collect data.

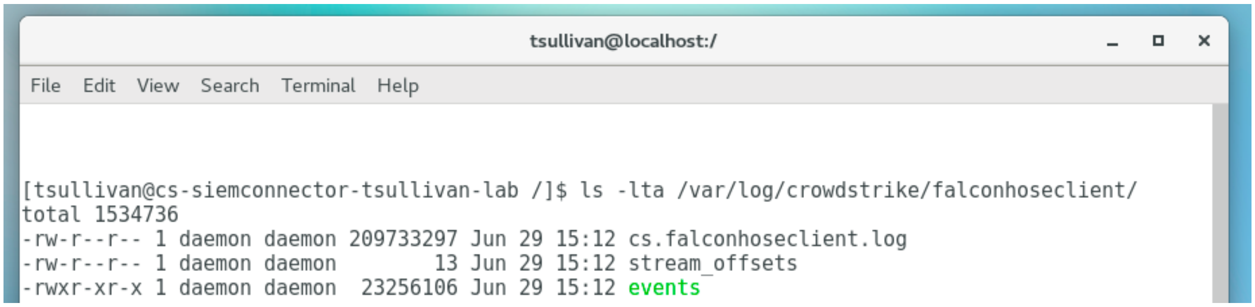
SIEM Connector Output File Check

In the previous configuration the output_path was set to the following path: /var/log/crowdstrike/falconhoseclient/events. To ensure that the path exists and examine the permissions run the following command:



```
[tsullivan@cs-siemconnector-tsullivan-lab /]$ ls -lta /var/log/crowdstrike/falconhoseclient/
```

In this case the file exists, is accessible and currently has data in it:



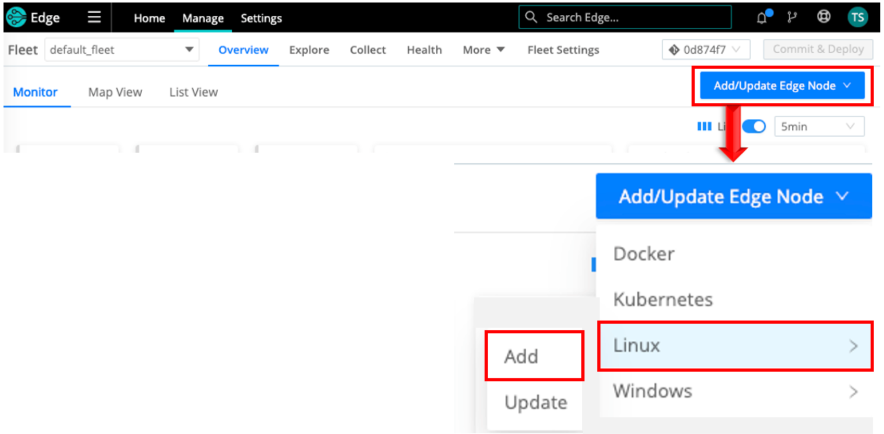
```
[tsullivan@cs-siemconnector-tsullivan-lab /]$ ls -lta /var/log/crowdstrike/falconhoseclient/
total 1534736
-rw-r--r-- 1 daemon daemon 209733297 Jun 29 15:12 cs.falconhoseclient.log
-rw-r--r-- 1 daemon daemon        13 Jun 29 15:12 stream_offsets
-rwxr-xr-x 1 daemon daemon  23256106 Jun 29 15:12 events
```

Cribl Edge Configurations

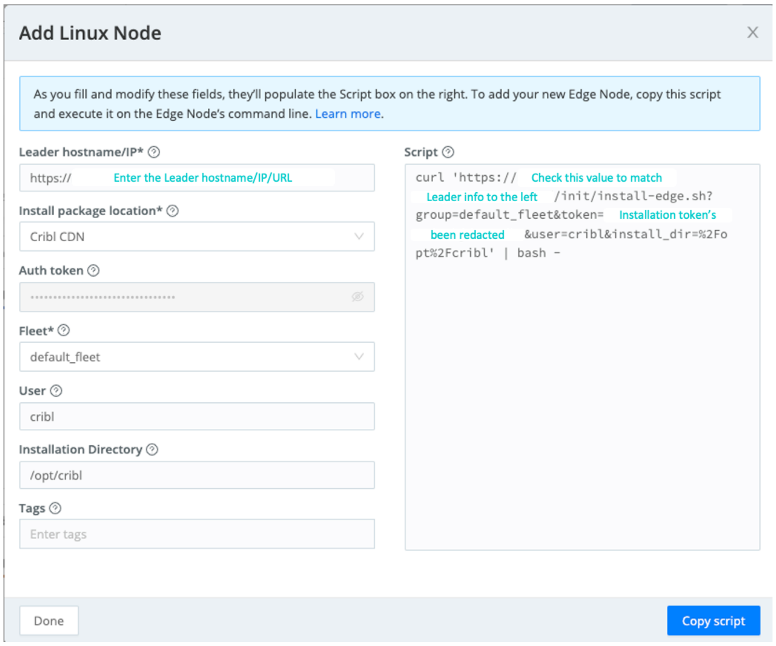
If the system that is running the CrowdStrike SIEM Connector is already running edge then skip to the next section. Otherwise follow the following process to deploy the Edge agent to the system that is running the SIEM connector.

Deploying the Edge Agent to the System

1. Navigate to the Edge platform, access a Fleet to do the agent deployment from and select ‘Add/Update Edge Node’ -> ‘Linux’ -> ‘Add’ in the top right corner.



2. Configure the proper deployment for your environment. Cribl documentation for Edge Deployment can be found here: https://docs.cribl.io/edge/deploy-planning .



Add Linux Node

As you fill and modify these fields, they'll populate the Script box on the right. To add your new Edge Node, copy this script and execute it on the Edge Node's command line. Learn more.

Leader hostname/IP*: https:// Enter the Leader hostname/IP/URL

Install package location*: Cribl CDN

Auth token: ································

Fleet*: default_fleet

User: cribl

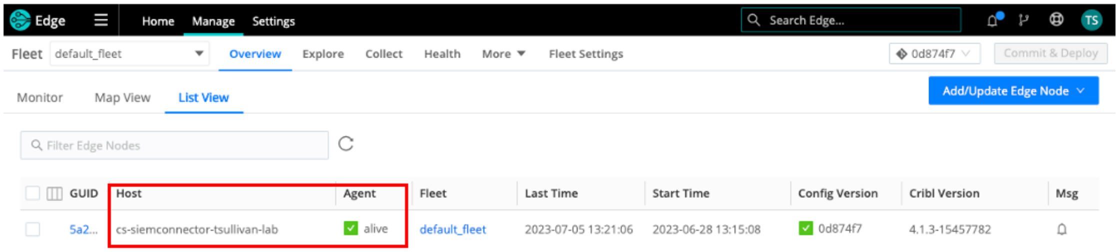
Installation Directory: /opt/cribl

Tags: Enter tags

Script:

```
curl 'https://  Check this value to match
  Leader info to the left   /init/install-edge.sh?
group=default_fleet&token=  Installation token's
  been redacted   &user=cribl&install_dir=%2Fo
pt%2Fcribl' | bash -
```

3. Deploy the Edge agent to the system and validate that it's properly communicating with the Fleet.

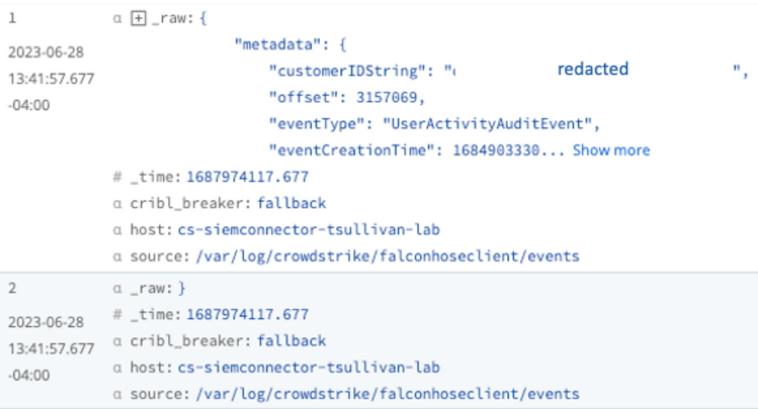


GUID	Host	Agent	Fleet	Last Time	Start Time	Config Version	Cribl Version	Msg
5a2...	cs-siemconnector-tsullivan-lab	✓ alive	default_fleet	2023-07-05 13:21:06	2023-06-28 13:15:08	✓ 0d874f7	4.1.3-15457782	🔔

Creating the Pre-Processing Pipeline in Cribl Edge

The JSON output of the CrowdStrike SIEM connector presents a small challenge that requires the use of a pre-processing pipeline. The output data is essentially designed to be independent JSON objects but the overall file format is not constructed as a JSON array or as a JSON object with nested JSON objects. The result can be that when Edge ingests the data that it won't recognize it as JSON. It will essentially split an object into 2 events: one will have all of the data and the second will typically be just a '}' bracket.

The following is an example of what this data collection would look like when it's first collected by Cribl Edge:



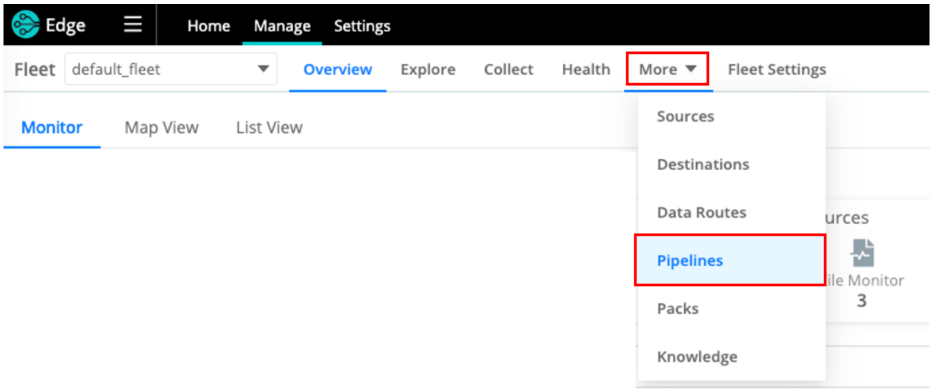
```
1               α ⊞ _raw: {
                            "metadata": {
2023-06-28                      "customerIDString": "(          redacted          ",
13:41:57.677                    "offset": 3157069,
-04:00                          "eventType": "UserActivityAuditEvent",
                                "eventCreationTime": 1684903330... Show more
                # _time: 1687974117.677
                α cribl_breaker: fallback
                α host: cs-siemconnector-tsullivan-lab
                α source: /var/log/crowdstrike/falconhoseclient/events
2               α _raw: }
                # _time: 1687974117.677
2023-06-28      α cribl_breaker: fallback
13:41:57.677    α host: cs-siemconnector-tsullivan-lab
-04:00          α source: /var/log/crowdstrike/falconhoseclient/events
```

The simplest way to address this is by using a Pre-Processing Pipeline. The first function will look for the events where the _raw value is just the single curly bracket ' } ' and remove them. The second will look for events where the _raw values that are larger than just a single curly bracket, add the curly bracket to the end, parse the response as JSON and remove everything but the event data.

