



CrowdStrike Falcon Endpoint Extension for QRadar

Installation and Configure Guide



Overview

This document describes how to install the CrowdStrike Falcon Endpoint app on the QRadar platform and how to use it. The CrowdStrike app (also referred to as an extension) on the QRadar platform enables the following capabilities:

- Ingest and view Detections from the CrowdStrike event streams API.
- Upload Custom IOCs to your watchlists.
- Network Contain Devices from the event details screen.
- Direct Links to the Falcon Platform to remediate further.
- Each CrowdStrike cloud can support multiple CIDs.

This extension facilitates establishing a connection to the CrowdStrike Event Streams API to receive event data and send it in QRadar for further analysis, tracking and logging. It is an upgrade to the existing application found here.

(<https://exchange.xforce.ibmcloud.com/hub/extension/fb00d5aa86cc7b5322e39d182d0445ca>)

The major differences for the v1 Version vs the v1.0.2 version are:

	V1.0.2	V1
API Credentials	OAuth2 Only	Legacy & Oauth2
Cloud Environments	US Commercial US Commercial 2 US GovCloud EU Cloud	US Commercial US GovCloud EU Cloud
Data Feed URLs	Multiple	Single
Multiple CIDs	Yes	No

Multitenancy – This extension ingests from a singular log source. You will need to extract custom properties from the events to perform domain separation. Covered here:

(https://www.ibm.com/support/knowledgecenter/SS42VS_7.3.2/com.ibm.qradar.doc/c_qradar_adm_tenant_domain_segment.html)



Contents:

Overview.....	2
Contents:	3
Getting Started	4
Enable Access to the Event Streams API	4
Proxy Considerations	6
QRadar Architecture	6
Initial Installation / Re-Installation / Manual Update:	8
QRadar Console and App Hosts	8
Application Configuration	11
Proxy Configuration (Optional).....	11
Cloud Setup.....	12
Usage.....	13
Usage: Event Details.....	14
Logging	17
Understanding the Event Streams API and Offset Values.....	18
The Anatomy of the Offset JSON File	19
Using Custom Offset Values	20
Troubleshooting and Support.....	21
Checking Configuration.....	21
Getting Support.....	22
Initial Deployment	22
Existing Deployment	22



Getting Started

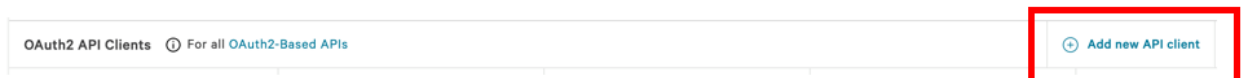
Prior to deploying the CrowdStrike Falcon Endpoint App ensure the following:

1. The latest version of the extension has been downloaded from IBM Security App Exchange
2. A QRadar system (App Host/Console) that the Extension will be deployed to has been identified.
3. An account with proper access to identified QRadar systems is available
4. Properly scoped API credentials have been created and recorded from the Falcon UI
5. (optional) – If the communication between QRadar and the Falcon platform will traverse a proxy server then appropriate configurations should be considered. 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 extension.

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):

The screenshot shows a modal window titled "Add new API client" with a close button (X) in the top right corner. The form contains the following fields:

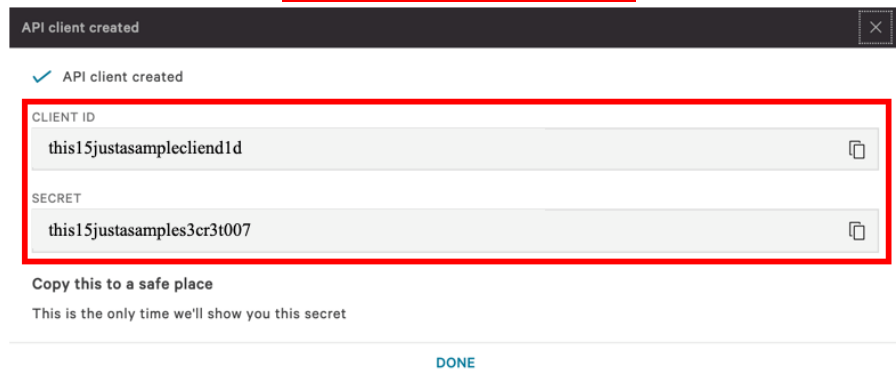
- CLIENT NAME:** A text input field containing "Qradar_App".
- DESCRIPTION:** A text area containing the text "This client is used for the CrowdStrike Falcon Endpoint App that is deployed to our QRadar Instances."
- API SCOPES:** A table with two columns: "Read" and "Write".



