CrowdStrike Falcon Event Streams Add-on

Installation and Configuration Guide

Release Tracking

V 2.0.9 – Released May 2021: Coding modifications for improved network communications, customer alert for inputs, custom action for inputs, additional search macros

Overview

This document outlines the deployment and configuration of the technology add-on for CrowdStrike Falcon Event Streams.

This technical add-on (TA) facilitates establishing a connecting to the CrowdStrike Event Streams API to receive event and audit data and index it in Splunk for further analysis, tracking and logging. It is a replacement for the previous TA "CrowdStrike Falcon Endpoint Add-on" (https://splunkbase.splunk.com/app/3944/#/overview) and does not serve or install as an upgrade.

The major differences for the Event Streams Add-on vs the Endpoint Add-on are:

	Event Streams Add-on	Endpoint Add-on
API Credentials	OAuth2 Only	Legacy Only
Cloud Environments	US Commercial US Commercial 2 US GovCloud EU Cloud	US Commercial
DataFeed URLs	Multiple	Single
Custom IOC	n/a	Limited functionality
Splunk: Python 3	Supported	Not Supported
Alert Actions	n/a	Limited functionality

Multitenancy - This TA is able to have multiple independent inputs enabled at the same time, each collecting data from different Falcon Instances and storing it in independent indexes.

Contents:

Getting Started

Prior to deploying the CrowdStrike Falcon Event Streams Technical Add-on (TA) ensure the following:

1. The latest version of the TA has been downloaded from Splunkbase
2. All Splunk systems that the TA will be deployed to have been identified
3. An account with proper access to identified Splunk systems is available
4. CrowdStrike support has enabled the Event Streams API for the instance (this API is disabled by default)
5. Properly scoped API credentials have been created and recorded from the Falcon UI
6. Any custom indexes being used have been created on the appropriate systems
7. (optional) – If the communication between Splunk and the Falcon platform will traverse a proxy server then appropriate configurations should be taken into account. If the connection will need to authenticate to the proxy then appropriate credentials should be created and available.

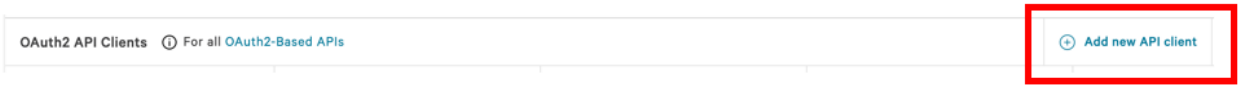
Enable Access to the Event Streams API

*Note this process is not required if there is an existing API client with proper access but it is recommended to leverage a dedicated account for the TA.

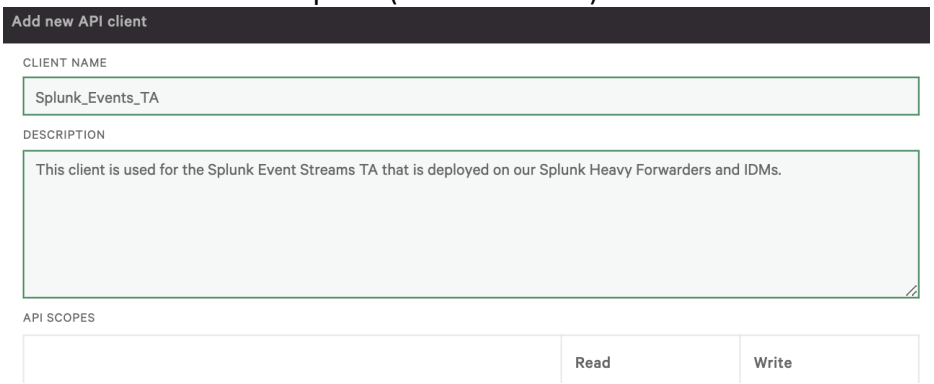
1. Log into the Falcon UI with an account that has administrator level permissions

2. Navigate to ‘Support’, ‘API Clients and Keys’ in the Falcon menu:

3. Select ‘Add new API Client’ to the right of ‘OAuth2 API Clients’:



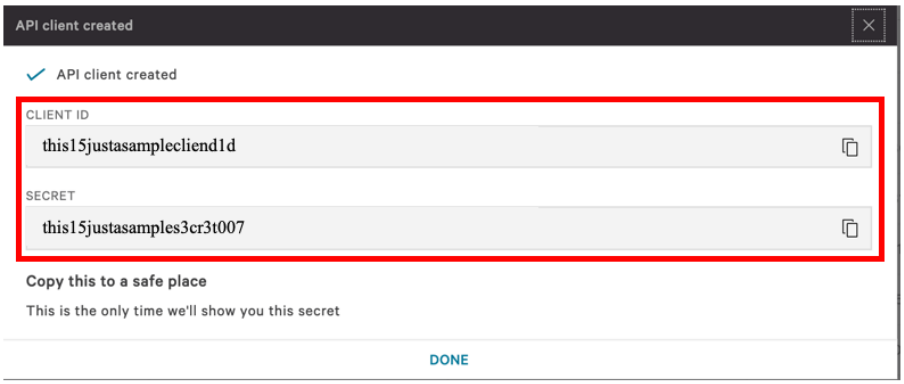
4. Provide a client name and description (recommended):



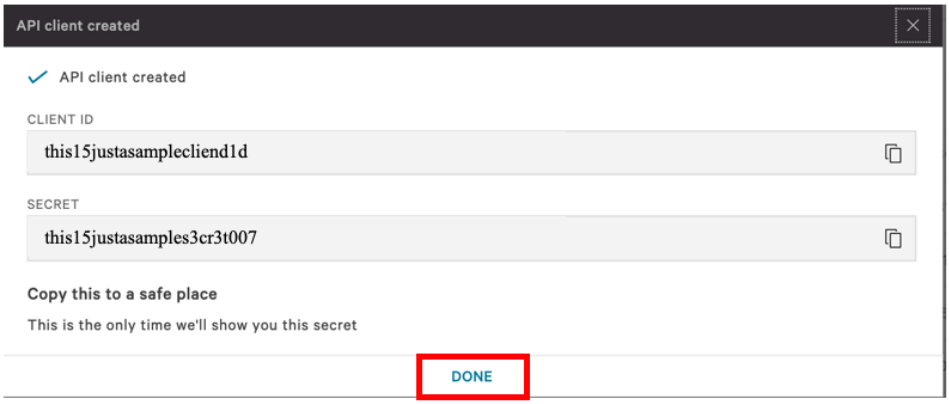
5. Under ‘API Scopes’ select the ‘Read’ check box next to ‘Event streams’:

6. Click ‘ADD’ to create the client:

7. A pop-up window will appear with the newly created Client ID and Secret
Ensure to record the secret correctly and store it in a safe place as this is the only time it will be visible/accessible



8. Once the credentials have successfully copied to a safe and secure location click ‘DONE’ to close the window:



Proxy Considerations

The CrowdStrike Technical Add-On establishes a secure persistent connection with the Falcon cloud platform. In some environments network devices may impact the ability to establish and maintain a secure persistent connection and as such these devices should be taken into account and configuration modifications should be done when necessary.

Ensure that the API URLs/IPs for the CrowdStrike Cloud environment(s) are accessible by the Splunk Heavy forwarder. For a complete list of URLs and IP address please reference CrowdStrike's API documentation.

The current base URLs for OAuth2 Authentication per cloud are:

US Commercial Cloud	: https://api.crowdstrike.com
US Commercial Cloud 2	: https://api.us-2.crowdstrike.com
US GovCloud	: https://api.laggar.gcw.crowdstrike.com
EU Cloud	: https://api.eu-1.crowdstrike.com

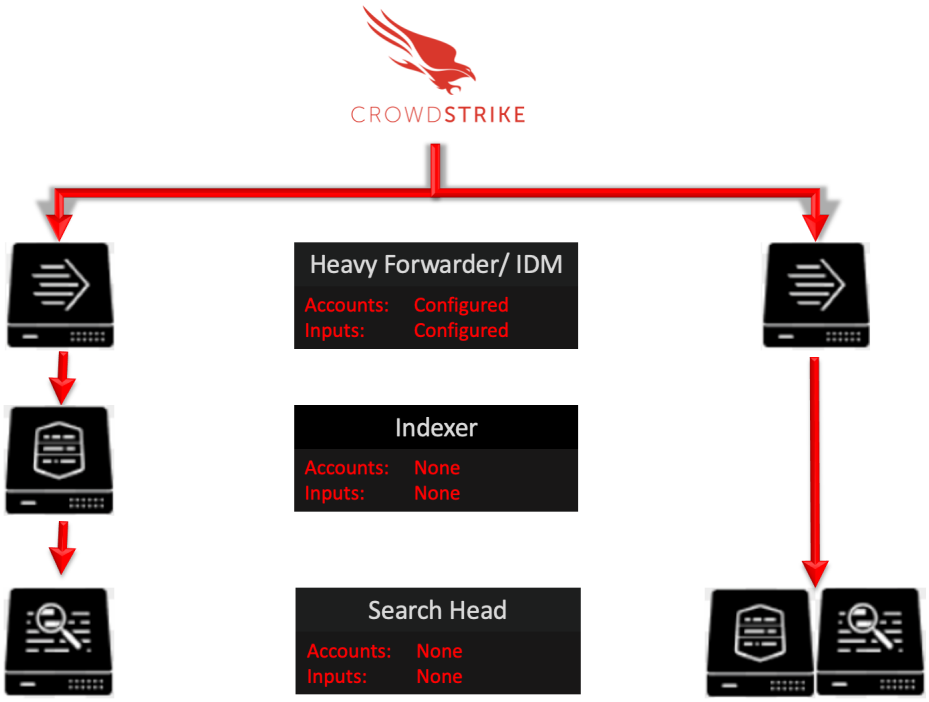
Splunk Architecture

Splunk Search Head(s) and Splunk Cloud: The TA should be installed to provide field mapping and search macro support. These are often required to support CrowdStrike Apps. The TA should be deployed without any accounts or inputs configured and any search macros should be properly configured for use.

Splunk Indexer(s): The TA can be installed to provide field mapping and search macro support. The TA should be deployed without any accounts or inputs configured and any search macros should be properly configured for use. If a custom index is going to be used, then it should be created here.