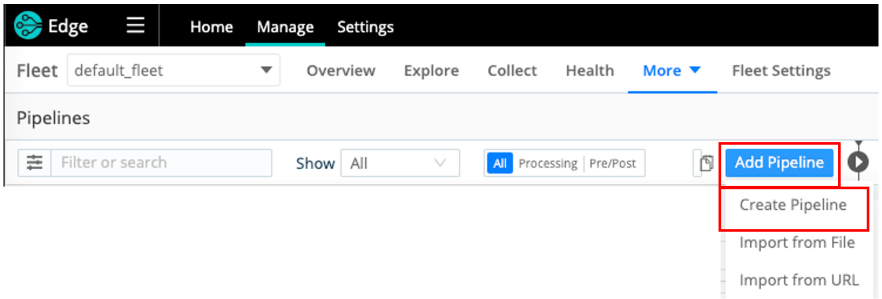
The filters being used in the provided example are simple but have been effective in processing SIEM connector data. There are certainly more advanced filters that could be constructed that may better align to an organization's requirements, such as identifying if specific fields are present in the data. The filters in these examples are merely examples and ensuring that the end configuration of the Pre-Processing pipeline meets published requirements is strongly encouraged.

Configure a Pre-Processing Pipeline as follows:

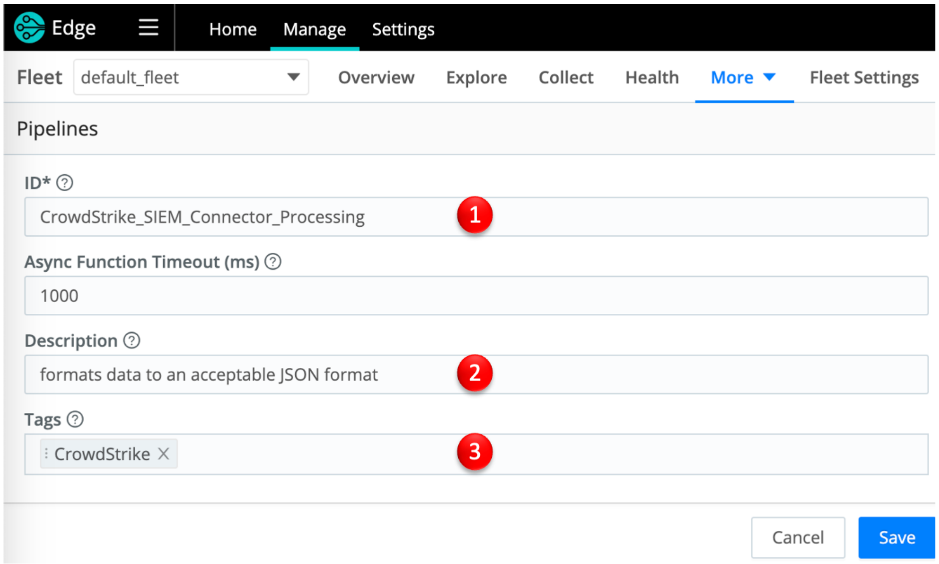
1. In the main menu select ‘More’ and then ‘Pipelines’



2. In the pipeline menu select ‘Add Pipeline’ and ‘Create Pipeline’ from the dropdown.



3. Complete the new pipeline configuration and select save.



Fleet: default_fleet

ID*: CrowdStrike_SIEM_Connector_Processing

Async Function Timeout (ms): 1000

Description: formats data to an acceptable JSON format

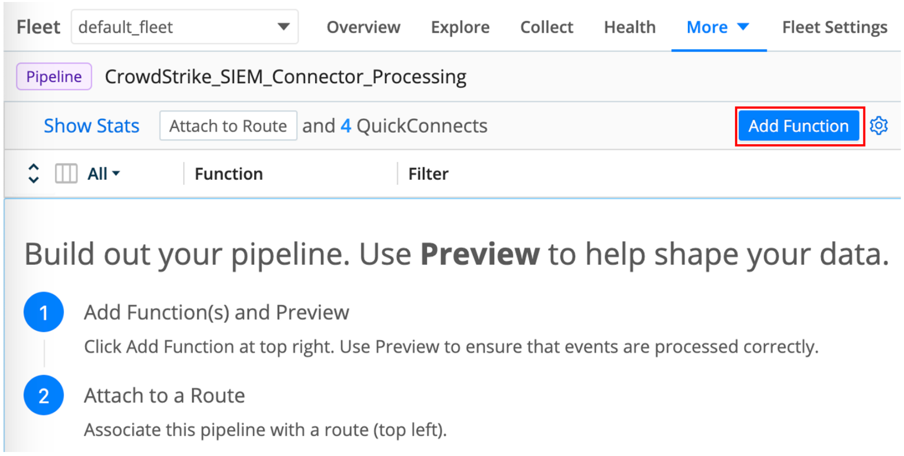
Tags: CrowdStrike

1. **ID**: Configure a name for the pipeline.
2. **Description**: (optional) Provide a description for the pipeline.
3. **Tags**: (optional) Provide a tag for the pipeline.

4. Build out the appropriate pipeline actions to handle the CrowdStrike SIEM connector data can best be accomplished by either building a new pipeline configuration (Step 4.1) or by leveraging the example pipeline configuration provided in Appendix A (Step 4.2).

4.1. **Build a new pipeline configuration**.

4.1.1. Select ‘Add Function’ in the new pipeline.



CrowdStrike_SIEM_Connector_Processing

and 4 QuickConnects

Build out your pipeline. Use **Preview** to help shape your data.

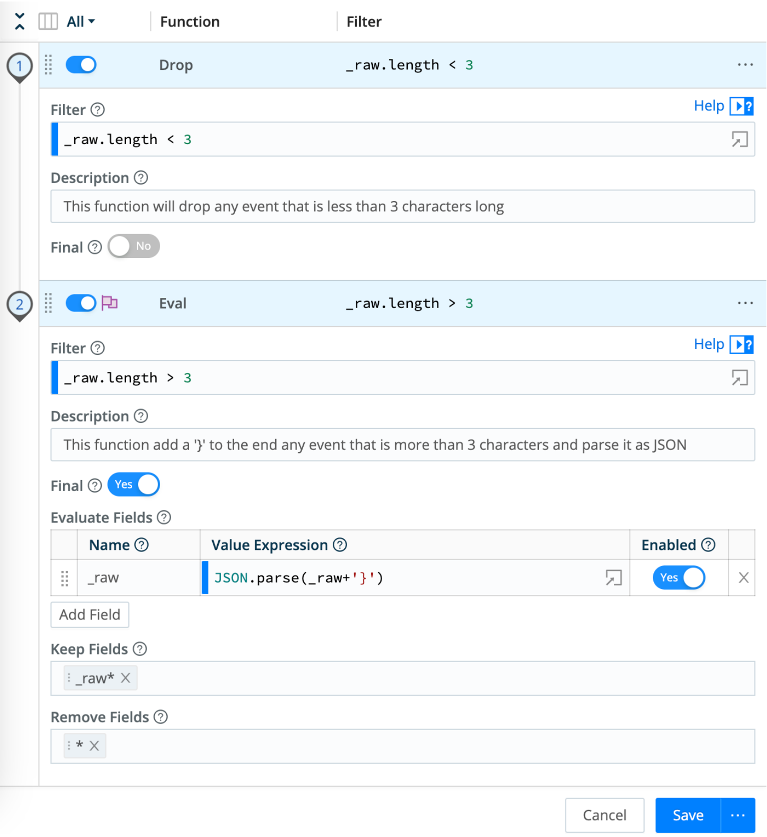
1 Add Function(s) and Preview

Click Add Function at top right. Use Preview to ensure that events are processed correctly.

2 Attach to a Route

Associate this pipeline with a route (top left).

4.1.2. Add the appropriate function to properly handle the SIEM Connector data, for example:



1 Drop `_raw.length < 3`

Filter: `_raw.length < 3`

Description: This function will drop any event that is less than 3 characters long

Final: No

2 Eval `_raw.length > 3`

Filter: `_raw.length > 3`

Description: This function add a '}' to the end any event that is more than 3 characters and parse it as JSON

Final: Yes

Evaluate Fields

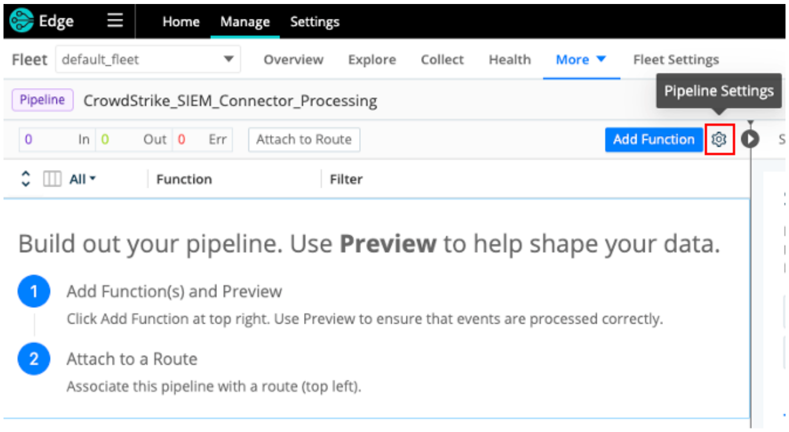
Name	Value Expression	Enabled
_raw	`JSON.parse(_raw+'}')`	Yes

Keep Fields: _raw*

Remove Fields: *

4.2. **Create a pipeline from the template in Appendix A**.

4.2.1. Select the ‘Pipeline Settings’ gear icon next to ‘Add Function’.



Build out your pipeline. Use **Preview** to help shape your data.

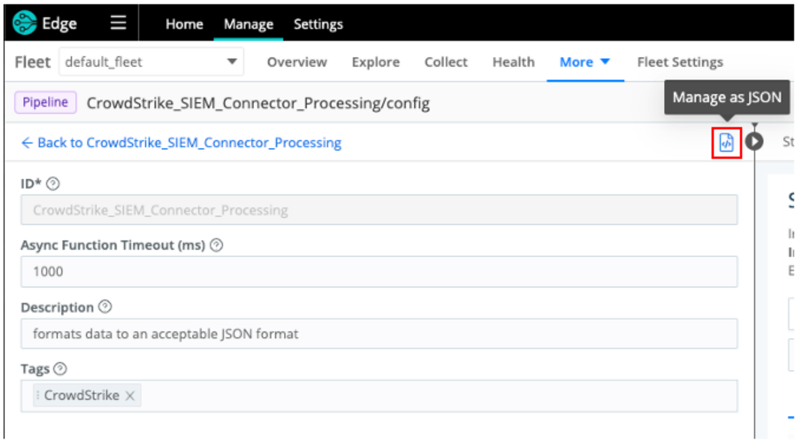
1 Add Function(s) and Preview

Click Add Function at top right. Use Preview to ensure that events are processed correctly.

2 Attach to a Route

Associate this pipeline with a route (top left).