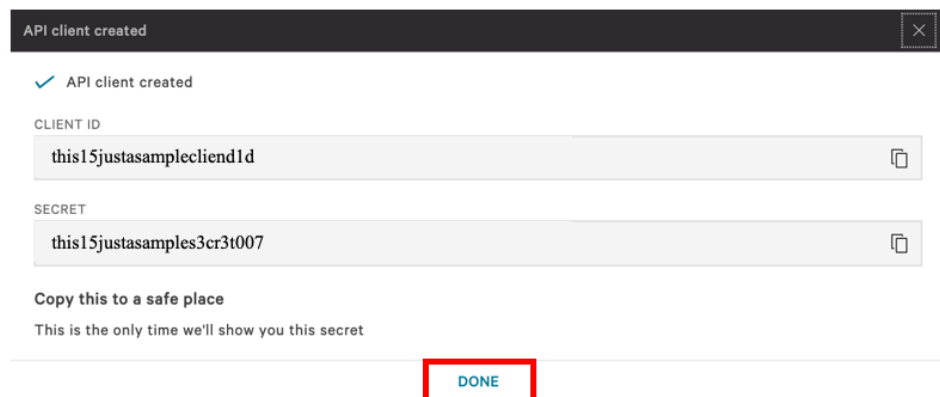
5. Under 'API Scopes' select the 'Read' check box next to the following:
 - a. Detections (Used to get the status of a detection)
 - b. Hosts (Used to get Containment Status)
 - c. Event Streams (Used to ingest Events)
6. Under 'API Scopes' select the 'Write' check box next to the following:
 - a. Detections (Used to Write the Status of a Detection)
 - b. Hosts (Used to Contain/Lift Containment)
 - c. IOCs (Indicators of Compromise)
7. Click 'ADD' to create the client:



8. 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



9. Once the credentials have successfully be copied to a safe and secure location click 'DONE' to close the window:





Proxy Considerations

The CrowdStrike Falcon Endpoint App 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 QRadar Console/App Host. 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

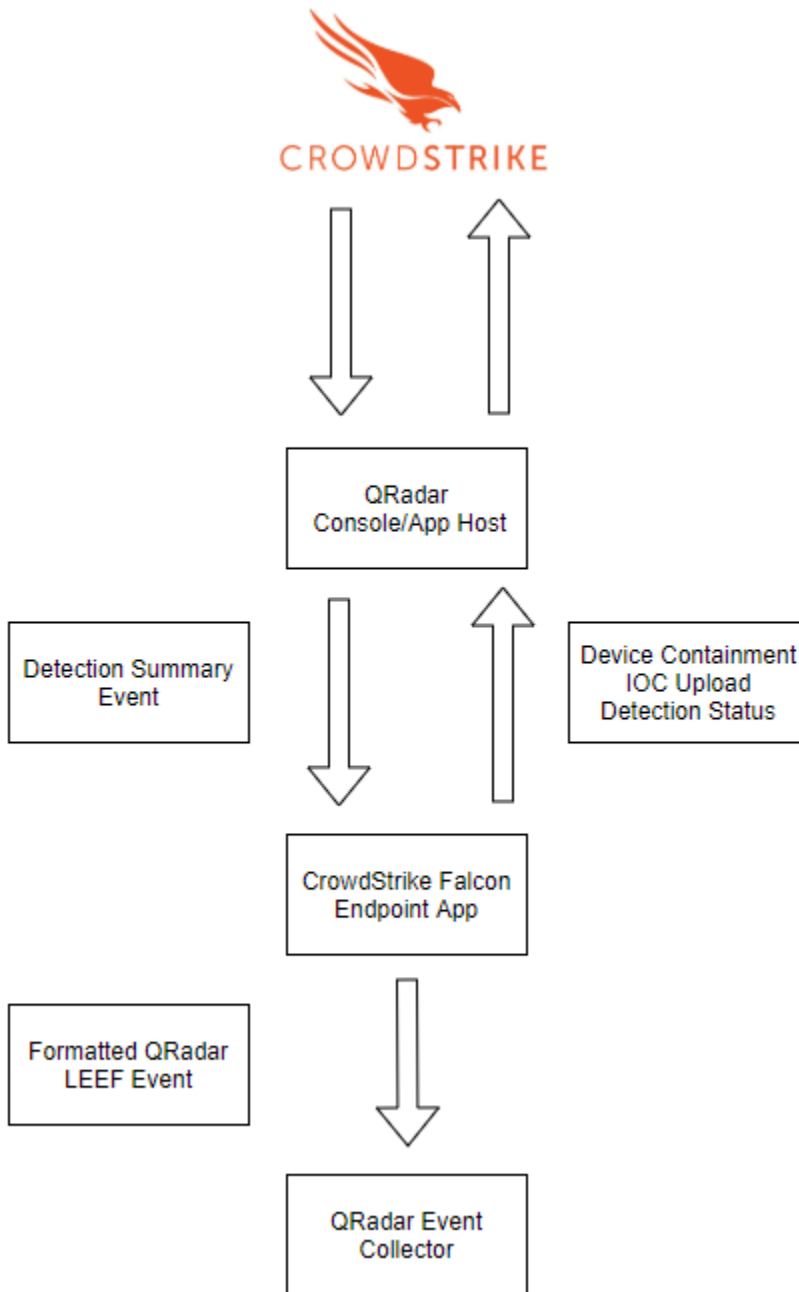
QRadar Architecture

QRadar Console: The QRadar Console provides the QRadar product interface, real-time event and flow views, reports, offenses, asset information, and administrative functions. In distributed environments, the QRadar Console is used to manage the other components in the deployment. If your environment is an "All In One" Appliance, install the extension to this machine.