Splunk Heavy Forwarder(s) & Information Data Managers (IDMs): The TA should be installed here as this is where the data from the Streaming API will be collected. The appropriate accounts or inputs should be properly configured for data collection. If the Heavy Forwarder is storing events prior to forwarding them to the Indexer and a custom index is being used, ensure that the index has been created on both the Heavy Forwarder as well as the Indexer(s).
Note: Due to python requirements the TA can only be installed on Heavy Forwarders and IDMs.

The following diagram shows the flow of data from the Streaming API and the Event Streams TA configuration within a distributed Splunk Enterprise and Splunk Cloud environment:



Splunk Enterprise: Distributed

Splunk Cloud

Add-On and Alert Action Logging

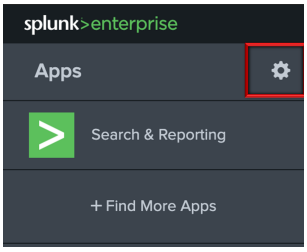
The Add-On logs can be found under: $Splunk/var/log/splunk/ and begin with 'ta_crowdstrike_falcon_event_streams' . These logs contain information about the configuration of the Add-On, API calls made to both CrowdStrike's API as well as the internal Splunk API's and other functionality

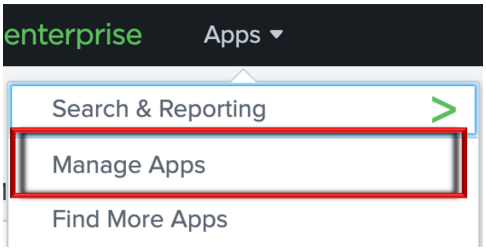
The Alert Action logs are separate from the Add-On logs but are also located under: $Splunk/var/log/splunk/ and begin with 'crowdstrike_event_streams_restart_input_modalert'

Initial Installation / Re-Installation / Manual Update
Heavy Forwarders, Information Data Managers, Indexers, Search Heads and Splunk Cloud

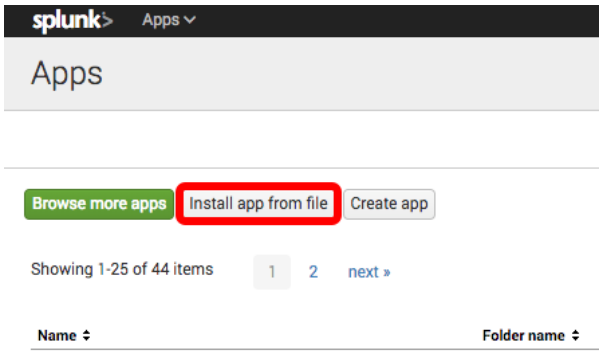
PERFORMING THIS ACTION REQUIRES A SYSTEM RESTART

1. From the Splunk home page or dropdown menu select ‘Manage Apps’:

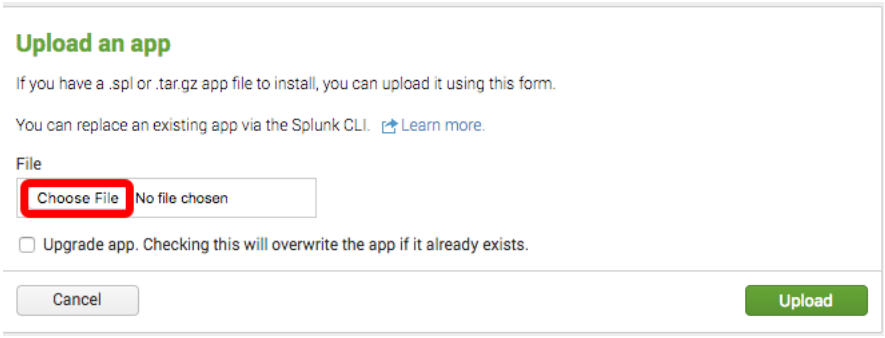




2. From the Manage Apps menu select ‘Install app from file’

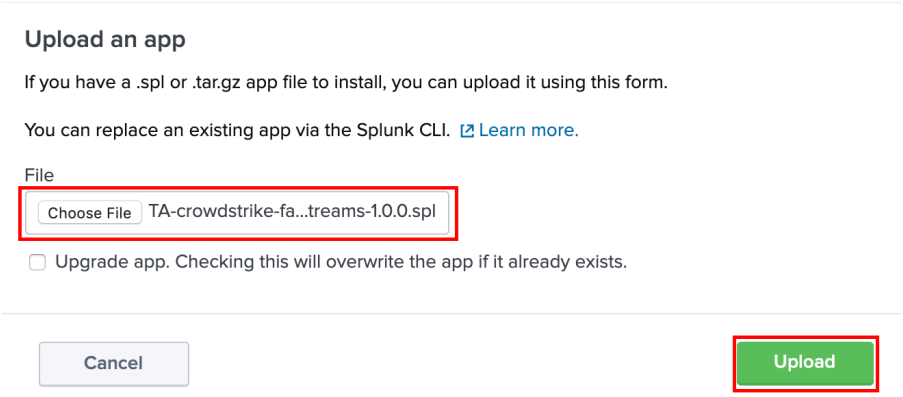


3. From the ‘Upload an app’ window, select ‘Choose File’ *
   *if upgrading or reinstalling an existing installation check the ‘Upgrade app’ checkbox



4. Select the downloaded Falcon Event Streams add-on file

5. Once the file is selected click ‘Upload’ to upload the add-on to system



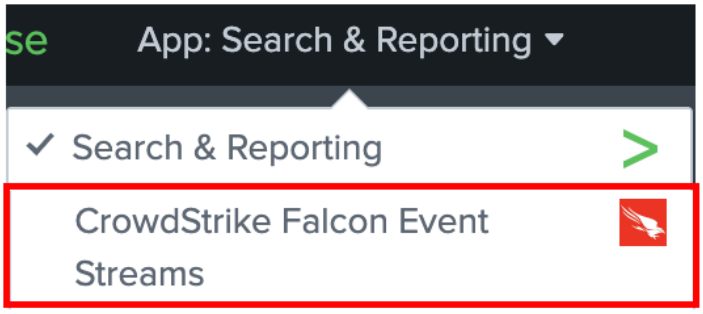
6. Once the add-on has been installed the system will require a restart for the add-on to complete installation.

----This concludes the Initial Installation / Re-Installation / Manual Update process----

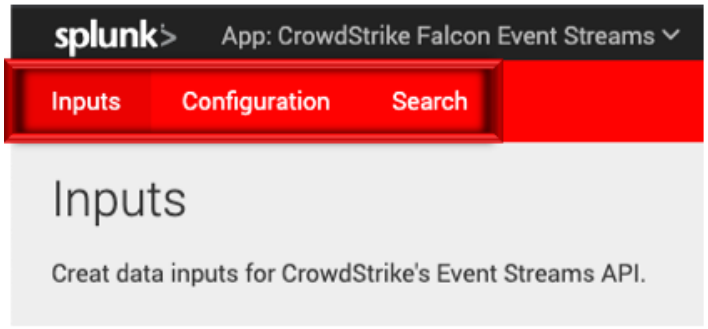
Heavy Forwarder/ Information Data Manager Configuration

This TA only supports connections to the Event Streams OAuth2 based API.

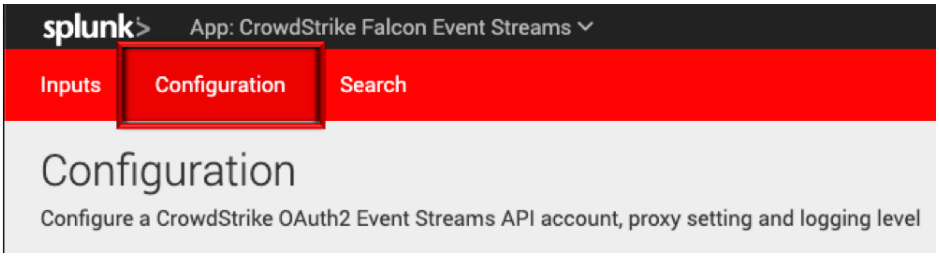
1. From the Splunk drop down menu select 'CrowdStrike Falcon Event Streams'



2. There are three sub menus within the add-on: 'Inputs', 'Configuration' and 'Search'

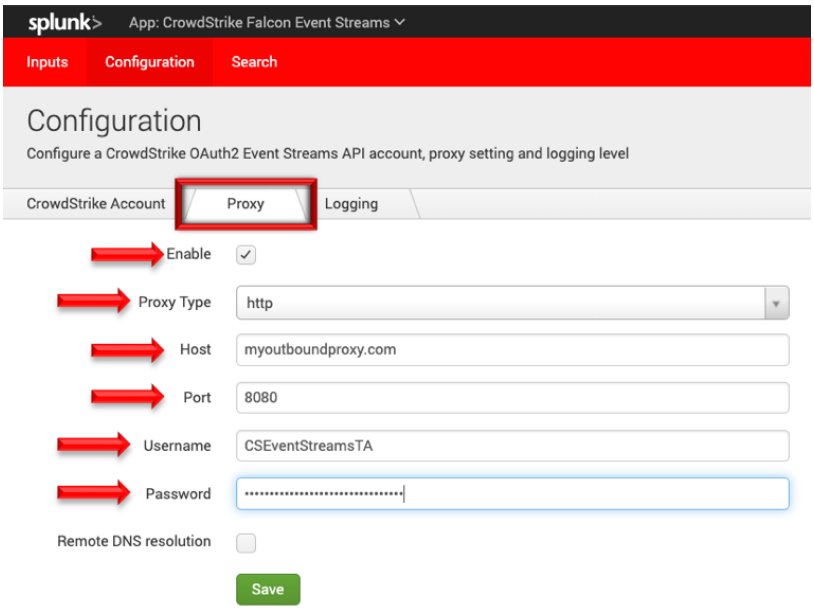


3. Select the submenu 'Configuration'



Proxy Configuration (Optional)

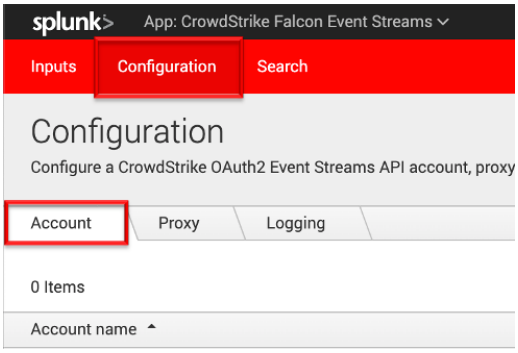
Select the ‘Proxy’ tab under ‘Configuration’ - Check the ‘Enable’ checkbox, select the Proxy Type from the drop down, enter the proxy host name, the proxy port and the credentials to allow communication.



