SIEM Connector Feature Guide

Last updated: November 10, 2020

Contents:

- Overview
  - System Requirements
- Understanding the SIEM connector
- Before You Begin
- Installing the SIEM Connector for a Single CID
  - 1. Download the SIEM Connector Installer
  - 2. Install the SIEM Connector
  - 3. Select an Output Type
  - 4. Add your API Credentials to the Configuration File
  - 5. Configure the SIEM Connector for Your Environment
  - 6. Start the SIEM Connector
- Installing the SIEM Connector for Multiple CIDs
  - 1. Download the SIEM Connector Installer
  - 2. Install the SIEM Connector
  - 3. Select an Output Type for Each CID
  - 4. Add your API Credentials to Each Configuration File
  - 5. Modify the Initialization Script to Use Multiple CIDs
  - 6. Configure the SIEM Connector for Your Environment
  - 7. Start the SIEM Connector
- Configuring the SIEM Connector for Your Environment
  - [Settings] Section
  - [Logging] Section
  - [Syslog] Section
  - [EventTypeCollection] Section
  - [EventSubTypeCollection] Section
  - Syslog Mappings
  - Splunk Configuration for the Falcon SIEM Connector Log
- Starting, Stopping, and Restarting the SIEM Connector
  - Start the SIEM Connector
  - Stop the SIEM Connector
  - Restart the SIEM Connector
- Updating the SIEM Connector
  - Updating to version 2.0
  - Updating to version 1.3.1 or later
- Uninstalling the SIEM Connector
- Troubleshooting
  - Configuration Errors
  - Connector/Client Errors
  - Transformation/Syslog Errors
- SIEM reference info
  - Default Directories
  - IPs for legacy streaming API

- Default Config Files

Overview

The Falcon SIEM Connector Feature Guide explains how to install and configure the Falcon SIEM Connector, a tool for gathering info from Falcon's event streams and sending them to your SIEM. The Falcon SIEM Connector:

- Transforms Falcon Streaming API data into a format that any log analysis tool can consume
- Maintains the connection to the Falcon Streaming API and your SIEM, in case either drops
- Manages the data-stream pointer to prevent data loss

Download the latest SIEM connector installer in Falcon at **Support** > **Tool Downloads**.

System Requirements

SUPPORTED OPERATING SYSTEMS

The latest version of the SIEM connector supports 64-bit versions of these OSes:

- CentOS/RHEL 6.x-7.x (64-bit)
- Ubuntu 14.x (64-bit)
- Ubuntu 16.04 (64-bit)
- Ubuntu 18.04 (64-bit)

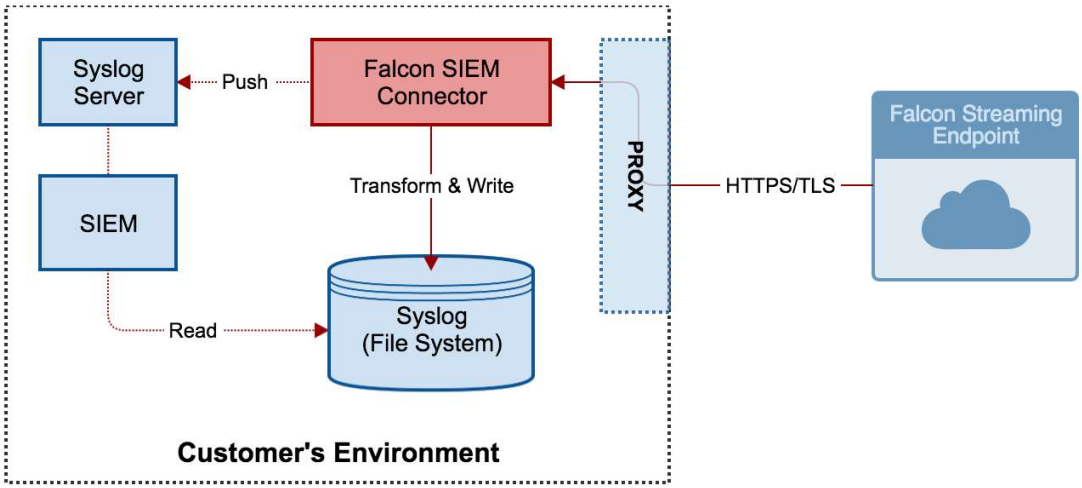
> For customers running version 1.3 of the SIEM connector: CentOS/RHEL 6.x-7.x (64-bit) and Ubuntu 14.x (64-bit) are the only supported operating systems.

OTHER REQUIREMENTS

- **Connectivity**: Internet connectivity and ability to connect the CrowdStrike Cloud
  - You can use the SIEM Connector through a proxy, but you must separately authenticate to the proxy. The SIEM Connector does not handle proxy authentication.
- **Communication**: Ability to communicate with Syslog Listener
- **Authorization**: Access to the Falcon Streaming API (contact support)
- **Time**: The date and time on the host running the Falcon SIEM Connector must be current

There are no specific RAM or processor requirements for running the Falcon SIEM Connector. The required disk space will vary depending on how long you wish to store output logs.

Understanding the SIEM connector



The Falcon SIEM Connector can be placed behind a proxy within your environment and will connect to the Falcon Streaming endpoint to authorize and discover available feeds. This information will then be used to stream the data (event feed) into your environment via UDP/TCP. The Connector will then:

- Transform and spool events into syslog. The syslog will be consumed by a log analysis tool.
- Transform and write events to syslog listener with a tool-agnostic payload.

Other features of the SIEM connector:

- Handles the mechanics to auto connect and reconnect to the Falcon Streaming API
  - Open long-lived connection to the Falcon Streaming API to receive streamed events
  - Save offset value of latest event received
  - When disconnected, automatically reconnect to the Falcon Streaming API and pass last offset value to continue receive events from that point
- Supports our OAuth2-based authentication (in SIEM connector version 2.0 and later)
- Built-in data transformation
  - By default, events are in nested JSON format
  - Options to transform data into Syslog, CEF, or LEEF format
- Flexible data integration options
  - Save nested JSON file to disk
  - Save Syslog/CEF/LEEF file to disk
  - Send Syslog/CEF/LEEF to Syslog listener

Before You Begin

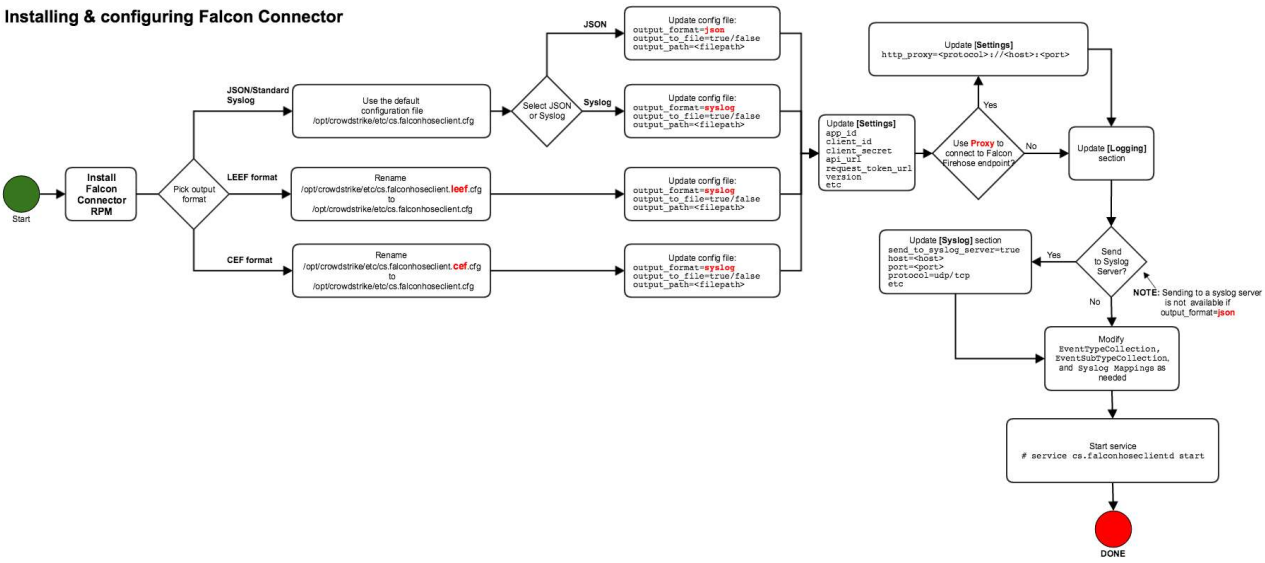
- You must contact support to enable the streaming APIs in your environment before using the SIEM connector.
- If your network requires it, allow ("whitelist") traffic between your network and the FQDNs for our APIs in the applicable cloud:

Cloud	URL
US-1	`api.crowdstrike.com` `firehose.crowdstrike.com`
US-2	`api.us-2.crowdstrike.com` `firehose.us-2.crowdstrike.com`
EU-1	`api.eu-1.crowdstrike.com` `firehose.eu-1.crowdstrike.com`
US-GOV-1	`api.lagger.gcw.crowdstrike.com` `firehose.lagger.gcw.crowdstrike.com`

Installing the SIEM Connector for a Single CID

In most cases, you should follow these steps to install the SIEM Connector. Most customers have a single customer ID (CID).

> If you have multiple CIDs, follow the steps in Installing the SIEM Connector for Multiple CIDs.



For info on specifying the **Settings** values, see the [Settings] Section.

Note: Administrative (root) permissions are required to install and configure the SIEM Connector. Administrative permissions are not required to run the SIEM Connector.