QRadar App Host: An App Host is a managed host that is dedicated to running apps. App Hosts provide extra storage, memory, and CPU resources for your apps without impacting the processing capacity of your QRadar Console. The extension can be installed here to alleviate ingestion issues and is recommended for customers with multiple CIDs.



The following diagram shows the flow of data from the CrowdStrike API and the Falcon Endpoint APP configuration within an example QRadar deployment.

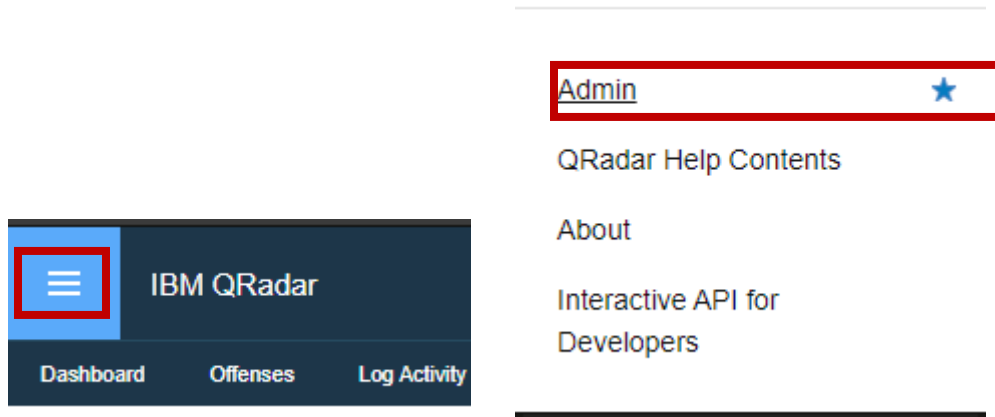




Initial Installation / Re-Installation / Manual Update: QRadar Console and App Hosts

PERFORMING THIS ACTION WILL RESET THE INDEX CAUSING THE INGESTION OF HISTORICAL EVENTS

1. From the QRadar home page or dropdown menu select 'Admin':



2. From the Admin menu select 'Extensions Management'

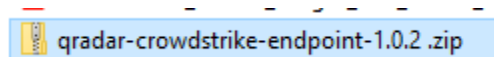




3. From the 'Extensions Management' window, select 'Add', then Browse for the Extension Zip File.

The screenshot shows a dialog box titled "Add a New Extension". Under the heading "From local storage:", there is a text input field containing the filename "qradar-crowdstrike-endpoint-1.0.2 .zip". To the right of this field is a "Browse" button, which is highlighted with a red rectangular box. Below the input field is a checked checkbox labeled "Install immediately". At the bottom of the dialog, there are two buttons: a blue "Add" button and a white "Cancel" button.

4. Select the downloaded Falcon Endpoint App file



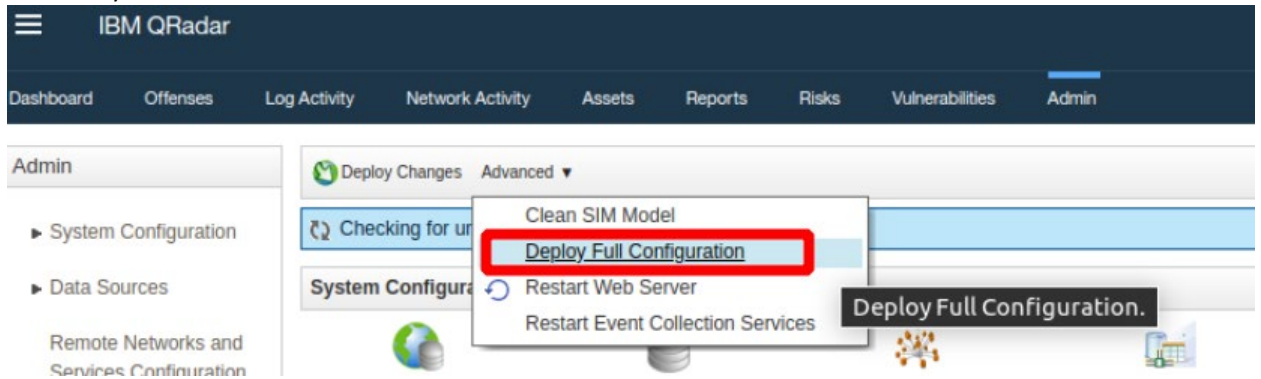
5. Once the file is selected click 'Add'

The screenshot shows the same "Add a New Extension" dialog box as in the previous image. In this view, the "Add" button at the bottom is highlighted with a red rectangular box, indicating it should be clicked.

