

# Fusion SOAR

Last updated: Jul. 18, 2025

## Overview

Falcon Fusion SOAR workflows help streamline analyst workflows by automating actions around specific and complex scenarios. You can create workflows to precisely define the actions you want Falcon to perform in response to incidents, detections, policies, cloud security findings, and more.

## Requirements

- **Subscriptions:**

- Workflow triggers are available for these subscriptions:
  - Falcon Next-Gen SIEM or Falcon Next-Gen SIEM 10GB
  - Falcon Insight XDR
  - Falcon Insight for ChromeOS
  - Falcon Prevent
  - Falcon Data Protection
  - Falcon Horizon
  - Falcon Cloud Workload Protection (CWP)
  - Falcon Discover
  - Falcon Exposure Management
  - Falcon FileVantage
  - Falcon Spotlight
  - Falcon Identity Threat Detection
  - Falcon Identity Threat Protection
  - Falcon for Mobile
  - Falcon Complete
  - Falcon OverWatch Elite
  - Falcon OverWatch (US-GOV-1 only)
- For actions based on event queries, you need these items:
  - Falcon Insight XDR subscription in a CID in the US-1, US-2, EU-1, or US-GOV-1 CrowdStrike cloud
  - Optional. To access data from a particular Falcon data source, you also need subscriptions for one or both of the following data sources on the CID:
    - Falcon for IT
    - Falcon Forensics
- For the **Write to log repo** action, which saves output, you need a Falcon Insight XDR subscription in a CID in the US-1, US-2, or EU-1 CrowdStrike cloud
- For the **HTTP Request** action, which makes on-demand API calls, you need a Falcon Next-Gen SIEM subscription.
- For actions that invoke on-premises APIs, you need either the Falcon Next-Gen SIEM or Falcon Next-Gen SIEM 10GB subscription on any supported 64-bit version of Windows or any supported version of macOS or Linux.

**Note:** This ability is not available for the US-GOV-1 or US-GOV-2 clouds. Also, it requires a host group to be configured to execute on-premises API requests. The hosts in the host group cannot be pods, containers, or 32-bit platforms.

- **Sensor support:**

- Windows
- macOS
- Linux
- Android
- iOS

- **Roles:**

- These roles can view **Workflows** pages, create and edit workflows, and save output:
  - Falcon Administrator
  - Workflow Author

- These roles can view the **Workflows** page:
  - Falcon Security Lead
  - Falcon Investigator
  - Falcon Analyst
  - Falcon Analyst - Read only
  - Vulnerability Manager
  - Exposure Management Manager
  - Exposure Assets Admin
  - Exposure Assets Analyst
- These roles can view, create, edit, and save workflows that have triggers based on inbound webhooks:
  - Falcon Administrator
  - Workflow Author

**Important:** To enable, disable, and run workflows that have triggers based on inbound webhooks, you must have the Falcon Next-Gen SIEM subscription. The Falcon Administrator role can then enable, disable, and run these workflows.

If you're creating custom roles, here is the permission required to enable, disable, and run these workflows:

csrn:workFlow:webhook-trigger START

- These roles can create event queries for the indicated data source:
  - Any data source: Falcon Administrator
  - Next-Gen SIEM data sources: XDR Administrator
  - The IT data source: IT Administrator
  - The Forensics data source: Forensic Investigator
- These roles can create and overwrite lookup files that are used with event queries:
  - Falcon Administrator
  - App Developer
- These roles can view all content in the content library:
  - Most roles

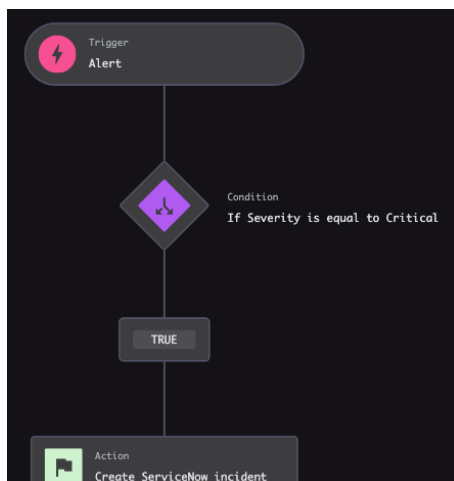
## Understand Fusion SOAR workflows

Workflows can help streamline and automate processes for several Falcon products. For example:

- Send out Slack notifications based on cloud container image assessment findings.
- Generate ServiceNow incident tickets when high-severity vulnerabilities are detected.
- Triage new endpoints seen in your environment.
- Run real time response (RTR) scripts to manage affected hosts when incidents are reported.
- Notify a user and request validation before continuing with an action.
- Schedule a workflow to perform an action regularly.

Workflows are built around a *trigger-condition-action* model. For example, a workflow that automatically generates ServiceNow incidents when critical detections are reported consists of these three components:

- **Trigger:** Falcon reports an endpoint detection.
- **Condition:** Detection has a severity of **Critical**.
- **Action:** Create a ServiceNow incident.



You can add familiar logic elements to build sophisticated workflows:

- **Else and Else if statements:** Conditional statements allow for multiple workflow branches
- **Else loops:** Conditional loops allow for multiple looping options
- **AND statements and OR statements:** Group multiple conditions within a single workflow branch
- **Parallel actions, conditions, and loops:** Create independent branches of actions, conditions, or loops within a single workflow
- **Sequential actions and loops:** Add actions or loops to execute in a specific order within a workflow branch

## Workflow triggers

Each workflow begins with a trigger.

For more info, see these topics:

- [Trigger types](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#vc13efaa]
- [Available triggers](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#n381e0e5]
- [Custom triggers based on inbound webhooks](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#mf5cdeeb]

### Trigger types

These are the types of triggers:

- **Event:** The workflow is triggered by an event in the Falcon environment, such as a new incident.
- **Schedule:** The workflow is triggered regularly, based on a defined schedule, such as hourly, daily, weekly, or monthly.  
Also, you can run this type of workflow immediately through the Falcon console or an API call, as shown in [Run a workflow on demand](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#xa3f58bb].
- **On demand:** The workflow is triggered directly through the Falcon console, by another workflow, or by an API call, as shown in [Run a workflow on demand](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#xa3f58bb].
- **Inbound webhook:** The workflow generates a unique webhook URL for the given workflow and is then triggered by an external system that uses that URL.  
For more info, see [Custom triggers based on inbound webhooks](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#mf5cdeeb] and [Create and manage triggers based on inbound webhooks](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#scedcaa6].

**Tip:** Control a workflow's behavior by adding one or more conditions. For more info, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].

**Note:** In Falcon Flight Control environments, you can create a workflow for a parent CID and apply it to child CIDs. However, the child CIDs can only access triggers that are available to the parent CID. A warning shows when a trigger is unavailable to a child CID.

### Available triggers

The event triggers available depend on your subscriptions.

You can look for triggers in the content library as explained in

[Library of actions, apps, Foundry app templates, playbooks, and triggers](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#qd6417a6]

or read this overview of the trigger categories.

- **3PI Data Connection:** For more info, see [Data Connectors](#) [/documentation/page/a76b8289/data-connectors].
- **Alert:** Falcon reports on these detection types:
  - **EPP Detection:** For more info, see [Endpoint Detection Monitoring \(New\)](#) [/documentation/page/nec99068/endpoint-detection-monitoring].
  - **Identity Detection:** For more info, see [Identity-based Incidents, Detections, and Risks](#) [/documentation/page/a78910d1/identity-based-incidents-detections-and-risks].
  - **Mobile detection:** For more info, see [Configuring Falcon for Mobile](#) [/documentation/page/dff3f04/configuring-falcon-for-mobile].
  - **Next-Gen SIEM Detection:** For more info, see [Detection Monitoring](#) [/documentation/page/ke886083/unified-detections-monitoring].
  - **Next-Gen SIEM Incident:** For more info, see [Incident Investigation](#) [/documentation/page/r2f1bac9/xdr-incident-investigation].
  - **Third Party Detection:** For more info, see [Third-party detections](#) [/documentation/page/ke886083/unified-detections-monitoring#y1bee1ed].
  - **Data Protection Detection:** For more info, see [Falcon Data Protection](#) [/documentation/page/o242b094/falcon-data-protection].
- **Asset management:** Falcon reports a change to an asset, including new assets seen in your environment, applications being installed or uninstalled, system resources reaching certain levels, and more. For more info, see [Asset Management: Assets](#) [/documentation/page/cda0a664/asset-management-assets] or [Asset Management: Applications](#) [/documentation/page/ab0b6dc5/asset-management-applications].
- **Audit event:** Trigger a workflow on any of these changes:
  - Incident
    - Assignment
    - Status
    - Comment
    - Tag
  - Policy (Prevention, Firewall, Sensor Update, Device Control, Response, Mobile, Identity Protection, and Airlock policies)

- Deleted
- Created
- Enabled
- Disabled
- Updated

○ Host

- Host unhidden
- Host containment lifted
- Host hidden
- Request to lift containment
- Containment requested
- Host contained

**Note:** If you include the Host Group attribute in a notification generated with one of these triggers, the Host Group ID appears in the notification, rather than the Host Group name.

○ Alert

- Tag
- Comment
- Parent incident
- Assignment
- Status
- Severity

○ General settings

- Channel file update controls

○ RTR Session

- Session end
- Session start

○ XIoT event

- Collection
- Schedule

- **Cloud security assessment:** Falcon reports a new cloud security finding
- **Custom IOA monitor:** Falcon reports a custom IOA detection. For more info, see [Monitor custom IOA detections and preventions](#) [/documentation/page/nec99068/endpoint-detection-monitoring#j9ed459d].
- **FileVantage change:** Falcon reports a file integrity change. For more info, see [Falcon FileVantage](#) [/documentation/page/ca042527/falcon-filevantage].
- **Host state:** Trigger a workflow when a host enters one of these states:
  - **Provisioning**
    - **Host fully provisioned:** When a host has downloaded all available channel files
  - **Support**
    - **Host in Reduced Functionality Mode (RFM):** When a sensor has entered RFM
  - **Visibility**
    - **Host connect:** When a host has connected to the cloud
    - **Host first seen:** When a host is seen on Falcon for the first time

**Note:** If you include the Host Group attribute in a notification generated with one of these triggers, the Host Group ID appears in the notification, rather than the Host Group name.

- **Identity account event:** Falcon reports an Identity account event, for example, a compromised password.
- **Kubernetes and containers:** Falcon reports a security finding related to your Kubernetes environment or containers in your cloud environment. This includes vulnerabilities and detections identified by the Image Assessment tool, container runtime detections, container drift detections, and detections triggered by Image Assessment prevention policies. For more info, see [Container Security](#) [/documentation/page/aa4fcce/container-security].
- **New incident:** Falcon reports an incident

**Note:** CrowdScore incidents can also be presented as XDR detections. To create a Falcon Fusion SOAR workflow based on CrowdScore incidents, the workflow trigger must be from the CrowdScore incident context, not the XDR detection context. When creating a workflow based on CrowdScore incidents, select a workflow trigger of **New incident**. For more info, see [CrowdScore incidents as Next-Gen SIEM incidents](#) [/documentation/page/dabdbd2a/incident-monitoring#r1e02695].

- **Phishing email:** A user reports a phishing email from a supported email service. For more info, see the CrowdStrike Store app [Email Phishing Connector built for Microsoft 365](#) [/store-y2/6shk72/2d800/358bd0/131ac/232020].

[Email Filtering Connector built for Microsoft 365 \(31018-V2/0000/274005074300007131E7433020\).](#)

- **Vulnerabilities user action:** A vulnerability management user selects **Create ticket** for a vulnerability, host, or remediation. For more info, see [Vulnerability Management Ticketing Workflows](#) [/documentation/page/e552ea54/vulnerability-management-ticketing-workflows].
- **Workflow execution:** Trigger a workflow off of a workflow. This option is useful if you want to set up notifications to know when workflows run or hit a failure point.

**Note:** In Flight Control environments, using this trigger makes the workflow execution customer ID available, which indicates the CID where the workflow executed and can be either the parent CID or the child CID.

- **Zero Trust Assessment:** Trigger a workflow based on Zero Trust Assessment (ZTA), such as a change in a host's assessment score or if a host fails a specific assessment. For more info, see [Zero Trust Assessment](#) [/documentation/page/bc52e49b/zero-trust-assessment].
- **Message Center:** Falcon Message Center has a new case, update, or comment. For more info, see [Message Center](#) [/documentation/page/f80e8fc0/message-center#v3f6c74e].

## Custom triggers based on inbound webhooks

Fusion SOAR provides many triggers to start your workflows. However, you can also create custom triggers that use inbound webhooks to integrate third-party tools, enabling real-time orchestration from those tools. For example, you can define triggers for events from external sources such as SIEMs, SOAR platforms, ticketing systems, and threat intelligence feeds.

For info about requirements for these triggers related to subscriptions, CrowdStrike clouds, and roles, see [Requirements](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#18d760c7].

For info about how to manage these actions, see

[Create and manage triggers based on inbound webhooks](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#scedcaa6].

## Limitations

Be aware of these limitations:

- A workflow can have only one webhook trigger.
- The webhook URLs are auto-generated and cannot be reused or edited.
- The payload must be raw JSON. Form-data, file uploads, and so on are not supported.
- There are no retry attempts for failed requests.
- There is no schema enforcement on incoming payloads.
- There is no support for multiple webhook variants per workflow.
- The maximum payload size is 1 MB.

## Workflow conditions

A condition consists of a *parameter*, an *operator*, and a *value*. A workflow compares an observed parameter to the value based on the operator. For example, a workflow has a condition that evaluates endpoint detection severity:

- **Parameter:** Severity
- **Operator:** Is greater than or equal to
- **Value:** High

If a detection has a severity of **High** or **Critical**, the condition is true and the workflow action delivers a Slack notification. For medium-severity detections, you can create another condition that sends emails instead.

**Note:** A workflow condition evaluates to either true or false. You must define what happens when the condition evaluates to true. However, it's optional to define what happens when the condition evaluates to false. If a workflow reaches a condition that evaluates to false but has nothing defined for that case, the workflow ends.

To set up conditions based on day or time, use time-based parameters. You can always create a time-based condition using the **Workflow execution time** parameter. Additional time-based parameters are available depending on the trigger or action. For example, with triggers, you can use these parameters: **Behavior timestamp**, **Last behavior**, **Start Time**, **End Time**, and **Last Activity**. With actions, examples include these action fields: **VirusTotal Last Analysis Date** and **VirusTotal Creation Date**. These parameters are only some of the available time-based parameters. The **includes** operator is the only operator available with time-based parameters. The workflow runs only on the days you specify. By default, the condition applies to the whole day. To specify a period of less than a day, choose times using the 12-hour clock format with AM or PM. To specify additional time periods, add an ELSE IF condition for each new time period.