1. Download the SIEM Connector Installer

1. Go to **Support > Tool Downloads**.
2. Download the SIEM Connector installer for your operating system:
   - CentOS: Download the latest `.rpm` installer
   - Ubuntu: Download the latest `.deb` installer

2. Install the SIEM Connector

1. Open a terminal.
2. Run the installation command, replacing `<installer package>` with the installer you downloaded:
   - CentOS:

     ```
     sudo rpm -Uvh <installer package>
     ```

   - Ubuntu:

     ```
     sudo dpkg -i <installer package>
     ```

SAMPLE CENTOS OUTPUT:

```
# sudo rpm -Uvh /path/to/file/cs.falconhoseclient-1.0.70-1.el7.centos.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:cs.falconhoseclient-1.0.70-1.el7.c################################# [ 50%]
Cleaning up / removing...
   2:cs.falconhoseclient-1.0.69-1.el7.c################################# [100%]
```

SAMPLE UBUNTU OUTPUT:

```
# sudo dpkg -i crowdstrike-cs-falconhoseclient_70-siem-release-1.0_amd64.deb
(Reading database ... 63084 files and directories currently installed.)
Preparing to unpack crowdstrike-cs-falconhoseclient_70-siem-release-1.0_amd64.deb ...
Unpacking crowdstrike-cs-falconhoseclient (70-siem-release-1.0) over (69-siem-release-1.0) ...
Setting up crowdstrike-cs-falconhoseclient (70-siem-release-1.0) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on the next reboot.
```

3. Select an Output Type

Your output type is defined by which of the sample configuration files you use. The sample configuration files are installed to `/opt/crowdstrike/etc/`. You can choose from these output formats:

- JSON (default)
- Syslog
- Common Event Format (CEF)
- Log Event Extended Format (LEEF)

JSON (DEFAULT)

JSON output is the default behavior. No changes are required.

The default output location is `/var/log/crowdstrike/falconhoseclient/output`

SYSLOG

1. On your device, edit the file `/opt/crowdstrike/etc/cs.falconhoseclient.cfg` in a text editor.
2. Change the value of `output_format` to read:

   ```
   output_format: syslog
   ```

COMMON EVENT FORMAT (CEF)

1. On your device, open the folder `/opt/crowdstrike/etc/`.
2. Rename the sample CEF config file to `/opt/crowdstrike/etc/cs.falconhoseclient.cfg`:

```
sudo mv /opt/crowdstrike/etc/cs.falconhoseclient.cef.cfg /opt/crowdstrike/etc/cs.falconhoseclient.cfg
```

LOG EVENT EXTENDED FORMAT (LEEF)

1. On your device, open the folder `/opt/crowdstrike/etc/`.
2. Rename the sample LEEF config file to `/opt/crowdstrike/etc/cs.falconhoseclient.cfg`.

```
sudo mv /opt/crowdstrike/etc/cs.falconhoseclient.leef.cfg /opt/crowdstrike/etc/cs.falconhoseclient.cfg
```

4. Add your API Credentials to the Configuration File

1. In the Falcon console, go to **Support > API Clients & Keys**.
2. Create an API client to use with the SIEM connector, and record its API client ID and API client secret.
3. Open `/opt/crowdstrike/etc/cs.falconhoseclient.cfg` (created in step 3) in a text editor.
4. Find the `[Settings]` section.
5. Edit these lines for your environment:
   - `app_id`: A unique app ID used to label your application. Max: 18 characters.
   - `client_id`: Your API client's ID
   - `client_secret`: Your API client's secret
6. Save your changes.

5. Configure the SIEM Connector for Your Environment

The rest of the configuration file defines how the SIEM connector formats data from the Streaming API into an appropriate format for your SIEM.Edit your `/opt/crowdstrike/etc/cs.falconhoseclient.cfg` file to include the data you want in the format that your SIEM requires. Review the instructions in Configuring the SIEM Connector for Your Environment for more details.

6. Start the SIEM Connector

After you've edited your `.cfg` file to include the data you want to provide to your SIEM, you're ready to start the SIEM Connector.

Run this command at a terminal:

- CentOS:

```
sudo service cs.falconhoseclientd start
```

- Ubuntu 14.x:

```
sudo start cs.falconhoseclientd
```

- Ubuntu 16.04 and later:

```
sudo systemctl start cs.falconhoseclientd.service
```

Installing the SIEM Connector for Multiple CIDs

If you manage multiple customer IDs (CIDs), you should follow these steps to install the SIEM Connector.

Most customers have a single customer ID. If you have a single CID, follow the steps in Installing the SIEM Connector for a Single CID.

Administrative (root) permissions are required to install and configure the SIEM Connector. Administrative permissions are not required to run the SIEM Connector.

1. Download the SIEM Connector Installer

1. Go to **Support > Tool Downloads**.
2. Download the SIEM Connector installer for your operating system:
   - CentOS: Download the latest `.rpm` installer
   - Ubuntu: Download the latest `.deb` installer

2. Install the SIEM Connector

1. Open a terminal.
2. Run the installation command, replacing `<installer package>` with the installer you downloaded:
   - CentOS:

     ```
     sudo rpm -Uvh <installer package>
     ```

   - Ubuntu:

     ```
     sudo dpkg -i <installer package>
     ```

Sample CentOS output:

```
# sudo rpm -Uvh /path/to/file/cs.falconhoseclient-1.0.70-1.el7.centos.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:cs.falconhoseclient-1.0.70-1.el7.c################################# [ 50%]
Cleaning up / removing...
   2:cs.falconhoseclient-1.0.69-1.el7.c################################# [100%]
```

Sample Ubuntu output:

```
# sudo dpkg -i crowdstrike-cs-falconhoseclient_70-siem-release-1.0_amd64.deb
(Reading database ... 63084 files and directories currently installed.)
Preparing to unpack crowdstrike-cs-falconhoseclient_70-siem-release-1.0_amd64.deb ...
Unpacking crowdstrike-cs-falconhoseclient (70-siem-release-1.0) over (69-siem-release-1.0) ...
Setting up crowdstrike-cs-falconhoseclient (70-siem-release-1.0) ...
Processing triggers for ureadahead (0.100.0-16) ...
ureadahead will be reprofiled on the next reboot.
```

3. Select an Output Type for Each CID

Each customer ID (CID) that you manage has its own output type and uses its own configuration file. Your output type is defined by which of the sample configuration files you use. You can choose from these output formats:

- JSON (default)
- Syslog
- Common Event Format (CEF)
- Log Event Extended Format (LEEF)

For each CID you want to manage:

1. Locate the sample config file for the output type you want.
   - JSON: `/opt/crowdstrike/etc/cs.falconhoseclient.cfg`
   - Syslog: `/opt/crowdstrike/etc/cs.falconhoseclient.cfg`
   - CEF: `/opt/crowdstrike/etc/cs.falconhoseclient.cef.cfg`
   - LEEF: `/opt/crowdstrike/etc/cs.falconhoseclient.leef.cfg`
2. Copy that sample config file.
3. Rename your copied file to use a unique label you choose.
   - The label is used to help you recognize which CID it represents. The log files for this CID also use the same label.
   - The file extension must remain `.cfg`, such as `example_customer.cfg`
4. Move your copied file to this directory: `/opt/crowdstrike/config/`

Repeat these steps for each CID that you want to add to the SIEM connector. Each CID's config file name must be unique.

4. Add your API Credentials to Each Configuration File

1. In the Falcon console, go to **Support > API Clients & Keys**.
2. Create an API client to use with the SIEM connector, and record its API client ID and API client secret.
3. For each of your `.cfg` files in `/opt/crowdstrike/config/`:
   1. Open `/opt/crowdstrike/etc/cs.falconhoseclient.cfg` in a text editor.
   2. Find the `[Settings]` section.
   3. Edit these lines for your environment:
      1. `app_id`: A unique app ID used to label your application. Max: 18 characters.
      2. `client_id`: Your API client's ID
      3. `client_secret`: Your API client's secret
   4. Save your changes.

5. Modify the Initialization Script to Use Multiple CIDs

To use multiple CIDs, you must modify the initialization script. The default initialization script uses a single CID.

UBUNTU

1. Open the configuration file in a text editor: `/etc/init/cs.falconhoseclientd.conf`
2. Locate the line beginning with `-config=`
3. Change the `.cfg` file name to use `cs.falconhoseclient_daemon.cfg`:
   - Before: `-config=/opt/crowdstrike/etc/cs.falconhoseclient.cfg`
   - After: `-config=/opt/crowdstrike/etc/cs.falconhoseclient_daemon.cfg`

CENTOS

1. Open the configuration file in a text editor: `/etc/init.d/cs.falconhoseclientd`
2. Locate the line beginning with `CONFIG=$CS_DIR`
3. Change the `.cfg` file name to use `cs.falconhoseclient_daemon.cfg`:
   - Before: `CONFIG=$CS_DIR/etc/cs.falconhoseclient.cfg`
   - After: `CONFIG=$CS_DIR/etc/cs.falconhoseclient_daemon.cfg`

6. Configure the SIEM Connector for Your Environment

The rest of the configuration file defines how the SIEM connector formats data from the Streaming API into an appropriate format for your SIEM.

For each CID, edit its `.cfg` file to include the data you want in the format that your SIEM requires. Review the instructions in Configuring the SIEM Connector for Your Environment for more details.

7. Start the SIEM Connector

After you've edited your `.cfg` files to include the data you want to provide to your SIEM, you're ready to start the SIEM Connector.

Run this command at a terminal:

- CentOS:

```
sudo service cs.falconhoseclientd start
```

- Ubuntu 14.x:

```
sudo start cs.falconhoseclientd
```

- Ubuntu 16.04 and later:

```
sudo systemctl start cs.falconhoseclientd.service
```

Configuring the SIEM Connector for Your Environment

1. If the SIEM Connector is running, stop the SIEM connector by running this command at a terminal:
   - CentOS:

     ```
     sudo service cs.falconhoseclientd stop
     ```

   - Ubuntu 14.x:

     ```
     sudo stop cs.falconhoseclientd
     ```

   - Ubuntu 16.04 or 18.04:

     ```
     sudo systemctl stop cs.falconhoseclientd.service
     ```

2. Open `/opt/crowdstrike/etc/cs.falconhoseclient.cfg` in a text editor.
3. Edit the lines in the configuration file for your environment.
   - The configuration file can interpret the options in any order.
   - Refer to the tables below for a full inventory of available configuration options.
4. Save your changes.
5. Start the SIEM Connector by running this command at a terminal:
   - CentOS:

     ```
     sudo service cs.falconhoseclientd start
     ```

   - Ubuntu 14.x:

     ```
     sudo start cs.falconhoseclientd
     ```

   - Ubuntu 16.04 or 18.04:

     ```
     sudo systemctl start cs.falconhoseclientd.service
     ```

[Settings] Section

This section contains all the client runtime specific configurations.