6. Select 'Install'
7. To Remove the extension, select uninstall from the extensions management window. Note that upon reinstall historical data will be ingested again.
8. Note: If you are installing the app on QRadar version 7.4.0, you need to perform a 'Deploy Full Configuration' after the app is successfully installed. This is a known issue in



QRadar version 7.4.0. (This extra step is not required if you are using another version of QRadar.)



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



Application Configuration

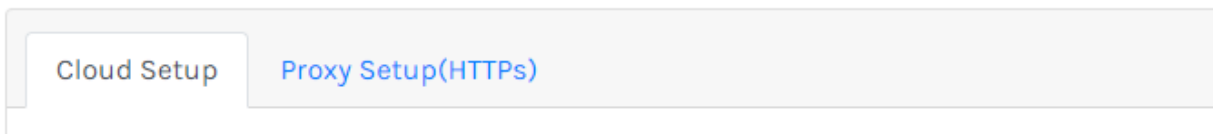
This Extension only supports connections to the OAuth2 based APIs.

1. From the Admin Menu, select apps and then 'Configure CrowdStrike Falcon Endpoint Integration'



2. There are two sub menus within the extension: 'Cloud Setup, and 'Proxy Setup (HTTPs)'

CrowdStrike Falcon Endpoint Configuration



Proxy Configuration (Optional)

Select the 'Proxy Setup' - Check the 'Use Proxy' checkbox, enter the proxy host name/imp, the proxy port and the credentials to allow communication if needed.

The screenshot shows the 'Proxy Setup(HTTPs)' configuration form. It has two tabs: 'Cloud Setup' and 'Proxy Setup(HTTPs)'. The 'Proxy Setup(HTTPs)' tab is selected. The form contains the following fields:

- Use Proxy
- Proxy Authentication
- Proxy IP: nyoutboundproxy.com
- Proxy Port: 8080
- Proxy User: qradar-user
- Proxy Password:



Cloud Setup

This extension only supports connections to the OAuth2 based APIs.

1. Select the appropriate cloud on the 'Cloud Setup Screen' and click Use Cloud
 - i. Endpoint API Host URL and APP ID are pre populated. If you would like to change the APP ID, ensure it is less than 32 characters and with no special characters.
2. Enter your Client ID, Client Secret. The client Description is included in events to assist with domain segmentation and multitenancy.

The screenshot shows the 'Cloud Setup' interface. At the top, there are two tabs: 'Cloud Setup' (selected) and 'Proxy Setup(HTTPs)'. On the left, a list of cloud options is shown: 'US Commercial Cloud' (highlighted in blue), 'Falcon on GovCloud', 'EU Cloud', and 'US Commercial Cloud 2'. To the right of this list, there is a checkbox labeled 'Use Cloud' which is checked. Below the checkbox, there are four input fields: 'Endpoint API Host URL' (pre-filled with 'https://api.crowdstrike.com'), 'Client ID' (pre-filled with 'thisistheclient'), 'Client Secret' (pre-filled with '*****'), and 'Client Description' (pre-filled with 'Account1'). At the bottom right of the form area, there is a blue button labeled 'Add Client'.

3. Click Save
4. Repeat Process for any additional clients you would like to include by pressing 'Add Client'
5. Events will now begin to ingest to 'Log Activity'

This concludes the Application Configuration process



Usage

After configuration is complete, the CrowdStrike Endpoint app will start ingesting events from the API and displaying them as QRadar events. Navigate to the Log Activity tab and filter the log source to show entries from “CrowdStrike Detection”.

To apply a filter – Click Add Filter, select Log Source [Indexed] and select CrowdStrike Detection.