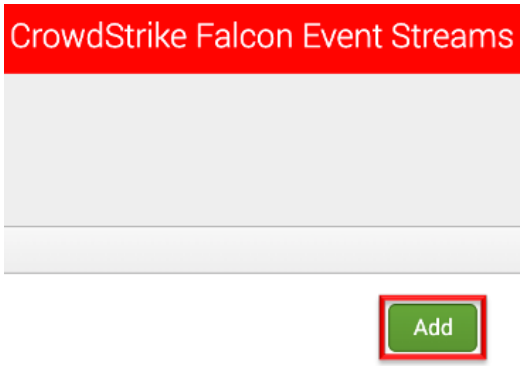
Event Streams TA Account Configuration

This TA only supports connections to the Event Streams OAuth2 based API.

1. Select the ‘Account’ tab under ‘Configuration’

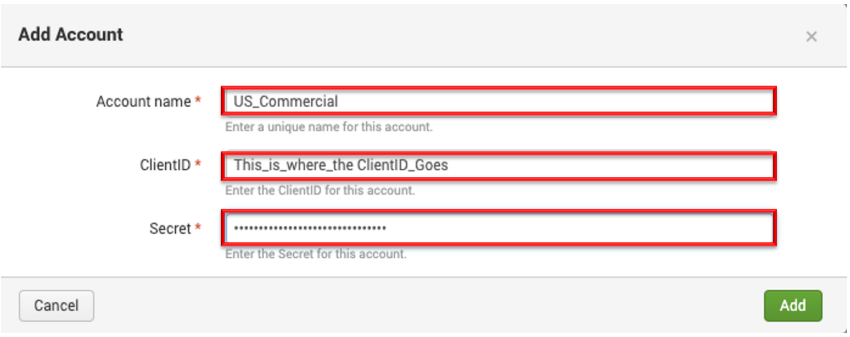


2. On the right-hand side select ‘Add’



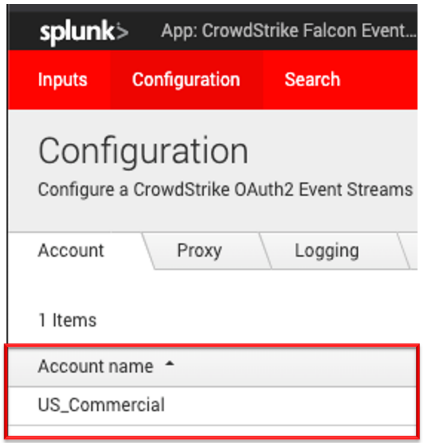
3. Configure the account for the Event Stream by providing the following:

   - **Account Name** – This is a unique name for the account within Splunk
   - **ClientID** – This is the ClientID for the API credential created
   - **Secret** – This is the Secret for the API credential created



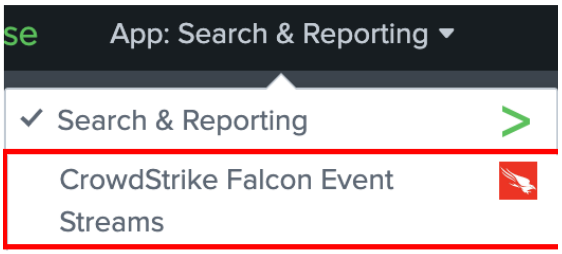
4. Once the information has been entered correctly click 'Add'

5. Validate that the account was saved successfully

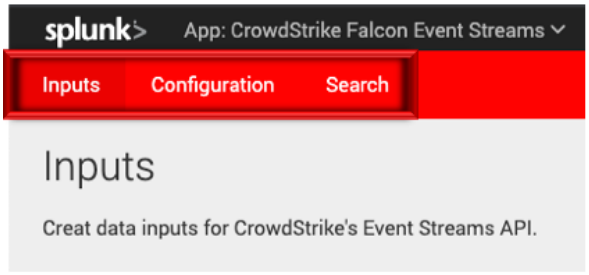


Event Streams TA Inputs Configuration

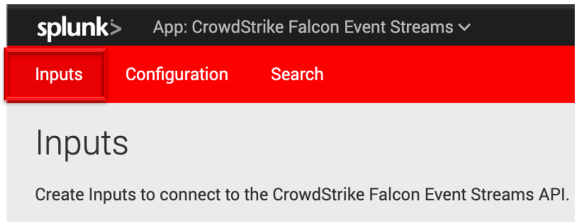
1. From the Splunk drop down menu select ‘CrowdStrike Falcon Event Streams’



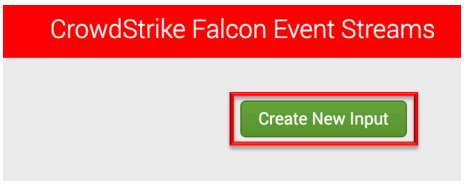
2. There are three sub menus within the add-on: ‘Inputs’, ‘Configuration’ and ‘Search’



3. Select the “Inputs” sub menu:

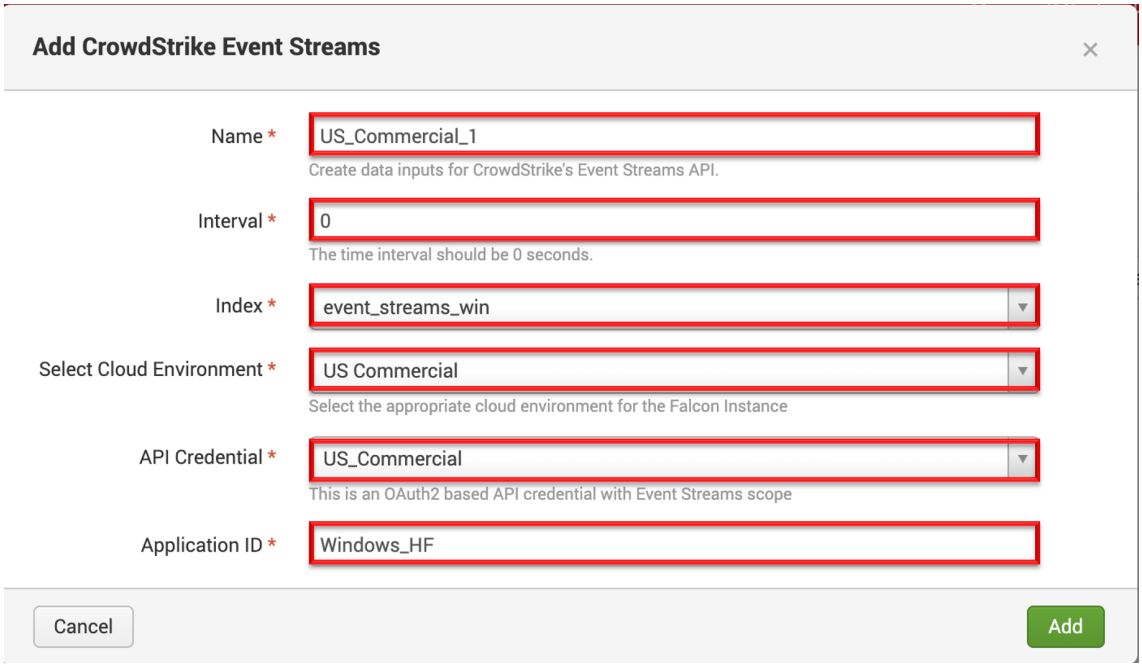


4. On the right-hand side, click ‘Create New Input’



5. Configure the input for the Event Stream by indicating the following:

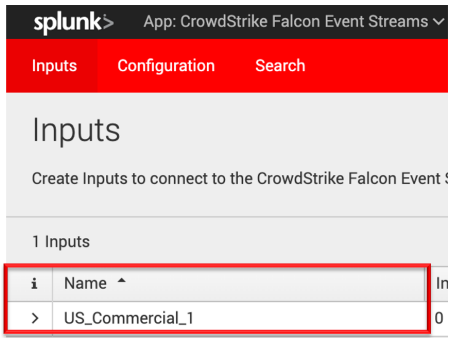
   - **Name** – The Splunk unique name for the input being configured
   - **Interval** – This field is not used in this TA and is set to 0 by default
   - **Index** – The index that the data will be stored in (must an existing index)
   - **Cloud Environment** – The CrowdStrike cloud environment the Falcon instance being connected to resides in
   - **API Credential** – The corresponding API credential for the Falcon instance in the select Cloud Environment
   - **Application ID** – An identifier for the API calls being made back to CrowdStrike (15 character maximum)



6. Once the Input parameters have been correctly configured click ‘add’*

***Newly created inputs are enabled by default**

7. Ensure that the Input has been successfully saved



This concludes the Heavy Forwarder/Information Data Manager Configuration process

Search Macro Configuration

Search macros are reusable chunks of Search Processing Language (SPL) that you can insert into other searches. Search macros can be any part of a search, such as an eval statement or search term, and do not need to be a complete command. You can also specify whether the macro field takes any arguments.
https://docs.splunk.com/Documentation/Splunk/8.0.3/Knowledge/Definesearchmacros

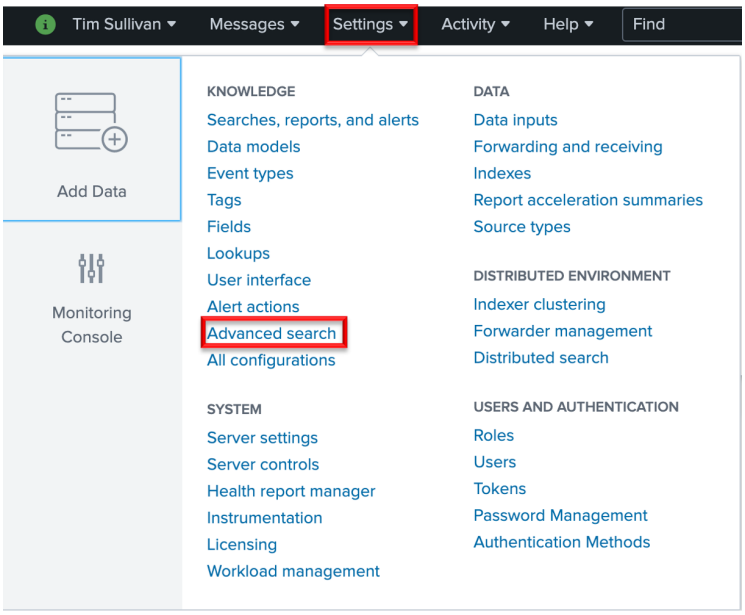
As of version 2.0.9 the Event Streams TA contains 4 search macros:

Name	Definition	Arguments
cs_es_get_index	index="*"	
cs_es_reset_action_logs	index=_internal sourcetype=splunkd component=sendmodalert action="restart_CS_ES_input"	
cs_es_ta_logs	index="_internal" sourcetype="tacrowdstrikefalconeventstreams:log"	
cs_es_tc_input(1)	index="_internal" sourcetype="tacrowdstrikefalconeventstreams:log" "OAuth2 Token was successfully" "for input: $input$"	input

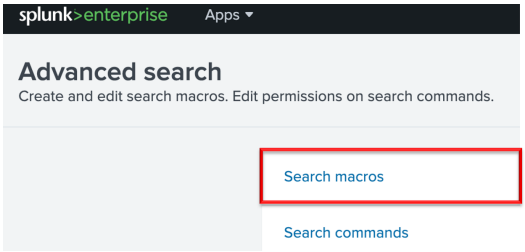
- **cs_es_get_index** (CrowdStrike Event Streams get index)A search macro that points to the index(es) that contain the data received from the Event Stream API. The default for this search macro is to point to all indexes to search for data but should be adjusted to reflect the specific index(es) that the Heavy Forwarder/IDMs are pushing the data to.
- **cs_es_reset_action_logs**: A search macro that provides access to the 'CrowdStrike Event Streams – Restart Input' alert action logs.
- **cs_es_ta_logs**: A search macro that provides access to the CrowdStrike Event Streams TA logs.
- **cs_es_tc_input(1)**: A search macro that's designed to work in conjunction with the 'CrowdStrike Event Streams – Restart Input' alert action. This search macro requires that an input name be declared.
    - The (1) in the search macro indicates that 1 variable needs to be provided – this would we the name of the CrowdStrike Event Stream TA input that is to be monitored.

The search macros can be modified as follows:

1. Select the ‘Settings’ dropdown menu in the Splunk bar and select ‘Advanced Search’

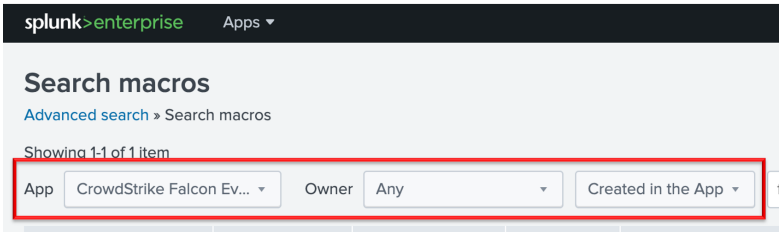


2. Select ‘Search macros’

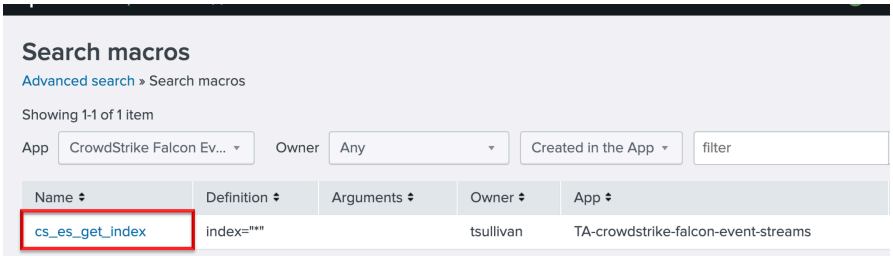


3. Configure the search settings as follows:

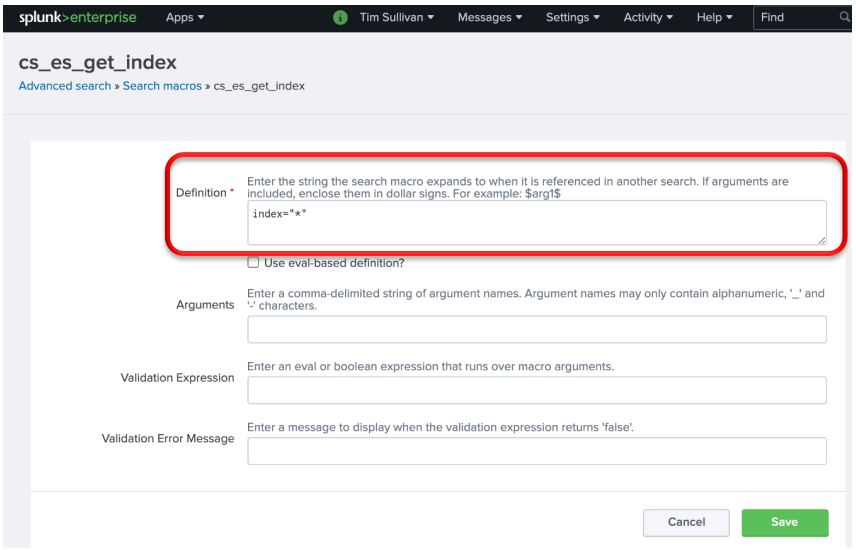
- **App** - CrowdStrike Falcon Event Streams
- **Owner** - Any
  - Created in the App



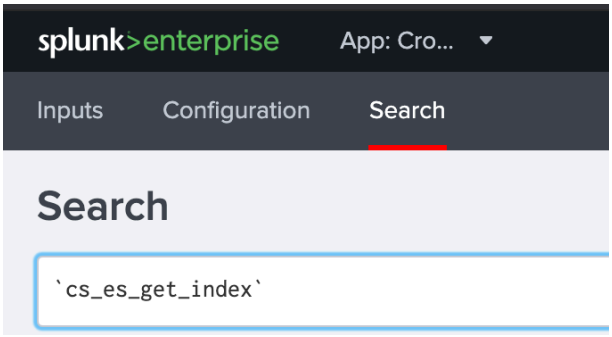
4. To modify a search macro, click on the name of the macro (cs_es_get_index for example):



5. Under 'Definition' enter the index or indexes that contain the Event Stream data to the right of "index=" – separate multiple indexes with the "OR" Boolean.



6. To leverage a search macro, open a search window within Splunk and enter the search macro enclose with backquotes: `cs_get_es_index` (the backquote key is the same key as a tilde on a US keyboard layout and should not be confused with a single quote)

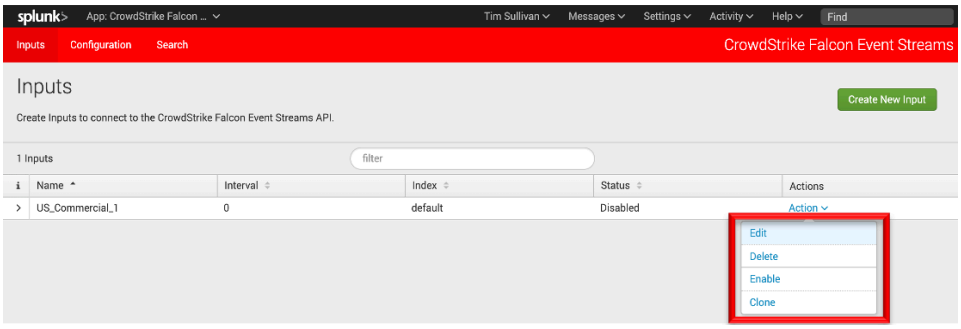


This concludes the Search Macro Configuration process

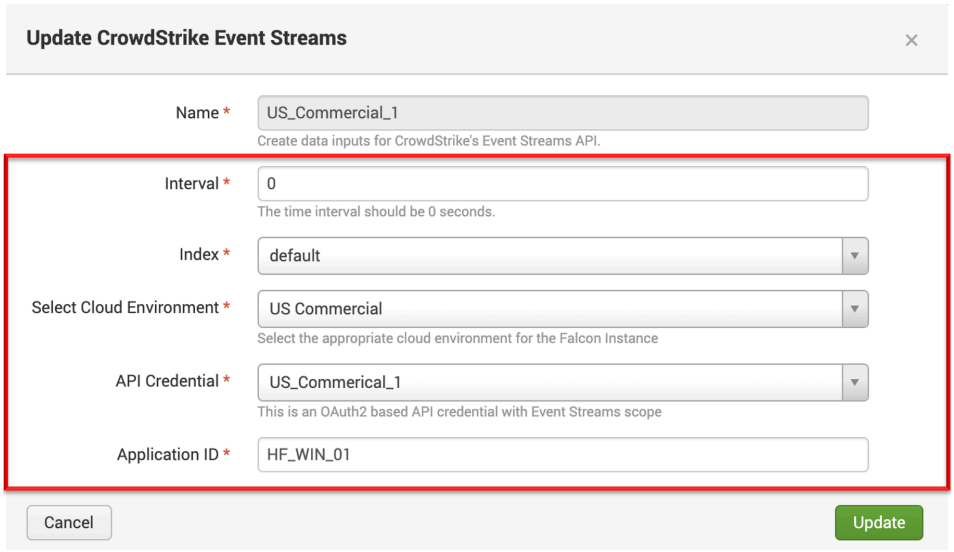
Modify, Remove or Clone Existing Settings

Inputs

1. Under the “Inputs” tab, under the “Action” column for an input, there is a pull-down menu with the options for “Edit”, “Delete”, “Enable” / ”Disable” or “Clone”