Key	Value	Description	Required?	Default
`version`	• `1`: *Deprecated* • `2`: *Deprecated* • `3`: OAuth2-based authentication — This is the only supported authentication method for the CrowdStrike streaming API	Authentication method	Y	`3`

`api_url`	`<protocol>://<host>:<port>/<paths>`	Defines the location of the Streaming API that the SIEM connector relies on.	Y	Specify a value based on your cloud: • `https://api.crowdstrike.com/sensors/entitie` • `https://api.us-2.crowdstrike.com/sensors/er` 2) • `https://api.eu-1.crowdstrike.com/sensors/er` 1) • `https://api.laggar.gcw.crowdstrike.com/sens` (US-GOV-1)
`app_id`	string	Identifies your API connection for troubleshooting. Only alphanumeric characters are valid (`a-z`, `A-Z`, `0-9`). Max: 32 characters.	Y	`FalconHoseClient`
`client_id`	string	API client ID to be used for client verification.	Y	n/a
`client_secret`	string	API client secret to be used for client verification.	Y	n/a
`request_token_url`	string	API endpoint that retrieves an auth token for your API connection	Y	Specify a value based on your cloud: • `https://api.crowdstrike.com/oauth2/token` (U • `https://api.us-2.crowdstrike.com/oauth2/tok` • `https://api.eu-1.crowdstrike.com/oauth2/tok` • `https://api.laggar.gcw.crowdstrike.com/oaut`
`connection_timeout`	number > 0 (seconds)	Amount of time (in seconds) client should wait for a connection to complete before considering a retry.	N	5 seconds
`read_timeout`	number > 0 (seconds)	Amount of time (in seconds) client should wait for server's response headers after fully writing the request.	N	8 seconds
`partition`	`all` or partition number (0-n)	Partition to be consumed by the running instance of the client application.	N	all
`http_proxy`	`<protocol>://<host>:<port>`	HTTP proxy to be used to connect to Falcon Streaming API endpoint. This can also be set in the environment variable `$HTTP_PROXY`.	N	n/a

output_format	syslog or json	syslog: will output syslog format with flat key=value pairs and uses the mapping configuration. Use syslog format if CEF/LEEF output is required. json: will output raw nested json format received from the Falcon Streaming API.	N	json
output_to_file	true or false	Enable/disable event output to file.	N	true
output_path	string	Location of file where event output should be written to.	N	/var/log/crowdstrike/falconhoseclient/output

[Logging] Section

Key	Value	Description	Required	Default
verbose_log	true or false	Enable/disable verbose logging.	N	true
max_size	number > 0 (MB)	Maximum individual log file size in Megabytes before rotation.	N	100MB
max_backups	number > 0 (count)	Number of backups to keep before purging.	N	10
max_age	number > 0 (days)	Maximum age (in days) of backup files before it is deleted.	N	30 days

[Syslog] Section

Key	Value	Description	Required	Default
send_to_syslog_server	true or false	Enable/disable push to syslog server. If you don't have a syslog server running, set this to false. Otherwise, the SIEM connector may fail to start.	Y	n/a
host	string (internet address)	Syslog/SIEM host address. It can be IP or host name.	If send_to_syslog_server is true	n/a
port	0-65535	Network port.	If send_to_syslog_server is true	n/a
protocol	udp or tcp	udp: User Datagram Protocol, connectionless transmission model \n . tcp: Transmission Control Protocol, guarantees delivery of data and sequence.	If send_to_syslog_server is true	n/a
tag	string	Syslog tag	N	n/a

`header_delim`	string	Header will be delimited by this value	N	n/a
`header_prefix`	string	Prefix that will be appended to syslog line	N	n/a
`key_val_delim`	string	Delimiter to be used between key and value i.e. if equal sign '=' is used then result will be key=value if a colon ':' is used then result will be key:value.	N	n/a
`field_delim`	string	Delimiter to be used to separate key and value pairs i.e. if pipe (¦) is used as `field_delim` and '=' is used as `key_val_delim`, then the result will be `key1=value1 PIPE key2=value2 PIPE key3=value3`	N	< space >
`val_enclosure`	string	String to be used to enclose the value of the key-value pairs i.e. if a single quote is used then the result will be key1='value1'	N	n/a
`time_fields`	comma-separated strings	Comma-separated strings of fields treated as times, to which the `time_format` configuration will be applied.	N	n/a
`time_format`	See Time Format Values table below.	See Time Format Values table below.	N	yyyy-MM-dd HH:mm:ss
`event_type_field`	string	Fields from nested JSON with dot notation e.g. metadata.eventType. This will be used for filtering and mapping. It is not recommended to change this field unless there is instruction to do so from CrowdStrike.	N	n/a
`event_subtype_field`	comma-separated strings	Fields from nested JSON with dot notation e.g. event.subType. This will be used for filtering and mapping. Please do not edit or adjust unless otherwise advised to do so by CrowdStrike.	N	n/a
`max_length`	number (bytes)	Maximum length of syslog line before being truncated. Truncation will happen atomically by field. Key-value pairs will not be appended unless the entire string will fit within the next line. No partial truncation will happen. The field will either show or not show.	N	n/a

TIME FORMAT VALUES

The following table shows values for the `time_format` syslog key above.

Key	Conversion
HH	2-digit hours
hh	2-digit hours
H	single digit hours
h	single digit hours
mm	minutes
ss	seconds
MMMM	Full month names e.g. January, February
MMM	3 characters month names e.g. Jan, Feb
MM	2 digit month i.e. 01, 02
M	1 digit month (when applicable) e.g. 1, 2, 12
pm	AM/PM
PM	AM/PM
ZZZZ	GMT Time offset e.g. -07:00
ZZZ	Timezone e.g. MST, PST
ZZ	Z notation of time offset e.g. Z07:00
yyyy	4-digit year
YYYY	4-digit year
YY	2-digit year
yy	2-digit year
DDDD	Full day name e.g. Monday, Tuesday
dddd	Full day name e.g. Monday, Tuesday
DDD	3-character day abbreviation e.g. Mon, Tue
ddd	3-character day abbreviation e.g. Mon, Tue
DD	2-digit day e.g. 02
dd	2-digit day e.g. 02
D	1-digit day e.g. 1
d	1-digit day e.g. 1

[EventTypeCollection] Section

This section specifies what event type to collect based on the event_type_field configuration in the [Syslog] section.

Key	Value	Description	Required	Default

One of these event types (specified by `event_type_field`): • `IncidentSummaryEvent` • `DetectionSummaryEvent` • `AuthActivityAuditEvent` • `UserActivityAuditEvent` • `RemoteResponseSessionStartEvent` • `RemoteResponseSessionEndEvent`	`true` or `false`	If `true`, that event type is collected. If `false`, that event type is ignored.	Y	n/a

[EventSubTypeCollection] Section

This section specifies what event type to collect based on the event_subtype_field configuration in the [Syslog] section.

Key	Value	Description
The key is being derived from the value of events coming in of which key is specified by `event_subtype_field`.	`true` or `false`	Currently supported event types (as key) are `DetectionSummaryEvent_DnsRequests`, `DetectionSummaryEvent_NetworkAccesses`, `DetectionSummaryEver` and `DetectionSummaryEvent_ExecutablesWritten`. If true, event will be collection, otherwise event will be skipped and not reported.

Syslog Mappings

MAPPING SECTION

The section names enclosed by square brackets, such as `[DetectionSummaryEvent]` and `[DetectionSummaryEvent_DnsRequests]` are derived from event types and event sub-types. The format is `[EventType_EventSubType]` , such as `[DetectionSummaryEvent_DnsRequests]`

MAPPING KEY-VALUE PAIRS

> **Important**: If the event type and/or event sub-type section is not specified, then the event WILL NOT be included. Additionally, if the field is not specified in the section for the event type and/or sub-type, the field will not be included.The key is the value to which the field will be mapped. For example:

```
externalID = event.SensorID
```

means that the value of `event.SensorID` will be using 'externalID' as a key. Thus, if `event.SensorID` is `ThisIsMySensorID`, the result will be:

```
externalID = "ThisIsMySensorID"
```

In the example above, we would also specify `val_enclosure = "` and `key_val_delim` with an equal sign `=` as values.If a configuration value is enclosed within a single quote, then the value will be taken as-is (literal). This is useful when we need to specify labels. For example:

```
aLiteralKey = 'This is a literal enclosed by single quote'
```

> **Tip**: Any value can be truncated to a character length limit (in version 1.2.0 and later of the SIEM Connector). Append `; max_length=<number_of_characters>` to the configuration line. For example:`commandLine = event.CommandLine; max_length=5`

MAPPING HEADER

In order to specify the event type/sub-type headers, you can use the following notation:

```
__header.{n}
```

where `{n}` is a number starting with 0. The header has the same rules as the mappings:

1. If the value of the header configuration is not enclosed by a single quote, the value will be taken from the incoming event for the specific event type/sub-type.
2. If the value of header is enclosed by single quote, the value will be taken as-is from the value enclosed by single quote.