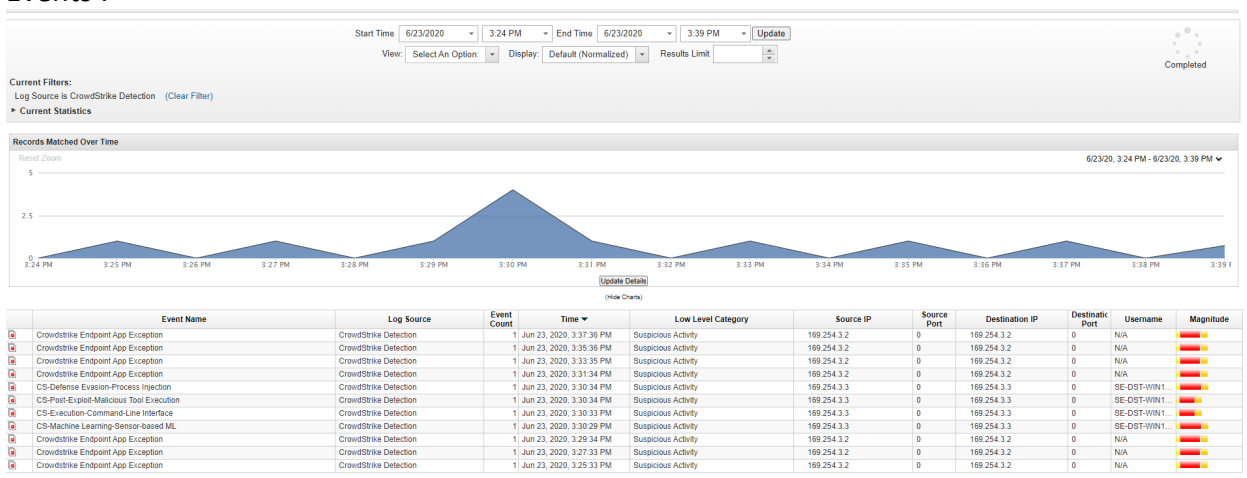
Add Filter

Parameter: Operator: Value:

Log Source Filter:

Log Source:

Once the filter is added, CrowdStrike detections will be listed as events after providing the time range in the View. Click View to choose various time range, by default it will be set to ‘Real Time Events’.




At install, historical events will be ingested based on your company retention. Once historical events are ingested, real time ingestion of events will begin and continue.



Usage: Event Details

1. Double-click on the logged event to see all the fields related to the event.

Event Information								
Event Name	CS-Execution-Command-Line Interface							
Low Level Category	Suspicious Activity							
Event Description	CrowdStrike Execution & Command-Line Interface							
Magnitude		(5)	Relevance	1	Severity	9	Credibility	5
Username	N/A							
Start Time	Jun 23, 2020, 3:43:49 PM	Storage Time	Jun 23, 2020, 3:43:49 PM	Log Source Time	Jun 23, 2020, 3:43:56 PM			
Agent ID (custom)	N/A							
Command Line (custom)	mysqldump --all-databases -u root -pPassword123!							
Computer Name (custom)	cs-se-djs-u14bh-bl							
Containment Status (custom)	normal							

Each Detection has following custom fields:

- a. Agent Id String
- b. Computer Name
- c. Detect Description
- d. Detect Name
- e. Detect Id
- f. Tactic
- g. Technique
- h. Objective
- i. Pattern Disposition Description
- j. Pattern Disposition Value
- k. File Name
- l. File Path
- m. IOC Value
- n. IOC Type
- o. MD5 String
- p. Machine Domain
- q. Parent Process Id
- r. Process Id
- s. Process Start Time
- t. Process End Time
- u. SHA256 String
- v. Severity Time
- w. Sensor Id
- x. Username
- y. Containment Status
- z. Command Line
- aa. Cloud URL
- bb. Client Id



- cc. Client Description
- dd. Falcon Host Link

2. There are 3 right click actions available. To access, mouseover the field value and right click.

- a. **Falcon Host Link:** Visit the Falcon UI for this detection Falcon Host Link
 - i. Action: Opens the CrowdStrike Falcon host url for the current detection.
 - ii. Location: Custom property value for Falcon Host Link
 - iii. Operation: Right Click
 - iv. Required Capabilities: None

FalconHost Link (custom)	https://falcon.crowdstrike.com/detection/7c0ef852f73847f757488d0a399943d0/2319294306354?_cid=5ddb0407bef249c19c7a9	Open CrowdStrike FalconHost URL
--------------------------	--	---------------------------------

- b. **Detection ID:** Update the Detection Status in CrowdStrike
 - i. Choose from The Options and Click 'Update CS Detection Status'
 - ii. Action: Opens a page to update the detection status of the detection to CrowdStrike Falcon server.
 - iii. Location: Custom property value for Detect ID
 - iv. Operation: Right Click
Required Capabilities: Admin