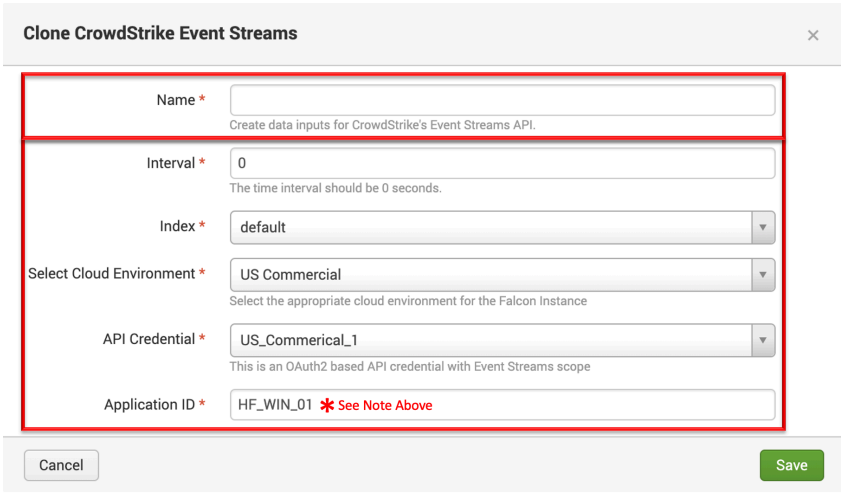
2. **Editing:** allows for changing all the input fields with the exception of the input’s original name



3. **Deleting:** allows for the input to be deleted

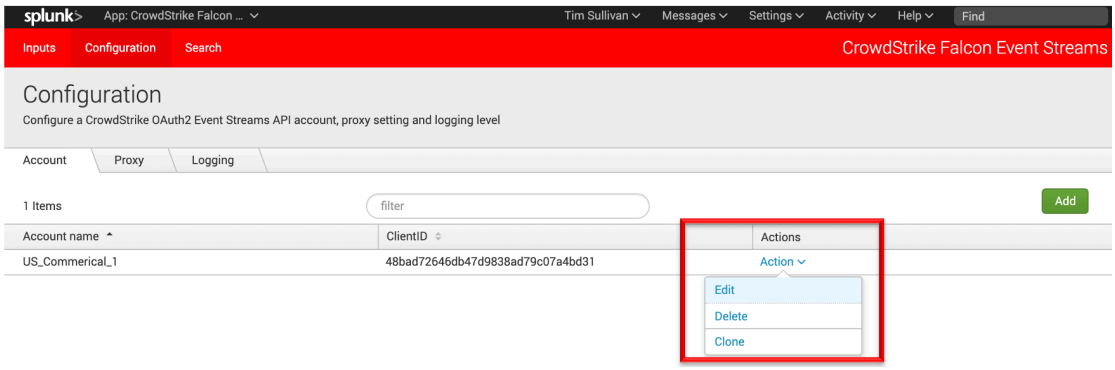
4. **Enabling/Disabling:** allows the input to enabled or disabled depending on the current state – the initial state is enabled

5. **Cloning**: allows all the settings of the input to be replicated with the exception of the "Name" field*
   *Note the Application ID must be unique per ClientID/Secret pair and should be changed if the credentials are being reused

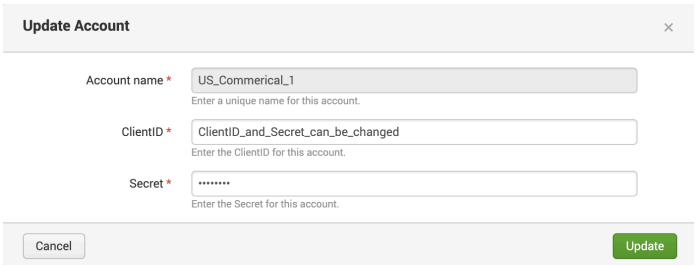


Configuration: Accounts

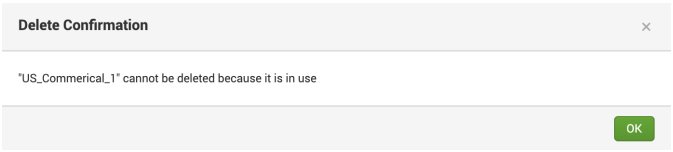
1. Under the “Configuration” sub menu, “Account” tab, and the “Actions” column for an account, there is a pull-down menu with the options for “Edit”, “Delete” or “Clone



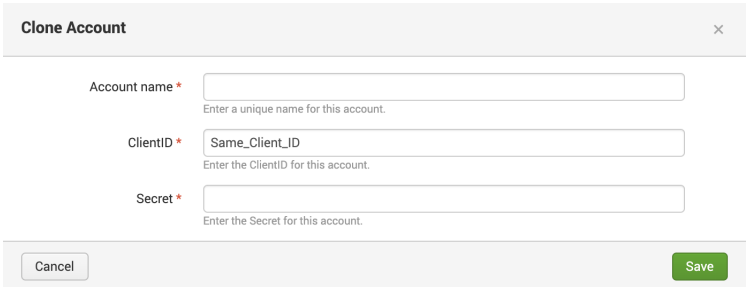
2. **Editing:** allows for the changing of the ClientID and Secret - the name is NOT able to be edited once created



3. **Deleting:** allows a configuration to be deleted however it has to be removed from all inputs before this can be accomplished

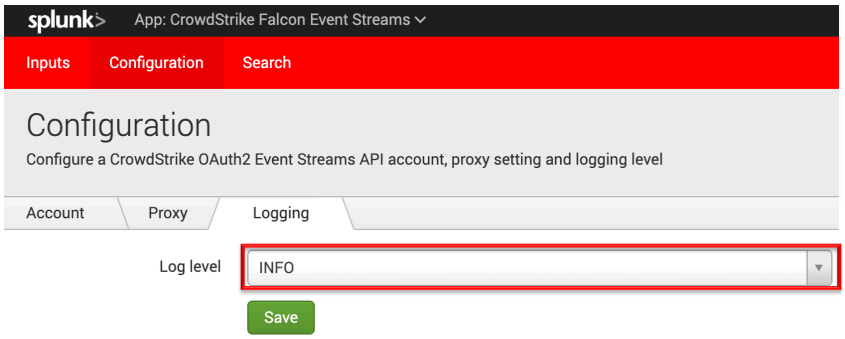


4. **Cloning**: allows for a second account to be created with the same ClientID as the original but requires a new Account Name and Secret to be entered

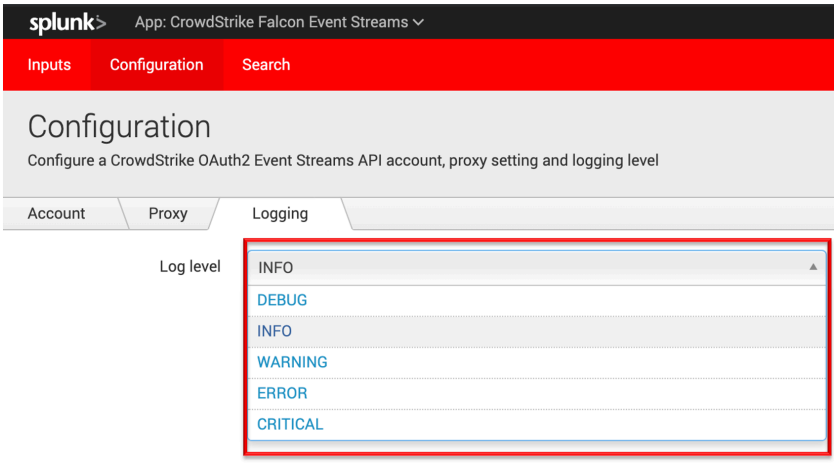


Configuration: Logging

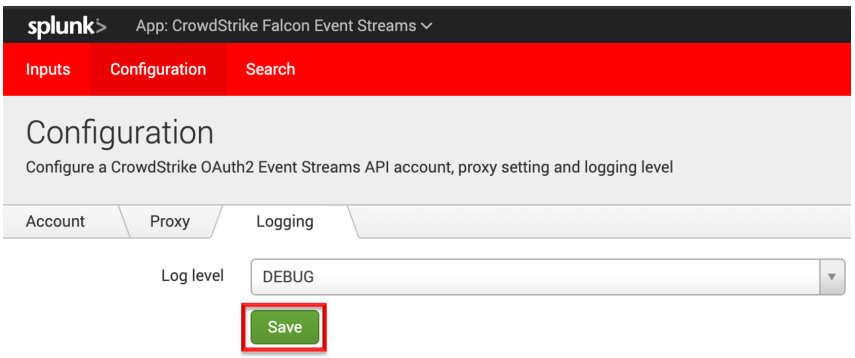
1. Under the “Configuration” sub menu, “Account” tab, and the “Actions” column for an account, there is a pull-down menu for setting the Log Level – which is ‘INFO’ by default



2. The TA provides the typical log levels available for a modular input. Those levels are (from most to least verbose): DEBUG, INFO, WARN, ERROR, FATAL.



3. Select the desired Log Level from the drop down and click “Save”



Custom and Calculated Fields

Custom Fields: ta_data

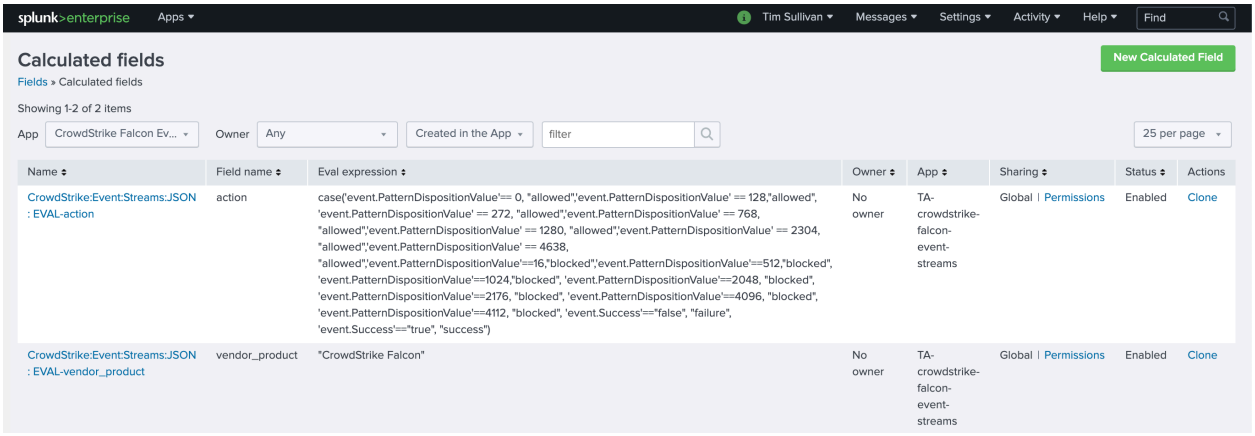
The Event Streams TA creates a custom information section and adds into all events to provide valuable information on the origin of the data and to assist in troubleshooting.

```
ta_data: { [-]
  Cloud_environment: us_commercial
  Feed_id: 0
  Input: bh1_es_linux
  Multiple_feeds: False
  TA_version: _US1_SES_B14
}
```