Example:

```
[DetectionSummaryEvent_DnsRequests]
__header.0 = metadata.eventType
__header.1 = 'DNS Request In A Detection Summary Event'
__header.2 = event.Severity

externalID = event.SensorId
spid = event.ProcessId
shost = event.ComputerName
suser = event.UserName
fname = event.FileName
filePath = event.FilePath
cs1Label = 'CommandLine'
cs1 = event.CommandLine
sntdom = event.MachineDomain
dhost = event.DnsRequests.DomainName
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'DnsRequestTime'
deviceCustomDate1 = event.DnsRequests.LoadTime
```

Splunk Configuration for the Falcon SIEM Connector Log

The Falcon SIEM Connector log may be indexed in Splunk by adding the log file location to the `inputs.conf` file and adding the indexing properties to the `props.conf` file.The default location for the Connector logs is `/var/log/crowdstrike/falconhoseclient/`. Using this as the location, add the following monitoring stanza to `inputs.conf`:

```
[monitor:///var/log/crowdstrike/falconhoseclient/output]
disabled = false
sourcetype = firehose
```

To configure basic attributes for event line breaking, timestamp extraction, and max lines per event (MAX_EVENTS), add the following stanza to `props.conf`:

```
[firehose]
BREAK_ONLY_BEFORE = ^{
DATETIME_CONFIG =
```

```
MAX_EVENTS = 2048
NO_BINARY_CHECK = true
TIME_PREFIX = (\"LoginTime":\s)|(\"ProcessStartTime\":\s)|(\"UTCTimestamp\":\s)
TRUNCATE = 0
category = Custom
disabled = false
pulldown_type = true
```

The above configuration corresponds to the following event types:

1. AuthActivityAuditEvent
2. DetectionSummaryEvent
3. LoginAuditEvent
4. UserActivityAuditEvent

Additional event types may not be covered.

The output connector log contains events formatted in JSON. Splunk line breaking is configured via regular expression, expressed as the left most opening brace ("{") marking the start of a new JSON event: i.e. each JSON event begins with a left brace ("{") at column one.

Once the configurations are in place, Falcon SIEM Connector log events will be indexed. You may need to restart Splunk to enable indexing. The following basic search may be used to return raw connector log events:

```
index=main source="/var/log/crowdstrike/falconhoseclient/output" sourcetype="firehose"
```

Be sure to select the search window in the time picker, or add earliest and latest times to the search, prior to running the above search.

Starting, Stopping, and Restarting the SIEM Connector

- By default, starting, stopping, and restarting the service all require root permission.
- When you start or restart the SIEM Connector service, the event output file is cleared.

Start the SIEM Connector

Run this command at a terminal:

- CentOS:

```
sudo service cs.falconhoseclientd start
```

- Ubuntu 14.x:

```
sudo start cs.falconhoseclientd
```

- Ubuntu 16.04 and later:

```
sudo systemctl start cs.falconhoseclientd.service
```

Stop the SIEM Connector

Run this command at a terminal:

- CentOS:

```
sudo service cs.falconhoseclientd stop
```

- Ubuntu 14.x:

```
sudo stop cs.falconhoseclientd
```

- Ubuntu 16.04 and later:

```
sudo systemctl stop cs.falconhoseclientd.service
```

Restart the SIEM Connector

Run this command at a terminal:

- CentOS:

```
sudo service cs.falconhoseclientd restart
```

- Ubuntu 14.x:

```
sudo restart cs.falconhoseclientd
```

- Ubuntu 16.04 and later:

```
sudo systemctl restart cs.falconhoseclientd.service
```

Updating the SIEM Connector

To update to a new version of the SIEM Connector:

1. Stop the SIEM Connector.
2. Make a backup copy of the directory containing your config files: `/opt/crowdstrike/config/`
3. Download and run the new version's installer.
4. Start the SIEM Connector.

Updating to version 2.0

Updating to version 2.0 of the SIEM connector involves several changes to support OAuth2-based authentication.

> Logs and error messages were improved in version 2.0, so their format may be different after you update.

1. Stop the SIEM connector
2. Create an API client for OAuth2-based authentication of the SIEM connector:
    1. In the Falcon console, go to **Support > API Clients and Keys**
    2. Click **Add new API client**, and provide a name and description
    3. Select the **read** scope for **Event Streams**
    4. Click **Add**, and record the API client ID and API client secret to use in the next step
3. Edit your client config file (default location: `/opt/crowdstrike/config`) with these changes:

    > Before editing, we recommend making a backup copy of your client config file outside the `/opt/crowdstrike` directory.

    1. Change the values for these fields:

Config file field	Correct value
`version`	`3`
`api_url`	Specify a value based on your cloud: ▪ `https://api.crowdstrike.com/sensors/entities/datafeed/v2` (US-1) ▪ `https://api.us-2.crowdstrike.com/sensors/entities/datafeed/v2` (US-2) ▪ `https://api.eu-1.crowdstrike.com/sensors/entities/datafeed/v2` (EU-1) ▪ `https://api.laggar.gcw.crowdstrike.com/sensors/entities/datafeed/v2` (US-GOV-1)

    2. Remove these fields and values: `api_key` and `api_uuid`
    3. Add a new line for each of these fields:

Config file field	Correct value

`request_token_url`	Specify a value based on your cloud: ▪ `https://api.crowdstrike.com/oauth2/token` (US-1) ▪ `https://api.us-2.crowdstrike.com/oauth2/token` (US-2) ▪ `https://api.eu-1.crowdstrike.com/oauth2/token` (EU-1) ▪ `https://api.laggar.gcw.crowdstrike.com/oauth2/token` (US-GOV-1)
`client_id`	Your API client ID
`client_secret`	Your API client secret
`app_id`	A unique app ID used to label your application. Only alphanumeric characters are valid (`a-z`, `A-Z`, `0-9`). Max: 32 characters.

4. Download the latest SIEM connector installer from **Support > Tool Downloads**
5. Install the SIEM connector as described above

   CentOS: if the installation command results in a file name conflict error, force install to overwrite the previous SIEM connector installation: `rpm -i <installer_package> --force`

6. Start the SIEM connector as described above

Updating to version 1.3.1 or later

The config files for SIEM Connector version 1.3.1 and later support the MITRE-based Falcon detection framework. The Streaming API contains the following fields to replace the DetectName field: Objective, Tactic, Technique, PatternDispositionDescription, and PatternDispositionValue. The DetectName field will be deprecated on January 16, 2019.

The config files of fresh installations of version 1.3.1 and later will have the new fields and be ready for any workflows you need to create or remap.

Upgrading to a SIEM Connector version that supports MITRE-based detections requires manual changes to the config file, because upgrading the SIEM Connector doesn't alter your config file. The sample config files in the etc/ directory will contain the sample configurations. The following config file update procedures provide guidance. Make changes to CEF and LEEF files to meet your requirements.

CEF CONFIG FILE UPDATE

1. Complete the following replacements:

Original	Replace with
`__header.1 = metadata.eventType`	`__header.1 = event.DetectName`
`cat = event.DetectName`	`cat = event.Tactic`

2. Add the following lines after `cat - event.Tactic`:
   - `act = event.Technique`
   - `reason = event.Objective`
   - `outcome = event.PatternDispositionDescription`
   - `CSMTRPatternDisposition = event.PatternDispositionValue`

LEEF CONFIG FILE UPDATE

Add the following lines after `srcMac = event.MACAdress`:

- `tactic = event.Tactic`
- `technique = event.Technique`
- `objective = event.Objective`
- `patternDisposition = event.PatternDispositionValue`
- `outcome = event.PatternDispositionDescription`

Uninstalling the SIEM Connector

Run this command at a terminal:

- CentOS:

```
sudo rpm -e cs.falconhoseclient
```

- Ubuntu:

```
sudo dpkg --remove crowdstrike-cs-falconhoseclient
```

Troubleshooting

When troubleshooting, refer to `/var/log/crowdstrike/falconhoseclient/output` for error lines.

Configuration Errors

Configuration errors are prefixed with ERROR[config] in the log file.

Error	Description/Resolution
`api_password setting is required`	Please specify valid `api_password` under [Settings] section.
`api_url setting is required`	Please specify valid `api_url` under [Settings] section.
`api_username setting is required`	Please specify valid `api_username` under [Settings] section.
`api_uuid setting is required`	Please specify valid `api_uuid` under [Settings] section.
`Invalid connection_timeout configuration value: <value>`	Please specify valid `connection_timeout` value under [Settings] section.
`Invalid log max size, defaulting to <size>MB`	Please specify valid `logging max_size` value under [Logging] section.
`Invalid partition configuration value: <value>`	Please specify valid partition value under [Settings] section.
`Invalid read_timeout configuration value: <value>`	Please specify valid `read_timeout` value under [Settings] section.
`Invalid send_to_syslog_server configuration value: <value>`	Please specify valid `send_to_syslog_server` value under [Syslog] section.
`Missing [Settings] section in configuration file. Unable to determine what format to produce`	Configuration file might be malformed. Please ensure that there is [Settings] section in the configuration file.
`Missing event_subtype_field configuration for Syslog.`	Please specify valid `event_subtype_field` value under [Syslog] section.
`Missing event_type_field configuration for Syslog.`	Please specify valid `event_type_field` value under [Syslog] section.
`Missing field_delim configuration for Syslog`	Please specify valid `field_delim` value under [Syslog] section.
`Missing header_delim configuration for Syslog`	Please specify valid `header_delim` value under [Syslog] section.
`Missing header_prefix configuration for Syslog`	Please specify valid `header_prefix` value under [Syslog] section.
`Missing host configuration for Syslog`	Please specify valid protocol host under [Syslog] section.
`Missing key_val_delim configuration for Syslog`	Please specify valid `key_val_delim` value under [Syslog] section.
`Missing max_length configuration for Syslog`	Please specify valid `max_length` value under [Syslog] section.
`Missing port configuration for Syslog`	Please specify valid port value under [Syslog] section.
`Missing protocol configuration for Syslog`	Please specify valid protocol setting under [Syslog] section.
`Missing send_to_syslog_server configuration`	Please specify valid `send_to_syslog_server` setting under [Syslog] section.
`Missing Settings section in configuration file.`	Configuration file might be malformed. Please ensure that there is [Settings] section in the configuration file.
`Missing tag configuration for Syslog`	Please specify valid tag value under [Syslog] section.
`Missing time_fields configuration for Syslog`	Please specify valid `time_fields` value under [Syslog] section.
`Missing val_enclosure configuration for Syslog`	Please specify valid `val_enclosure` value under [Syslog] section.