4.2.2. Select the ‘Manage as JSON’ icon in the right corner.



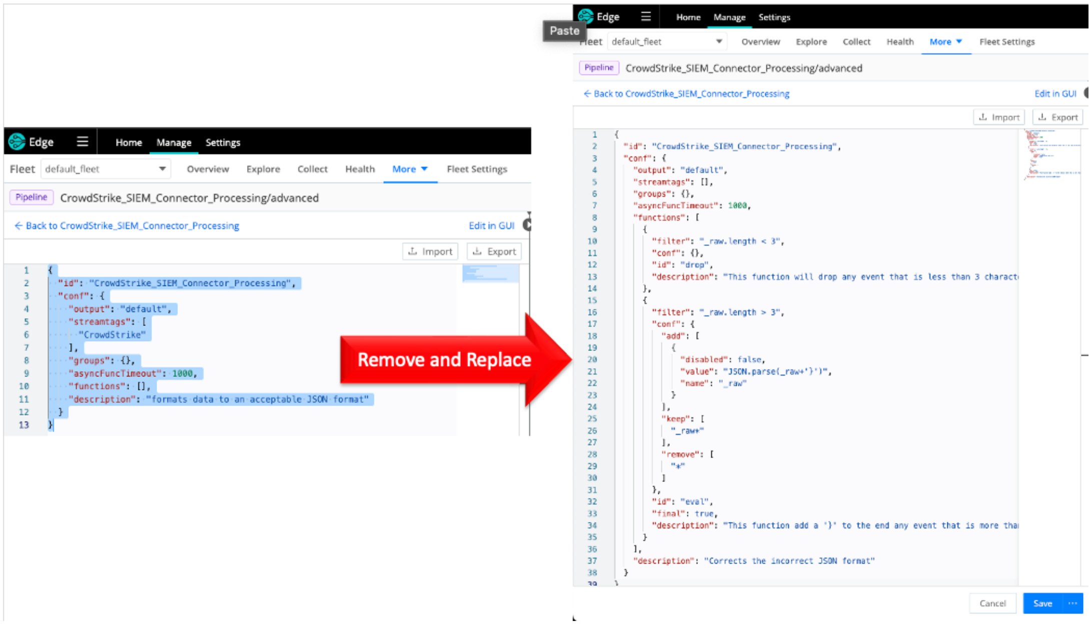
ID*: CrowdStrike_SIEM_Connector_Processing

Async Function Timeout (ms): 1000

Description: formats data to an acceptable JSON format

Tags: CrowdStrike

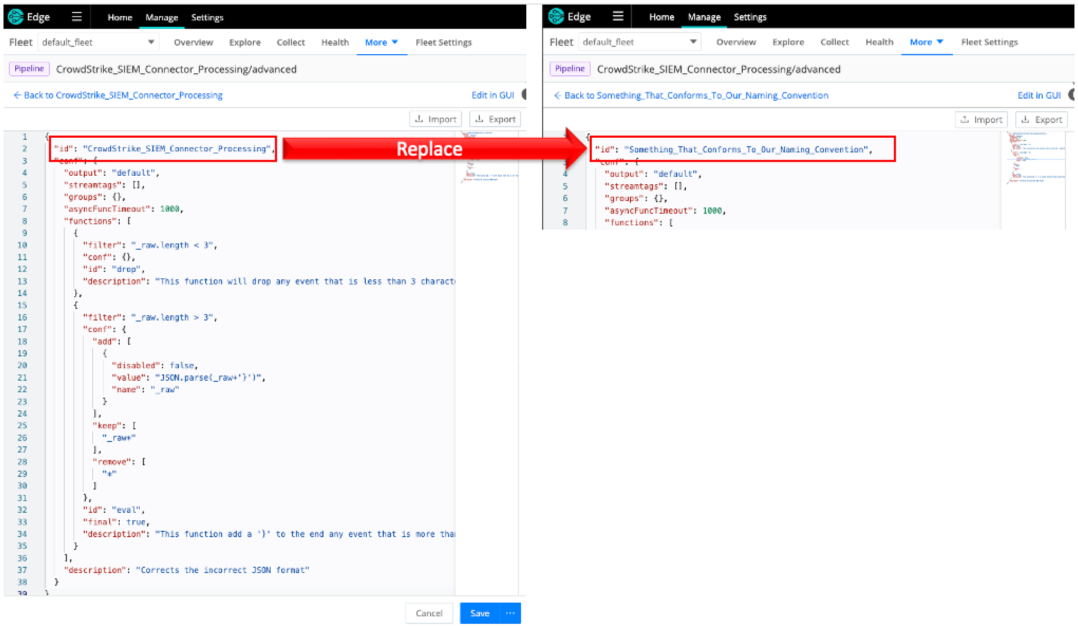
4.2.3. Select and remove all the existing text and then cut and paste the example JSON text in appendix A.



```
{
  "id": "CrowdStrike_SIEM_Connector_Processing",
  "conf": {
    "output": "default",
    "streamtags": [
      "CrowdStrike"
    ],
    "groups": {},
    "asyncFuncTimeout": 1000,
    "functions": [],
    "description": "formats data to an acceptable JSON format"
  }
}
```

```
{
  "id": "CrowdStrike_SIEM_Connector_Processing",
  "conf": {
    "output": "default",
    "streamtags": [],
    "groups": {},
    "asyncFuncTimeout": 1000,
    "functions": [
      {
        "filter": "_raw.length < 3",
        "conf": {},
        "id": "drop",
        "description": "This function will drop any event that is less than 3 charact
      },
      {
        "filter": "_raw.length > 3",
        "conf": {
          "add": [
            {
              "disabled": false,
              "value": "JSON.parse(_raw+'}')",
              "name": "_raw"
            }
          ],
          "keep": [
            "_raw*"
          ],
          "remove": [
            "*"
          ]
        },
        "id": "eval",
        "final": true,
        "description": "This function add a '}' to the end any event that is more tha
      }
    ],
    "description": "Corrects the incorrect JSON format"
  }
}
```

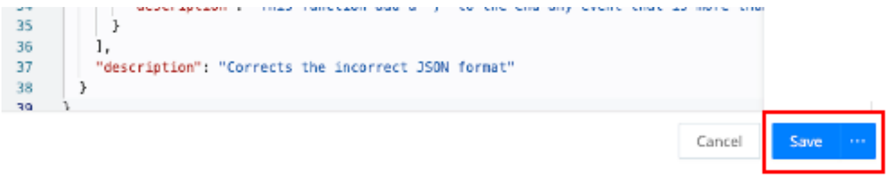
4.2.4. **PRIOR TO SAVING** (Optional) The ‘id’ field value can be changed so that it matches the name that was originally given to the pipeline.


Edge
Home
Manage
Settings
Fleet
default_fleet
Overview
Explore
Collect
Health
More
Fleet Settings
Pipeline
CrowdStrike_SIEM_Connector_Processing/advanced
← Back to CrowdStrike_SIEM_Connector_Processing
← Back to Something_That_Conforms_To_Our_Naming_Convention
Edit in GUI
Import
Export
Replace
Cancel
Save

```
{
  "id": "CrowdStrike_SIEM_Connector_Processing",
  "conf": {
    "output": "default",
    "streamtags": [],
    "groups": {},
    "asyncFuncTimeout": 1000,
    "functions": [
      {
        "filter": "_raw.length < 3",
        "conf": {},
        "id": "drop",
        "description": "This function will drop any event that is less than 3 charact
      },
      {
        "filter": "_raw.length > 3",
        "conf": {
          "add": [
            {
              "disabled": false,
              "value": "JSON.parse(_raw+'}')",
              "name": "_raw"
            }
          ],
          "keep": [
            "_raw*"
          ],
          "remove": [
            "*"
          ]
        },
        "id": "eval",
        "final": true,
        "description": "This function add a '}' to the end any event that is more tha
      }
    ],
    "description": "Corrects the incorrect JSON format"
  }
}
```

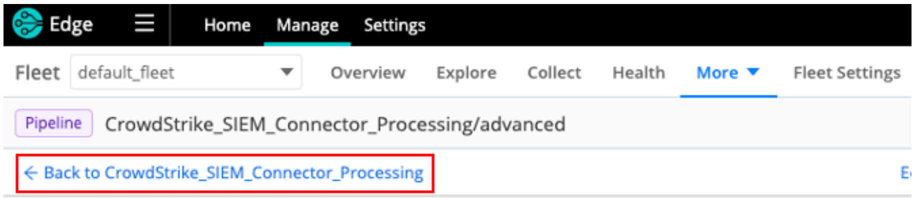
```
{
  "id": "Something_That_Conforms_To_Our_Naming_Convention",
  "conf": {
    "output": "default",
    "streamtags": [],
    "groups": {},
    "asyncFuncTimeout": 1000,
    "functions": [
```

4.2.5. Select ‘Save’ in the bottom right corner.

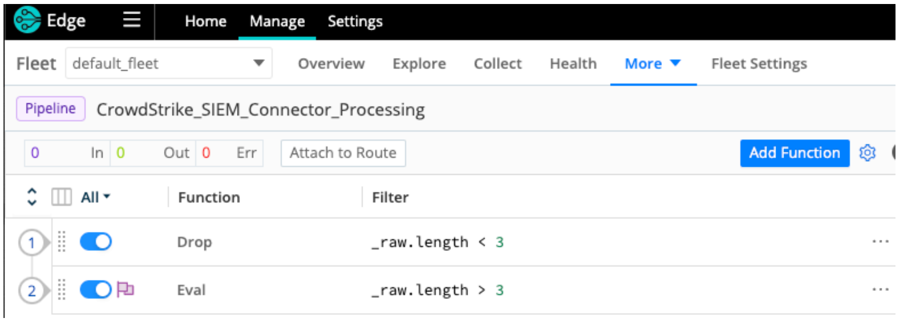


```
    }
  ],
  "description": "Corrects the incorrect JSON format"
}
```

4.2.6. Select the ‘Back to *whatever_name_given_to_pipeline*’.

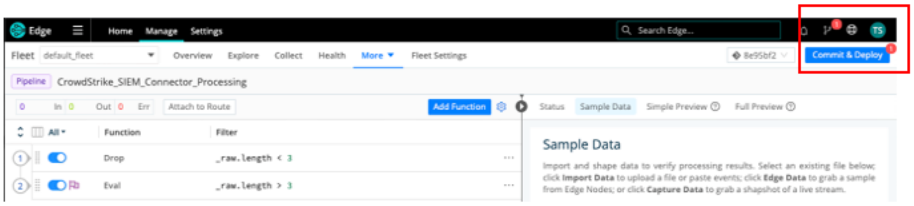


4.2.7. Validate that the pipeline was imported properly.



	Function	Filter
1	Drop	_raw.length < 3
2	Eval	_raw.length > 3

5. Commit and deploy the changes in the top right corner of the page.



	Function	Filter
1	Drop	_raw.length < 3
2	Eval	_raw.length > 3

Sample Data

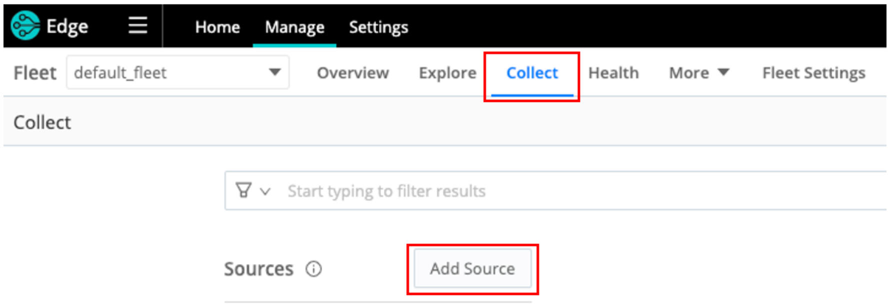
Import and shape data to verify processing results. Select an existing file below; click **Import Data** to upload a file or paste events; click **Edge Data** to grab a sample from Edge Nodes; or click **Capture Data** to grab a shapshot of a live stream.