For more info, see:

- [Using multiple workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#pf96034d]
- [Conditions reference](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z9bd9cff]
- [Conditions in Advanced mode using CEL expressions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#x6a68fc9]

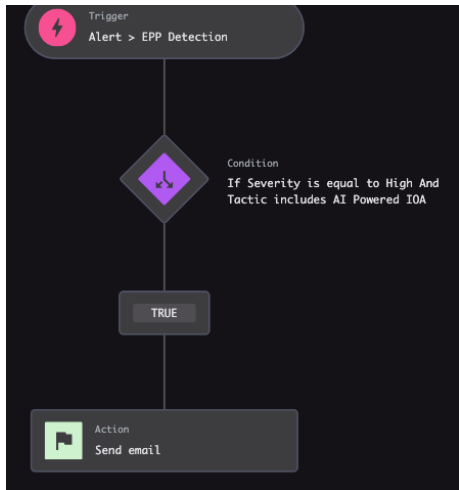
## Using multiple workflow conditions

You can use multiple workflow conditions joined with AND operators or OR operators to create complex logic in your workflows.

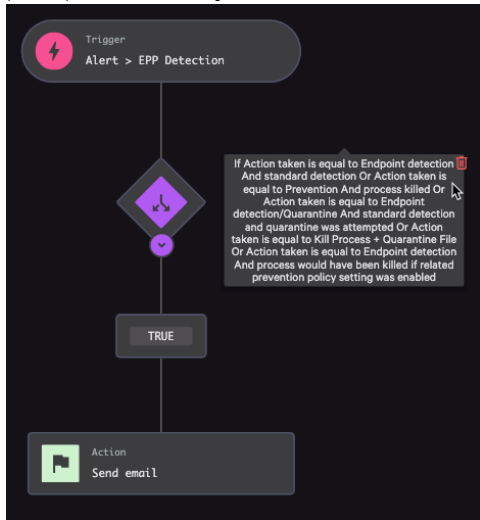
- With AND operators, the overall workflow condition evaluates to true only if all included conditions are true.
- With OR operators, the overall workflow condition evaluates to true if any one or more conditions are true.

For example, this sample workflow sends a notification by email when an endpoint detection is high severity and associated with a specified tactic. Both conditions must evaluate to **true** for the workflow to send the email.





For a more complex example, this sample workflow sends a notification by email when any one of several detections or preventions take place in Falcon Prevent: a detection occurs, a prevention kills a process, a file is quarantined, a combined action of killing a process and quarantining a file, or a process would have been killed but a policy setting was not enabled. You could create individual workflows or ELSE IF operators for each of these items, but using an OR operator can be simpler for your analysts to understand at a glance.



Logical expression	Workflow representation	Description
(hosttag == "critical system") OR (Severity == critical)	(Parameter: Hosttag Operator: equals Value: Critical System) OR (Parameter: Severity Operator: equals Value: Critical)	If the first condition evaluates to <b>true</b> OR the second condition evaluates to <b>true</b> , then the condition block evaluates to <b>true</b> , and the workflow proceeds along the THEN branch. If neither condition evaluates to <b>true</b> , then the condition block evaluates to <b>false</b> , and the workflow proceeds from the ELSE branch, if any.
(Severity >= high AND hostgroup == server) OR Severity == critical	(Parameter: Severity Operator: greater than or equal to Value: High AND Parameter: Hostgroup Operator: equals Value: server) OR (Parameter: severity Operator: equals Value: Critical)	If both severity and hostgroup conditions evaluate to <b>true</b> , OR if the second severity condition evaluates to <b>true</b> , then the workflow proceeds along the THEN branch.
hosttag == "critical system" OR Severity == critical OR (Severity >= high AND hostgroup == server)	(Parameter: Hosttag Operator: equal to Value: critical system) OR (Parameter: Severity Operator: equals to Value: Critical) OR (Parameter: Severity Operator: greater than or equal to Value: High AND Parameter: Hostgroup Operator: equals to	If any of the three conditions evaluates to <b>true</b> , then the workflow proceeds along the THEN branch. The third condition combines two conditions with an AND operator, so both must be true for the combined condition to evaluate to <b>true</b> .

	<div>Operator: equal to Value: server )</div>	
<div>username == exampleUser AND (FilePath matches "*Documents" OR ParentProcessFilePath matches "exampleFilePath")</div>	<div>(Parameter: Username Operator: equal to Value: exampleUser) AND (Parameter: FilePath Operator: matches Value: *Documents) OR (Parameter: Username Operator: equal to Value: exampleUser) AND (Parameter: ParentProcessFilePath Operator: matches Value: exampleFilePath)</div>	<div>Combine two expressions with OR to evaluate expressions like A AND (B OR C). If either expression evaluates to <b>true</b>, the workflow proceeds along the THEN branch.</div> <div><div>Condition</div><div><div>IF</div><div>User name is equal to exampleUser</div><div></div></div><div><div>AND</div><div>File path matches *Documents</div><div></div></div><div><div>+</div>Add condition line</div><div><div></div>Delete group</div><div><div>OR</div><div>User name is equal to exampleUser</div><div></div></div><div><div>AND</div><div>Parent process file path is equal to exampleFilePath</div><div></div></div><div><div>+</div>Add condition line</div><div><div></div>Delete group</div><div><div>+</div>Add OR condition</div><div><div>Cancel</div><div>Next</div></div></div>

Conditions reference

Conditions include several operators that vary depending on the selected parameter:

- [is equal to \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y64852b0\]](#)
- [is not equal to \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#rd58e247\]](#)
- [is greater than \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#h0deca01\]](#)
- [is less than \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ze0006c7\]](#)
- [is greater than or equal to \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#qa4781d8\]](#)
- [is less than or equal to \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ia47a26\]](#)
- [exists \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#f719c149\]](#)
- [does not exist \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#hb3cb01a\]](#)
- [includes \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p1a252f6\]](#)
- [does not include \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#s9667be5\]](#)
- [matches \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ja2b1035\]](#)
- [does not match \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#dd10c950\]](#)

is equal to

Data types	Description	Example
<div><ul style="list-style-type: none"><li>Strings (case-sensitive)</li><li>Whole numbers</li><li>Categorical data, such as Technique, Platform, and Severity</li><li>Populated data, such as Hostnames and Platform</li></ul></div>	<div>True when the observed parameter is exactly the same as the value you provide</div>	<div>To only process endpoint detections where an IOC Type is an MD5 hash, configure the condition with these settings: <b>Parameter: IOC Type</b> <b>Operator: is equal to</b> <b>Value: MD5 hash</b></div> <div><div>Parameter</div><div>If</div><div>IOC Type</div><div>Operator</div><div>is equal to</div><div>Value</div><div>MD5 hash</div><div>Select a value</div></div>

is not equal to

Data types	Description	Example
<div><ul style="list-style-type: none"><li>Strings (case-sensitive)</li><li>Whole numbers</li></ul></div>		<div>To only execute a workflow path for endpoint detections not occurring on Microsoft Windows hosts, apply these settings:</div>

<ul style="list-style-type: none"><li>• Categorical data, such as Technique, Platform, and Severity</li><li>• Populated data, such as Hostnames and Platform</li></ul>	True when the observed parameter is not the same as the value you provide	<b>Parameter: Platform</b> <b>Operator: is not equal to</b> <b>Value: Windows</b>
--	---	---

is greater than

Data types	Description	Example
Data with a logical ordering: <ul style="list-style-type: none"><li>• Numerical data</li><li>• Ordinal category data, such as Severity</li></ul>	True when the observed parameter is larger than the value you provide--or follows it for ordinal categories. Equivalent to the mathematical relation ">" as in "5 > 3"	To process detections having high or critical severity, apply these settings: <b>Parameter: Severity</b> <b>Operator: is greater than</b> <b>Value: Medium</b>

is less than

Data types	Description	Example
Data with a logical ordering: <ul style="list-style-type: none"><li>• Numerical data</li><li>• Ordinal category data, such as Severity</li></ul>	True when the observed parameter is smaller than the value you provide--or comes before it for ordinal categories. Equivalent to the mathematical relation "<" as in "2 < 3"	To only process incidents having a score smaller than 2.4, apply these settings: <b>Parameter: Incident score</b> <b>Operator: is less than</b> <b>Value: 2.4</b>

is greater than or equal to

Data types	Description	Example
Data with a logical ordering: <ul style="list-style-type: none"><li>• Numerical data</li><li>• Ordinal category data, such as Severity</li></ul>	True when the observed parameter is larger than or the same as the value you provide--or includes or follows it for ordinal categories	To process all detections with a severity of medium or higher, apply these settings: <b>Parameter: Severity</b> <b>Operator: is greater than or equal to</b> <b>Value: Medium</b>

is less than or equal to

Data types	Description	Example
Data with a logical ordering: <ul style="list-style-type: none"><li>• Numerical data</li><li>• Ordinal category data, such as Severity</li></ul>	True when the observed parameter is smaller than or the same as the value you provide--or includes or comes before it for ordinal categories	To only process incidents having a score of 2.4 or smaller, apply these settings: <b>Parameter: Incident score</b> <b>Operator: is less than or equal to</b> <b>Value: 2.4</b>

exists

Data types	Description	Example
<ul style="list-style-type: none"><li>• Optional categorical data, such as Grand parent process command line, Source endpoint object GUID, and Custom IOA rule name</li><li>• Populated data, such as Tags</li></ul>	True when the observed parameter exists in the workflow events	To only process identity protection alerts with a source endpoint object GUID, apply these settings: <b>Parameter: Source endpoint object GUID</b> <b>Operator: exists</b>

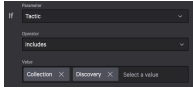
does not exist

Data types	Description	Example
<ul style="list-style-type: none"><li>• Optional categorical data, such as Grand parent process command line, Source endpoint object GUID, and Custom IOA rule name</li><li>• Populated data, such as Tags</li></ul>	True when the observed parameter does not exist in the workflow events	To only process endpoint detections where the grandparent process command line does not exist, apply these settings: <b>Parameter: Grand parent process command line</b> <b>Operator: does not exist</b>

includes

--



Data types	Description	Example
<ul style="list-style-type: none"> <li>Strings you select from the provided list</li> <li>Strings you enter individually and press Return after (case-sensitive)</li> <li>Time-based windows specified using a time-based parameter with a trigger or an action field, such as Workflow execution time, Last Activity, or VirusTotal Last Analysis Date</li> </ul> <p><b>Note:</b> Each item in the <b>Value</b> field should have an <b>X</b> symbol after it, as shown in the <b>Example</b> column. Click the <b>X</b> to remove the item.</p>	<p>True when the observed parameter equals one of the items in the list you provide</p> <p>If the parameter is a string, its value must equal one of the items in the list you provide.</p> <p>If the parameter is an array, one of the values in the array must equal one of the items in the list you provide.</p> <p>In cases where the parameter value will not be exactly equal to one of the items in the list you provide, use the <b>matches</b> operator. See <a href="#">matches [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ja2b1035]</a></p>	<p>To only process endpoint detections where the tactic is either <b>Collection</b> or <b>Discovery</b>, apply these settings:</p> <p><b>Parameter:</b> <b>Tactic</b>  <b>Operator:</b> <b>includes</b>  <b>Value:</b> Collection  Discovery</p> 

#### does not include

Data types	Description	Example
<ul style="list-style-type: none"> <li>Strings you select from the provided list</li> <li>Strings you enter individually and press Return after (case-sensitive)</li> </ul> <p><b>Note:</b> Each item in the <b>Value</b> field should have an <b>X</b> symbol after it, as shown in the <b>Example</b> column for <a href="#">includes [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p1a252f6]</a>. Click the <b>X</b> to remove the item.</p>	<p>True when the observed parameter does not equal any of the items in the list you provide</p> <p>If the parameter is a string, its value must not equal any of the items in the list you provide.</p> <p>If the parameter is an array, no value in the array can equal any of the items in the list you provide.</p> <p>In cases where you want to compare the parameter value to only part of a string, use the <b>does not match</b> operator. See <a href="#">does not match [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#dd10c950]</a></p>	<p>To only process endpoint detections where the technique is neither <b>Adware</b> nor <b>Adware/PUP</b>, apply these settings:</p> <p><b>Parameter:</b> <b>Technique</b>  <b>Operator:</b> <b>does not include</b>  <b>Value:</b> Adware Adware/PUP</p>

#### matches

Data types	Description	Example
Strings (case-sensitive)	<p>True when the observed parameter matches your wildcard filter pattern</p> <p><b>Note:</b> This pattern is not a regular expression. The asterisk is the only supported wildcard and represents any text. To form the pattern, combine a string with an asterisk before it, after it, or both. The wildcard filter pattern can have one of these formats:</p> <ul style="list-style-type: none"> <li>Data that starts with a string and ends with an asterisk: &lt;string&gt;*</li> <li>Data that starts with an asterisk and ends with a string: *&lt;string&gt;</li> <li>Data that starts with an asterisk, has a string, and then ends with an asterisk: *&lt;string&gt;*</li> </ul> <p>You cannot insert asterisks in the middle of the string. To match the start and end of a value, create a condition to match the start; then click <b>And</b> to create another one to match the end.</p>	<p>To only process endpoint detections where the command line starts with C:\windows\System32\wscript.exe apply these settings:</p> <p><b>Parameter:</b> <b>Command Line</b>  <b>Operator:</b> <b>matches</b>  <b>Value:</b> C:\windows\System32\wscript.exe*</p> <p>To only process the workflow branch for incidents when the domain of the host ends with labs.example.com, apply these settings:</p> <p><b>Parameter:</b> <b>Domain</b>  <b>Operator:</b> <b>matches</b>  <b>Value:</b> *labs.example.com</p>

#### does not match

Data types	Description	Example
Strings (case-sensitive)	<p>True when the observed parameter does not match your wildcard filter pattern</p> <p><b>Note:</b> This pattern is not a regular expression. The asterisk is the only supported wildcard and represents any text. To form the pattern, combine a string with an asterisk before it, after it, or both. The wildcard filter pattern can have one of these formats:</p> <ul style="list-style-type: none"> <li>Data starts with a string and ends with an asterisk: &lt;string&gt;*</li> <li>Data starts with an asterisk and ends with a string: *&lt;string&gt;</li> <li>Data starts with an asterisk, has a string, and then ends with an asterisk: *&lt;string&gt;*</li> </ul> <p>You cannot insert asterisks in the middle of the string. To match the start and end of a value, create a condition to match the start; then click <b>And</b> to create another one to match the end.</p>	<p>To process a workflow branch only for endpoint detections whose <b>File Path</b> parameter does not include the directory \known_samples\, apply these settings with asterisks before and after the value:</p> <p><b>Parameter:</b> <b>File path</b>  <b>Operator:</b> <b>does not match</b>  <b>Value:</b> *\known_samples\*</p>

Conditions in Advanced mode using CEL expressions

**Important:** This feature is available to all customers as a beta release. It includes CrowdStrike extensions, which are Common Expression Language (CEL) functions that start with the `cs.` prefix. CrowdStrike extensions are subject to change. During this beta release period, these changes might not be backward compatible. Any changes will be communicated. After general release, changes to CrowdStrike extensions will be backward compatible unless noted otherwise.