`Unable to check if we want to output to file through output_to_file configuration`	Missing `output_to_file` under [Settings] section to specify whether or not client should output the event to file.
`Unable to create output file: <output_file>, writing to service log: <service_log>`	Please specify valid `output_path` under [Settings] section to output event into and make sure that write permission is given to service.
`Unable to retrieve log max size, defaulting to <size>MB`	Please specify valid `logging max_size` under [Logging] section.
`Unable to retrieve output_path, default to service log`	Please specify valid `output_path` under [Settings] section to output event into and make sure that write permission is given to service.
`Unsupported protocol for Syslog. Supported protocols are udp or tcp`	Please specify to use udp/tcp for syslog protocol under [Syslog] section.
`Version setting '<version>' is invalid`	Invalid version setting under [Settings] section.
`Version setting is required`	Missing version setting under [Settings] section.

Connector/Client Errors

Error	Description/resolution
`Discovery failed with HTTP: <http_code>, Payload: <payload>`	The server is returned a failure as a HTTP status code, usually in the 4xx range. Confirm that your credentials are correct and that your device's time is accurate.
`Failed to convert offsets [<offsets>]`	Failure occurred while converting offset to be stored. Please contact CrowdStrike Customer Support with the log file.
`Failed to handle feed for partition <partition_no>: <event> - <message>`	Failure occurred while performing event transformation or posting to syslog remote server. `<message>` contains more information about this error. Please ensure that syslog remote server is configured properly and please contact CrowdStrike Customer Support with the log file.
`Failed to read from i/o buffer - <message>`	Failure occurred while digesting events. `<message>` contains more information about this error. This can occur for several reasons (1) Network interruption occurs during ingestion of event: Retries will be attempted. (2) Falcon Streaming Client service is interrupted by shutdown: Retries will be attempted after start up.
`Failed to retrieve partition/offset <partition_no> with last <format> data <event>`	Failure occurred while parsing event data to retrieve offset. Please contact CrowdStrike Customer Support with the log file.
`Failed to save offsets [<offsets>] to file`	Failure occurred while attempting to save offset to file. Please ensure that permission is granted to service to write to file system under `/opt/crowdstrike/*`.
`Missing data feed URL for <configuration>`	In some cases, when connector/client recovered from ungraceful disconnects i.e. power outage, hot reboot etc, server still maintains the existing session with the given `app_id`. Certain configurations in client's environment may maintain the opened connection to server longer than expected, this will in turn cause the endpoint to be unaware of client's disconnections. Falcon Streaming Client/Connector will keep retrying until streaming is successful. In order to get around this quicker, `app_id` can be modified into a different `app_id` and restart the service.

`No resource discovered for<configuration>`	In some cases, when connector/client recovered from ungraceful disconnects i.e. power outage, hot reboot etc, server still maintains the existing session with the given `app_id`. Certain configurations in client's environment may maintain the opened connection to server longer than expected, this will in turn cause the endpoint to be unaware of client's disconnections. Falcon Streaming Client/Connector will keep retrying until streaming is successful. In order to get around this quicker, `app_id` can be modified into a different `app_id` and restart the service.
`Partition streaming failed - HTTP: <http_code> - <message>`	Server is returning status code other than 200 during event streaming. The `<http_code>` is usually in the 4xx range. In most cases, HTTP code will be 401 which means unauthorized access has been attempted. `<message>` part contains more information about this error. Retries will be performed. Please ensure that credentials provided are correct.
`Partition streaming failed @ GET: <partition_no> - <message>`	Error returned during the attempt to stream partition from the underlying HTTP GET. Usually caused by underlying network connectivity which renders the client unable to reach the Falcon Streaming endpoint part contains more information about this error. Retries will be performed. Please ensure that endpoint is reachable from client machine.
`Timed out, last heartbeat received <time> ago`	Timed out occurred while waiting for Falcon Streaming endpoint heartbeat. Retries will be attempted. Please make sure client machine is able to reach Falcon Streaming endpoint.
`Unable to parse dataFeedURL from <configuration>`	This is cause by server returning unexpected URL format. Please contact CrowdStrike Customer Support with the log file.

Transformation/Syslog Errors

Configuration errors are prefixed with ERROR[syslog] in the log file.

Error	Description/resolution
`Missing event (sub)type for <event>`	Please contact CrowdStrike Customer Support with the log file.
`Unable to obtain section key for event: <event>`	Please contact CrowdStrike Customer Support with the log file.
`Unable to parse <event_value> as time with format(<format>)`	Please review your time_fields configuration, the field might not have time value.
`Missing field name:<field_name> in the feed type: <feed_type>`	Please review your mapping configuration with the given `field_name` under `feed_type` section.
`Unexpected JSON value in array format from key <key>, value <value>`	Please contact CrowdStrike Customer Support with the log file.

SIEM reference info

Default Directories

- Installation: `/opt/crowdstrike`
- Service script:
  - CentOS: `/etc/init.d/cs.falconhoseclientd`
  - Ubuntu: `/etc/init/cs.falconhoseclientd`
- Logs: /`var/log/crowdstrike/falconhoseclient/`

IPs for legacy streaming API

If your SIEM connector credentials use our legacy (key-based) API authentication, you can use the IP addresses in Cloud IP Addresses to allowlist TLS traffic by IP address instead of by FQDN. See the **Streaming API** section for the applicable cloud.

Default Config Files

These are blank copies of the default config files for the SIEM connector in CEF, These are sample events that are included in the default `.cfg` file. Adapt these examples to your environment and the data you want to send to your SIEM.