--- End of Section ---

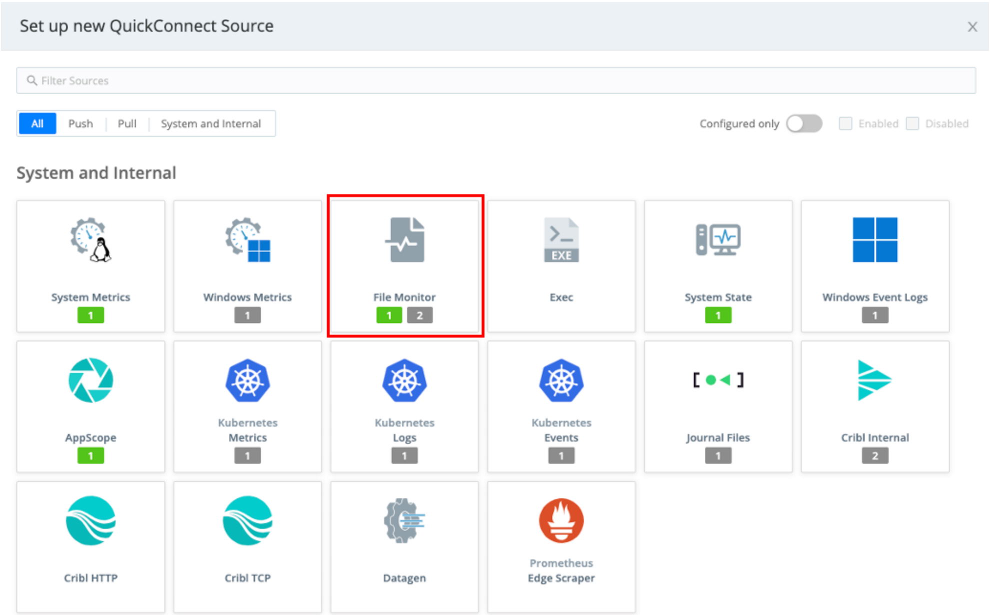
Configuring Data Collection in Edge

Data can be collected by using an existing collection or by creating a new one. In the interest of simplicity this document will assume a new collection needs to be created.

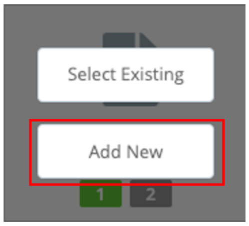
1. In the Fleet menu select 'Collect' and then 'Add Source'.



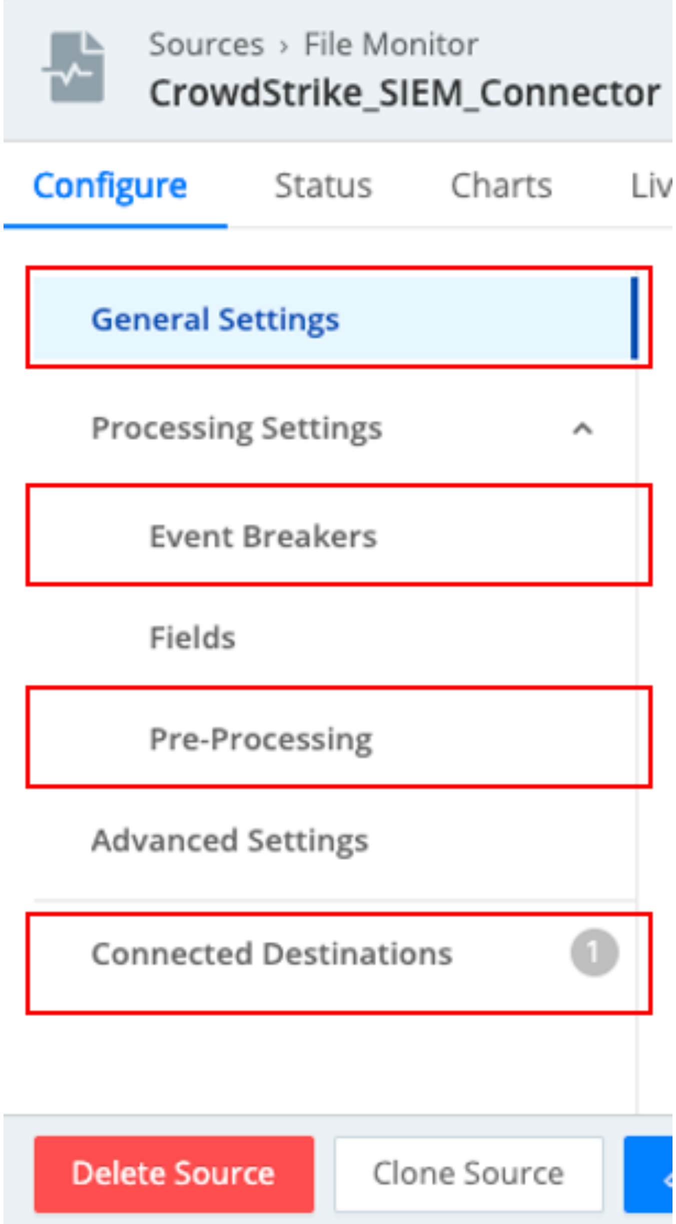
2. In the 'Set up new QuickConnect Source' menu select 'File Monitor':



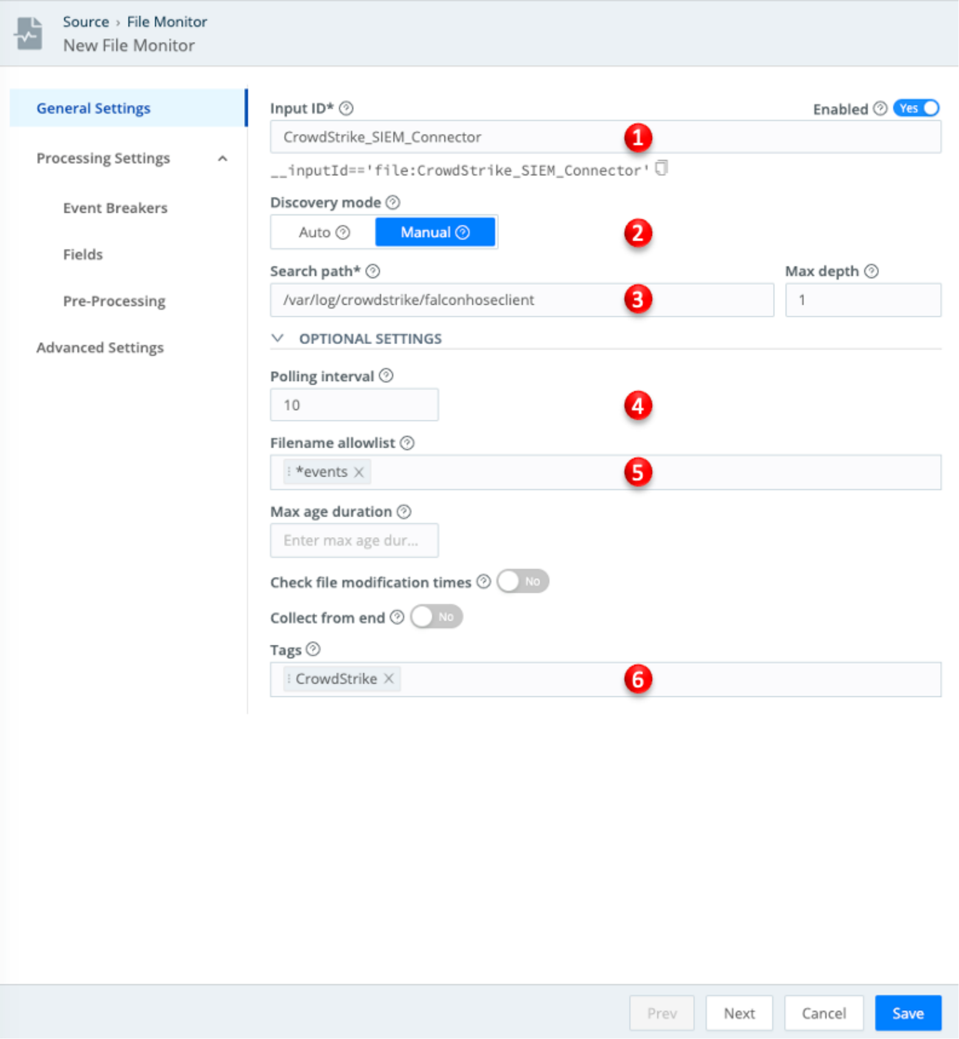
3. Select 'Add New' to create a new file collection.



4. There are 4 areas of concern for this Source configuration – Under Configure those areas are: **General Settings**, **Event Breakers**, **Pre-Processing** and **Destination**.



5. **General Settings**: Configure the new File Monitor.



Input ID*: CrowdStrike_SIEM_Connector

Enabled: Yes

__inputId=='file:CrowdStrike_SIEM_Connector'

Discovery mode: Auto ● Manual

Search path*: /var/log/crowdstrike/falconhoseclient

Max depth: 1

OPTIONAL SETTINGS

Polling interval: 10

Filename allowlist: *events

Max age duration: Enter max age dur...

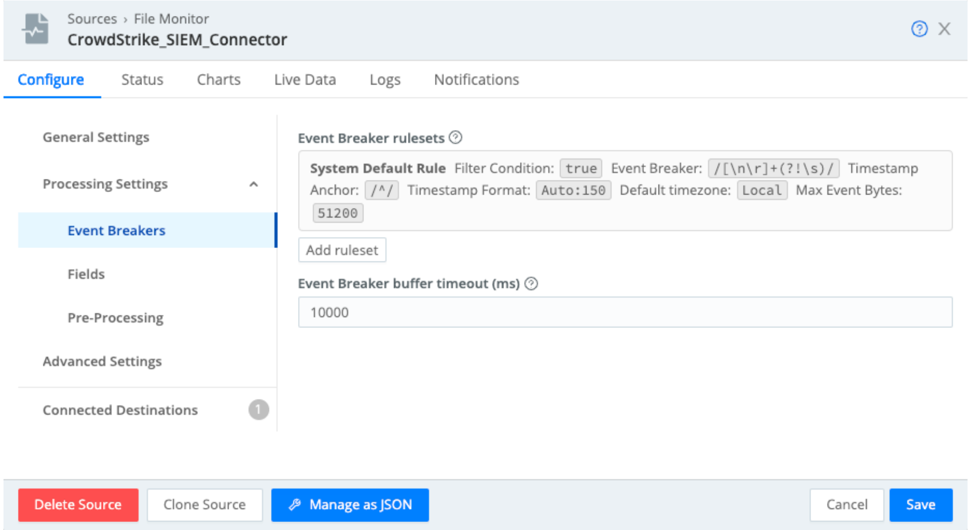
Check file modification times: No

Collect from end: No

Tags: CrowdStrike

1. **Input ID**: Configure an input name for the file monitor data collection.
2. **Discovery mode**: Set the discovery mode to ‘Manual’ to configure the Search Path.
3. **Search Path**: Configure the path to the output file location configured in the SIEM Connector configuration.
4. **Polling Interval**: Configure the interval the Edge agent should use to collect the data.
5. **Filename allowlist**: Configure the name of the output file as it was configured in the SIEM Connector configuration.
6. **Tags**: Optional – Gives the source a tag for filtering and grouping with in Edge.