- **ta_data** - The name of the data section that provides the custom TA data
- **Cloud_environment** – The cloud environment selected for the Input
- **Feed_id** – The id number for the data URL feed
- **Input** – The name of the configured Input that received the data
- **Multiple_feeds** – Indicates if the Event Stream contains multiple data URLs
- **TA_version** – Data pulled from the TA configuration file and indicates the version of the TA

Calculated Fields

There are two calculated fields created in this TA:

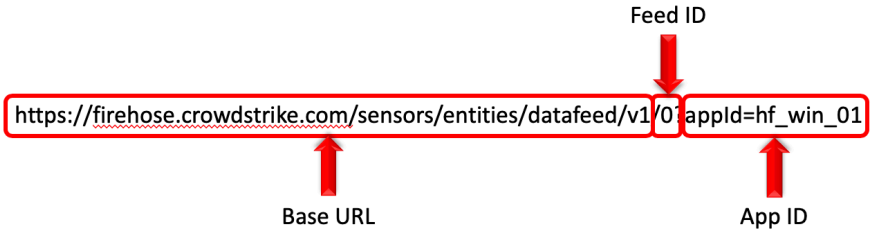


- **Action** – This field is calculated to be able to map to the 'Action' field in both the authentication and malware CIM (Common Information Model) tables
    - Authentication – The 'event.Success' field for authentication events is evaluated to provide the correct value
    - Malware – the numerical value of the 'event.PatternDispositionValue' is evaluated to provide the correct value
- **Vendor_product** – is calculated based on the source type to indicate that it was from CrowdStrike's Falcon platform

Understanding the Event Streams API and Offset Values

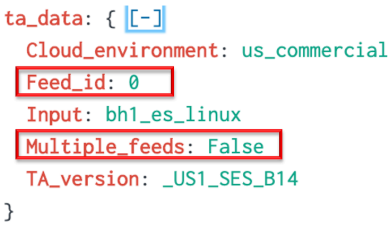
The CrowdStrike Event Streams API provides a substantial amount of data. In some instances, the amount of data is large enough that it is not feasible for a single URL to provide it all and the information is broken up into multiple data URL feeds. This is transparent to the end user for the most part and takes place during the API authentication process. Once the credential is authenticated the API will provide a list of data URL feeds that the client needs to connect to for data collection. All data URL feed connection must be successfully established and maintained to ensure all the appropriate data is being collected.

The data URL feed format is as follows:



- **Base URL** – The cloud environment's base URL for the CrowdStrike Event Stream API gateway
- **Feed ID** – The numerical count of the data feed (count starts a '0')
- **App ID** – The App ID assigned in the TA Input configuration

The TA will examine the API response to determine the number of URL feeds and attempt to create and maintain an independent connect to each one. As events are processed from the URL feed(s) the TA will include the associated Feed_id (single URL feeds will always be '0') and if there were multiple feeds presented in the 'ta_data' section:



Each event within a URL feed contain a unique numerical value called an 'offset' value. This value is used as a unique identifier for event within that URL feed. It is visible in the 'metadata' section of the Splunk event:

In the event that the network connection is disrupted the TA will leverage this information as the marker to determine the last event processed. Since the TA is able to support multiple inputs it uses the name of the Input as the unique identifier and then relates the data feed URL and offset values with it.

Input_name{datafeedURL:offset}

This information is then stored by the TA in two independent locations:

- Splunk KV Store: The first location is within the Splunk KV (key:value) store. This is an internal Splunk location that the TA will call via API to both read and write data.

(for more information please reference: https://docs.splunk.com/Documentation/Splunk/8.0.4/Admin/AboutKVstore)

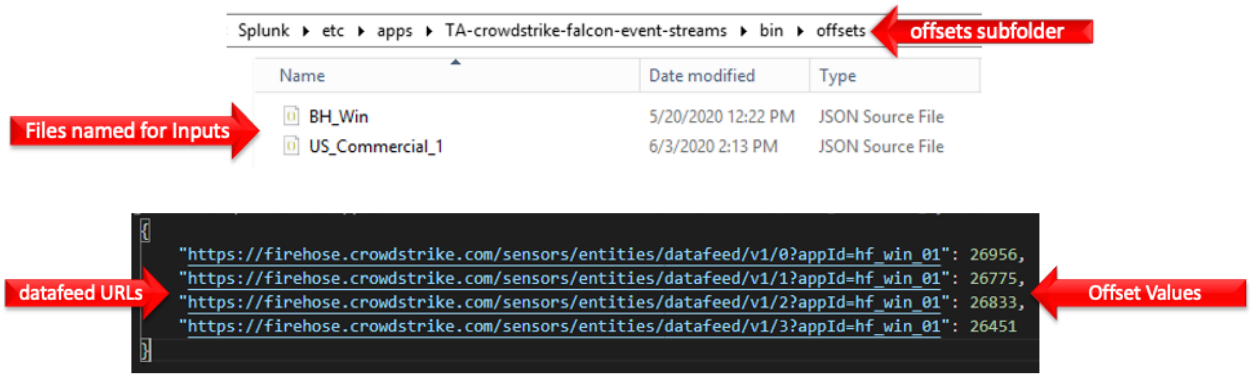
- The 'offsets' folder within the TA itself ($SPLUNK/etc/apps/TA-crowdstrike-falcon-event-streamds/bin/offsets): The second location is within the TA's '/bin/offsets/' folder on the heavy forwarder/IDM. Within this folder are JSON files named after the configured inputs. As events are processed within the TA it will attempt to write the information out to the file after it's written to the KV store.

The data is stored in two location for the following reasons:

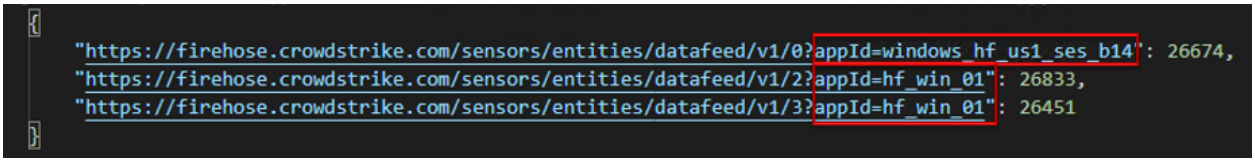
1. **To provide redundancy of data** – Should the data in the KV store or the JSON file become corrupted the data still resides in the other to be used by the TA
2. **To enable back up/transfer of the data** – In the event that there is a catastrophic failure of the Heavy forwarder or the TA is relocated to another Heavy Forwarder that data can be restored/transferred as necessary
3. **To customize data ingestion** – The JSON file can be manipulated so that data collection can begin at specific offsets

The Anatomy of the Offset JSON File

The Offset JSON files are stored in the $SPLUNK/etc/apps/TA-crowdstrike-falcon-event-streams/bin/offsets folder. The file names are the names of their corresponding inputs (**note that these files are not deleted when the input is removed**). Each JSON file will have the datafeed URLs and offset values that have been associated with that Input.



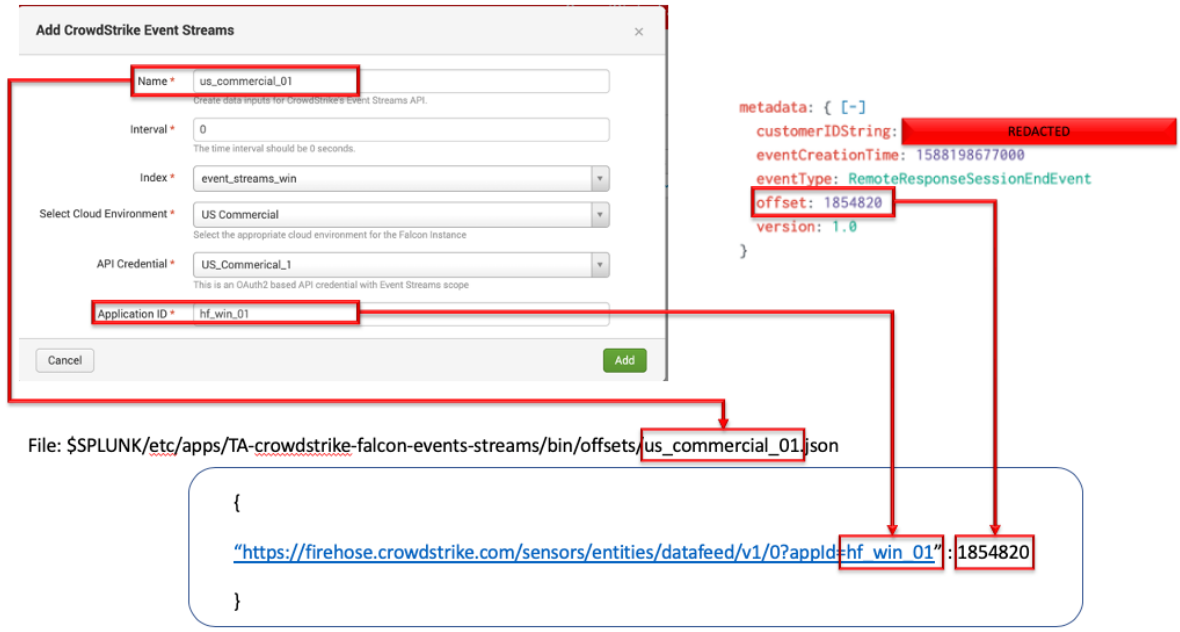
In the event that an existing Input's AppID is modified or if a new Input's name matches an existing file name but the AppID does not match the AppID value in the datafeedURL there may be multiple entries present.



Using Custom Offset Values

The TA will search both the Splunk KV store and the JSON offset files to find the Input name and then look within that data for the data URL feed. If the data URL feed is located, it will retrieve the associated offset value. In the event that both the KV Store and the JSON file return a value, the TA will use the highest value returned. There are ways to manipulate the offset value that the TA will use when making its calls. This is accomplished through modifying the values in the JSON file and/or the Input. Some example use cases are:

1. **Migrate from existing TA/CrowdStrike SIEM connector** - The value assigned to the data URL feed in the JSON file can be added or manipulated manually while the specific Input is disabled. If a customer is migrating from another Event Stream source (such as the CrowdStrike SIEM collector) and knows the last offset value for the Feed ID (if it's a single feed then it will just be '0') they can create the JSON file and add the entry so that when the input is enabled it will start collecting from that point on.