BASIC CONFIG FILE

```
[Settings]
version = 3
api_url = https://api.crowdstrike.com/sensors/entities/datafeed/v2
request_token_url = https://api.crowdstrike.com/oauth2/token
app_id = SIEM-Connector-v2.0.0
# API Client ID
client_id =
# API Client Secret
client_secret =
# Amount of time (in seconds) we will wait for a connect to complete.
connection_timeout = 10
# Amount of time to wait (in seconds) for a server's response headers after fully writing the request.
read_timeout = 30
# Specify partition number 0 to n or 'all' (without quote) for all partitions
partition = all
http_proxy =
# 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 = 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/output
# Offset file full filepath and filename
offset_path = /var/log/crowdstrike/falconhoseclient/stream_offsets
[Output_File_Rotation]
# If the output is writing to a file, then the settings below will govern output file rotation
#
# If true, then the rotation rules will apply. If not, the client will continue to write to the same file.
rotate_file = true
# Maximum individual output file size in MB
max_size = 500
# Number of backups of the output file to be stored
max_backups = 10
# Maximum age of backup output files before it is deleted in DAYS
max_age = 30
[Logging]
verbose_log = true
# Maximum individual log file size in MB
max_size = 500
# Number of backups to be stored
max_backups = 10
# Maximum age of backup files before it is deleted in DAYS
max_age = 30
[Syslog]
send_to_syslog_server = false
host = localhost
port = 514
protocol = udp
# CEF/LEEF Headers, header_prefix will be appended before any other header information
# Within each mapping section, we can add __header.{n} (note double underscore) where n is consecutive integer
# starting with 0 which will be added sequentially.
# Value of headers can be:
#   1. As specified: enclose by single-quote
#   2. Field value: just specify which field name
header_delim = |
header_prefix = CEF:0|CrowdStrike|FalconHost|1.0|
# Character Escaping Setting
# Syntax Guidelines:
#   - Enclose characters with double-quote i.e. "|"
#   - From and To characters are delimited by colon
#   - Character(s) that needs to be escaped is placed on the left side of a colon (:) and character to replace with is on the right
i.e. "from":"to"
#   - Multiple character escape setting is delimited by a common i.e. "from1":"to1","from2":"to2" and so on
#   - header_prefix setting (above) will not be escaped
escape_header = "|":"\|","\\":"\\\\"
escape_ext = "\\":"\\\\","=":"\=","\n":"\\n","\r":"\\r"
# Delimiter separating key and value, example: if the delimiter is '='(equal): filename=abc.txt
```

```
key_val_delim = =

# Delimiter separating 2 key-value pairs , example: if the delimiter is ','(comma): filename=abc.txt,domain=www.google.com

# Note: For space just leave it empty

field_delim =

val_enclosure =

# These fields will be converted to time format, field name should be the key on the mapping section (RFC3339)

time_fields = deviceCustomDate1

time_format = MMM dd yyyy HH:mm:ss

# This will be use for filtering

event_type_field = metadata.eventType

event_subtype_field = event.subType

# Max length of syslog line in bytes

max_length = 2048

# Send retry interval in seconds (applicable only for TCP)

retry_interval_secs = 10

# Static order fields

keys_ordered = true

[EventTypeCollection]

DetectionSummaryEvent = true

AuthActivityAuditEvent = true

UserActivityAuditEvent = true

RemoteResponseSessionStartEvent = true

RemoteResponseSessionEndEvent = true

# ------------------------------------------------------------------------------------------------------------------

# Below configurations only applies if syslog is ENABLED (under Syslog: enabled=true

# ------------------------------------------------------------------------------------------------------------------

[EventSubTypeCollection]

# Format: <EvenType>_<EventSubType> = true/false

DetectionSummaryEvent_DnsRequests = true

DetectionSummaryEvent_NetworkAccesses = true

DetectionSummaryEvent_DocumentsAccessed = true

DetectionSummaryEvent_ScanResults = true

DetectionSummaryEvent_ExecutablesWritten = true

DetectionSummaryEvent_QuarantineFiles = true

# FIELD MAPPINGS

# Section name format: <EventType> OR <EventType>_<EventSubType>

# Reserved keys:

#       __header.{n} where n is integer starting with 0

#

# There are 2 possible values for the mapping

#       1. Literals which will be used as-is (for labelling) should be enclosed by single quotes

#       2. Value based on incoming event

#

# If field mapping is not specified, then field will not appear in the results

# DetectName has been deprecated because CrowdStrike now supports MITRE framework

[DetectionSummaryEvent]

__header.0 = metadata.eventType

__header.1 = metadata.eventType

__header.2 = event.Severity

cat = event.Tactic
```

```
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
cn1Label = 'ParentProcessId'
cn1 = event.ParentProcessId
dhost = event.ComputerName
duser = event.UserName
msg = event.DetectDescription
fname = event.FileName
filePath = event.FilePath
cs5Label = 'CommandLine'
cs5 = event.CommandLine
fileHash = event.MD5String
dntdom = event.MachineDomain
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
rt = metadata.eventCreationTime
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
outcome = event.PatternDispositionValue
[DetectionSummaryEvent_DnsRequests]
__header.0 = 'DNS Request In A Detection Summary Event'
__header.1 = 'DNS Request In A Detection Summary Event'
__header.2 = event.Severity
cat = event.Tactic
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
dntdom = event.MachineDomain
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'DNS Request Time'
deviceCustomDate1 = event.DnsRequests.LoadTime
rt = metadata.eventCreationTime
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
```

```
outcome = event.PatternDispositionValue

[DetectionSummaryEvent_NetworkAccesses]

__header.0 = 'Network Access In A Detection Summary Event'

__header.1 = 'Network Access In A Detection Summary Event'

__header.2 = event.Severity

cat = event.Tactic

externalId = event.SensorId

cn2Label = 'ProcessId'

cn2 = event.ProcessId

dhost = event.ComputerName

duser = event.UserName

fname = event.FileName

filePath = event.FilePath

cs5Label = 'CommandLine'

cs5 = event.CommandLine

dntdom = event.MachineDomain

src = event.NetworkAccesses.LocalAddress

c6a2 = event.NetworkAccesses.LocalAddress

dst = event.NetworkAccesses.RemoteAddress

c6a3 = event.NetworkAccesses.RemoteAddress

spt = event.NetworkAccesses.LocalPort

dpt = event.NetworkAccesses.RemotePort

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink

cn3Label = 'Offset'

cn3 = metadata.offset

deviceCustomDate1Label = 'Network Access Timestamp'

deviceCustomDate1 = event.NetworkAccesses.AccessTimestamp

rt = metadata.eventCreationTime

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[DetectionSummaryEvent_DocumentsAccessed]

__header.0 = 'Document Access In A Detection Summary Event'

__header.1 = 'Document Access In A Detection Summary Event'

__header.2 = event.Severity

cat = event.Tactic

externalId = event.SensorId

cn2Label = 'ProcessId'

cn2 = event.ProcessId

dhost = event.ComputerName

duser = event.UserName

fname = event.FileName

filePath = event.FilePath

dntdom = event.MachineDomain

cs2Label = 'AccessedDocFileName'

cs2 = event.DocumentsAccessed.FileName

cs3Label = 'AccessedDocFilePath'
```

```
cs3 = event.DocumentsAccessed.FilePath

cs5Label = 'CommandLine'

cs5 = event.CommandLine

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink

cn3Label = 'Offset'

cn3 = metadata.offset

deviceCustomDate1Label = 'Document Accessed Timestamp'

deviceCustomDate1 = event.DocumentsAccessed.Timestamp

rt = metadata.eventCreationTime

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[DetectionSummaryEvent_ScanResults]

__header.0 = 'AV Scan Results In A Detection Summary Event'

__header.1 = 'AV Scan Results In A Detection Summary Event'

__header.2 = event.Severity

cat = event.Tactic

externalId = event.SensorId

cn2Label = 'ProcessId'

cn2 = event.ProcessId

dhost = event.ComputerName

duser = event.UserName

fname = event.FileName

filePath = event.FilePath

fileHash = event.MD5String

dntdom = event.MachineDomain

cs2Label = 'ScanResultEngine'

cs2 = event.ScanResults.Engine

cs1Label = 'ScanResultName'

cs1 = event.ScanResults.ResultName

cs4Label = 'ScanResultVersion'

cs4 = event.ScanResults.Version

cs5Label = 'CommandLine'

cs5 = event.CommandLine

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink

cn3Label = 'Offset'

cn3 = metadata.offset

rt = metadata.eventCreationTime

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[DetectionSummaryEvent_ExecutablesWritten]

__header.0 = 'Executable Written In A Detection Summary Event'

__header.1 = 'Executable Written In A Detection Summary Event'
```

```
__header.2 = event.Severity
cat = event.Tactic
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
dntdom = event.MachineDomain
cs2Label = 'WrittenExeFileName'
cs2 = event.ExecutablesWritten.FileName
cs3Label = 'WrittenExeFilePath'
cs3 = event.ExecutablesWritten.FilePath
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'ExeWrittenTimestamp'
deviceCustomDate1 = event.ExecutablesWritten.Timestamp
rt = metadata.eventCreationTime
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
outcome = event.PatternDispositionValue
[DetectionSummaryEvent_QuarantineFiles]
__header.0 = 'Quarantined Files In A Detection Summary Event'
__header.1 = 'Quarantined Files  In A Detection Summary Event'
__header.2 = event.Severity
cat = event.Tactic
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
dntdom = event.MachineDomain
cs2Label = 'QuarantineFileSHA256'
cs2 = event.QuarantineFiles.SHA256HashData
cs3Label = 'QuarantineFilePath'
cs3 = event.QuarantineFiles.ImageFileName
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
```

```
cn3 = metadata.offset

deviceCustomDate1Label = 'ExeWrittenTimestamp'

deviceCustomDate1 = event.ExecutablesWritten.Timestamp

rt = metadata.eventCreationTime

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[UserActivityAuditEvent]

__header.0 = metadata.eventType

__header.1 = event.OperationName

__header.2 = '1'

cat = metadata.eventType

destinationTranslatedAddress = event.UserIp

duser = event.UserId

deviceProcessName = event.ServiceName

cn3Label = 'Offset'

cn3 = metadata.offset

outcome = event.Success

rt = metadata.eventCreationTime

[AuthActivityAuditEvent]

__header.0 = event.OperationName

__header.1 = event.OperationName

__header.2 = '1'

cat = metadata.eventType

destinationTranslatedAddress = event.UserIp

duser = event.UserId

deviceProcessName = event.ServiceName

cn3Label = 'Offset'

cn3 = metadata.offset

outcome = event.Success

deviceCustomDate1Label = 'Timestamp'

deviceCustomDate1 = event.UTCTimestamp

rt = metadata.eventCreationTime

[RemoteResponseSessionStartEvent]

__header.0 = metadata.eventType

__header.1 = 'Remote Response Session Start event'

__header.2 = '1'

cat = metadata.eventType

cn3Label = 'Offset'

cn3 = metadata.offset

rt = metadata.eventCreationTime

dhost = event.HostnameField

duser = event.UserName

sessionStartTimestampLabel = 'RemoteResponseSessionStartTimestamp'

sessionStartTimestamp = event.StartTimestamp

[RemoteResponseSessionEndEvent]

__header.0 = metadata.eventType

__header.1 = 'Remote Response Session End event'
```

```
__header.2 = '1'
cat = metadata.eventType
cn3Label = 'Offset'
cn3 = metadata.offset
rt = metadata.eventCreationTime
dhost = event.HostnameField
duser = event.UserName
sessionEndTimestampLabel = 'RemoteResponseSessionEndTimestamp'
sessionEndTimestamp = event.EndTimestamp
```