6. **Processing Settings – Event Breakers**: This configuration can leverage the default event breaker.



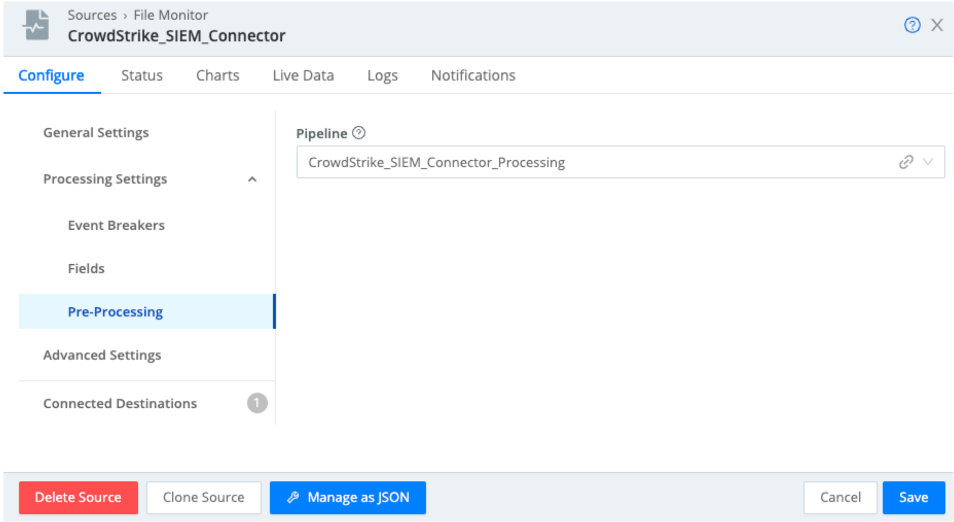
Event Breaker rulesets

System Default Rule Filter Condition: `true` Event Breaker: `/[\n\r]+(?!\s)/` Timestamp Anchor: `/^/` Timestamp Format: `Auto:150` Default timezone: `Local` Max Event Bytes: `51200`

Event Breaker buffer timeout (ms)

10000

7. **Processing Settings – Pre-Processing**: In this configuration, the pre-processing pipeline that was created earlier in this document need to be selected.

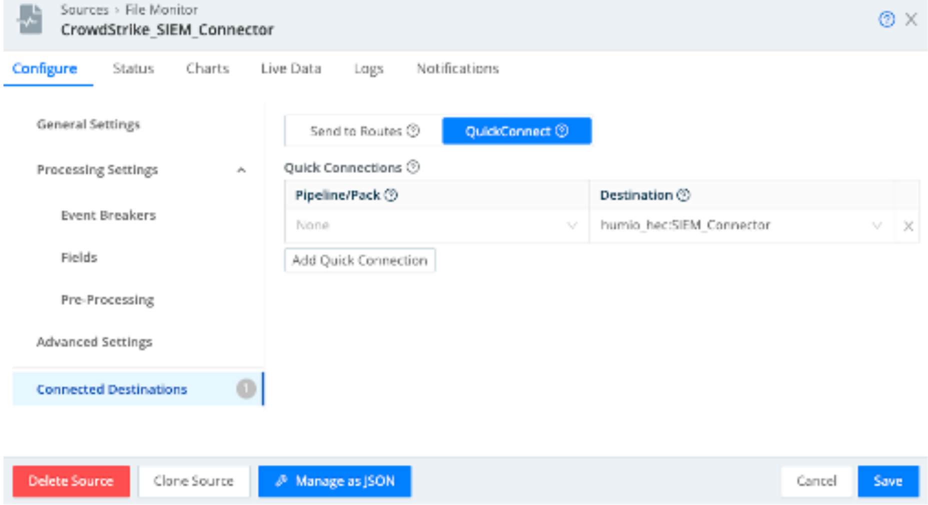


Pipeline

CrowdStrike_SIEM_Connector_Processing

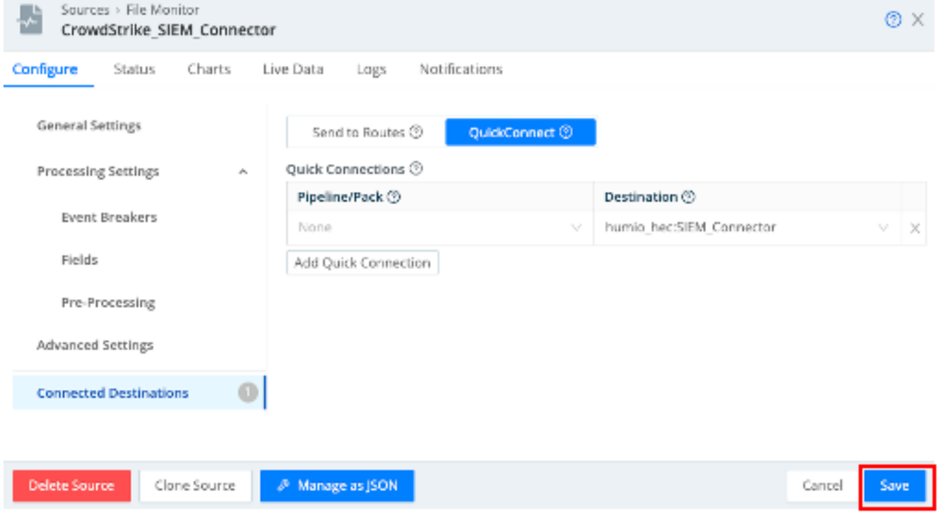
NOTE: FAILURE TO CONFIGURE THIS CAN RESULT IN NOT DATA BEING COLLECTED

8. **Connected Destinations** – This configuration can be to send the data to Routes for processing or to a QuickConnect destination. This example has the data being sent to a QuickConnect LogScale Destination.



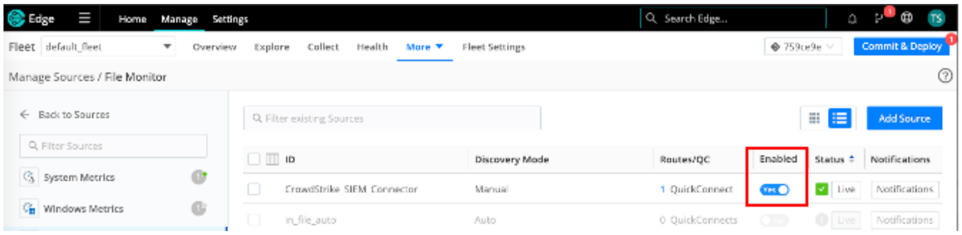
Pipeline/Pack	Destination
None	humio_hec:SIEM_Connector

9. Once these configurations are completed select 'Save' in the bottom right corner.



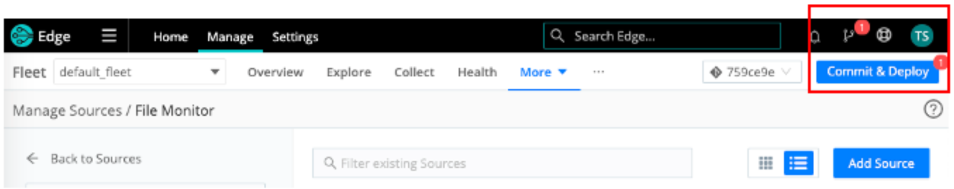
Pipeline/Pack	Destination
None	humio_hec:SIEM_Connector

10. Optional – If this data is not scheduled to be collected at the time of configuration the source and be disabled in the Manage Sources/File Monitor area.



ID	Discovery Mode	Routes/QC	Enabled	Status	Notifications
CrowdStrike_SIEM_Connector	Manual	1 QuickConnect	Yes	✓ Live	Notifications
in_file_auto	Auto	0 QuickConnects	No	Live	Notifications

11. Commit and deploy the changes in the top right of the page.



--- End of Section ---

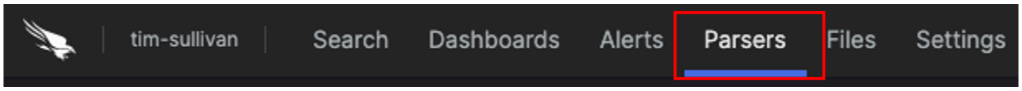
Sending to Falcon LogScale

This data can be sent to Falcon LogScale leveraging the LogScale Destination in Edge. The LogScale HEC token and parser should be configured prior to configuring the Cribl Edge LogScale destination.

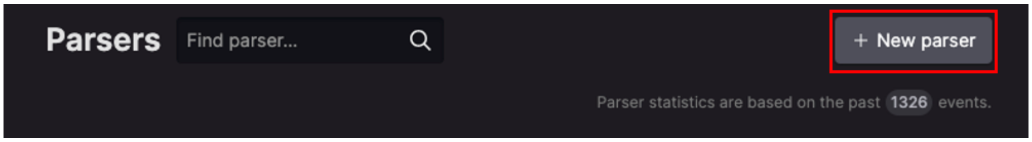
Configure LogScale

The information coming from Cribl Edge will be received by Falcon LogScale using an HEC input. This can be an existing HEC input but it’s recommended that a dedicated token and dedicated parser be configured for this data collection.

1. Creating a dedicated Parser is recommended as the first step so that it can be assigned to the token. In the LogScale UI select ‘Parsers’ in the top menu.

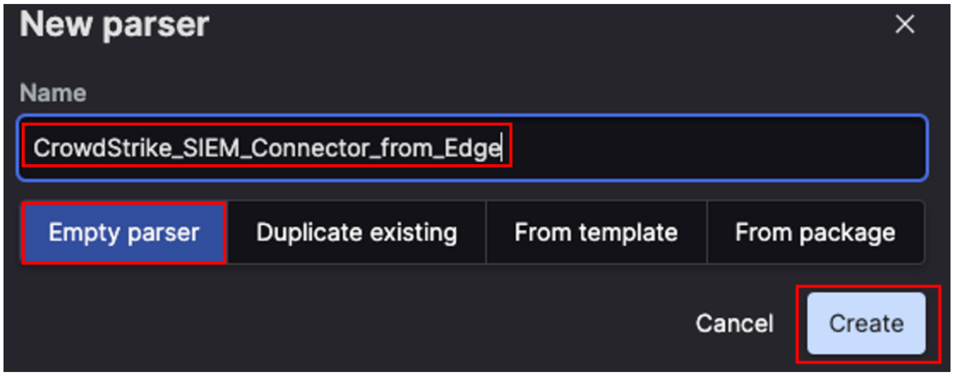


2. Select ‘New Parser’ in the ‘Parsers’ page.



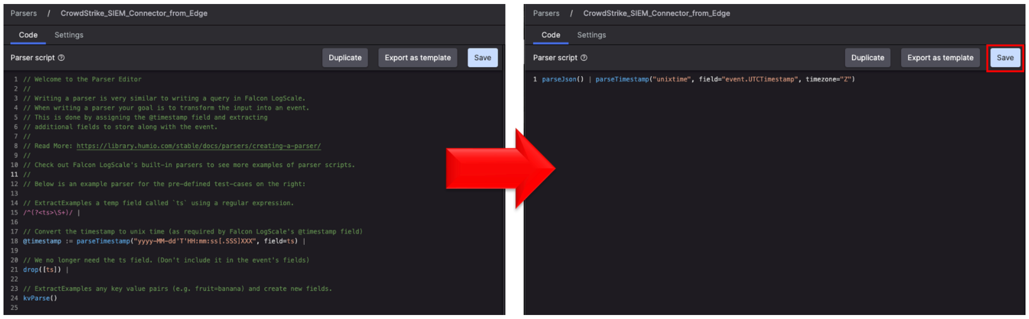
Parser statistics are based on the past 1326 events.