2. **Recovery from failure/ Moving TA to a different Heavy Forwarder/ Migrating to a Splunk IDM** – In the event that the TA is being installed on a new system but there has already been data collection the JSON file can be relocated to the new system. An Input with the same name and App ID can be created and the TA should retrieve the offset data from the previous Input. *Note: Duplicate App IDs should not be used at the same time. It is possible to change the Input and App ID name but the

file name and the App ID in the data URL feed string must match the new information.

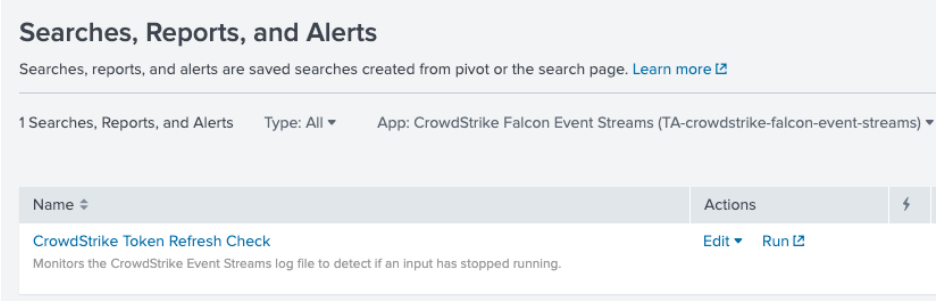
3. **Selective data pull** – In the event that a specific offset value is available for retrieval an only a specific value/values need to be retrieved a new Input can be created with a modified JSON file name, updated AppID and specific offset. The input should be pointed at a dedicated index so as to not create duplicate data in another index.

Token Refresh Check Alert and Restart Input Alert Action

In order to help mitigate issues with API connectivity and data transfers, v2.0.9 introduced a custom search and alert action that can be configured to identify if the connection can become blocked or unstable and data is no longer able to be collected. A few examples of situation that can cause this to occur are:

- Network congestion
- Network devices that block persistent connections (and reconnect attempts) after a prolonged period of time
- Malformed connection communications such as a data connection within the API being closed but the connection to the API gateway remains active
- Internal Splunk errors
- Input accidentally disabled

A properly functioning input should attempt to refresh its OAuth2 token every 20 minutes. The 'CrowdStrike Token Refresh Check' alert is designed to look for OAuth2 issue and refresh logs within the TA to help determine if the input is still processing data correctly. This is accomplished by leveraging the **cs_es_tc_input(1)** search macro to look for OAuth2 issue/refresh events within a 60 minute time window and ensure that there are at least 2 events. The search macro takes an input name by default so an alert is considered specific to that input. In an environment with multiple inputs, it's recommended to configure alerts for all active inputs. If there are not at least 2 events the alert should fire and take the alert actions that have been properly configured.

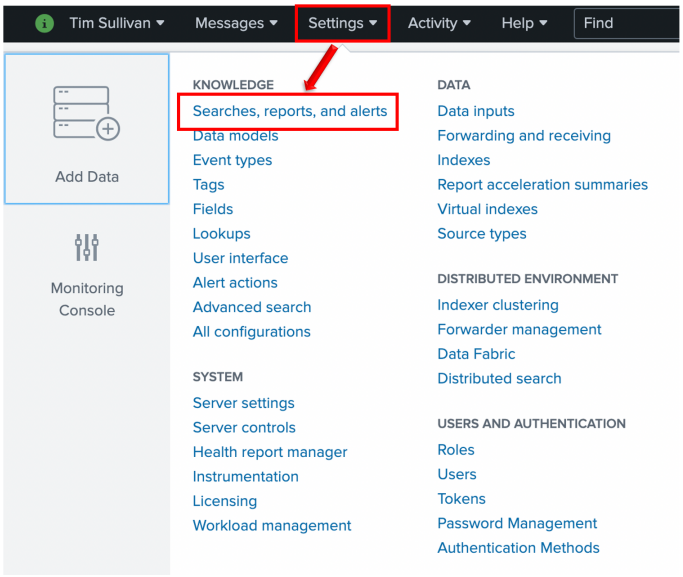


- **CrowdStrike Token Refresh Check**: This alert is designed to detect if there have been more than 2 token refresh/issue logs within the past 60 minutes (default settings)

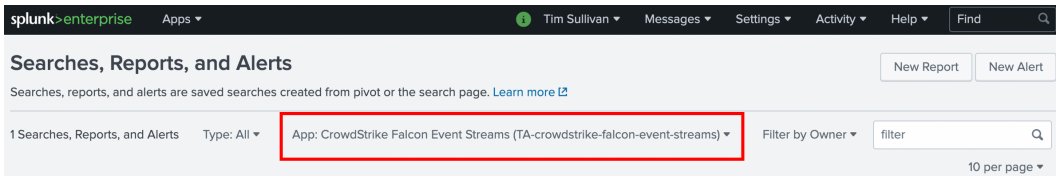
The combination of this alert and the alert action is designed to cycle an input through the 'disable' and 'enable' actions via the Splunk REST API. In order for this to be successful the account that is performing this action should have the proper level of access to those REST endpoints. Typically, this is an account with 'admin' or 'system' level access.

Configuring the custom alert to restart an input

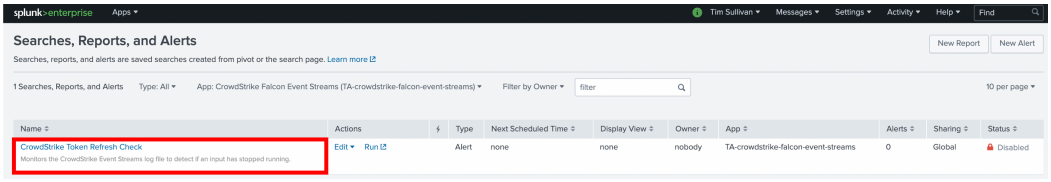
1. Under ‘Settings’ select ‘Searches, reports, and alerts’



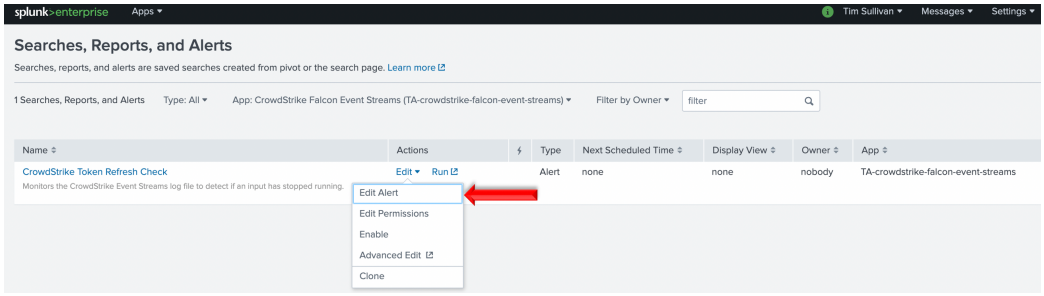
2. Ensure that the CrowdStrike Falcon Event Streams TA is selected as the ‘App’:



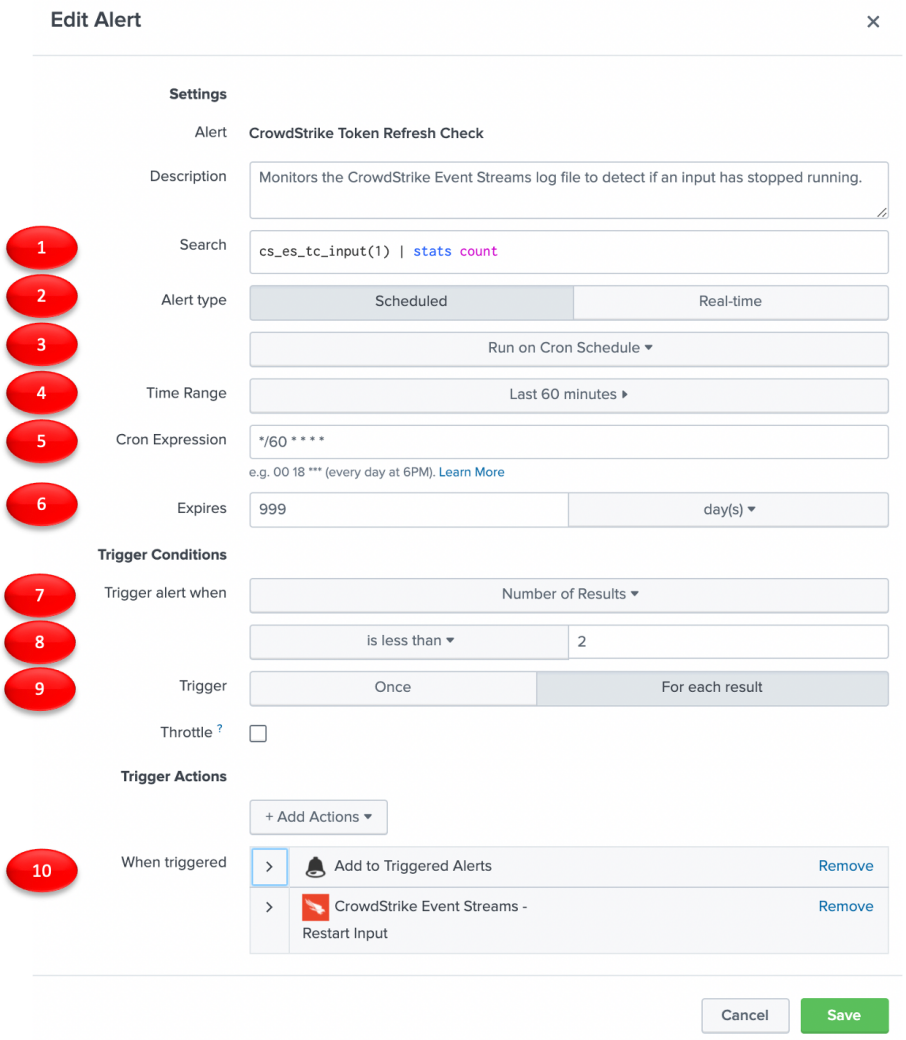
3. Locate the ‘CrowdStrike Token Refresh Check’ alert:



4. Under ‘Actions’ select ‘Edit’:



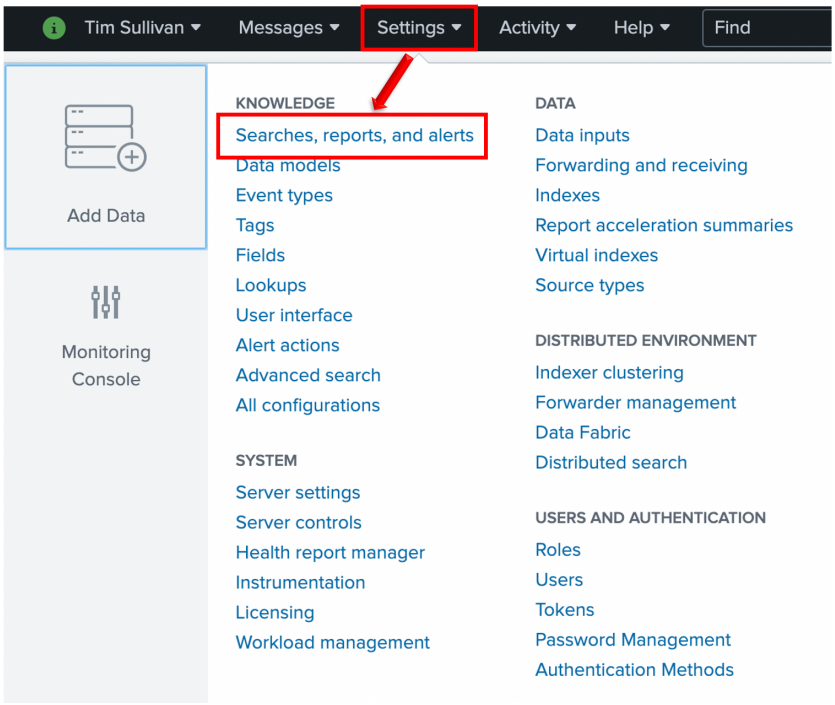
5. Configuring the Alert and associated actions:



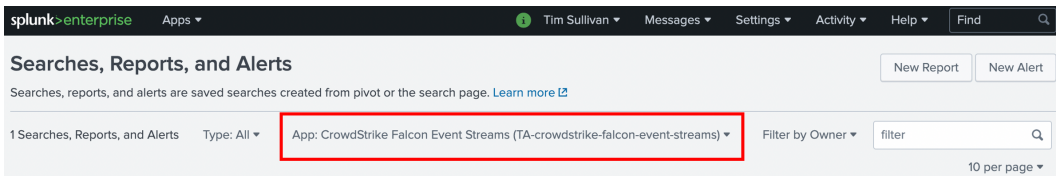
1. **Search:** This is the search that will be used to look for the log information
2. **Alert Type**: This alert is designed to be a scheduled alert
3. **Schedule configuration** – the default configuration is a Cron Schedule
4. **Time Range**: The default time range is 60 minutes which should have at least 2 logs
5. **Cron Expression**: The default Cron expression sets the search to run every 60 minutes
6. **Expires**: The alert is set to expire in 999 days
7. **Trigger alert when**: The default is set to the number of results for the search
8. **Number of results evaluation** – The default is set to less than 2 results in 60 minutes
9. **Trigger**: The default configuration is for each result
10. **When Triggered**: There are two default actions:
    a. **'Add to Triggered Alerts'**
    b. **'CrowdStrike Event Streams Restart Input'**

Enabling the custom alert to restart an input

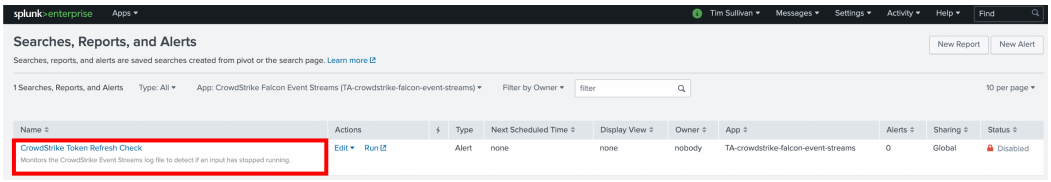
1. Under ‘Settings’ select ‘Searches, reports, and alerts’



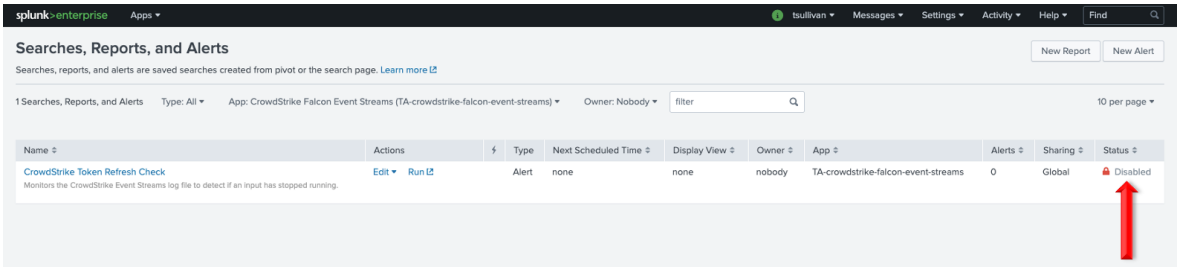
2. Ensure that the CrowdStrike Falcon Event Streams TA is selected as the ‘App’:



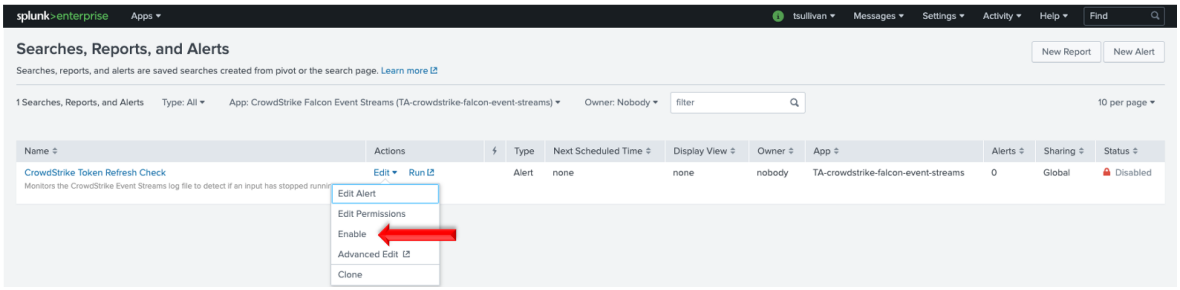
3. Locate the ‘CrowdStrike Token Refresh Check’ alert:



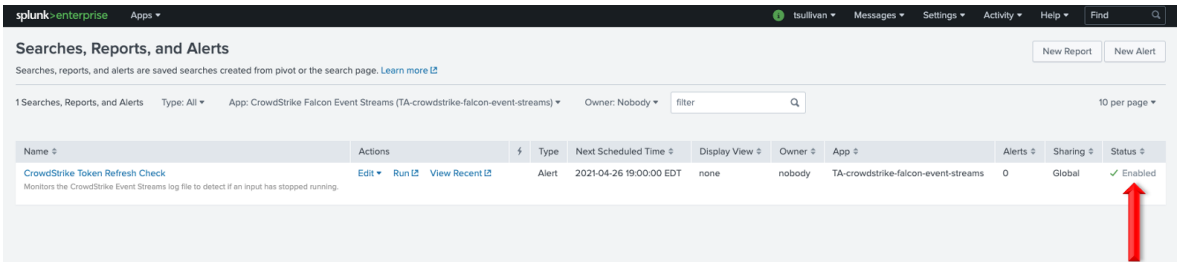
4. Verify that the alert is not currently enabled:



5. Under ‘Actions’ select ‘Enable’:



6. Ensure that the alert status has changed to ‘Enabled’:



Troubleshooting and Support

CrowdStrike provides support for the TA's code, the functionality of that code and authentication to the API endpoint(s). The following topics fall outside of that scope:

1. Network connectivity issues unrelated to authentication response from the CrowdStrike API endpoint
2. Tagging and CIM mapping (these are considered feature requests and will be evaluated by the integrations team)

Checking Configuration

Unable to establish connection:

1. Ensure that the Event Stream API has been enabled for the CID
2. Ensure that the proper Cloud Environment has been selected
3. Ensure that the OAuth2 credential has been scoped correctly
4. Ensure that the OAuth2 credential has been entered correctly
5. Ensure that network devices aren't blocking or tearing down the connection

No data is present:

1. Ensure that the Input is enabled
2. Ensure that the Index has been created on the Indexer(s)
3. If leveraging the Search Macro ensure that it's been configured correctly
4. Ensure that events have taken place since the connection was established

Getting Support

Prior to contacting CrowdStrike support please review the following:

Initial Deployment

1. Ensure that the Event Stream API has been enabled by CrowdStrike support
2. Ensure that the OAuth2 credential information have been entered correctly
3. Ensure that the OAuth2 credential has been scoped correctly
4. Set the TA log level to 'DEBUG'
5. Repeat and record the action(s) that are associated with the issue you are reporting
6. Download the all log files containing 'ta_crowdstrike_falcon_event_streams' under the $Splunk/var/log/splunk/ directory
7. Record the following information about the Splunk system:
    - Splunk environment type
    - Splunk version
    - TA version
8. Identify the types of networks devices that the connection will traverse and ensure that they have been properly configured
9. Collect API audit logs from the Falcon instance for the time frame when the issue is occurring
10. Navigate to https://supportportal.crowdstrike.com/
11. Provide (at a minimum) the information from steps 5-9

Existing Deployment

1. Set the TA log level to 'DEBUG'
2. Disable and re-enable TA Inputs
3. Download the all log files containing 'ta_crowdstrike_falcon_event_streams' under the $Splunk/var/log/splunk/ directory
4. Record the following information about the Splunk system:
    - When was the last successful connection
    - If a TA or Splunk update performed around the same time frame
    - Splunk environment type
    - Splunk version
    - TA version
5. Identify the types of networks devices that the connection will traverse and ensure that they are still properly configured
6. Collect API audit logs from the Falcon instance for the time frame when the issue began occurring
7. Navigate to https://supportportal.crowdstrike.com/
8. Provide (at a minimum) the information from steps 3-6