CEF FORMAT CONFIG FILE

```
[Settings]
version = 3
api_url = https://api.crowdstrike.com/sensors/entities/datafeed/v2
request_token_url = https://api.crowdstrike.com/oauth2/token
app_id = SIEM-Connector-CEF-v2.0.0
# API Client ID
client_id =
# API Client Secret
client_secret =
# Amount of time (in seconds) we will wait for a connect to complete.
connection_timeout = 10
# Amount of time to wait (in seconds) for a server's response headers after fully writing the request.
read_timeout = 30
# Specify partition number 0 to n or 'all' (without quote) for all partitions
partition = all
http_proxy =
# 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
# 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/output
# Offset file full filepath and filename
offset_path = /var/log/crowdstrike/falconhoseclient/stream_offsets
[Output_File_Rotation]
# If the output is writing to a file, then the settings below will govern output file rotation
#
# If true, then the rotation rules will apply. If not, the client will continue to write to the same file.
rotate_file = true
# Maximum individual output file size in MB
max_size = 500
# Number of backups of the output file to be stored
```

```
max_backups = 10

# Maximum age of backup output files before it is deleted in DAYS

max_age = 30

[Logging]

verbose_log = true

# Maximum individual log file size in MB

max_size = 500

# Number of backups to be stored

max_backups = 10

# Maximum age of backup files before it is deleted in DAYS

max_age = 30

[Syslog]

send_to_syslog_server = false

host = localhost

port = 514

protocol = udp

# CEF/LEEF Headers, header_prefix will be appended before any other header information

# Within each mapping section, we can add __header.{n} (note double underscore) where n is consecutive integer

# starting with 0 which will be added sequentially.

# Value of headers can be:

#   1. As specified: enclose by single-quote

#   2. Field value: just specify which field name

header_delim = |

header_prefix = CEF:0|CrowdStrike|FalconHost|1.0|

# Character Escaping Setting

# Syntax Guidelines:

#   - Enclose characters with double-quote i.e. "|"

#   - From and To characters are delimited by colon

#   - Character(s) that needs to be escaped is placed on the left side of a colon (:) and character to replace with is on the right
i.e. "from":"to"

#   - Multiple character escape setting is delimited by a common i.e. "from1":"to1","from2":"to2" and so on

#   - header_prefix setting (above) will not be escaped

escape_header = "|":"\|","\\":"\\\\"

escape_ext = "\\":"\\\\","=":"\=","\n":"\\n","\r":"\\r"

# Delimiter separating key and value, example: if the delimiter is '='(equal): filename=abc.txt

key_val_delim = =

# Delimiter separating 2 key-value pairs , example: if the delimiter is ','(comma): filename=abc.txt,domain=www.google.com

# Note: For space just leave it empty

field_delim =

val_enclosure =

# These fields will be converted to time format, field name should be the key on the mapping section (RFC3339)

time_fields = deviceCustomDate1

time_format = MMM dd yyyy HH:mm:ss

# This will be use for filtering

event_type_field = metadata.eventType

event_subtype_field = event.subType

# Max length of syslog line in bytes

max_length = 2048

# Send retry interval in seconds (applicable only for TCP)

retry_interval_secs = 10

# Static order fields
```

```
keys_ordered = true

[EventTypeCollection]

DetectionSummaryEvent = true

AuthActivityAuditEvent = true

UserActivityAuditEvent = true

RemoteResponseSessionStartEvent = true

RemoteResponseSessionEndEvent = true

# ------------------------------------------------------------------------------------------------------------------

# Below configurations only applies if syslog is ENABLED (under Syslog: enabled=true

# ------------------------------------------------------------------------------------------------------------------

[EventSubTypeCollection]

# Format: <EvenType>_<EventSubType> = true/false

DetectionSummaryEvent_DnsRequests = true

DetectionSummaryEvent_NetworkAccesses = true

DetectionSummaryEvent_DocumentsAccessed = true

DetectionSummaryEvent_ScanResults = true

DetectionSummaryEvent_ExecutablesWritten = true

DetectionSummaryEvent_QuarantineFiles = true

# FIELD MAPPINGS

# Section name format: <EventType> OR <EventType>_<EventSubType>

# Reserved keys:

#       __header.{n} where n is integer starting with 0

#

# There are 2 possible values for the mapping

#       1. Literals which will be used as-is (for labelling) should be enclosed by single quotes

#       2. Value based on incoming event

#

# If field mapping is not specified, then field will not appear in the results

# DetectName has been deprecated because CrowdStrike now supports MITRE framework

[DetectionSummaryEvent]

__header.0 = metadata.eventType

#__header.1 = event.DetectName

__header.1 = event.Tactic

__header.2 = event.Severity

externalId = event.SensorId

cn2Label = 'ProcessId'

cn2 = event.ProcessId

cn1Label = 'ParentProcessId'

cn1 = event.ParentProcessId

dhost = event.ComputerName

duser = event.UserName

msg = event.DetectDescription

fname = event.FileName

filePath = event.FilePath

cs5Label = 'CommandLine'

cs5 = event.CommandLine

fileHash = event.MD5String

dntdom = event.MachineDomain

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink
```

```
cn3Label = 'Offset'
cn3 = metadata.offset
rt = metadata.eventCreationTime
src = event.LocalIP
smac = event.MACAddress
cat = event.Tactic
act = event.Technique
reason = event.Objective
outcome = event.PatternDispositionValue
CSMTRPatternDisposition = event.PatternDispositionDescription
[DetectionSummaryEvent_DnsRequests]
__header.0 = 'DNS Request In A Detection Summary Event'
#__header.1 = event.DetectName
__header.1 = event.Tactic
__header.2 = event.Severity
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
dntdom = event.MachineDomain
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'DNS Request Time'
deviceCustomDate1 = event.DnsRequests.LoadTime
rt = metadata.eventCreationTime
src = event.LocalIP
smac = event.MACAddress
cat = event.Tactic
act = event.Technique
reason = event.Objective
outcome = event.PatternDispositionValue
CSMTRPatternDisposition = event.PatternDispositionDescription
[DetectionSummaryEvent_NetworkAccesses]
__header.0 = 'Network Access In A Detection Summary Event'
#__header.1 = event.DetectName
__header.1 = event.Tactic
__header.2 = event.Severity
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
```

```
filePath = event.FilePath
cs5Label = 'CommandLine'
cs5 = event.CommandLine
dntdom = event.MachineDomain
c6a2 = event.NetworkAccesses.LocalAddress
dst = event.NetworkAccesses.RemoteAddress
c6a3 = event.NetworkAccesses.RemoteAddress
spt = event.NetworkAccesses.LocalPort
dpt = event.NetworkAccesses.RemotePort
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'Network Access Timestamp'
deviceCustomDate1 = event.NetworkAccesses.AccessTimestamp
rt = metadata.eventCreationTime
src = event.LocalIP
smac = event.MACAddress
cat = event.Tactic
act = event.Technique
reason = event.Objective
outcome = event.PatternDispositionValue
CSMTRPatternDisposition = event.PatternDispositionDescription
[DetectionSummaryEvent_DocumentsAccessed]
__header.0 = 'Document Access In A Detection Summary Event'
#__header.1 = event.DetectName
__header.1 = event.Tactic
__header.2 = event.Severity
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
dntdom = event.MachineDomain
cs2Label = 'AccessedDocFileName'
cs2 = event.DocumentsAccessed.FileName
cs3Label = 'AccessedDocFilePath'
cs3 = event.DocumentsAccessed.FilePath
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
deviceCustomDate1Label = 'Document Accessed Timestamp'
deviceCustomDate1 = event.DocumentsAccessed.Timestamp
rt = metadata.eventCreationTime
src = event.LocalIP
```

```
smac = event.MACAddress
cat = event.Tactic
act = event.Technique
reason = event.Objective
outcome = event.PatternDispositionValue
CSMTRPatternDisposition = event.PatternDispositionDescription
[DetectionSummaryEvent_ScanResults]
__header.0 = 'AV Scan Results In A Detection Summary Event'
#__header.1 = event.DetectName
__header.1 = event.Tactic
__header.2 = event.Severity
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
fname = event.FileName
filePath = event.FilePath
fileHash = event.MD5String
dntdom = event.MachineDomain
cs2Label = 'ScanResultEngine'
cs2 = event.ScanResults.Engine
cs1Label = 'ScanResultName'
cs1 = event.ScanResults.ResultName
cs4Label = 'ScanResultVersion'
cs4 = event.ScanResults.Version
cs5Label = 'CommandLine'
cs5 = event.CommandLine
cs6Label = 'FalconHostLink'
cs6 = event.FalconHostLink
cn3Label = 'Offset'
cn3 = metadata.offset
rt = metadata.eventCreationTime
src = event.LocalIP
smac = event.MACAddress
cat = event.Tactic
act = event.Technique
reason = event.Objective
outcome = event.PatternDispositionValue
CSMTRPatternDisposition = event.PatternDispositionDescription
[DetectionSummaryEvent_ExecutablesWritten]
__header.0 = 'Executable Written In A Detection Summary Event'
#__header.1 = event.DetectName
__header.1 = event.Tactic
__header.2 = event.Severity
externalId = event.SensorId
cn2Label = 'ProcessId'
cn2 = event.ProcessId
dhost = event.ComputerName
duser = event.UserName
```

```
fname = event.FileName

filePath = event.FilePath

dntdom = event.MachineDomain

cs2Label = 'WrittenExeFileName'

cs2 = event.ExecutablesWritten.FileName

cs3Label = 'WrittenExeFilePath'

cs3 = event.ExecutablesWritten.FilePath

cs5Label = 'CommandLine'

cs5 = event.CommandLine

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink

cn3Label = 'Offset'

cn3 = metadata.offset

deviceCustomDate1Label = 'ExeWrittenTimestamp'

deviceCustomDate1 = event.ExecutablesWritten.Timestamp

rt = metadata.eventCreationTime

src = event.LocalIP

smac = event.MACAddress

cat = event.Tactic

act = event.Technique

reason = event.Objective

outcome = event.PatternDispositionValue

CSMTRPatternDisposition = event.PatternDispositionDescription

[DetectionSummaryEvent_QuarantineFiles]

__header.0 = 'Quarantined Files In A Detection Summary Event'

#__header.1 = event.DetectName

__header.1 = event.Tactic

__header.2 = event.Severity

externalId = event.SensorId

cn2Label = 'ProcessId'

cn2 = event.ProcessId

dhost = event.ComputerName

duser = event.UserName

fname = event.FileName

filePath = event.FilePath

dntdom = event.MachineDomain

cs2Label = 'QuarantineFileSHA256'

cs2 = event.QuarantineFiles.SHA256HashData

cs3Label = 'QuarantineFilePath'

cs3 = event.QuarantineFiles.ImageFileName

cs5Label = 'CommandLine'

cs5 = event.CommandLine

cs6Label = 'FalconHostLink'

cs6 = event.FalconHostLink

cn3Label = 'Offset'

cn3 = metadata.offset

deviceCustomDate1Label = 'ExeWrittenTimestamp'

deviceCustomDate1 = event.ExecutablesWritten.Timestamp

rt = metadata.eventCreationTime

src = event.LocalIP
```

```
smac = event.MACAddress

cat = event.Tactic

act = event.Technique

reason = event.Objective

outcome = event.PatternDispositionValue

CSMTRPatternDisposition = event.PatternDispositionDescription

[UserActivityAuditEvent]

__header.0 = metadata.eventType

__header.1 = event.OperationName

__header.2 = '1'

cat = metadata.eventType

destinationTranslatedAddress = event.UserIp

duser = event.UserId

deviceProcessName = event.ServiceName

cn3Label = 'Offset'

cn3 = metadata.offset

outcome = event.Success

rt = metadata.eventCreationTime

[AuthActivityAuditEvent]

__header.0 = event.OperationName

__header.1 = event.OperationName

__header.2 = '1'

cat = metadata.eventType

destinationTranslatedAddress = event.UserIp

duser = event.UserId

deviceProcessName = event.ServiceName

cn3Label = 'Offset'

cn3 = metadata.offset

outcome = event.Success

deviceCustomDate1Label = 'Timestamp'

deviceCustomDate1 = event.UTCTimestamp

rt = metadata.eventCreationTime

[RemoteResponseSessionStartEvent]

__header.0 = metadata.eventType

__header.1 = 'Remote Response Session Start event'

__header.2 = '1'

cat = metadata.eventType

cn3Label = 'Offset'

cn3 = metadata.offset

rt = metadata.eventCreationTime

dhost = event.HostnameField

duser = event.UserName

sessionStartTimestampLabel = 'RemoteResponseSessionStartTimestamp'

sessionStartTimestamp = event.StartTimestamp

[RemoteResponseSessionEndEvent]

__header.0 = metadata.eventType

__header.1 = 'Remote Response Session End event'

__header.2 = '1'

cat = metadata.eventType

cn3Label = 'Offset'
```

```
cn3 = metadata.offset

rt = metadata.eventCreationTime

dhost = event.HostnameField

duser = event.UserName

sessionEndTimestampLabel = 'RemoteResponseSessionEndTimestamp'

sessionEndTimestamp = event.EndTimestamp
```