3. Select ‘Empty parser’, provide a name for the new parser and select ‘Create’:



Name: CrowdStrike_SIEM_Connector_from_Edge

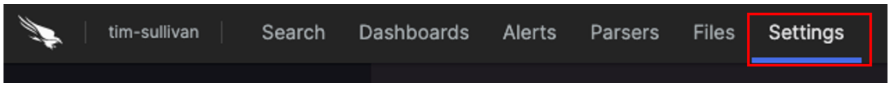
4. The parser used in this document is provided in Appendix B and is simply parsing the data as JSON and identifying the timestamp and timezone information. A more detailed parser can be created if desired.
   In the parser windows, remove the existing text and past in the parser from Appendix B and select 'Save'.



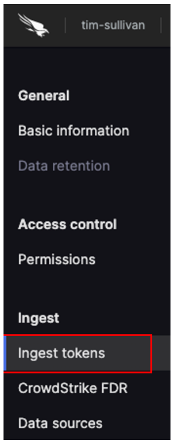
```
// Welcome to the Parser Editor
//
// Writing a parser is very similar to writing a query in Falcon LogScale.
// When writing a parser your goal is to transform the input into an event.
// This is done by assigning the @timestamp field and extracting
// additional fields to store along with the event.
//
// Read More: https://library.humio.com/stable/docs/parsers/creating-a-parser/
//
// Check out Falcon LogScale's built-in parsers to see more examples of parser scripts.
//
// Below is an example parser for the pre-defined test-cases on the right:

// ExtractExamples a temp field called `ts` using a regular expression.
/^(?<ts>\S+)/ |

// Convert the timestamp to unix time (as required by Falcon LogScale's @timestamp field)
@timestamp := parseTimestamp("yyyy-MM-dd'T'HH:mm:ss[.SSS]XXX", field=ts) |

// We no longer need the ts field. (Don't include it in the event's fields)
drop([ts]) |

// ExtractExamples any key value pairs (e.g. fruit=banana) and create new fields.
kvParse()
```

```
parseJson() | parseTimestamp("unixtime", field="event.UTCTimestamp", timezone="Z")
```

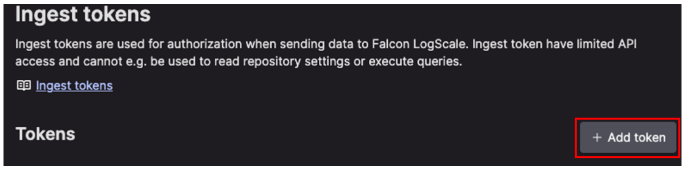
5. In the LogScale UI select 'Settings' from the menu.



6. In the menu on the left, select 'Ingest tokens'.



7. In the 'Ingest tokens' page, select 'Add token' from the middle window.



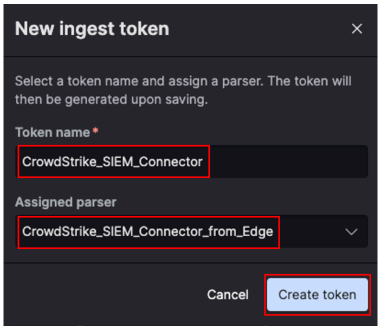
Ingest tokens

Ingest tokens are used for authorization when sending data to Falcon LogScale. Ingest token have limited API access and cannot e.g. be used to read repository settings or execute queries.

Ingest tokens

Tokens

8. In the ‘New ingest token’ popup window: provide a Token name, assign the parser that was created for this data and select ‘Create token’.



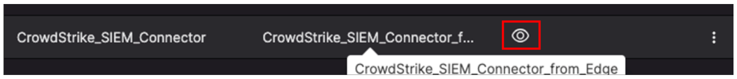
New ingest token

Select a token name and assign a parser. The token will then be generated upon saving.

Token name*: CrowdStrike_SIEM_Connector

Assigned parser: CrowdStrike_SIEM_Connector_from_Edge

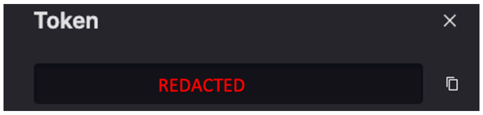
9. In the list of ingest tokens, locate the newly created token and select the eye icon to display the token value.



CrowdStrike_SIEM_Connector | CrowdStrike_SIEM_Connector_f...

CrowdStrike_SIEM_Connector_from_Edge

10. Record the token value for use in the Edge Destination configuration.



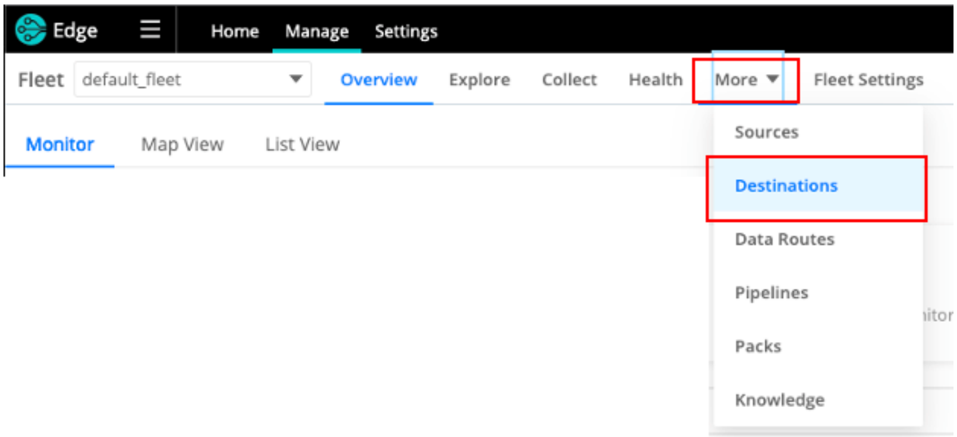
Token

REDACTED

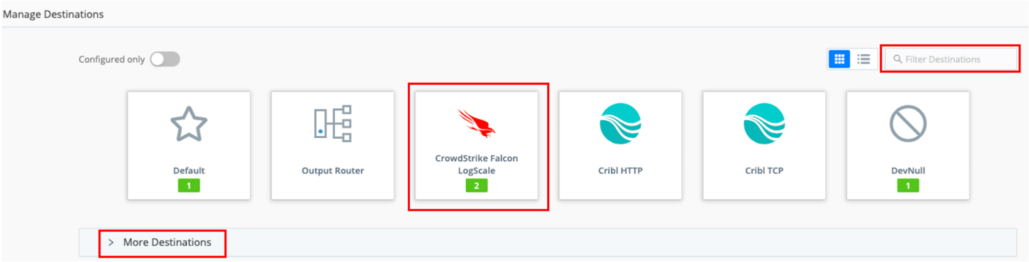
Configure Cribl Edge

Cribl Edge has a dedicated CrowdStrike Falcon LogScale Destination. Prior to configuring this Destination, a Falcon LogScale HEC token must have been created to provide authentication and it is also recommended to have a dedicated parser for parsing the incoming data.

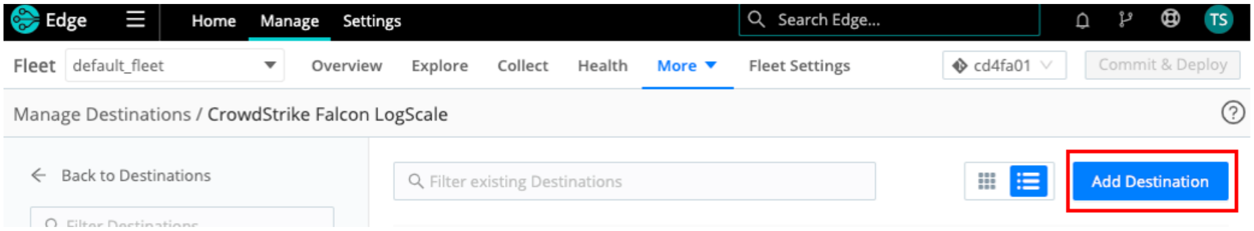
1. From the Fleet menu in Cribl Edge select ‘More’ and then ‘Destinations.



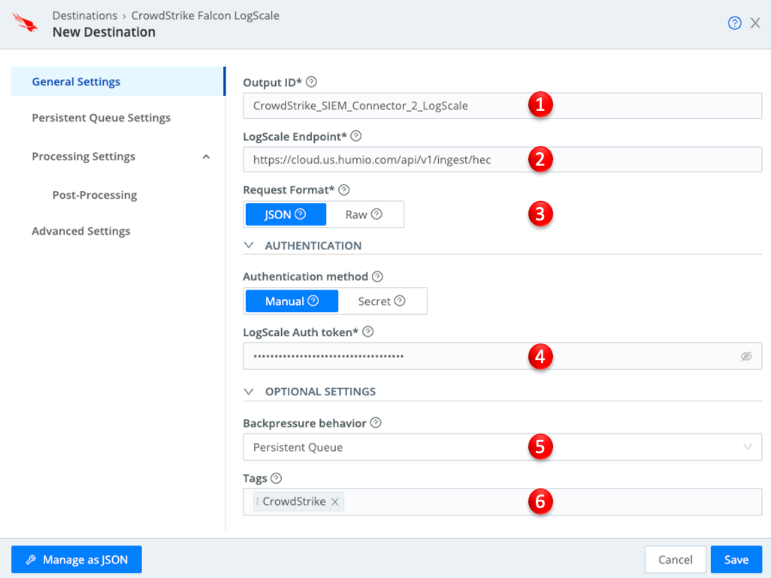
2. Locate the ‘LogScale’ Destination icon. *NOTE: If the icon is not visible, select ‘More Destinations’ or use the ‘Filter Destinations’ search box.



3. Under the LogScale Destination select ‘Add Destination’ in the right corner.



4. Complete the Destination configuration.



Output ID*: CrowdStrike_SIEM_Connector_2_LogScale

LogScale Endpoint*: https://cloud.us.humio.com/api/v1/ingest/hec

Request Format*: JSON | Raw

Authentication method: Manual | Secret

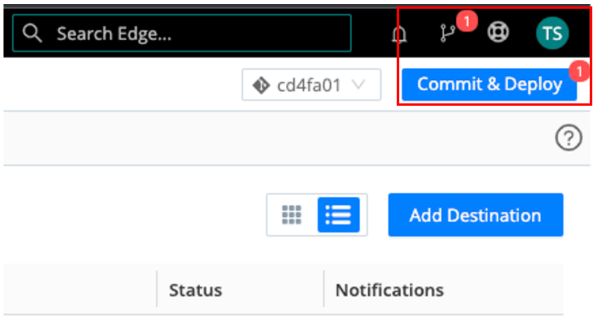
LogScale Auth token*: ••••••••••••••••••••••••••••••••••••

Backpressure behavior: Persistent Queue

Tags: CrowdStrike

1. **Output ID**: Configure an output name for the LogScale destination.
2. **LogScale Endpoint**: Set the HEC URL for the LogScale instance.
3. **Request Format**: Set the format of the data, this document's process outputs the data in JSON format.
4. **Authentication Method & LogScale Auth token**: Configure authentication method as 'Manual' and provide the LogScale HEC token.
5. **Backpressure behavior**: Configure desired the backpressure behavior.
6. **Tags**: Optional – Gives the destination a tag for filtering and grouping with in Edge.