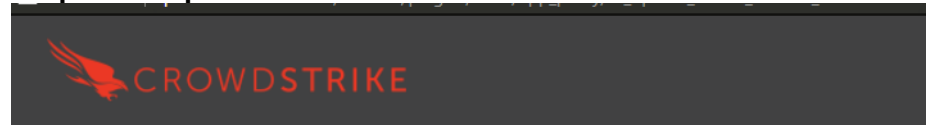
CrowdStrike Detection Status Update

True Positive

- c. **Sensor ID:** Check Containment Status for a device and Contain it.
 - i. Action: Opens a page to update the containment status of the detection to CrowdStrike Falcon server.
 - ii. Location: Custom property value for Sensor ID
 - iii. Operation: Right Click



iv. **Required Capabilities: Admin**



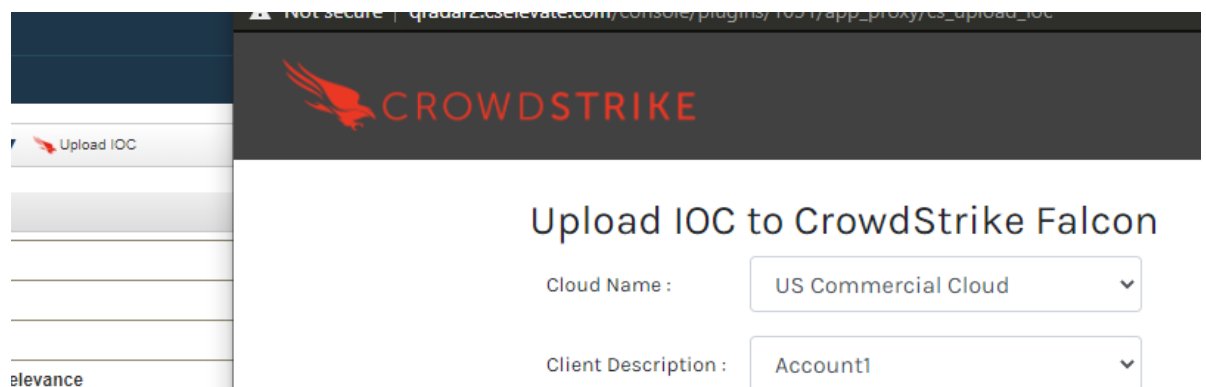
CrowdStrike Device Contain Status Update

Current Device Contain Status: **normal**

Host Name: **cs-se-djs-u14bh-bl**

Select

- d. The Upload IOC (Indicator Of Compromise) button is visible on the toolbar to upload an IoC.
 - i. Action: Create a new IoC and upload it to CrowdStrike Falcon server
 - ii. Location: Event Details Toolbar
 - iii. Operation: Click
 - Required Capabilities: Admin




- iv. Cloud Name: Describes the CS Cloud to upload the IOC to
- v. Client Description: Describes the Client Credentials for the selected cloud.
- vi. Type: Describes the type of Indicator
- vii. Policy: Describes the action when this indicator is found.
- viii. Value: Value of the Indicator
- ix. Description: Description of the Indicator



Logging

1. During Normal Operation you will see events flow into qradar on the 'Log Activity' Screen. When there are any Application Errors, you will see this as part of the logging as well with the event "CrowdStrike Endpoint App Exception" The payload information in the event details will show the specific error.

	Event Name
	Crowdstrike Endpoint App Exception

Payload Information

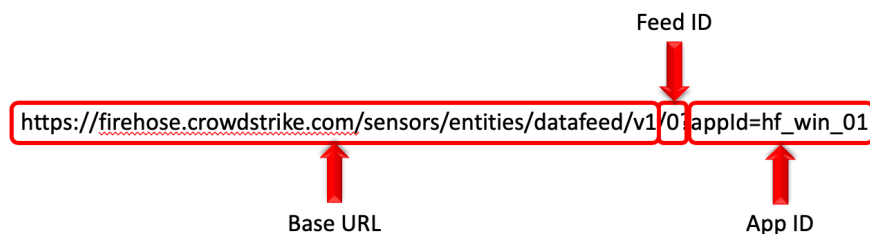
```
utf hex base64  
 Wrap Text  
Jun 26 15:39:18 CrowdStrikeDetection LEEF:1.0|IBM|CrowdStrikeDetection|1.0|2000007|ExceptionDetails=Error when getting JWT token: 'access_token'
```



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 Client configuration

The extension will examine the API response to determine the number of URL feeds and attempt to create and maintain an independent connect to each one.

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 and is logged in the main application logging within the apps docker container.

If the network connection is disrupted the extension will leverage this information as the marker to determine the last event processed. Since the extension can support multiple clients it uses the description of the client as the unique identifier and then relates the data feed URL and offset values with it.

This information is then stored by the extension:

- The ‘`cs_config.json`’ folder within the App Extension Itself.
 - From Within the Docker Container at `/store/cs_config.json`. See the troubleshooting guide for more info on connecting to the docker container.



The Anatomy of the Offset JSON File

The Offset JSON files are stored in the docker container at `/store/cs_config.json`. At times these may need to be updated. This is usually the case during a fresh install of the application as the offsets are reset to zero. When upgrading from an old version of this app you may want to view your current offset so that you can resume your stream at the correct spot. The JSON file will have the data feed URLs and offset values that have been associated with that Client. To change these offset values, connect to the docker container using the method outlined in the troubleshooting guide. Then Follow these steps.