In Fusion SOAR conditions, you can use operators such as **is equal to** and **exists** to examine a parameter and possibly a literal, or static, value. In addition, you can use functions to manipulate data and write more expressive conditions. For example, you can determine whether an IP address is v4 or v6 with these expressions:

Operation	Expression	Result
Check whether the address is IPv4	<code>cs.ip.isV4('4.4.4.4')</code>	True
Check whether the address is IPv6	<code>cs.ip.isV6('2001:0db8:85a3:0000:0000:8a2e:0370:7334')</code>	True

For another example, you can check whether an array is empty. If a trigger returns an array, such as Source IP addresses, it's a good practice to check the size of the array before trying to act on its contents. Here is the expression to check:

```
size(data[source.ips]) > 0
```

This feature uses expressions based on CEL.

To use these expressions, when creating or editing a condition, click **Advanced mode**.

With this feature, you can also compare variables. For example, you can compare an IP address from event search results with one retrieved for a particular host.

For more info about using CEL in Fusion SOAR, see [CEL expressions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ie35e6c4].

Workflow actions

In addition to supporting the same notification channels as [Falcon Notifications](#) [documentation/page/b76dcde1/falcon-notifications-1692362323515.739], Fusion SOAR workflows support an expanded collection of potential actions.

**Note:** Available actions depend on the trigger type, your subscriptions, and your CrowdStrike Store app integrations. In addition, in Flight Control environments, when you create a workflow for a parent CID and apply it to child CIDs, the child CIDs can only access actions that are available to that parent CID. However, even some of the actions that the parent CID can access aren't available to its child CIDs. The Falcon console warns you when an action is not available to a child CID. Some warnings occur depending on whether an action takes place in the parent or in the child. For example, with Falcon Intelligence, actions happen in the parent. So a parent with Falcon Intelligence can create a workflow with an Intelligence action and apply the workflow to a child without warnings even if the child does not have an Intelligence subscription. However, with Falcon Insight XDR, the containment action happens in the child. So if a parent has Falcon Insight XDR and creates a workflow with a containment action, applying that workflow to a child without a Falcon Insight XDR subscription results in a warning.

**Tip:** Workflows now support variables in input fields. When you're configuring actions, you can insert variables in text fields that are followed by an **Insert variable** link. For example, if your action sends an email, the **Subject** text field supports variables such as **Customer ID**, **Workflow name**, and **Workflow execution timestamp**. In addition to using the **Insert variable** link, entering a dollar sign followed by a brace, `$`{, displays the variables menu. You can navigate the menu with arrow keys or filter it by typing part of a variable name. Be aware that the character count validation for the subject field ignores some hidden characters that are counted by a separate validation later. As a result, if variables used in the subject are longer than what was initially validated, running a workflow might produce an error. For example, if the value of `${Workflow.Description}` variable is greater than 100 characters, then the workflow returns an error on execution.

Find actions to add to your workflows by searching or browsing the action panel. Alternatively, use the content library discussed in [Library of actions, apps, Foundry app templates, playbooks, and triggers](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#qd6417a6].

For more info, see these topics:

- Find actions [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#f051f0dd]
- Add an action to your workflow [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#q0359ccd]
- Grouping of custom actions [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#rc25b385]
- More about specific actions and types of actions [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#611afd2]

Find actions

You can find the actions when searching in several ways:

- By the action name
- By words in the action descriptions

When you search, the most relevant results are shown in the **Top results** section. After that, all of the results are shown in lists.

Before and after searching, you can refine the search by filtering. To see the options, click **Filter**. These are the options:

- Hide unavailable actions**  
Show all the actions or just the ones that make sense in the current context by enabling or disabling **Hide unavailable actions**. Actions might be unavailable because of a missing prerequisite, plugin, or permissions. Showing all actions allows you to see the possibilities if you are able to resolve the requirements.
- Group by**  
Group the actions by vendor to show only the actions relevant to the given vendor.  
Group actions by use case, such as **Endpoint security** or **Identity & Access**, which shows all the actions relevant to the given use case.
- Vendor**  
Select specific vendors to show only the actions relevant to those vendors.

## Add an action to your workflow

After you find an action, complete these steps to add the action to your workflow:

1. Click the action.
2. Optional. Check what input the action requires by clicking **View schema** and then **Input schema**.  
For more info about JSON schema, see [Manage action input, action output, and on-demand triggers](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d].
3. Check the output so you know what to expect from the command by clicking **Output schema**.
4. Configure the action.
5. Click **Next**.

## Grouping of custom actions

If you create any actions—such as actions based on event queries or in Foundry apps, these actions go in the group named **Other (Custom, Foundry, etc.)**.

## More about specific actions and types of actions

You can use many of the actions you find in the console without additional info. However, some of them have usage details that aren't obvious. This list provides more info about those actions:

- **Enrichment:** Incorporate device or third-party vendor data for enriched context
  - **Get customer details:** This action is available only in parent CIDs in Flight Control environments.
- **Real time response:** Execute [Real Time Response commands](#) [/documentation/page/b8c1738c/real-time-response#k893b7c0]

**Note:** For info about enabling multifactor authentication for workflows with RTR actions, see [Real Time Response \(RTR\)](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ua24dff0].

**Tip:** Review details about executed RTR commands at [Audit logs > Audit logs > RTR](#) [/activity/real-time-response/audit-logs].

- Execute commands aligned with your RTR role:
  - **Get file** (get)
    - Retrieves host files.
  - **Kill process** (kill)
    - Takes a process ID as input and stops the process from running.
    - A common subsequent RTR command workflow action is **Remove file**.
  - **Process memory dump** (memdump)
    - Available with Windows only.
    - Requires a condition that specifies the sensor platform is equal to Windows. For more info, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].
    - Takes a process ID as input.
    - Output is the local file path of the memory dump file saved on the host.
    - Common subsequent RTR command workflow actions include:
      - **Get file** then **Remove file**
      - **Kill process** then **Remove file**
  - **Put file** (put)
    - Available with Windows, Mac, and Linux only
    - Requires a condition that specifies the sensor platform is equal to Windows, Mac, or Linux. For more info, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].
    - Takes an executable file name from the list of **"PUT" files** uploaded to **Host setup and management > Response and containment > Response scripts and files**.
    - A destination path that specifies where to download the file must be provided.
  - **Put and run file** (put-and-run)
    - Available with Windows and Mac only
    - Requires a condition that specifies the sensor platform is equal to Windows or Mac. For more info, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].
    - Takes an executable file name from the list of **"PUT" files** uploaded to **Host setup and management > Response scripts and files**.
    - Supports optional command line parameters that are passed when the file runs.
  - **Remove file** (rm)

**Note:** You can't **rm** actively running executables or protected system files, such as anything under C:\windows\system32.

Attempts to do so produce a failed workflow action and this error: Access to the path is deniedTo improve workflow efficacy:

    - Set up other workflow actions to run in parallel as the **rm** action.

- Avoid setting up critical actions after an **rm** action. If the **rm** action fails, subsequent actions are skipped.

- **Retrieve active network connections** (`netstat`)

- **Retrieve running processes** (`ps`)

- **Run file** (`run`)

- Takes the file provided in the trigger event as input. For more info, see [Workflow triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#x19234b1\]](#).
- Supports optional command line parameters that are passed when the file runs.

- Execute Falcon scripts.

- Select the script name, and then provide arguments as required.
- Available with Windows only.
- Requires a condition that specifies the sensor platform is equal to Windows. See [Workflow conditions \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba\]](#).
- Only users with the RTR Administrator role can add Falcon script actions to workflows.

- Execute RTR custom scripts. For more info, see [Managing custom response scripts \[documentation/page/b8c1738c/real-time-response#hdf249ae\]](#).

**Note:** Only users with the RTR Administrator role can add custom-script actions to workflows.

- **Identity Protection:** For example use cases for the following Identity Protection workflow actions, see [Identity Protection in Fusion SOAR \[documentation/page/d5505b0c/identity-protection-in-falcon-fusion-workflows\]](#).

- **Watch or unwatch users and endpoints**

Endpoints must be in an Active Directory domain monitored by Identity Protection. CrowdStrike recommends adding a condition that filters out non-Windows platforms before adding this action to a workflow. For more info, see

[Workflow conditions \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba\]](#).

- Active Directory actions

- **Reset user Active Directory password**
- **Active Directory - Enable Account**
- **Active Directory - Disable Account**
- **Active Directory - Unlock User**

- Get additional user or endpoint context from Falcon Identity Protection.

- **Get user identity context**

- This action provides identity context for a given user identifier. There are a number of available identifiers that you can use based on the trigger, including:
  - User object SID
  - User name
  - User object SID

**Note:** This action executes a query that searches for an entity which matches all fields with the OR condition. Therefore, we recommend that you select only one identifier to use in your workflow.

- This action returns information from Identity about the user such as risk severity, privileges, group membership, etc. This info can be used in further conditions in your workflow.

- **Get endpoint identity context**

- This action provides identity context for a given entity. There are a number of available identifiers that you can use based on the trigger, including:
  - Endpoint SID
  - Endpoint object GUID
  - Sensor ID
  - Endpoint name

**Note:** This action executes a query that searches for an entity which matches all fields with the OR condition. Therefore, we recommend that you select only one identifier to use in your workflow.

- This action returns information from Identity about the endpoint such as risk severity, classification, group membership, etc. This info can be used in further conditions in your workflow.

- Entra ID response actions

**Note:** The CrowdStrike Store Microsoft Entra ID SOAR Actions plugin must be configured for this action to appear. For more info, see [Integrate with Microsoft Entra ID \[documentation/page/dfe838e5/crowdstrike-store-app-integrations#x7f2ece9\]](#).

- **Entra ID - Add User to Group**
- **Entra ID - Disable User**
- **Entra ID - Enable User**
- **Entra ID - Remove User from Group**

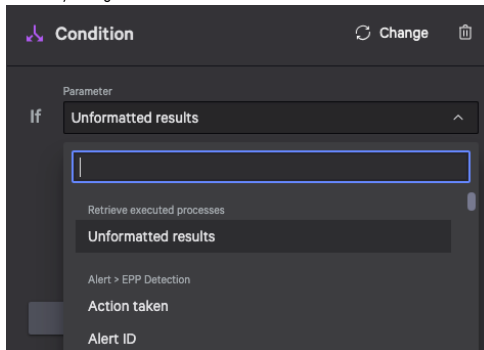
- **Entra ID - Revoke Existing Refresh Tokens**
- **Entra ID - Revoke Existing Sign-in Sessions**
- **Entra ID - Mark User as Risky (requires Microsoft Entra ID P2 license)**
- **Entra ID - Unmark User as Risky (requires Microsoft Entra ID P2 license)**

- **Event search:** Run an event search to gather data for actions or triggers.

For example, you could find users logged in during the last day. Check each user against the Identity Protection watchlist. Then, for the users on the watchlist, send the users email to notify them that their systems will be contained, and then contain the systems. Alternatively, when a detection occurs, you could get the process associated with the detection and find all network connections associated with that process. Then add this info to a report and distribute it in an email.

**Note:** These actions require the Falcon Insight XDR subscription.

After the event search action runs, the results are available to use in an action or condition. The results vary based on the search action. To get the results when you are defining an action that accepts a file, such as the **Create Jira issue** action, in the **Data to include** list, find the name of the event search action. To get the results when you are defining a condition, in the **Parameter** list, find the name of the event search action. For example, for the condition shown below, under the name of the event search action, **Retrieve executed processes**, the only option to access the results is **Unformatted results**. Another you might see is **Full results download URL**.



**Tip:** To expose fields from the results one at a time, use the results in a loop. For example, using the **Find logins for a user account across all hosts** search returns a list that includes the systems where the user is logged in. You can then loop over the results and perform actions for each of those systems. When defining the loop, in the **Input source** list, find the name of the event search action. Related playbooks:

[Identity Protection Watchlist Update after Credential Access Detection](#) [/documentation/page/zb98cbcd/identity-protection-watchlist-update-after-credential-access-detection-0]

: Automatically adds users and hosts to the Falcon Identity Protection watchlist in response to identity-based incidents that use MITRE ATT&CK® Credential Access tactics.

[ServiceNow Ticket Creation for High Severity Detections](#) [/documentation/page/r4d0b59b/servicenow-ticket-creation-for-high-severity-detections-0]

: Creates ServiceNow ticket for high severity detections with recent executed processes and command line history attached to the ticket.

**Note:** An event search action times out if it runs more than 1800 seconds, which is 30 minutes. The workflow execution fails with the timeout.

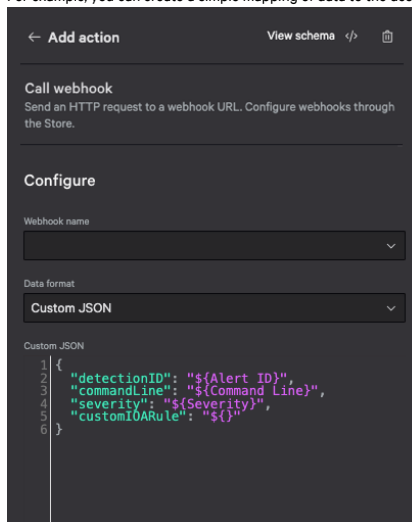
- **Call webhook:** Send an HTTP request to a webhook URL

The **Call webhook** action supports customizable input through its **Custom JSON** field. This support provides flexibility in exporting and managing data by allowing you to define a mapping to tailor data to the downstream tool processing the webhook.

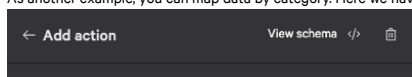
When you add a **Call webhook** action, you have these **Data format** options for the data to export:

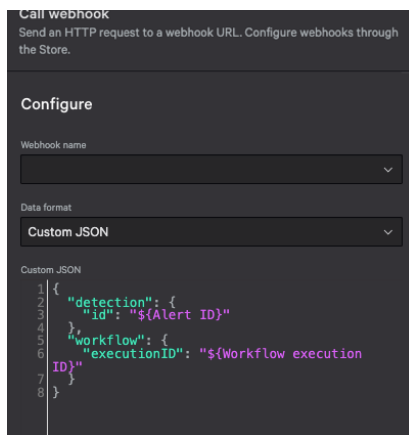
- **Default**  
Using this format option, which was previously the only one available, displays the **Data to include** list that allows you to select multiple items to use.
- **Custom JSON**  
Using this new format option displays a field where you enter your JSON to map Falcon data to the desired custom JSON format. To help you map data to the desired fields, click **Insert variable** and select variables.

For example, you can create a simple mapping of data to the desired property names.



As another example, you can map data by category. Here we have two categories: One for detection data and one for workflow data.