5. Commit and deploy the configuration.



6. This destination can now be leveraged in Routes or QuickConnect configurations.

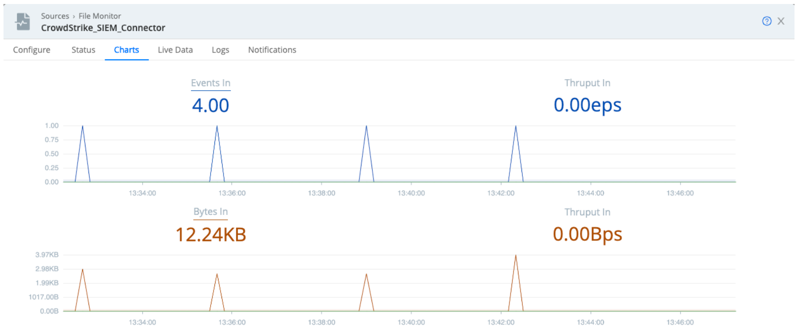
--- End of Section ---

Basic Troubleshooting

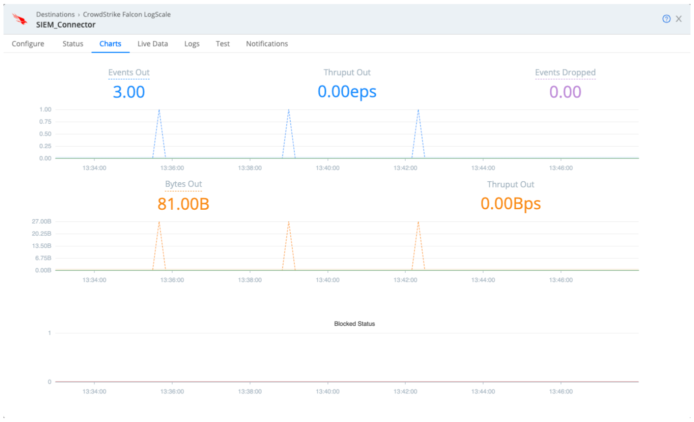
“There doesn’t appear to be data coming into the File Monitoring Source”

This can be cause by multiple issues, the most common causes are:

1. Ensure that the firewall on the Linux host running the CrowdStrike SIEM Connector is not blocking communication between the CrowdStrike API and the SIEM Connector code and that the firewall is not blocking communication between the Cribl Edge client and the Cribl Edge Cloud.
2. Ensure that the CrowdStrike SIEM Connector is properly configured and that there are events being created in the appropriate output file in the appropriate output location.
3. Validate that the Cribl Edge client is configured to collect data from the correct output file in the correct output location.
4. Validate that the CrowdStrike SIEM Connector is running.
5. Validate that the Cribl Edge File Monitor Source is enabled.
6. Check that the Pre-Processing Pipeline is properly configurated and has been configured in the Cribl Edge File Monitor Source.
7. Check the ‘Charts’ section of the Cribl Edge File Monitor Source to see if there are signs of events being collected.



8. Check the 'Charts' section of the Cribl Edge Destination to see if there are signs of events being sent to the proper destination.

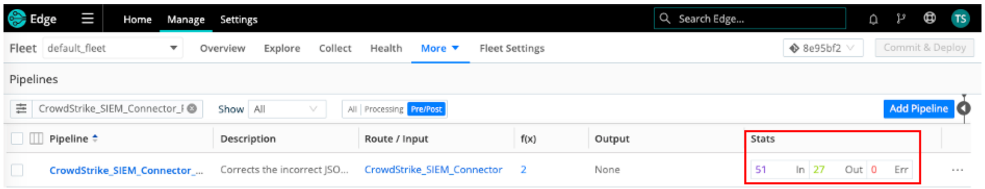


Destinations › CrowdStrike Falcon LogScale
SIEM_Connector

Configure | Status | Charts | Live Data | Logs | Test | Notifications

Events Out: 3.00
Thruput Out: 0.00eps
Events Dropped: 0.00
Bytes Out: 81.00B
Thruput Out: 0.00Bps

9. Check the status for the Pre-Processing Pipeline to see if there are any errors and if there are events being passed in and out.



Fleet: default_fleet

Pipelines

CrowdStrike_SIEM_Connector_F | Show: All | All | Processing | Pre/Post

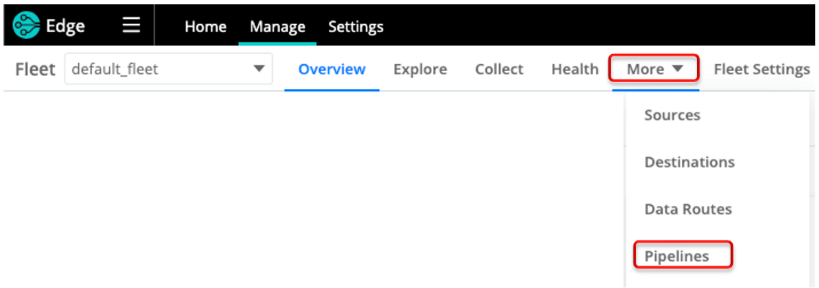
Pipeline	Description	Route / Input	f(x)	Output	Stats
CrowdStrike_SIEM_Connector_...	Corrects the incorrect JSO...	CrowdStrike_SIEM_Connector	2	None	51 In 27 Out 0 Err

“I’m not sure that the Pre-Processing Pipeline is working correctly”

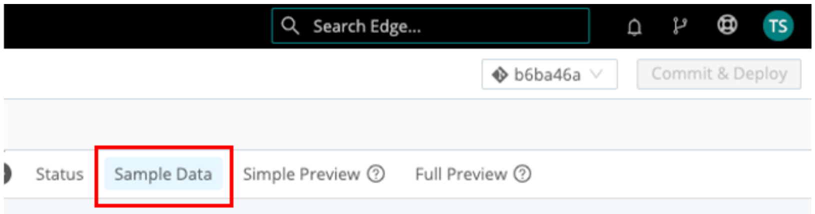
The most efficient way to test and validate that the Pre-Processing Pipeline configuration will produce the desired output is to test it with sample data from the CrowdStrike SIEM Connector. This sample is best if it’s collected from the SIEM Connector system that will ultimately be supplying the data. For how to capture sample data in Cribl Edge refer to the documentation: https://docs.cribl.io/edge/data-preview/#capturing-sample-data .
Alternately the data from the SIEM connector file can be imported into Edge.

Importing Sample Data from a File

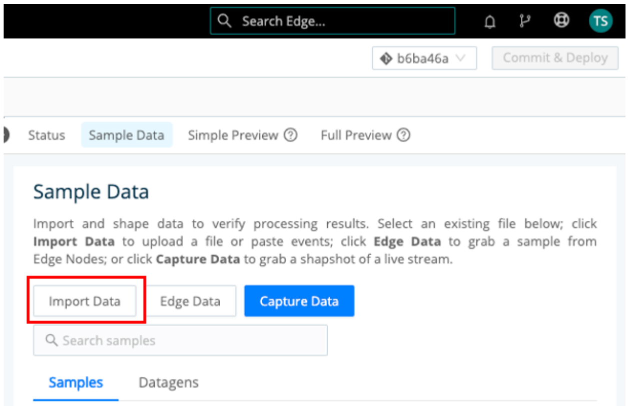
1. Navigate to the Pre-Processing Pipeline under ‘More’ – ‘Pipelines’ and select the pipeline.



2. If necessary, expand the sidebar section and select ‘Sample Data’.



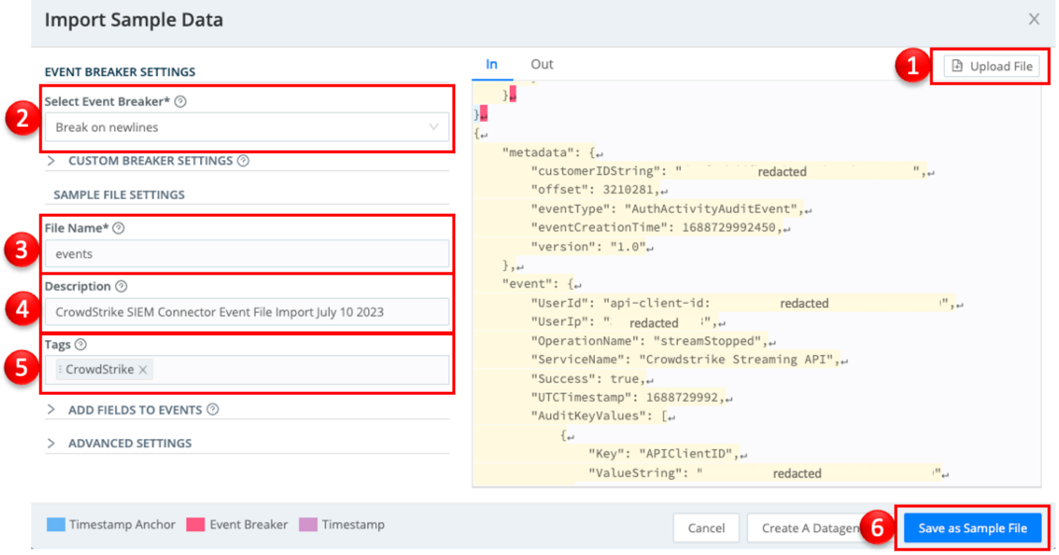
3. If uploading the sample data from a file, select ‘Import Data’



Sample Data

Import and shape data to verify processing results. Select an existing file below; click **Import Data** to upload a file or paste events; click **Edge Data** to grab a sample from Edge Nodes; or click **Capture Data** to grab a shapshot of a live stream.

4. Importing sample file from the SIEM connector requires that it be accessible on the local system that’s accessing the Edge UI.