1. Open the file `/store/cs_config.json` with a file editor.
 - a. `vi /store/cs_config.json`
2. Find the Key `stream_status`
3. Edit the `next_offset` for your desired stream.
 - a. Press `I` to enter edit mode
 - b. Press `ESC` to exit edit mode
 - c. Press `:wq ENTER` to save.
4. Save and restart the docker container.
 - a. `exit`
 - b. `docker restart <<CONTAINER_ID>>`
5. The stream will be started at your selected offset.

```
],
"interval": "2",
"stream_status": {
  "https://api.crowdstrike.com-879c7339ed8a4535bd5bf24e91275745-1": {
    "status": "active",
    "next_offset": 17790836
  }
}
```



Using Custom Offset Values

The extension will search for `/store/cs_config.json` 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 the JSON fails to return a value, the extension will use index zero, ingesting historical events.

1. **Migrate from app deployment (Version 1.0.1)** Check the `/store/cs_config.json` file by connecting to the docker container with instructions found in the troubleshooting guide. Install the new app and enter credentials, modifying the current offset to the stored value from version 1.0.1
2. **Recovery from failure/ Moving extension to a different App Host/ Migrating to a QRadar Deployment** – In the event that the extension is being installed on a new system but there has already been data collection the JSON file should be checked for offsets and these should be migrated over.
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 Client can be created with a modified description, updated AppID and specific offset.



Troubleshooting and Support

CrowdStrike provides support for the extensions 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. Configuration Of Offenses and Domain Separation

[Checking Configuration](#)

Unable to establish connection: (No Events, Logging Errors)

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



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. Repeat and record the action(s) that are associated with the issue you are reporting
5. Check the Troubleshooting Guide to Retrieve All Logs
6. Record the following information about the QRadar system:
 - QRadar environment type
 - QRadar version
 - Extension version
7. Identify the types of networks devices that the connection will traverse and ensure that they have been properly configured
8. Collect API audit logs from the Falcon instance for the time frame when the issue is occurring
9. Navigate to <https://supportportal.crowdstrike.com/>
10. Provide (at a minimum) the information from steps 5-8

Existing Deployment

1. Disable and re-enable Cloud Clients
2. Check the Troubleshooting Guide to Retrieve All Logs
3. Record the following information about the QRadar system:
 - When was the last successful connection?
 - If an Extension or QRadar update performed around the same time frame
 - QRadar environment type
 - QRadar version
 - Extension version
4. Identify the types of networks devices that the connection will traverse and ensure that they are still properly configured
5. Collect API audit logs from the Falcon instance for the time frame when the issue began occurring
6. Navigate to <https://supportportal.crowdstrike.com/>
7. Provide (at a minimum) the information from steps 3-5



The document guides you through the steps to view and export the logs created by the CrowdStrike Qradar extensions. The logs are helpful for debugging and troubleshooting execution of the apps.

Steps to get the logs

1. SSH into the QRadar instance

Syntax:

```
ssh -i <pemfile> <user>@<QRadarMachine>
```

Example:

```
root@qradar731 ~# ssh -i "id_rsa.pem" ec2-user@qradar731.amazonaws.com
Last login: Fri Feb  7 09:32:30 2020 from 10.0.0.1
ec2-user@ip-10-0-0-1: ~]$
ec2-user@ip-10-0-0-1: ~]$
```

2. Search the application ID for CrowdStrike extension

Syntax:

```
sudo /opt/qradar/bin/contentManagement.pl -a search -c 100 -r "CrowdStrike"
```

After running this, application IDs for both Intel and Endpoint will be visible

Example:

```
[root@qradar731 ~]# sudo /opt/qradar/bin/contentManagement.pl -a search -c 100 -r "CrowdStrike"
[INFO] Initializing Content Management Tool...
[INFO] (ContentManagementCLI) Start Time: 2020-04-24 17:01:20
[INFO] Starting search process
[INFO] Search results:
[INFO] - [Id] - [Name] - [Description]
[INFO] - [1153] - [CrowdStrike Falcon Intel] - [Qradar Extension to Ingest CrowdStrike IOCs into Qradar]
[INFO] - [1157] - [CrowdStrike Falcon EndPoint] - [Qradar Extension to ingest CrowdStrike Falcon EndPoint detections into Qradar]
[root@qradar731 ~]#
```

3. Find the relevant container where the CrowdStrike extension is running. This can be found by the application IDs of both the apps.