- **Sleep:** Wait a specified time before proceeding to the next action

**Note:** When a trigger occurs, status and all other info is permanently set. Using a Sleep action in a workflow to wait for any of the trigger info to change doesn't work because that info never changes.

- **Next-Gen SIEM actions:** With Next-Gen SIEM, several actions are available to use third-party data. However, the following actions are not available in US-GOV-1:
  - Darkfeed
  - Falcon for Mobile built for Microsoft Intune
  - IPQS Threat & Risk Scoring
  - Iris Threat Intelligence
  - Rubrik Security Cloud
  - SecurityScorecard Cyber Risk Ratings
  - VirusTotal

## Library of actions, apps, Foundry app templates, playbooks, and triggers

To help you find Fusion SOAR workflow options that you can use to set up or modify a workflow, visit the library at [Fusion SOAR > Fusion SOAR > Content library \[/workflow/fusion/content-library\]](#).

The content library provides a single location where you can find actions, apps, Foundry app templates, playbooks, and triggers. You can then explore what's possible, learn how to use content more effectively, or more quickly find what you need to define a workflow for your particular scenario. For more info, see [Workflow actions \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p7d19355\]](#) and [Fusion SOAR Playbooks \[/documentation/page/faa65a8c/falcon-fusion-playbooks\]](#).

To find items that most interest you, you have several options:

- List, sort, and browse
- Search  
You can search for a whole phrase or just the beginning of a phrase.
- Filter in several ways:
  - By vendor, so you can focus on items that work with the selected vendor
  - By use case, such as **Application security** or **Exposure management**
  - By multiple use cases at the same time

On the **Apps** tab, you see these items for the apps:

- Names and descriptions of the apps
- Their use cases
- When they were last updated
- Their status: Installed or –, meaning not installed

When you click an app, you see these items:

- The app description
- Actions in the app
  - To see descriptions plus input and output schemas, if any, for an action, expand its listing  
For more info about schemas, see [Manage action input, action output, and on-demand triggers \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d\]](#)
- Configuration instructions for the app
- A **Configure app** button you can click to set up the app or modify its configuration

On the **App Templates** tab, you see these items for the Foundry app templates:

- Names and descriptions of the app templates

- Names and descriptions of the app templates
- Their use cases
- When they were last updated
- The number of actions they include

When you click an app template, you see these items:

- The app template description
- A **Deploy in Foundry** button that takes to you to Foundry to deploy the app template

On the **Playbooks** tab, you see these items for the playbooks:

- Names and descriptions of the playbooks
- Their use cases
- When they were last updated
- The number of actions they include

When you click a playbook, you see these items:

- The playbook description
- The playbook definition
- The trigger and actions used in the playbook
- An **Open in Fusion SOAR** button you can click to open the playbook and customize it in Fusion SOAR.

On the **Actions** tab, you see these items for the actions:

- Names and descriptions of the actions

**Note:** This tab includes native actions, actions used in apps, and actions used in app templates. It does not include custom actions—actions you create based on event queries or in Foundry apps.

- Their use cases
- When they were last updated

When you click an action, you see these items:

- The action description
- The input schema and output schema, if any, for the action

For more info about schemas, see

[Manage action input, action output, and on-demand triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d1\]](#).

On the **Triggers** tab, you see these items for the triggers:

- Names and descriptions of the triggers

**Note:** Only triggers based on events appear on this tab. Scheduled triggers, on-demand triggers, and triggers based on inbound webhooks are not included.

- Their use cases
- When they were last updated

When you click a trigger, you see these items:

- The trigger description
- The output schema, if any, for the trigger

For more info about schemas, see

[Manage action input, action output, and on-demand triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d1\]](#).

## Looping in workflows

Workflows can loop through items found in a list of data in the output of certain triggers and actions. For each item, the workflow can make a decision or take an action based on that item. The lists depend on the trigger or action. For example, the **Alert > EPP Detection** trigger output includes a list of **Host groups**. For an example of an action, under the **Threat Graph** type, the **Get devices associated with a sha256 hash** action produces a list of hosts.

Here are some possible uses for loops:

- Polling for results  
The loop polls an API and then uses the **Sleep** action to wait for a response. When the API returns a specific response or status code, the **Loop break** action stops the loop. The workflow continues to its next component.
- Paginating API responses  
For APIs that support pagination, use a While loop to make requests until a response contains the desired data. Using this technique, you avoid pulling all the data at once. The While loop would contain a nested loop to check the data for a certain value. When that value is found, a **Loop break** action in the nested loop breaks out of both loops.
- Making loop data available to the rest of a workflow  
For a loop with sequential iterations, you can preserve data from the loop for use after the loop. The output is an array of objects where each object corresponds to a loop iteration. You can then use the entire array or its components in other parts of the workflow.
- Using a variable within a loop  
With the **Create variable** action, define a variable that you then update within the loop using the **Update variable** action.  
For a loop with sequential iterations, this variable is available outside the loop and has the value set in the last loop iteration.

## General considerations

- Loop types

- **For each:** Iterate over a defined dataset concurrently or sequentially until the dataset is used—or a **Loop break** action occurs.
- **While:** Iterate sequentially over loop-scoped data until a specified condition is met or a **Loop break** action occurs.

**Tip:** In addition to defining a condition using the **Parameter** and **Operator** dropdown lists, you can define a condition using Common Expression Language (CEL) expressions. Using these expressions, you can manipulate data and write more expressive conditions. To use these expressions, when creating or editing a condition, click **Add condition using CEL**. For more info about these expressions, see [CEL expressions \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ie35e6c4\]](/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ie35e6c4).

With this loop type, you also have the option to end a loop based on the number of iterations and time in the loop.

**Important:** Most parameters you set in a **While condition** are not updated with each iteration. To use those parameters:

1. Create a custom variable that you set to the value of the desired parameter before the loop. Use the **Create variable** action.
2. Use that custom variable as the parameter in the **While condition**.
3. Update the value of the custom variable inside the loop to match the current value of the desired parameter. Use the **Update variable** action.

- Processing orders

- **At the same time / concurrently**

- Available only to **For each** loops.
- Runs iterations in batches of 500 with the iterations run at the same time. Batches run one at a time. The batch size is subject to change.
- Concurrent loops do not output data that you can directly use in subsequent actions or loops, however you can use the **Write to log repo** action to retain the output.
- Any custom variable updated in a concurrent loop doesn't retain the update after the loop iteration

- **One after the other / sequentially**

- Available to **For each** loops and **While** loops.
- Runs each iteration in order.
- Loops end when all iterations are complete or a **Loop break** action occurs.  
**While** loops can also end based on a condition, the number of iterations, and time in the loop.
- Sequential loops output data arrays that you can use in subsequent actions or loops.
- Any custom variable updated in a sequential loop retains the update made in the last loop iteration

- Nested loops

- A loop within a loop is a nested loop. Nesting only goes one level: Nested loops can't contain loops.
- A loop can have one or more loops nested in it. With multiple nested loops, the loops must be on different branches and loop on different items.

- Related actions

- **Loop break**

Use this action after a condition to possibly end the loop.

**Note:** This action is available only for loops with sequential iterations.

When you use this action within a nested loop, specify whether to end the parent loop or the grandparent loop.

- **Create variable**

Use this action to create a variable.

**Note:** Use integer variables whenever comparing whole numbers. Don't use integer variables as input in action fields.

- **Update variable**

Use this action to update a variable inside a loop.

**Note:** For a loop with sequential iterations, variables are available outside the loops and have the value set in the last loop iteration. For a loop with concurrent iterations, you can update and use variables inside the loop; however, the values set in the loop are not retained for use after the loop.

- Output availability

- You can make the output of actions in a sequential loop available to use later in the workflow.

The **End Loop** item identifies the loop's output.

The output is an array of objects, where each object corresponds to an iteration of the loop and each field corresponds to a chosen loop output key.

You can then use the entire array or its separate fields in other parts of the workflow.

**Important:** In the output, some fields might be empty if conditions in the loop create branches and those branches use actions that populate different fields.

- To retain the output for a concurrent loop, use the **Write to log repo** action.
- In a loop with multiple actions, the output from each action is only available to subsequent actions in that loop. For example, the second action can use the output of the first action. However, the first action can't use the output of the second action.
- The output of an action not in a loop is available to any actions that come after it on the same branch.

- Default iteration limits

To prevent indefinite loops when you use a **While** loop or a **For each** loop with sequential processing, loops have a default limit of running for 50 iterations or 60 minutes—whichever comes first. To adjust these values, select the **Limit iterations** option and enter your desired values.



## Max iterations

A loop can run, or iterate, up to 100,000 times. If a loop attempts to iterate more than 100,000 times, both the loop and its workflow go into an error state. Such a workflow is not available to retry. See [Retry a failed execution](#) [\[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#t8c378c1\]](#). To continue a workflow even when a loop fails, use the **Continue workflow on loop iteration failure** option when you create loops. See [Loop iteration failure](#) [\[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#s5eef7d0\]](#).

With nested loops, both the outer loop and the nested loop can iterate up to 100,000 times each.

## Workflow execution time

In concurrent loops, loop iterations are processed in batches, one batch at a time. As a result, larger loops can take hours to complete.

In both concurrent and sequential loops, the amount of time for a workflow to complete depends on the number of actions in each loop and, if those actions aren't completed quickly enough, the timeouts for those actions.

To reduce a workflow's execution time, decrease the amount of work in each loop.

## Parallel loops and sequential loops

Consider loops A, B, and C.



Loops A and C are parallel:

- They run at the same time.
- There is no dependency between the loops. If loop A fails, loop C still runs.
- The output of any actions in loop A is not available in loop C.
- The output of any actions in loop C is not available in loop A.

Loops A and B are sequential:

- Loop A must finish before loop B can start.
- If loop A fails, loop B might not start—depending on the **Continue workflow on loop iteration failure** option. For more info, see [Loop iteration failure](#) [\[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#s5eef7d0\]](#).
- The output of any actions in loop A is not available in loop B.

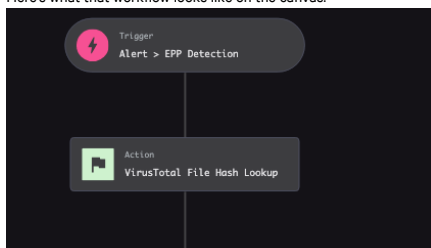
## Loop examples

The possibilities for looping are numerous. Here are examples to give you some ideas about those possibilities.

### Contain all devices with a file hash considered malicious by VirusTotal

In this example, the workflow looks for hashes that are considered malicious. It then gets devices that have that hash. The loop iterates through the list of devices and contains those devices. Next, the workflow assigns the alert to a user, adds a comment to the alert, and sends an email that indicates how many devices were contained.

Here's what that workflow looks like on the canvas.



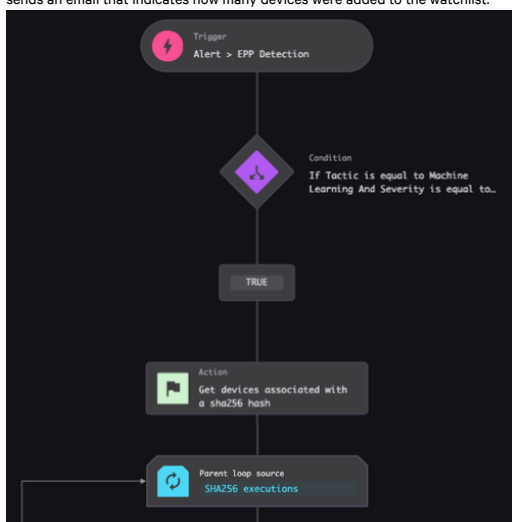


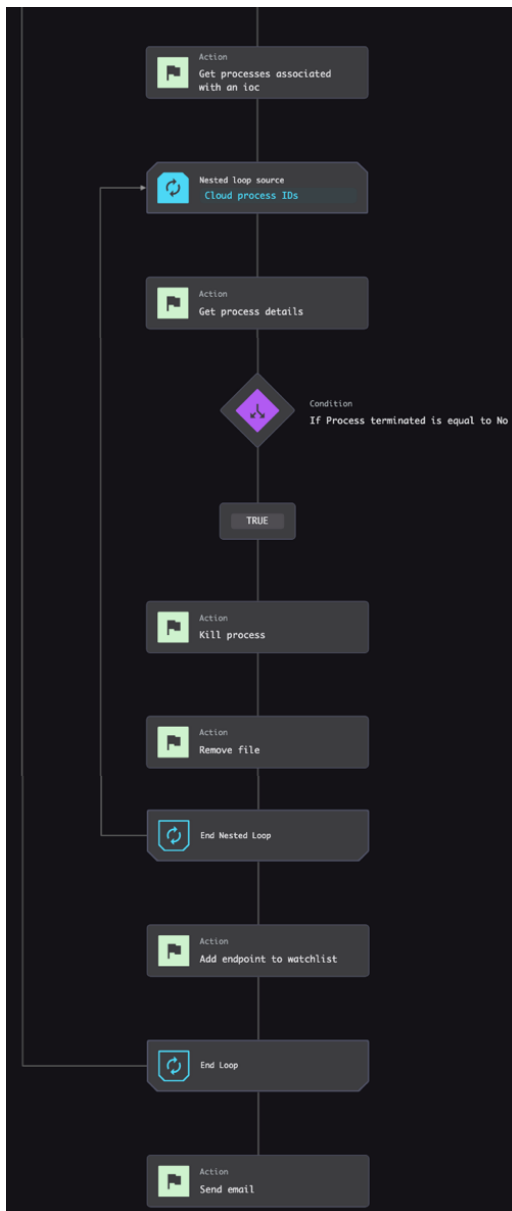
To create this workflow, the high-level steps are as follows.

1. Create a trigger for an **Alert > EPP detection**.
2. Add the **VirusTotal File Hash Lookup** action for the **Executable SHA256** found in the alert.
3. Create a condition to check whether the **VirusTotal file Malicious Count** exceeds the threshold, which is 10 in this example.
4. Add the **Get devices associated with a sha256 hash** action and the **Executable SHA256** hash.
5. From that action, create a loop with an input type of **SHA256 executions**.
6. In that loop, add the **Contain device** action with **Device ID** set to **Host ID**.
7. After the loop, add the **Assign alert to user** action and specify the user.
8. Add the **Assign comment to alert** action and specify a comment to indicate that a workflow took action.
9. Add the **Send email** action.

### Terminate process, remove file, and add host to watchlist for processes associated with an IOC across devices

This example uses nested loops. The outer loop iterates through devices getting processes associated with an indicator of compromise, or IOC. For each device, the inner loop iterates through that list of processes to kill the process, remove the file, and add the host to the Falcon Identity Protection watchlist. Next, the workflow sends an email that indicates how many devices were added to the watchlist.