LEEF FORMAT CONFIG FILE

```
[Settings]

version = 3

api_url = https://api.crowdstrike.com/sensors/entities/datafeed/v2

request_token_url = https://api.crowdstrike.com/oauth2/token

app_id = SIEM-Connector-LEEF-v2.0.0

# API Client ID

client_id =

# API Client Secret

client_secret =

# Amount of time (in seconds) we will wait for a connect to complete.

connection_timeout = 10

# Amount of time to wait (in seconds) for a server's response headers after fully writing the request.

read_timeout = 30

# Specify partition number 0 to n or 'all' (without quote) for all partitions

partition = all

http_proxy =

# 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

# 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/output

# Offset file full filepath and filename

offset_path = /var/log/crowdstrike/falconhoseclient/stream_offsets

[Output_File_Rotation]

# If the output is writing to a file, then the settings below will govern output file rotation

#

# If true, then the rotation rules will apply. If not, the client will continue to write to the same file.

rotate_file = true

# Maximum individual output file size in MB

max_size = 500

# Number of backups of the output file to be stored

max_backups = 10

# Maximum age of backup output files before it is deleted in DAYS

max_age = 30
```

```
[Logging]

verbose_log = true

# Maximum individual log file size in MB

max_size = 500

# Number of backups to be stored

max_backups = 10

# Maximum age of backup files before it is deleted in DAYS

max_age = 30

[Syslog]

send_to_syslog_server = false

host = localhost

port = 514

protocol = udp

# CEF/LEEF Headers, header_prefix will be appended before any other header information

# Within each mapping section, we can add __header.{n} (note double underscore) where n is consecutive integer

# starting with 0 which will be added sequentially.

# Value of headers can be:

#   1. As specified: enclose by single-quote

#   2. Field value: just specify which field name

header_delim = |

header_prefix = LEEF:1.0|CrowdStrike|FalconHost|1.0|

# Character Escaping Setting

# Syntax Guidelines:

#   - Enclose characters with double-quote i.e. "|"

#   - From and To characters are delimited by colon

#   - Character(s) that needs to be escaped is placed on the left side of a colon (:) and character to replace with is on the right
i.e. "from":"to"

#   - Multiple character escape setting is delimited by a common i.e. "from1":"to1","from2":"to2" and so on

#   - Order of escapes will be respected

#   - header_prefix setting (above) will not be escaped

escape_header = "|":"\|","\\":"\\\\"

escape_ext = "\\":"\\\\","=":"\=","\n":"\\n","\r":"\\r","\t":"\\t"

# Delimiter separating key and value, example: if the delimiter is '='(equal): filename=abc.txt

key_val_delim = =

# Delimiter separating 2 key-value pairs , example: if the delimiter is ','(comma): filename=abc.txt,domain=www.google.com

# Note: For space just leave it empty

field_delim = \t

val_enclosure =

# These fields will be converted to time format, field name should be the key on the mapping section (RFC3339)

time_fields = devTime

time_format = yyyy-MM-dd HH:mm:ss

# This will be use for filtering

event_type_field = metadata.eventType

event_subtype_field = event.subType

# Max length of syslog line in bytes

max_length = 2048

# Send retry interval in seconds (applicable only for TCP)

retry_interval_secs = 10

# Static order fields

keys_ordered = false

[EventTypeCollection]
```

```
DetectionSummaryEvent = true

LoginAuditEvent = true

AuthActivityAuditEvent = true

UserActivityAuditEvent = true

CustomerIOCEvent = true

RemoteResponseSessionStartEvent = true

RemoteResponseSessionEndEvent = true

# ------------------------------------------------------------------------------------------------------------------

# Below configurations only applies if syslog is ENABLED (under Syslog: enabled=true

# ------------------------------------------------------------------------------------------------------------------

[EventSubTypeCollection]

# Format: <EvenType>_<EventSubType> = true/false

DetectionSummaryEvent_DnsRequests = true

DetectionSummaryEvent_NetworkAccesses = true

DetectionSummaryEvent_DocumentsAccessed = true

DetectionSummaryEvent_ScanResults = true

DetectionSummaryEvent_ExecutablesWritten = true

# FIELD MAPPINGS

# Section name format: <EventType> OR <EventType>_<EventSubType>

# Reserved keys:

#       __header.{n} where n is integer starting with 0

#

# There are 2 possible values for the mapping

#       1. Literals which will be used as-is (for labelling) should be enclosed by single quotes

#       2. Value based on incoming event

#

# If field mapping is not specified, then field will not appear in the results

# DetectName has been deprecated because CrowdStrike now supports MITRE framework

[DetectionSummaryEvent]

__header.0 = event.Tactic

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

cat = metadata.eventType

sev = event.Severity

usrName = event.UserName

resource = event.ComputerName

domain = event.MachineDomain

description = event.DetectDescription

commandLine = event.CommandLine

fileName = event.FileName

filePath = event.FilePath

sha256 = event.SHA256String

md5 = event.MD5String

url = event.FalconHostLink

src = event.LocalIP

srcMAC = event.MACAddress

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription
```

```
outcome = event.PatternDispositionValue

[DetectionSummaryEvent_DnsRequests]

__header.0 = event.Tactic

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = event.DnsRequests.LoadTime

cat = event.subType

dnsRequestDomain = event.DnsRequests.DomainName

requestType = event.DnsRequests.RequestType

usrName = event.UserName

resource = event.ComputerName

domain = event.MachineDomain

url = event.FalconHostLink

src = event.LocalIP

srcMAC = event.MACAddress

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[DetectionSummaryEvent_NetworkAccesses]

__header.0 = event.Tactic

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = event.NetworkAccesses.AccessTimestamp

cat = event.subType

proto = event.NetworkAccesses.Protocol

dst = event.NetworkAccesses.RemoteAddress

srcPort = event.NetworkAccesses.LocalPort

dstPort = event.NetworkAccesses.RemotePort

connDir = event.NetworkAccesses.ConnectionDirection

usrName = event.UserName

resource = event.ComputerName

domain = event.MachineDomain

url = event.FalconHostLink

src = event.LocalIP

srcMAC = event.MACAddress

tactic = event.Tactic

technique = event.Technique

objective = event.Objective

patternDisposition = event.PatternDispositionDescription

outcome = event.PatternDispositionValue

[DetectionSummaryEvent_DocumentsAccessed]

__header.0 = event.Tactic

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = event.DocumentsAccessed.Timestamp

cat = event.subType

docAccessedFileName = event.DocumentsAccessed.FileName

docAccessedFilePath = event.DocumentsAccessed.FilePath

usrName = event.UserName

resource = event.ComputerName

domain = event.MachineDomain
```

```
url = event.FalconHostLink
src = event.LocalIP
srcMAC = event.MACAddress
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
outcome = event.PatternDispositionValue
[DetectionSummaryEvent_ScanResults]
__header.0 = event.Tactic
devTimeFormat='yyyy-MM-dd HH:mm:ss'
devTime = event.ProcessStartTime
cat = event.subType
scanResultEngine = event.ScanResults.Engine
scanResultName = event.ScanResults.ResultName
scanResultVersion = event.ScanResults.Version
scanResultDetected = event.ScanResults.Detected
usrName = event.UserName
resource = event.ComputerName
domain = event.MachineDomain
url = event.FalconHostLink
src = event.LocalIP
srcMAC = event.MACAddress
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
outcome = event.PatternDispositionValue
[DetectionSummaryEvent_ExecutablesWritten]
__header.0 = event.Tactic
devTimeFormat='yyyy-MM-dd HH:mm:ss'
devTime = event.ExecutablesWritten.Timestamp
cat = event.subType
exeWrittenFileName = event.ExecutablesWritten.FileName
exeWrittenFilePath = event.ExecutablesWritten.FilePath
exeWrittenCompany = event.ExecutablesWritten.Company
usrName = event.UserName
resource = event.ComputerName
domain = event.MachineDomain
url = event.FalconHostLink
src = event.LocalIP
srcMAC = event.MACAddress
tactic = event.Tactic
technique = event.Technique
objective = event.Objective
patternDisposition = event.PatternDispositionDescription
outcome = event.PatternDispositionValue
[LoginAuditEvent]
__header.0 = event.OperationName
devTimeFormat='yyyy-MM-dd HH:mm:ss'
```

```
devTime = event.LoginTime

cat = metadata.eventType

usrName = event.UserId

src = event.UserIp

result = event.Success

[UserActivityAuditEvent]

__header.0 = event.OperationName

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

cat = metadata.eventType

usrName = event.UserId

src = event.UserIp

serviceName = event.ServiceName

success = event.Success

[AuthActivityAuditEvent]

__header.0 = event.OperationName

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

cat = metadata.eventType

usrName = event.UserId

src = event.UserIp

serviceName = event.ServiceName

success = event.Success

[CustomerIOCEvent]

__header.0 = 'Indicator of Compromise'

cat = metadata.eventType

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

commandLine = event.CommandLine

resource = event.ComputerName

fileName = event.FileName

filePath = event.FilePath

dnsRequestDomain = event.DomainName

dstIPv4 = event.IPv4

dstIPv6 = event.IPv6

md5 = event.MD5String

sha1 = event.SHA1String

sha256 = event.SHA256String

[RemoteResponseSessionStartEvent]

__header.0 = 'Remote Response Session Start event'

cat = metadata.eventType

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

usrName = event.UserName

[RemoteResponseSessionEndEvent]

__header.0 = 'Remote Response Session End event'

cat = metadata.eventType

devTimeFormat='yyyy-MM-dd HH:mm:ss'

devTime = metadata.eventCreationTime

usrName = event.UserName
```