Syntax:

```
sudo docker ps
```

Example:

```
[root@qradar731 ~]# sudo docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED             STATUS
bb104df7c02        qregistry.service.consul:5444/qapp/1157:1.0.2-20200424162135  "sh /secret_env_unwr..." 55 minutes ago    Up 55 m
92-449a-8a9e-0c5068d46303-S0.db0b709a-a46e-42f8-93fc-9bb44ea7408f  "sh /secret_env_unwr..." 4 hours ago       Up 4 h
9c5319b80d9        qregistry.service.consul:5444/qapp/1153:1.0.1-20200424123944  "sh /secret_env_unwr..." 4 hours ago       Up 4 h
92-449a-8a9e-0c5068d46303-S0.d271dcd7-b593-4573-9dc1-1ad1032d79b9  "sh secret_env_unwr..." 5 hours ago       Up 5 h
5ef931e627b0e        docker-registry.service.consul:15443/qaauth:0.0.16           "sh secret_env_unwr..." 5 hours ago       Up 5 h
92-449a-8a9e-0c5068d46303-S0.de493c4f-2f24-405d-8609-d543eb66f185  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
59b74a0e7708        qregistry.service.consul:5444/qapp/1002:1.0.13-20200323055341  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
92-449a-8a9e-0c5068d46303-S0.b22595a1-a143-4e68-b926-a6ae69d10087  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
8639d9289cf3        qregistry.service.consul:5444/qapp/1001:1.1.1-20200323055040  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
92-449a-8a9e-0c5068d46303-S0.b0792096-81ad-475b-81d1-7a7d8aa72c4b  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
6ecfed71db89        docker-registry.service.consul:15443/qbert/mesos-consul:0.4.3  "/bin/mesos-consul..." 5 hours ago       Up 5 h
92-449a-8a9e-0c5068d46303-S0.9c0381c0-381c-46ea-bc79-fd4d35567aa5  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
80772f4053be        docker-registry.service.consul:15443/qbert/nginx-consul:2.4   "/scripts/launch.sh"    5 hours ago       Up 5 h
pps_proxy
53e35d1082e        docker-registry.service.consul:15443/qbert/nginx-consul:2.4   "/scripts/launch.sh"    5 hours ago       Up 5 h
7951c342c88f        docker-registry.service.consul:15443/qbert/service-launcher:0.1.9  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
92-449a-8a9e-0c5068d46303-S0.77469968-d02c-4463-943c-be765c10cb99  "sh /secret_env_unwr..." 5 hours ago       Up 5 h
87acfaf8a995        qregistry:1.7.0         "sh /qregistry/boo..." 6 hours ago       Up 5 h
```

4. *Bash* into the container using the 'CONTAINER ID'

Syntax:



sudo docker exec -it <app-container-id> /bin/bash

Example:

```
[ec2-user@ip-10.10.10.10 ~]$ sudo docker exec -it 822f13d424df /bin/bash
bash-4.1#
bash-4.1#
bash-4.1#
```

5. Locate the logs created by the CrowdStrike extension

Syntax:

ls store/log/

Example:

```
[root@7bb104df7c02 /]# ls store/log
app.log  poll.log  startup.log  supervisord.log
```

6. View the content of the log with `less` command. The content can be navigated with up/down keys. Press 'Q' key to exit the display

Syntax:

less store/log/app.log

Example:

```
2020-02-06 05:30:08,669 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Received 1000 events
2020-02-06 05:30:08,669 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Sending 1000 events to the QRadar console at
2020-02-06 05:30:08,802 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] More events are available. Fetching...
2020-02-06 05:30:08,802 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Getting data by pagination. URL is https://expander.expance.co/apl/v1/behavior/risky-flows?filter(created_after)=2019-12-06T23:59:59Z&filter(created_before)=2020-02-04T23:59:59Z&page[offset]=0000&page[limit]=1000
2020-02-06 05:30:10,623 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Received 1000 events
2020-02-06 05:30:10,623 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Sending 1000 events to the QRadar console at
2020-02-06 05:30:10,729 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] More events are available. Fetching...
2020-02-06 05:30:10,730 [abstract_qpylib.log] [MainThread] [INFO] - 127.0.0.1 [APP_ID|1010][NOT:0000000000] Getting data by pagination. URL is https://expander.expance.co/apl/v1/behavior/risky-flows?filter(store/log/app.log)
```

7. Exit from the app container

Syntax:

exit

Example:

```
[root@7bb104df7c02 /]# exit
exit
```

8. Export the log files from app container to the host

Syntax:

sudo docker cp <app-container-id>:/store/log/ <target-directory>

Example:

```
[root@qradar731 /]# sudo docker cp 7bb104df7c02:/store/log/ /home/qradar/CrowdStrike-Qradar-Logs/
You have new mail in /var/spool/mail/root
[root@qradar731 /]# cd /home/qradar/CrowdStrike-Qradar-Logs/
[root@qradar731 CrowdStrike-Qradar-Logs]# ls
log
[root@qradar731 CrowdStrike-Qradar-Logs]# cd log
[root@qradar731 log]# ls
app.log  poll.log  startup.log  supervisord.log
[root@qradar731 log]#
```