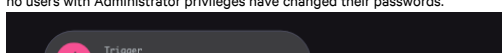


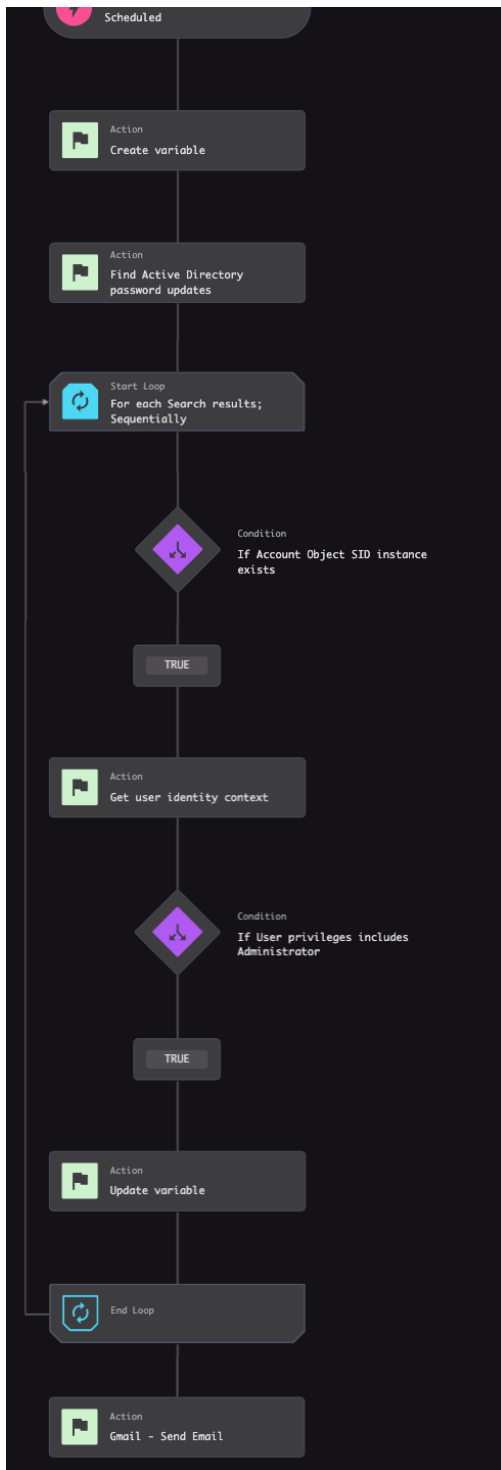
For this workflow, the high-level steps are as follows.

1. Create a trigger for an **Alert** with a subcategory of **EPP detection**.
2. Create a condition to check whether both the **Tactic** is **Machine Learning** and the **Severity** is **High**.
3. Create an action of the **Threat Graph** type with the **Get devices associated with a sha256 hash** action and the **Executable SHA256** hash.
4. From that action, create a loop with an input type of **SHA256 executions**.
5. In that loop, create an action of the **Threat Graph** type with the **Get processes associated with an ioc** action.
6. Still in that loop, create another loop with an input type of **Cloud process IDs**.
7. In the inner loop, complete these steps:
  - a. Create an action of the **Threat Graph** type with the **Get process details** action and set the **Cloud process ID** field to the **Cloud process IDs instance**.
  - b. Create a condition to check whether **Process terminated** is **No**.
  - c. Create an action of the **Real time response** type with the **Kill process** action with **Device ID** set to **Host ID instance** and **Process ID** set to **OS Process ID**.
  - d. Create an action of the **Real time response** type with the **Remove file** action with **Device ID** set to **Host ID instance** and **File Path** set to **File path**.
8. After the inner loop, create an action of the **Identity Protection** type with the **Add endpoint to watchlist** action and set **Endpoint ID** to **Host ID instance**.
9. After both loops, create an action of the **Notifications** type with the **Send email** action to indicate a workflow took action and to include data such as **Device count** and any other desired info.

#### Check hourly for users with Administrator privileges changing their passwords and send email

This example creates a variable and then uses a **For Each** loop to check whether any users who have Administrator privileges have changed their passwords. If any of these users have changed their passwords, they are added to the variable. After the loop finishes, the variable is used in the body of an email. The email is empty if no users with Administrator privileges have changed their passwords.





For this loop, the high-level steps are as follows.

1. Create a scheduled trigger that runs hourly.
2. Add the **Create variable** action. We'll use this variable to collect a list of users with Administrator privileges who have changed their passwords. The variable will be an array called users. Here's the JSON schema that defines the variable:

```

{
  "type": "object",
  "properties": {
    "users": {
      "items": {
        "type": "string"
      },
      "type": "array"
    }
  }
}
  
```

3. Add the **Find Active Directory password updates** action.
4. Add a loop with these settings:  
**Loop type:** For each  
**Loop source:** Search results  
**Processing order:** One after the other / sequentially

5. Inside the loop, complete the steps as follows:

- a. Add a condition to check if the Account Object SID instance exists.
  - b. If it does exist, add a condition to check if user privileges includes Administrator.
  - c. Add the **Update variable** action.  
If the user does have Administrator privileges, we update our variable to include the user.  
In **Variable**, select the array variable we created: **Users**. In **Value**, include **Users** and **User name**.
6. After the loop, add the **Gmail - Send Email** action.  
Include the variable in the email by setting **Message Body** to `${Users}`.

### Loop iteration failure

In concurrent loops, a loop tries to run for each item of input, up to a maximum of 100,000 iterations. If any of those iterations does not complete successfully, the workflow behavior depends on the **Continue workflow on loop iteration failure** option. This option is available to select only when you define a concurrent loop.

Continue workflow on loop iteration failure option	Description
Selected	Even if a loop iteration fails or a loop has zero iterations because of a lack of input, the workflow continues to execute any nodes after the loop itself. If any iterations fail, the entire execution shows as failed even if actions after the loop succeeded. Select the option to treat the loop executions as "best effort."
Deselected	<p>The workflow does not proceed any further. Don't select this option in either of these cases:</p> <ul style="list-style-type: none"> <li>• Actions after the loop depend on the success of the loop</li> <li>• Actions after the loop should run only if all loop iterations are successful</li> </ul>

## Event queries, or saved searches, as workflow actions

You can create workflow actions that query various Falcon data sources. The available data sources depend on your Falcon subscriptions.

This feature improves security efficiency and reduces response time by eliminating manual enrichment and correlation tasks.

By creating workflow actions that run queries, security engineers can perform activities such as these:

- Query data in a repository from a Fusion SOAR workflow to eliminate a manual, redundant task done by an analyst or take an action based on the output of the query
- Create a scheduled workflow that completes a proactive threat hunt by running a series of event queries and then sends an alert if suspicious event data is found

For info about requirements for event queries related to subscriptions, CrowdStrike clouds, and roles, see [Requirements \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#8d760c7\]](#).

For info about how to manage these actions, see [Create and manage actions based on event queries \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#t1e23404\]](#).

### Limitations

Be aware of these limitations:

- You can only query repositories that CrowdStrike provides. Querying data sources you host is not possible.
- While the email action available in Fusion SOAR does allow you to append query results to the message body, the results are limited to 5000 characters and in JSON.  
However, you can attach a file that is up to 10 MB in size.
- Search actions are available only in the CIDs where they were created.
- By default, query results are limited to 200 rows.  
You can increase this limit to 10,000 rows by specifying a tail in the query. To specify a tail, pipe the query to `tail(x)`:  
`| tail(x)`  
where `200 < x <= 10000`.  
Similarly, you can pipe a query to `head(x)`.
- The query result supports only primitive types—that is, types that represent a single value.  
For example, `{key: foo}` is supported, but `{key: [foo, bar]}` is not.
- An action based on an event query times out if it runs more than 1800 seconds, which is 30 minutes. The workflow execution fails with the timeout.

## Setup

### Planning and preparation

As you prepare to create and refine workflows, there are important considerations for each action type you want a workflow to perform.

### Incident and endpoint detection updates

Review how incidents and endpoint detections have been assigned to your users, tagged, and moved through statuses to identify what actions you might want to

automate.

## Real Time Response (RTR)

- Review how your organization has remediated incidents and detections in the past to establish which kinds of actions you might want to automate on specific kinds of detections.
- It's essential that you craft workflows that involve remediation with specific conditions to avoid taking action on a large number of hosts.
- Consider creating or updating RTR custom scripts. Custom scripts with the **Share with workflows** toggle enabled are available as RTR actions in workflows. For more info, see [Managing custom response scripts](#) [/documentation/page/b8c1738c/real-time-response#hdf249ae].
- Ensure your Response Policies, set using [Host setup and management > Response and containment > Response policies](#) [/configuration/real-time-response/policies], are configured to allow the actions you're expecting from your workflows.
- You can enable additional security by going to [Support and resources > Resources and tools > General settings](#) [/configuration/general-settings]. In the **Security** menu, select **Re-authentication for critical actions**, and then enable these settings:
  - **Real Time Response (RTR) identity verification**
  - **Before enabling a Fusion SOAR workflow with any RTR action**

With these settings enabled, if a workflow has an RTR action, when you enable that workflow or run it on demand, Fusion SOAR prompts you to verify your identity using multifactor authentication, or MFA.

## Network containment

**Important:** Apply network containment actions with extreme caution. When added to a broad trigger, essential hosts might be unexpectedly isolated from the network. To reduce this risk, configure triggers that meet very specific conditions.

## VirusTotal lookup

See [Set up VirusTotal integration](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations].

## Notifications

- Think about what notifications you want to set up and review your existing [Falcon notifications](#) [/documentation/page/b76dcde1/falcon-notifications-1692362323515.739], noting potential crossover that might create duplicate notifications for your users.
- Carefully consider what people in your organization need to know about and the best way to reach them. Be precise about what triggers notifications and who receives them so you don't fire off too many for your users to handle.
- Set up plugin integrations for notification channels, if needed:
  - [Jira](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#g7581864]
  - [Microsoft Teams](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#p0e9e45b]
  - [PagerDuty](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#xd8f65f4]
  - [ServiceNow](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#fa465454]

**Tip:** When you create a workflow, refer to

[Creating ServiceNow incidents with host details](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#f4d5181f] or [Vulnerability Management Ticketing Workflows](#) [/documentation/page/e552ea54/vulnerability-management-ticketing-workflows].

- [Slack](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#n94f6582]
- [Webhook](#) [/documentation/page/dfe838e5/crowdstrike-store-app-integrations#pc43a512]

## Assets

You can trigger workflows for unmanaged assets, unsupported assets, and new external assets using an asset management trigger.

- Triage unmanaged and unsupported assets. These are assets that don't have the Falcon sensor installed.
  - Think about how best to determine whether assets new to your environment belong there and which conditions typically inform those decisions.
  - Carefully consider the conditions and triage actions to decide which ones you might want to automate, such as when to recommend a sensor install on an unmanaged asset, or to change an unmanaged asset to unsupported.

**Note:** Any workflows that include the **Move to unsupported** or **Move to unmanaged** action must also include a condition that specifies the **Entity type** as either **Unmanaged** or **Unsupported** or else the workflow won't be completed.

- Trigger a workflow when new external assets are added to your ecosystem.

Asset actions aren't available in CrowdStrike government clouds.

## Testing


- Start by turning on a few key workflows and reviewing the execution log to get a sense of how often they're triggered. See [Monitor executions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z9a11c10].
- Quickly turn off a workflow if it's triggered too frequently.

- Edit workflows as needed:
  - Add or change conditions to refine how often they're triggered.
  - Update actions to refine what happens when they're triggered.

# Manage workflows

## View workflows

To see and manage workflows, go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).

Click **Open menu**  for any workflow to access options to view, edit, and more.

## Create a workflow

Through the process, you choose a trigger, add conditions to refine the trigger, and define actions to be performed when the exact trigger conditions are met. As you work through creating a workflow, the **Workflow preview** shows you where you are in the process and the attributes of the workflow you're creating.

**Note:** If you add an invalid item to your workflow, a warning or error is shown in the **Issues** panel.

Workflows can be straightforward and sequential, or can accommodate specific scenarios and needs by adding Else If conditions, Else actions, or loops to create branches.

The Fusion SOAR workflow builder consists of the workflow *canvas* and various panels.



- The canvas is a visual representation of the workflow functionality.
- The panels let you define individual elements in your workflow.

## Navigating the workflow canvas

If you are creating or editing a workflow, you can navigate the workflow canvas several ways.

To move the workflow within the canvas, click and hold a spot in the canvas, and then drag it.





To zoom in or out on a particular spot in a workflow, hover over that spot and use your mouse's scroll wheel.

To edit an item, click it and, depending on the item, click **Edit**  or **Back** .

To see options to delete an item or to add more items immediately after it, hover over the item.

Also, you can select items in the workflow by using the **Tab** key.

When tabbing, press **Enter** to act on the selected item. Within a workflow, pressing **Enter** could show details for the item and allow you to edit it, delete the item, or present items you can add after the current item. Consider these scenarios:

- If pressing **Tab** selects an action item, pressing **Enter** displays the panel where you can edit the action.
- If pressing **Tab** selects , pressing **Enter** deletes the item.
- If pressing **Tab** selects , , or  at the bottom of an item, pressing **Enter** shows the items you could add at that point.

Outside a workflow, tabbing provides access to the menu bar, where you also press **Enter** to act on the selected item.

## Create the workflow

**Tip:** For ideas about what to include in your workflow, use the [content library](#). For more info, see [Library of actions, apps, Foundry app templates, playbooks, and triggers \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#qd647a6\]](#)

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
2. Click **Create workflow**.
3. For Flight Control environments, users with the Falcon Administrator role in the parent CID can choose which child CIDs the workflow applies to. Those workflows can trigger for any child CID, even if the users don't have permissions in the child CIDs. However, users with the Workflow Author role in a parent CID can only create workflows that trigger for that parent CID: Those users aren't able to create workflows for that CID's child CIDs.

**Tip:** When creating a workflow that you will use with multiple CIDs, you can set up the workflow in a child CID to test and verify it first by triggering it and checking the execution log in the parent CID. Then, after verification, apply the workflow to more CIDs. For more info, see [Edit a workflow's CIDs in a Flight Control environment \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u8fc34f6\]](#).

When applying a workflow to child CIDs, you have these options:

Option	Description
All child CIDs	Applies the workflow to all current and future child CIDs. However, you can explicitly exclude some CIDs.
Specific CIDs	Applies the workflow only to the CIDs you specify.
Only current CID {CID_name}	Applies the workflow only to the current CID.

If you define a workflow in a parent CID, the definition of the workflow, its execution log, and its audit log are only shown in the parent CID and not any of its child CIDs.

4. Select whether to create a workflow from scratch, by using a playbook, or by importing a workflow, and then click **Next**.

- From scratch: Define all of the settings of the workflow yourself.
- Playbooks: To help you create your workflow, several playbooks are available. They serve as templates that you modify to quickly set up workflows in your environment. Playbooks simplify and automate common use cases and demonstrate workflow possibilities. Each playbook includes its own setup steps. For more info, see [Fusion SOAR Playbooks \[documentation/page/faa65a8c/falcon-fusion-playbooks\]](#).
- Import: If you have exported a workflow, you can import its definition file.  
For more info, see [Export or import a workflow \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#rfd59569\]](#).  
If you import a workflow, skip the rest of these steps and follow the steps in [Import a workflow \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#e555e82d\]](#) instead.