Import Sample Data

EVENT BREAKER SETTINGS

Select Event Breaker*: Break on newlines

CUSTOM BREAKER SETTINGS

SAMPLE FILE SETTINGS

File Name*: events

Description: CrowdStrike SIEM Connector Event File Import July 10 2023

Tags: CrowdStrike

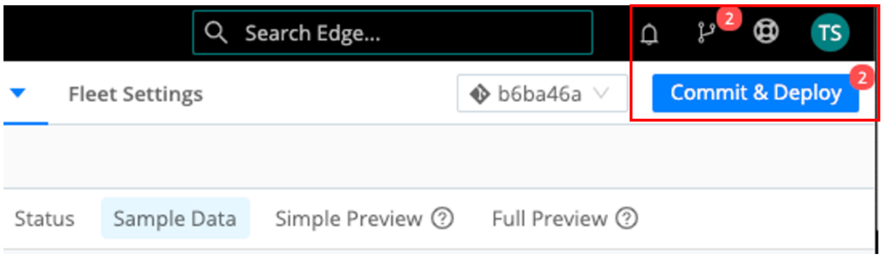
ADD FIELDS TO EVENTS

ADVANCED SETTINGS

```
    }
}
{
    "metadata": {
        "customerIDString": "            redacted                ",
        "offset": 3210281,
        "eventType": "AuthActivityAuditEvent",
        "eventCreationTime": 1688729992450,
        "version": "1.0"
    },
    "event": {
        "UserId": "api-client-id:         redacted             ",
        "UserIp": "  redacted   ",
        "OperationName": "streamStopped",
        "ServiceName": "Crowdstrike Streaming API",
        "Success": true,
        "UTCTimestamp": 1688729992,
        "AuditKeyValues": [
            {
                "Key": "APIClientID",
                "ValueString": "         redacted              "
```

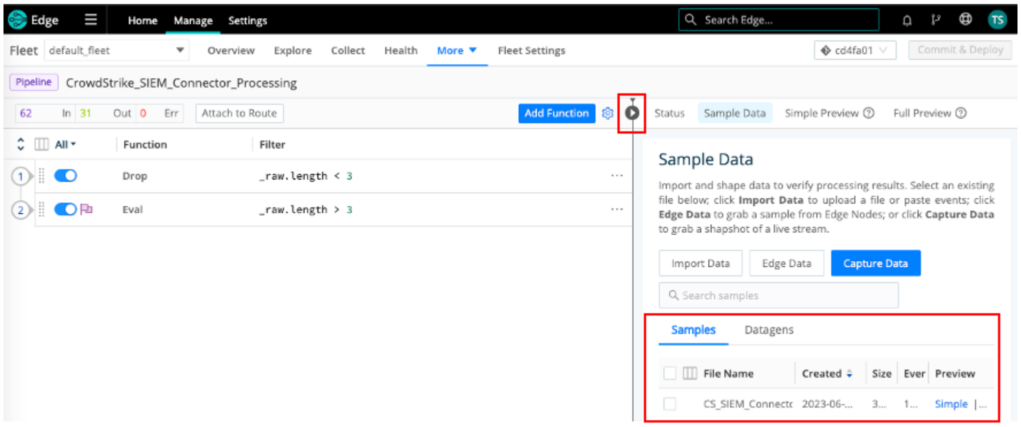
1. **Upload file**: Upload a file from the local system being used to access Edge.
2. **Select Event Breaker**: Set the event breaker to ‘Break on newlines’.
3. **File Name**: The name of the file that was uploaded.
4. **Description**: (optional) A description of the sample data.
5. **Tags**: (optional) Tags within Edge for grouping purposes.
6. **Save as Sample File**: Once the data looks correct save the sample file.

5. Commit and deploy the changes.



Using Sample Data for Testing Output

1. If necessary, expand the sidebar section, locate the sample data and select ‘Simple’ under ‘Preview’.



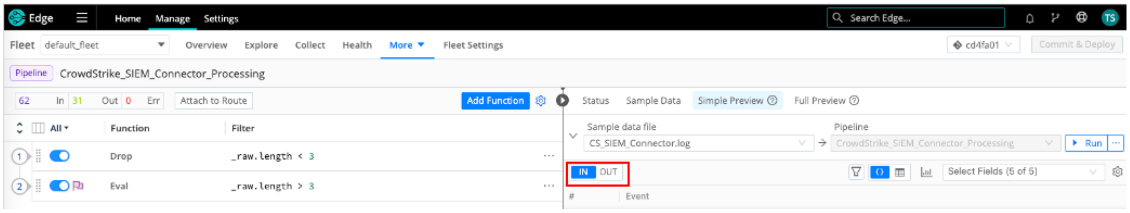
#	Function	Filter
1	Drop	`_raw.length < 3`
2	Eval	`_raw.length > 3`

Sample Data

Import and shape data to verify processing results. Select an existing file below; click **Import Data** to upload a file or paste events; click **Edge Data** to grab a sample from Edge Nodes; or click **Capture Data** to grab a shapshot of a live stream.

File Name	Created	Size	Ever	Preview
CS_SIEM_Connectc	2023-06-...	3...	1...	Simple |...

2. The ‘In’ and ‘Out’ selections can now be used to view the data as it will look coming into the Pre-Processing Pipeline and also how it will look coming out.

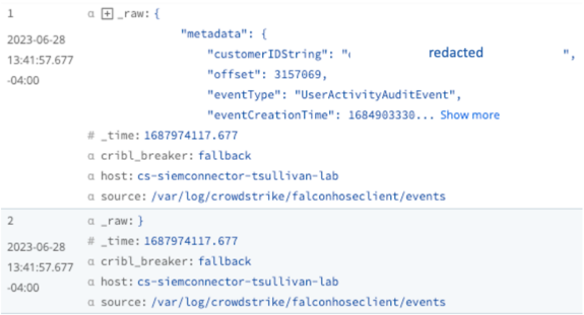


#	Function	Filter
1	Drop	`_raw.length < 3`
2	Eval	`_raw.length > 3`

Sample data file: CS_SIEM_Connector.log

Pipeline: CrowdStrike_SIEM_Connector_Processing

3. The ‘IN’ view of the data should look similar to the following.



```
1                  a ⊞ _raw: {
                              "metadata": {
2023-06-28                        "customerIDString": "           redacted              ",
13:41:57.677                      "offset": 3157069,
-04:00                            "eventType": "UserActivityAuditEvent",
                                  "eventCreationTime": 1684903330... Show more
                   # _time: 1687974117.677
                   a cribl_breaker: fallback
                   a host: cs-siemconnector-tsullivan-lab
                   a source: /var/log/crowdstrike/falconhoseclient/events
2                  a _raw: }
                   # _time: 1687974117.677
2023-06-28         a cribl_breaker: fallback
13:41:57.677       a host: cs-siemconnector-tsullivan-lab
-04:00             a source: /var/log/crowdstrike/falconhoseclient/events
```

4. The ‘OUT’ view of the data should look like the corrected JSON.

--- End of Section ---

Support

The documentation is provided as an example of how Cribl Edge can be used in conjunction with the CrowdStrike SIEM connector. Support for this process depends on where the issue is taking place.

For issues specific to the CrowdStrike SIEM connector:

- Review the published documentation to ensure that the SIEM connector has been properly deployed and configured on a supported operating system:
- If necessary open a support ticket with CrowdStrike Support at https://supportportal.crowdstrike.com. Include specific information about the SIEM Connector deployment, configuration and the issue(s) that are currently present. Provide an available log file and any other information outlined in the SIEM connector documentation.

For issues specific to Cribl Edge:

- Review the appropriate support option(s) here: https://cribl.io/support/.

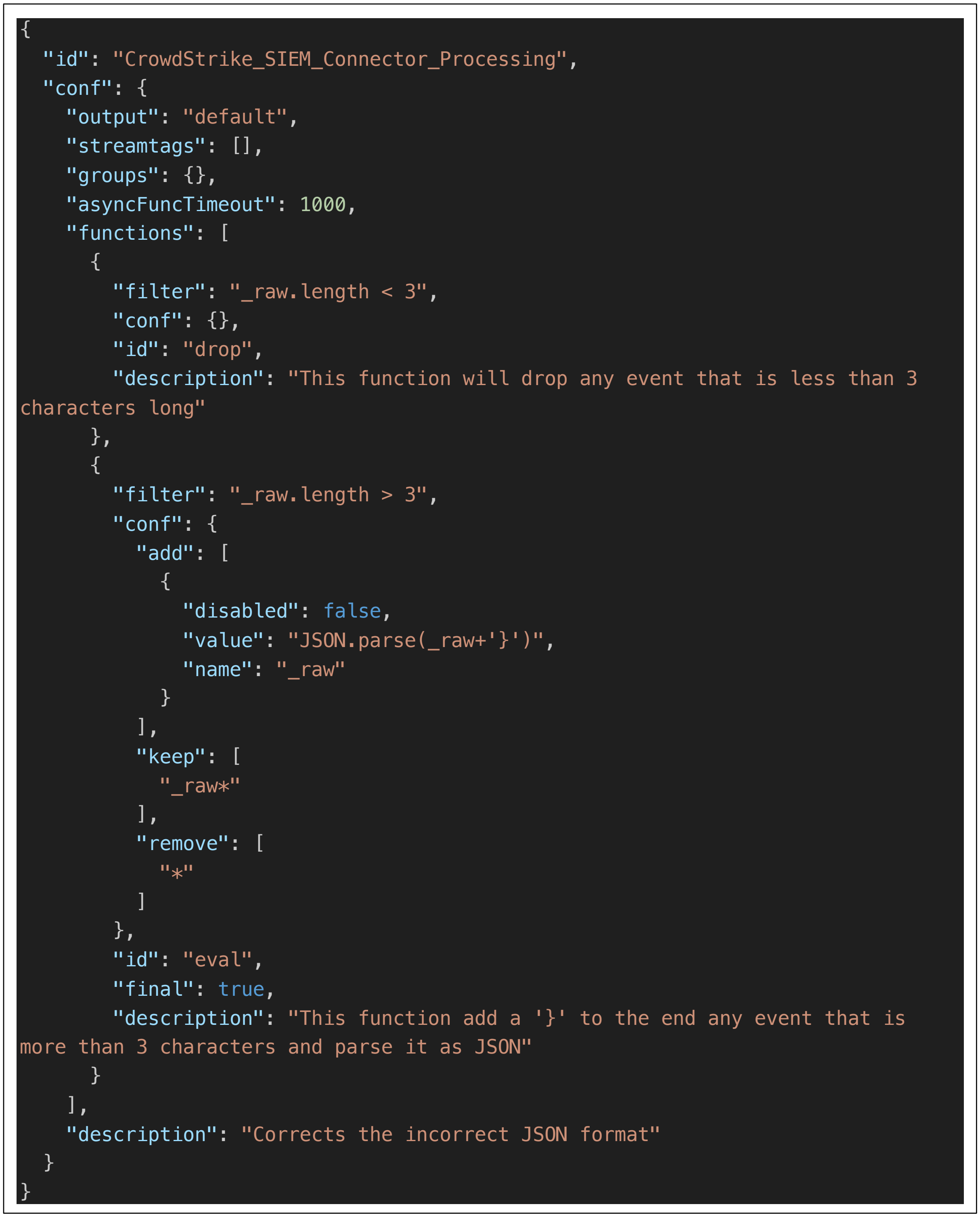
For issues related to the process outlined in this documentation:

- Ensure that both platforms are functioning correctly.
- If necessary open a support ticket with CrowdStrike Support at https://supportportal.crowdstrike.com.
  - Provide log files from the SIEM connector deployment.
  - Provide screenshots of Cribl Edge configuration.
  - Provide examples/screenshots of live data collection within Cribl Edge.

Due to the nature of this process CrowdStrike may not be able to resolve all support requests.

Appendix A

Pre-Processing Pipeline JSON Example: CrowdStrike_SIEM_Connector_Processing



```
{
  "id": "CrowdStrike_SIEM_Connector_Processing",
  "conf": {
    "output": "default",
    "streamtags": [],
    "groups": {},
    "asyncFuncTimeout": 1000,
    "functions": [
      {
        "filter": "_raw.length < 3",
        "conf": {},
        "id": "drop",
        "description": "This function will drop any event that is less than 3
characters long"
      },
      {
        "filter": "_raw.length > 3",
        "conf": {
          "add": [
            {
              "disabled": false,
              "value": "JSON.parse(_raw+'}')",
              "name": "_raw"
            }
          ],
          "keep": [
            "_raw*"
          ],
          "remove": [
            "*"
          ]
        },
        "id": "eval",
        "final": true,
        "description": "This function add a '}' to the end any event that is
more than 3 characters and parse it as JSON"
      }
    ],
    "description": "Corrects the incorrect JSON format"
  }
}
```

Appendix B

Basic LogScale Parser Example:

```
parseJson() | parseTimestamp("unixtime", field="metadata.eventCreationTime",
timezone="Z")
```