5. If creating a workflow from scratch, find an event-based trigger or select a trigger type:

- **Event:** The workflow is triggered by an event in the Falcon environment, such as a new incident.  
To find an event-based trigger, use the search field or browse the lists of use cases. Either way, you can filter the triggers shown and adjust the sort. For info about event triggers, see [Workflow triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z19234b1\]](#).
- **Scheduled workflow:** The workflow is triggered regularly, based on a defined schedule, such as hourly, daily, weekly, or monthly. These options are available to specify the details of the workflow schedule:
  - **How often:** Schedule a workflow to run hourly, daily, weekly, or monthly.
  - **Start time:** Specify a time for the workflow to start. For example, if you specify a daily workflow, you can set it to start at 2 AM each day.
  - **Time zone:** Specify the time zone that should be used for running the scheduled workflow. The workflow settings, such as start time, adhere to this time zone, not the time zone of a Falcon console user or an asset like an affected host.
  - **On these days:** For weekly or monthly scheduled workflows, select one or more days of the week or month to run the workflow. For example, you can specify a monthly workflow to run on the 1st and 15th days of the month or you can specify a weekly workflow to run on Tuesday and Thursday of each week.
  - **Add start or end date:** Optionally, specify a date to start or end the schedule for the workflow runs. If specified, the workflow runs according to the specified schedule only during the selected timeframe.

**Skip if a previous execution is still in progress:** Select if you want to skip an iteration of the scheduled workflow if the previous iteration is still in progress. For example, if you have an hourly scheduled workflow, you might not want to start a new iteration if the previous hour's workflow is still in progress.

**Tip:** You can also run this type of workflow immediately, as shown in [Run a workflow on demand \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#xa3f58bb\]](#).

- **On demand:** The workflow is triggered directly through the Falcon console, by another workflow, or by an API call, as shown in [Run a workflow on demand \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#xa3f58bb\]](#). When setting up this trigger, you have these options:
  - If you prefer not to require any actions or fields when a user runs the workflow, go to the next step.
  - Specify the required items for the workflow to run by defining a JSON schema to specify the mandatory actions and fields.  
You have several ways you can define a JSON schema:

- To find a sample to use, click **Search schema samples**. For example, if you want to get an alert ID or run a device query, there are samples that show you how.
- To provide and convert JSON into JSON schema, click **Generate schema** and paste in the sample JSON.
- To create the schema one property at a time, click **Add property** (+).

Here's an example JSON schema that requires a specific device as input:

```
{
  "type": "object",
  "properties": {
    "deviceId": {
      "type": "string",
      "format": "aid"
    }
  }
}
```

You could use that input in a workflow to contain a device and create a Jira ticket to investigate the situation.

For more info about JSON schema, see [Manage action input, action output, and on-demand triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#v566df3d\]](#).


- **Inbound webhook:** The workflow generates a unique webhook URL for the given workflow and is then triggered by an external system that uses that URL. For more info, see [Custom triggers based on inbound webhooks \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#mf5cdeeb\]](#) and [Create and manage triggers based on inbound webhooks \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#scedcaa6\]](#).

6. Click **Next**.

- Your selected trigger appears on the workflow canvas.
- To change the trigger selection, click the trigger element on the canvas and click **Back** ← to return to trigger selection.

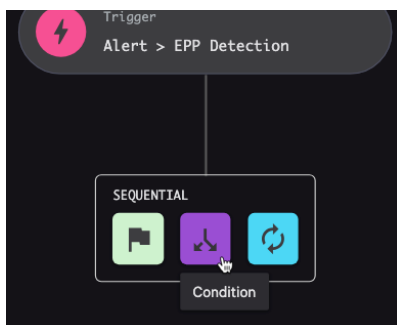
**Tip:** Throughout the process of building a workflow, the details panel shows helpful tips, warnings, and error messages.

- **Undo** and **Redo** buttons are available.

7. In the **Add next** panel or on the workflow canvas, select **Condition** .







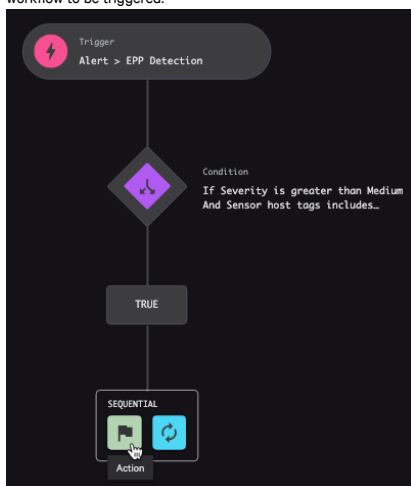
8. In the panel, make your selections to define the condition and click **Next**.

For info about conditions, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].

**Note:** Conditions with a **Matches** or **Does not match** operator provide a blank field for you to enter your value. You must include an asterisk (\*) at the start, end, or both of the value entered.

9. Refine the trigger further by adding more conditions. Click **Next** when all conditions have been added.

In the example, an additional condition has been added to show how multiple conditions are grouped to visually indicate that all must be met for the workflow to be triggered.



10. In the **Add next** panel or on the workflow canvas, select **Action**  to add the action to be performed when this condition is met.

11. Select and define an action and click **Next**.

**Note:** Some workflow actions aren't available based on their support for Flight Control environments. For example, Real Time Response (RTR) custom scripts are available in workflows, but only to 1 CID at a time. So even if you assign a custom script to the workflow for a parent CID and apply that workflow to child CIDs, the action for the workflow is only available in the workflow at the parent level.

**Note:** Workflow actions that require specific subscriptions are only available to the child CIDs with those subscriptions.

For an introduction to actions, see [Workflow actions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p7d19355].

For more info about how to set up some of the action options, see these topics:

- [Manage on-demand workflows that are used as actions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#adccbd4]
- [Create and manage actions based on event queries](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#t1e23404]
- [Create actions in Foundry that use APIs for on-premises tools](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d434618d]

12. Review the workflow in the panel and canvas. With a trigger, conditions, and actions defined, this is a complete workflow. If the **Issues** panel opens, resolve any warnings or errors that you see before continuing.

Click **Save and exit**.

- Enter a name and description for the workflow.
- Optional. Set **Status** to **On**.

13. Optional. Define workflow output to save data for later use. For more info, see


[Save data using workflow output](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c0b19bdf].

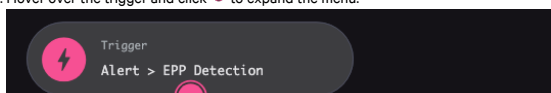
14. Click **Save and exit**.

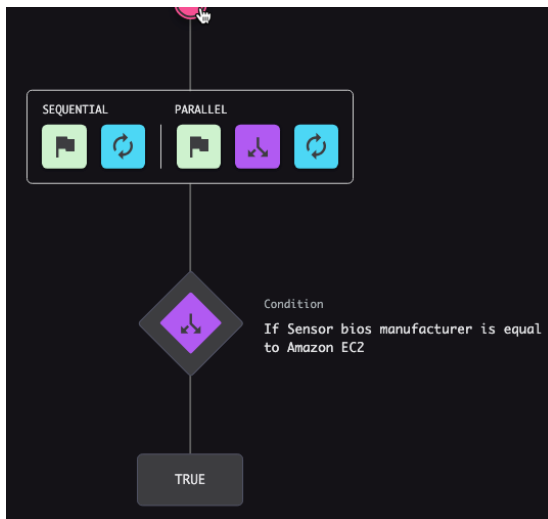
## Add sequential or parallel elements to a workflow

Add sequential or parallel elements to your workflows by adding them to the canvas and defining them.

**Note:** When a workflow contains an action to update the triggering object, that change isn't reflected in subsequent actions within the workflow execution. For example, if one action within a workflow adds a tag to an incident, and another action sends a Slack message with incident details, including tags, the Slack message won't contain the tag added by the workflow.

1. Hover over the trigger and click  to expand the menu.





2. Select an option:

**Note:** You can define an action based on an event query. For more info, see [Event queries, or saved searches, as workflow actions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#cd22d93e].

**Note:** Whenever you have parallel loops, make sure they're iterating through different data sources.


- Sequential action: Add an action to the current workflow branch
- Sequential loop: Add a loop to the current workflow branch
- Parallel action: Create a branch with an action
- Parallel condition: Create a branch with a condition
- Parallel loop: Create a branch with a loop

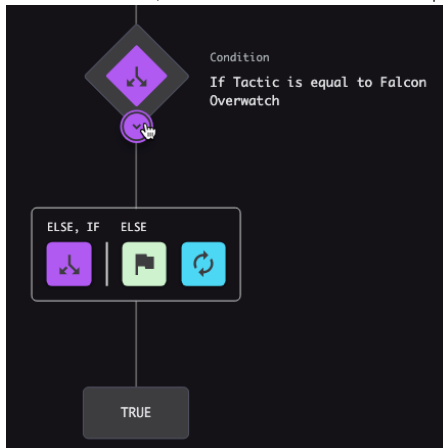
**Tip:** If you pick an option but then decide you want a different one, click [Change](#).

3. Define and save your elements.

## Add ELSE IF conditions and ELSE actions

**ELSE, IF** conditions and **ELSE** actions let you create separate workflow branches for different conditions.

1. On the workflow canvas, hover over the condition and click  to expand the menu.



2. Select an option:

- Else If condition: Creates a condition branch that is checked if the condition you branch from is *not* met  
For info about conditions, see [Workflow conditions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#yd588dba].
- Else action: Creates an action that is performed if the condition you branch from is *not* met
- Else loop: Creates a loop that is performed if the condition you branch from is *not* met




**Tip:** If you pick an option but then decide you want a different one, click [Change](#).

3. In the details panel, define and save your elements.

## Add loops

Add loops to your workflows by adding them to the canvas and defining them in the details panel. Loops can be sequential or parallel. You can also nest loops. Add loops after triggers, conditions, and actions. The items available to loop through depend on the trigger, condition, or action.

For more info, see [Looping in workflows](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#k9bd6faf].

1. Hover over the trigger, condition, or action where you want to add a loop and then, to expand the menu, click , , or .

2. Select an option:

**Note:** Not all of these options appear all the time. They appear based on context.


**Note:** Whenever you have parallel loops, iterate through different data sources.

For info about the **Continue workflow on loop iteration failure** option, see

[Loop iteration failure \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#s5eef7d0\]](#).

- Loop: Create a **While** loop or a **For each** loop to iterate through items
- Parallel loop: Create a branch with a loop to run at the same time
- Sequential loop: Add a loop to execute in a specific order in the current workflow branch

**Tip:** If you pick an option but then decide you want a different one, click **Change**.

3. To nest a loop inside another loop, hover over the beginning of the loop where this new loop should be nested and click .

4. Define the loops and finish defining the workflow.

**Note:** The first condition or action that you set in a loop must be based on items the loop is iterating through. If not, you can get an error that prevents you from continuing to the next step. For example, with a trigger of **Alert > EPP Detection** followed by a loop that iterates through **Host tags**, the first condition or action in the loop should be defined using **Host tags instance**. However, if you set that first condition or action to be the **Contain device** action, you'll get an error because the **Contain device** action doesn't involve the items being looped through.

## Duplicate a workflow

Instead of building workflows from scratch, you can duplicate and modify existing workflows.


**Note:** For Flight Control CIDs, you can duplicate a workflow only within the currently selected CID.


For info about workflow elements and Fusion SOAR, see

[Create a workflow \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o4bc3767\]](#).

### Duplicate the workflow

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).

2. Click **Open menu**  for the workflow you want to duplicate.

You can also click **Open menu**  for the workflow you want to duplicate on the **Execution log** tab or in the **Execution details** panel of the workflow.

3. Click **Duplicate workflow**.

The duplicated workflow with "Copy of" prefixed to the **Workflow name** opens in Fusion SOAR.

4. Update and rename the duplicated workflow.

For info about how to edit workflow elements in Fusion SOAR, see

[Edit a workflow element \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#jb86e89d\]](#).

5. Click **Save and exit** to save the updated workflow.

**Note:** Duplicated workflows are not saved until you click **Save and exit**.

**Tip:** You can click the name of a workflow on the **All workflows** tab to manage, edit, or duplicate the workflow and view details such as **Version history**, **Total executions**, and, for duplicated workflows, **Duplicated from**.

## Export or import a workflow

You can export and import Fusion SOAR workflow definition files. With this ability, you have these options:



- Manage your workflows using source control
- Share workflows between CIDs more easily

### Export a workflow

To export a workflow definition:

**Note:** Exports are not available for workflows created from Falcon Foundry workflow templates.

1. Start an export from either of these locations:

- The **All workflows** tab
  - a. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
  - b. Click **Open menu**  for the workflow you want to export.
  - c. Click **Export workflow**.
- The **Workflow details** page
  - a. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
  - b. Click the name of the workflow you want to export.
  - c. Click **Open menu**  and then select **Export workflow**.

- 2. Enter a name and location for the workflow definition file.
- 3. Save the file.  
The workflow definition is saved to a YAML file.
- 4. If the workflow contains email addresses, some of the addresses are changed to example@crowdstrike.com. Customize the addresses for use with a given CID when you import the workflow through the Falcon console. Otherwise, if you plan to use the API to import the workflow, edit the YAML file now to use addresses with email domains that are valid for the CID where you plan to import the workflow. For more info about the API, see [Fusion SOAR Workflow APIs \[documentation/page/z028de1a/fusion-workflow-apis\]](#).
- 5. If the workflow is triggered based on a schedule that includes an end date, edit the YAML file to remove the line that contains end\_date. Otherwise, importing the file can cause an error if the date is in the past.

Import a workflow

You can only import workflow definitions when these requirements are met:

- The workflow doesn't use actions from a third-party plugin or from Falcon Foundry
- If the workflow has any first-party actions or triggers that require a Falcon subscription, that subscription is available in the CID where the workflow is being imported
- If the workflow requires a plugin from the CrowdStrike Store, the plugin is available in the CID where the workflow is being imported

An import might not be successful based on the availability of external dependencies in a given CID environment.

Custom event queries are specific to the CIDs where they were created. Consequently, if a workflow is from a different CID and contains actions based on custom event queries, you can't import it.

To import a workflow definition:

- 1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).
- 2. Click **Create workflow**.
- 3. For Flight Control environments, users with the Falcon Administrator role in the parent CID can choose which child CIDs the workflow applies to. Those workflows can trigger for any child CID, even if the users don't have permissions in the child CIDs. However, users with the Workflow Author role in a parent CID can only create workflows that trigger for that parent CID: Those users aren't able to create workflows for that CID's child CIDs.

**Tip:** When creating a workflow that you will use with multiple CIDs, you can set up the workflow in a child CID to test and verify it first by triggering it and checking the execution log in the parent CID. Then, after verification, apply the workflow to more CIDs. For more info, see [Edit a workflow's CIDs in a Flight Control environment \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u8fc34f6\]](#).

When applying a workflow to child CIDs, you have these options:

Option	Description
All child CIDs	Applies the workflow to all current and future child CIDs. However, you can explicitly exclude some CIDs.
Specific CIDs	Applies the workflow only to the CIDs you specify.
Only current CID {CID_name}	Applies the workflow only to the current CID.

If you define a workflow in a parent CID, the definition of the workflow, its execution log, and its audit log are only shown in the parent CID and not any of its child CIDs.

- 4. Click **Import workflow** and then click **Next**.
- 5. Click **Upload workflow file**, and then browse to and select the workflow definition file to use.
- 6. Click **Import workflow**.
- 7. Enter a unique name for the new workflow.
- 8. Click **Import workflow**.  
If the import is successful, a page appears that allows you to customize the workflow.

**Note:** This customization page appears even if no customization or configuration is needed.

- 9. Click **Customize workflow**.
  - If no changes are needed, click **Continue**.
  - If the workflow has actions that need configuration, you are stepped through each item that you must configure. When you complete the required steps, click **Continue**.
  - If the workflow includes a trigger based on an inbound webhook and the trigger uses authentication, enter the authentication information.
- 10. Click **Save and exit**.

Resolve workflow issues

The Fusion SOAR **Issues** panel shows warnings and errors for invalid workflows. Whether creating or editing a workflow, you need to resolve any warnings and errors before saving. If the save button is grayed out, undo your changes until you return to a valid workflow, and then save.

Here are some common user actions that generate warning or error messages:

- Changing the trigger type
- Replacing an action node with a different action node creates an error that requires undefined properties to be removed or replaced
- Deleting an action node when the outputs are referenced as inputs in downstream nodes results in undefined properties

## Edit a workflow element

You can edit elements while creating or editing a workflow.

In some cases, more details about editing are available:

- [Edit an on-demand workflow that is used as an action \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y61d0d02\]](#)
- [Edit an action based on an event query \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ye9dafa6\]](#)

In general though, complete these steps:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
2. Click the name of the workflow that has items to edit.
3. Click **Edit**.
4. Select an element on the canvas.  
A panel opens so you can edit the element.
5. Make updates in the panel and click **Next**.  
To exit the editing view for an element, click **Cancel**.
6. Optional. Define or edit workflow output to save data for later use by clicking **Workflow details**. For more info, see [Save data using workflow output \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c0b19bdf\]](#).
7. Click **Save and exit**.
8. Click **Update workflow**.

## Delete a workflow element

You can delete elements while creating or editing a workflow.

In some cases, more details about editing are available:

- [Delete an on-demand workflow that is used as an action \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#x7d0445d\]](#)
- [Delete an action based on an event query \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r3d8c3d2\]](#)

In general though, complete these steps:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
2. Click the name of the workflow that has items to delete.
3. Click **Edit**.
4. Hover over an element on the canvas to delete.

**Note:** You cannot delete a trigger, but you can change it. Click the trigger and then click **Back** ← to go to trigger selection.

5. Click **Delete**  .

## Run a workflow on demand

Depending on how a workflow is set up, you might be able to run the workflow on demand. These workflows are available for running on demand:

- On demand workflows  
These workflows use an **On demand** trigger. For more info, see [Workflow triggers \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z19234b1\]](#).
- Scheduled workflows  
These workflows use a **Schedule** trigger. For more info, see [Workflow triggers \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z19234b1\]](#).  
You can run scheduled workflows on demand instead of waiting for the next scheduled run.

**Note:** If you run a scheduled workflow on demand, the next scheduled run is not skipped. The schedule remains the same.

If you're using these types of workflows in Flight Control environments, be aware of these behaviors:

- As with other workflows, you can choose which child CIDs an on-demand workflow applies to. The action to run the on-demand workflow is available only in the parent CID. Other workflows in the parent CID can use that action and run the action against all the children CIDs. However, the action is not directly available to workflows in the child CID.
- If an on-demand workflow has been applied to several child CIDs, whenever you click **Execute workflow**, you must select exactly one CID where the workflow will run.

Fusion SOAR provides several ways you can run a workflow on demand now or later, as explained in these sections:

- [From the page that lists all workflows \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#a06afbe6\]](#)
- [From the execution log \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#df673a16\]](#)
- [From the execution log, using a workflow's execution details \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o105de59\]](#)
- [From a workflow's page \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#je08623a\]](#)
- [From an action in another workflow \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c274a4dd\]](#)
- [Using an API call \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#h6886afda\]](#)

- [Using an API call \(documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#10ef382\)](#)
- [As a response action in an incident \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#10ef382\]](#)

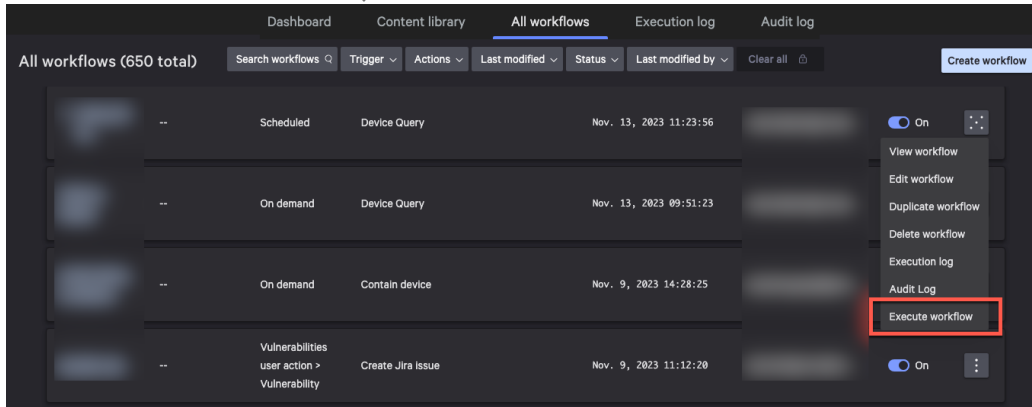
## From the page that lists all workflows

Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).

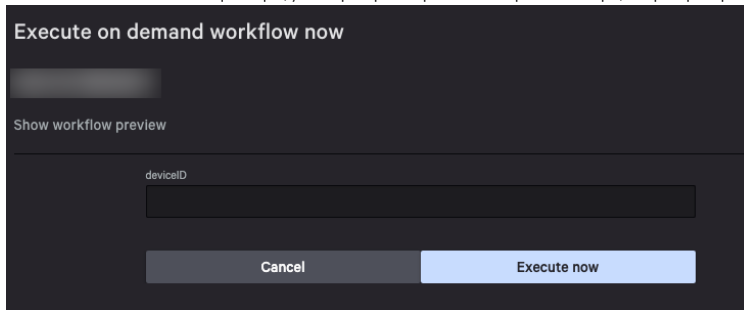
You can immediately run workflows that meet all of these requirements:

- The **Trigger** column contains **Scheduled** or **On demand**
- The **Last modified by** column contains an email address
- The **Status** column shows **On**

To run such a workflow immediately, click **Open menu**  and select **Execute workflow**.



For on-demand workflows that require input, you are prompted to provide that input. For example, this prompt requests a deviceID.



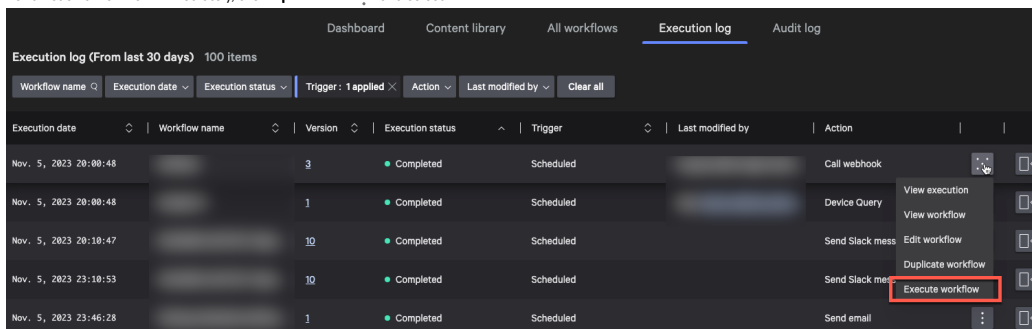
## From the execution log

Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#) and click the **Execution log** tab.

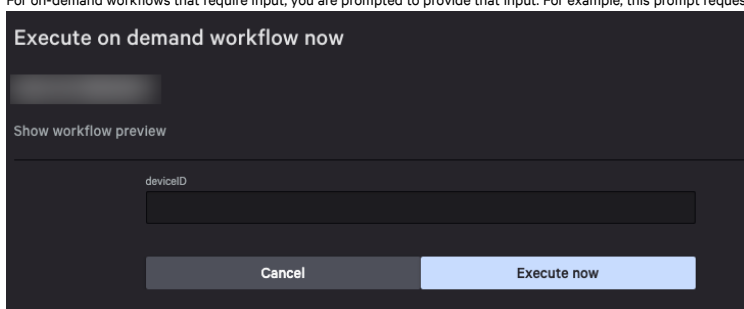
You can immediately run workflows that meet all of these requirements:

- The **Trigger** column contains **Scheduled** or **On demand**
- The **Last modified by** column contains an email address

To run such a workflow immediately, click **Open menu**  and select **Execute workflow**.



For on-demand workflows that require input, you are prompted to provide that input. For example, this prompt requests a deviceID.



From the execution log, using a workflow's execution details

Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#) and click the **Execution log** tab.

You can immediately run workflows that meet all of these requirements:

- The **Trigger** column contains **Scheduled** or **On demand**
- The **Last modified by** column contains an email address

To run such a workflow immediately, click **View execution details**, click **Open menu**, and select **Execute workflow**.

DashboardContent libraryAll workflowsExecution logAudit log

Execution log (From last 30 days) 3 items

Workflow nameExecution dateExecution statusTrigger: 1 appliedActionLast modified by: 1 appliedClear all

Execution date	Workflow name	Version	Execution status	Trigger	Last modified by	Action
Nov. 16, 2023 12:06:48		6	Completed	Scheduled	Device Query, {	View execution details
Nov. 15, 2023 16:25:39		1	Completed	Scheduled	Device Query, Send email	
Nov. 15, 2023 16:25:34		1	Completed	Scheduled	Device Query, Send email	

For on-demand workflows that require input, you are prompted to provide that input. For example, this prompt requests a deviceID.

Execute on demand workflow now

Show workflow preview

deviceID

CancelExecute now

From a workflow's page

Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).

You can immediately run workflows that meet all of these requirements:

- The **Trigger** column contains **Scheduled** or **On demand**
- The **Last modified by** column contains an email address
- The **Status** column shows **On**

To run such a workflow immediately, click the workflow name, click **Open menu**, and select **Execute now**.

View only - Edit

Manage workflow

Trigger

Scheduled

Action

Device Query

Action

Send email

Execute now

For on-demand workflows that require input, you are prompted to provide that input. For example, this prompt requests a deviceID.

Execute on demand workflow now

Show workflow preview

deviceID

CancelExecute now




From an action in another workflow

When you create a workflow that includes an on-demand trigger, Fusion SOAR automatically creates an action based on that workflow. To start that workflow from another workflow, add the action based on the workflow in the other workflow.

For example, assume whenever you contain a device, you must also create a Jira ticket to follow up. You could create an on-demand workflow that does both the containment and the Jira ticket creation. Other workflows could then trigger the on-demand workflow rather than do the containment directly.

**Note:** When you include one of these actions in a workflow, you introduce dependencies on the workflow that is the basis for the action. For info about how to manage these workflows, see [Manage on-demand workflows that are used as actions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ladccbd4].

Call an on-demand workflow from a workflow:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].
2. Verify that the on-demand workflow that the action is based on is enabled.  
For an action based on a workflow to be available, its workflow must be enabled.
3. Click **Create workflow** or click the name of workflow where you want to add the action.  
If you're editing a workflow, click **Edit**.
4. In the workflow canvas, hover over the trigger, condition, or action where you want to add the action and then, to expand the menu, click , , or .
5. Add the action for the on-demand workflow.

**Note:** The name of the action is the name of the on-demand workflow.

6. Define or edit the rest of the workflow as needed and save it.

## Using an API call

If a workflow uses a **Schedule** trigger or an **On demand** trigger, in addition to running the workflow on demand from the Falcon console, you can use the `/workflows/entities/execute` API endpoint.

For more info about this API, see [Fusion SOAR Workflow APIs](#) [/documentation/page/z028de1a/fusion-workflow-apis].

## As a response action in an incident

Execute a Falcon Fusion SOAR on-demand workflow in the incident workbench, in the **On-demand workflows** section of an applicable entity's summary panel. For more info, see [Incident Investigation](#) [/documentation/page/r2f1bac9/xdr-incident-investigation].

# Create and manage triggers based on inbound webhooks

The **Add trigger** panel provides an **Inbound Webhook** option. When you use this option, Fusion SOAR generates a unique webhook URL for the given workflow. External systems can then use that URL to send JSON payloads and trigger the workflow automatically.

For more info, see [Custom triggers based on inbound webhooks](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#mf5cdeeb].

For info about requirements for these triggers related to subscriptions, CrowdStrike clouds, and roles, see [Requirements](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#l8d760c7].

For info about working with these actions, see these topics:

- [Create a trigger based on an inbound webhook](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ebd0980c]
- [Edit a trigger based on an inbound webhook](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y066f03e]  
[View the URL for an existing trigger based on an inbound webhook](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p4f67dd]
- [Inspect webhook issues](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#gb0c3974]

## Create a trigger based on an inbound webhook

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].
2. Click **Create workflow**, select how to create the workflow, and continue to the workflow canvas.
3. In the **Add trigger** panel, click **Inbound webhook**.
4. Enter a name and optionally a description.
5. Select the HTTP method.
6. Select the authentication type.  
Anyone trying to start the workflow from an external system will have to provide any authentication info entered here to trigger the workflow.
  - Basic: Enter a username and password.
  - HMAC: The hash algorithm is automatically selected. Enter a secret, select the signature encoding, and enter the signature header name, which is the header in the request, such as `X-Signature`, that will contain the signature.  
The message digest is used in forming the signature. The request body is always included in this digest. You can also include **Timestamp** and **Message ID** in the digest.  
When you enable HMAC authentication, for each request from a tool outside of Fusion SOAR, you must use the same hash algorithm, secret, and encoding to compute the signature that you place in the specified header.
  - API key: Enter a key and select whether the key will be found in the requests' header or body.
7. Complete the fields for the chosen authentication type.
8. If you'd like to set a particular response body or response code, click **Add advanced configuration** and enter the desired info.  
By default, the response body is empty and the response code is 200.



#### 9. Click **Generate URL**.

Fusion SOAR creates the URL that you'll use in an external system to trigger the current workflow.

The generated webhook endpoint URL is specific to the workflow where its trigger was created. You can't use the URL with other workflows.

10. Copy the URL to immediately add to the external system or to save for later.

11. Create a JSON schema for the request payload to enforce a format for the payload.

Enter a sample payload and then click **Generate schema**.

**Tip:** To avoid enforcing a format and just pass through the entire payload, enter empty curly braces, {}, as the sample payload. When you use this technique, the workflow execution log shows the entire webhook payload. Also, if you use this technique, you can use data transformation functions to extract or transform the webhook payload. For more info, see [CEL expressions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ie35e6c4].

12. Click **Next** to add the trigger.

13. Define the rest of the workflow.

## Edit a trigger based on an inbound webhook

You can change the trigger's name, description, authentication configuration, advanced configuration, and payload schema.

For info about the authentication options, see

[Create a trigger based on an inbound webhook](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ebd0980c].

**Note:** If you want a different webhook URL with the same trigger definition, create a new workflow using the desired trigger definition and generate a new URL.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].

2. Find and click the workflow.

3. Click **Edit**.

4. Click the trigger.

5. Make the desired changes.

6. Click **Next**.

## View the URL for an existing trigger based on an inbound webhook

If you have already created a trigger based on an inbound webhook and want to see the URL again to copy it:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].

2. Find and click the workflow.

3. Click the trigger.

A panel opens.

4. In the panel, locate the **Webhook URL** entry.

## Inspect webhook issues

To inspect issues with a webhook, you have these options:

- Execution log

Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion] and click the **Execution log** tab.

- Events in LogScale

Go to [Investigate > Search > Advanced event search](#) [/investigate/search] and run this query to get all of the webhook ingestion errors:

```
#repo = fusion
| "#event_simpleName" = FusionWorkflowEvent
| "webhook_ingest_log_type" = IngestError
```



Possible issues and their causes:

- **Authentication:** The request failed authentication.

Common causes:

- Missing or incorrect API key
- Invalid Basic Auth credentials
- Invalid HMAC signature because of the wrong secret, the encoding, or the message content
- Missing required headers, such as X-Signature, X-Timestamp, or X-Message-ID

Log subtype: Authentication

- **ParsingPayload:** The request body could not be parsed.

Common causes:

- Malformed JSON, possibly a syntax error
- Incorrect Content-Type header; should be application/json
- Empty or improperly encoded body

Log subtype: ParsingPayload

- **Internal:** An unexpected error occurred during webhook processing.

...or an exception that occurred during message processing.

Common causes:

- Server-side exception or infrastructure issue
- Workflow misconfiguration
- Fusion service failure

Log subtype: Internal

- **Schema Binding Failure:** Data available in a condition or an action does not match the schema.

Common causes:

- Missing required fields, such as `project.id`
- Incorrect field names or data types
- Mismatched object or array structures

- **Replay Protection Failure (HMAC):** The request was rejected because of a replay detection.

Common causes:

- Expired or future-dated timestamp
- Time sync issues between sender and Fusion

- **Missing Required Headers:** A required header for validation was not included.

Common causes:

- Missing signature header, `X-Signature`
- Missing timestamp or message ID when HMAC replay protection is enabled

- **Unsupported HTTP Method:** The request used an invalid HTTP method.

Common causes:

- Sending GET instead of POST
- Endpoint expecting JSON but receiving form data

- **Rate Limiting:** The system rejected the request because of the excessive frequency.

Common causes:

- Too many requests sent in a short period, which is rare in typical use

- **Misconfigured Webhook Trigger:** The webhook trigger is not properly set up.

Common causes:

- Missing or invalid webhook URL
- Trigger was deleted or is inactive
- Workflow is not enabled

## Manage action input, action output, and on-demand triggers

To define the input and output of various actions, use JSON schema to indicate the expected formats. Similarly, for on-demand workflows that require input, use JSON schema to prompt for that input.

The schema is a JSON structure formatted according to draft 7 of the JSON Schema standard. For more info, see [JSON Schema \[https://json-schema.org/\]](https://json-schema.org/).

For more info, see these topics:

- [View a schema \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o64b29aa\]](#)
- [Understand a schema \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#t713a4b8\]](#)
  - [Basic properties \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#w35a121e\]](#)
  - [Format options, including usage for CrowdStrike data types \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#hce2e631\]](#)
- [Use an array \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#v38028db\]](#)
- [Set a field to populate \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o905571a\]](#)
  - [Example: List host groups in the trigger for an on-demand workflow \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#v0f0b0c7\]](#)
  - [Example: List hosts when adding an action based on an event query \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d0acde98\]](#)
- [Require any of a set of possible values \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#zaf8d4da\]](#)

### View a schema

To chain actions together and build better conditions, you must understand the input and output of the actions you're using. By viewing a schema, you see which fields are required for its action, the properties for those fields, and their descriptions.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).
2. If you're creating a workflow, click **Create workflow**, select how to create the workflow, continue to the workflow canvas, and add an action. If you're editing a workflow, find and click an existing workflow that has actions you want to see schemas. Click **Edit** and select the action with the schema you want to see.

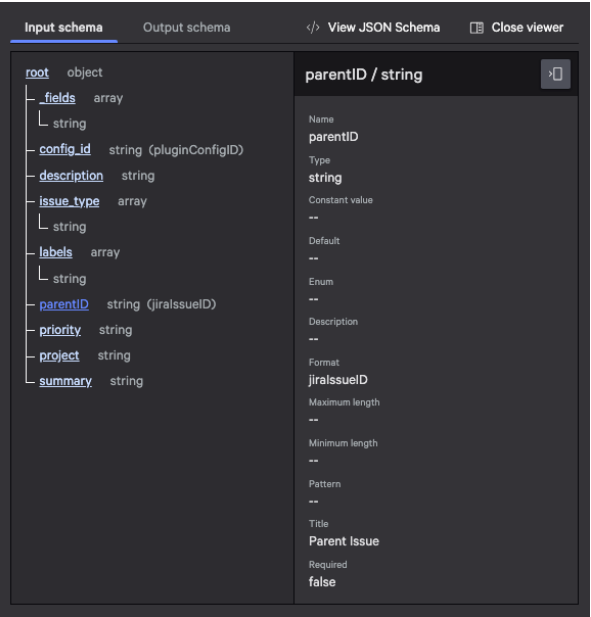
**Tip:** If you create or edit a workflow with an **On demand** trigger, you can see sample schemas by clicking **Search schema samples**.

3. Click **View schema**.

The schema viewer opens, showing a form-based version of the schema.  
To view the raw JSON, click **View JSON Schema**.

**Note:** The form-based schema builder is not available when the JSON is more complex and can't be represented in the form-based version.

The form-based schema builder shows the fields and their types. To see a field's properties, click the field. If a field is an array, click **View JSON Schema** to see its properties.  
This image shows the properties for the `parentID` field.



Understand a schema

Each field in a schema can have numerous properties. Edit properties using the form-based schema builder or directly in the JSON. Use properties to provide context for a field, restrict inputs, and more.  
For more info, see these topics:

- [Basic properties](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#w35a121e]
- [Format options, including usage for CrowdStrike data types](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#hce2e631]

Basic properties

Here are some of the basic properties you might encounter as you work with JSON Schema. For more info about the JSON Schema standard, see [Introduction to JSON Schema](#) [https://json-schema.org/learn/getting-started-step-by-step#intro].

Property as seen in the JSON viewer	Corresponding property	Description
Name	name	Names the field.
Type	type	<p>Defines the data type that a field must use.</p> <p>Each data type has specific properties.</p> <p>string is the most common scalar type to specify a text field.</p> <p>Here are other properties you might want to use with the string type:</p> <ul style="list-style-type: none"><li>• <code>maxLength</code>: the maximum number of characters a text field can contain</li><li>• <code>minLength</code>: the minimum number of characters a text field can contain</li><li>• <code>pattern</code>: a regular expression to define the format of the text field</li><li>• <code>format</code>: a predefined format for the text field, such as email or date</li></ul>
Constant value	const	<p>Restricts a property to a single value.</p> <p>Applicable across all data types.</p>
Default	default	<p>Sets the default value of the field.</p> <p>Fusion SOAR uses this value unless the user provides a different value.</p> <p>Applicable across all data types.</p>
Enum	enum	<p>Specifies a list of values a property can have.</p> <p>For example, this list validates alert severities:</p> <div><pre>{   "enum": ["low", "medium", "high", "critical"] }</pre></div> <p>A value of <code>info</code> is not valid because it isn't in the list.</p>

Description	description	Indicates the purpose of the field and provides context to the end-user when they are viewing the schema. Applicable across all data types.
Format	format	Specifies the format the data field must use. For more info, see <a href="#">Format options, including usage for CrowdStrike data types</a> [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#hce2e631] .
Maximum length	maxLength	Sets the maximum length of the string that can be entered. Applicable to string types only.
Minimum length	minLength	Sets the minimum length of the string that can be entered. Applicable to string types only.
Maximum value	maximum	Sets the maximum value for an integer. Applicable to integer and number types only.
Minimum value	minimum	Sets the minimum value for an integer. Applicable to integer and number types only.
Pattern	pattern	Specifies a regular expression pattern to ensure that a value matches a specific format, such as an email address or phone number. Applicable to string types only.
Title	title	Specifies the display label shown in forms that appear in the Falcon console.
Required	required	Indicates which child fields in an object are required. The required property can be at these locations: <ul style="list-style-type: none"><li>• At the same level as the properties field</li><li>• In an anyOf property used at the same level as the properties field</li></ul>
Examples	examples	Defines reference values
Properties	properties	Specifies and validates the child fields of an object type. In the key-value pairs in the object, each key names a field and the corresponding value defines JSON schema to govern that field.
Items	items	Specifies and validates the child fields of an array type. In the key-value pairs in the array, each key names a field and the corresponding value defines JSON subschema to govern that field.

Format options, including usage for CrowdStrike data types

The format field supports these commonly used values. Not all of these values are in the JSON schema specification.

Format value	Description
date	Date
date-time	Date time
domain	Domain
email	Email
ipv4	IP version 4
ipv6	IP version 6
md5	MD5 hash
sha256	SHA256 hash
time	Time
uuid	UUID
url	URL

CrowdStrike also uses the standard format field to indicate the type of data a field represents, which is critical for some workflow actions. For example, to contain a device you need its sensor ID because the contain action requires a sensor ID as input. A sensor ID field is "type": "string", but you also need "format": "aid" in the JSON. As you build a workflow, you can only select the contain action if the data you are working with has a type with format "aid". So to build an on-demand workflow that takes a sensor ID and contains a device, the input JSON schema must set the format to "aid".

The format field supports these values that are specific to CrowdStrike to indicate types of data.

Format value	Description
--------------	-------------

Format value	Description
aid	Sensor ID
deviceTag	Device Tag
hostGroupID	Host group ID
hostGroupName	Host group name
hostname	Host name
incidentID	Incident ID
investigatableID	Alert ID
localfilepath	Local file path
oktaUserID	Okta user ID
platform	OS platform
responseUserID	User name
rtrFileName	RTR put file name

### Use an array

While the form-based schema builder simplifies many aspects of editing a JSON schema, it does not support arrays of scalars. To define or edit an array of scalars, you must edit the JSON directly. However, you can use the schema builder with arrays of objects.

Here is an example of an array of sensor IDs.

```
{
  "type": "object",
  "items": {
    "type": "object",
    "properties": {
      "devices": {
        "description": "A list of sensor IDs",
        "type": "array",
        "title": "Sensor IDs",
        "items": {
          "format": "aid",
          "type": "string",
          "pattern": "^[A-Za-f0-9]{32}$"
        }
      }
    },
    "required": [
      "devices"
    ]
  }
}
```

Here's an example of JSON that complies with this JSON schema:

```
{
  "devices": ["0302188e5dc9490e861af9c40bc23c15", "94e8963899f74ffdb330065738faa35a"]
}
```

However, this example JSON does not comply with the schema. Neither ID is the required 32 characters, plus the second ID contains special characters.

```
{
  "devices": ["bc23c15", "%7!4(ffdb33006a35a)"]
}
```

### Set a field to populate

To set a field to populate with valid options, use the `x-cs-pivot` property. This property specifies a set of data to look up as someone, for example, configures a field while using a workflow or adds an action based on an event query to a workflow.

**Note:** This feature is only available when you are editing the JSON schema directly. It isn't supported using the form-based schema builder.

To get various types of data, here are entity values you can use with `x-cs-pivot`:

Entity value	Description
devices.groups.name	Host group names
devices.hostname.raw	Host names in the CID
devices.platform_name	Platforms used by the CID's hosts

patterns.technique	MITRE techniques
patterns.tactic	MITRE tactic names
users.email	Falcon user emails
users.id	Falcon user IDs
devices.tags	Device tags
incidents.tags	Crowdscore Incident tags

For more info, see these topics:

- [Example: List host groups in the trigger for an on-demand workflow \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#v0f0b0c7\]](#)
- [Example: List hosts when adding an action based on an event query \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d0acde98\]](#)

### Example: List host groups in the trigger for an on-demand workflow

By setting up a populated list in the trigger for an on-demand workflow, you can simplify creation of the workflow and running the workflow.

This schema looks up host groups for the current CID by using x-cs-pivot with the devices.groups.name entity value:

```
{
  "type": "object",
  "properties": {
    "hostGroup": {
      "type": "string",
      "description": "A CrowdStrike host group name",
      "format": "hostGroupName",
      "title": "Host Group",
      "x-cs-pivot": {
        "entity": "devices.groups.name"
      }
    }
  },
  "required": [
    "hostGroup"
  ]
}
```

Assume you set up an on-demand workflow and include that JSON in the schema for the **On demand** trigger, as shown here.

The screenshot shows the 'JSON Schema editor' interface. On the left, there's a 'Trigger' section with a lightning bolt icon and the text 'On demand'. The main area is a code editor showing the JSON schema. The schema is as follows:

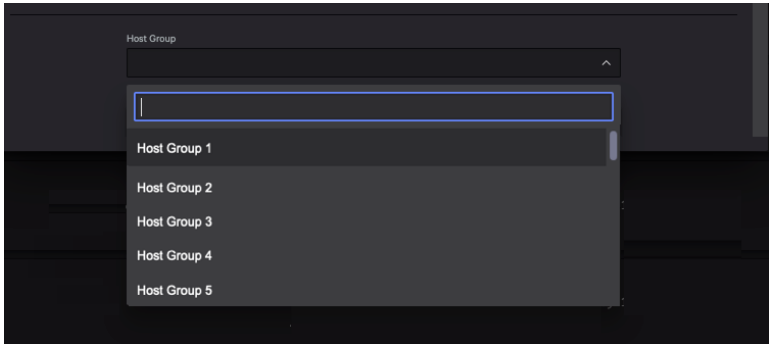
```
1 {
2   "type": "object",
3   "properties": {
4     "hostGroup": {
5       "type": "string",
6       "description": "A CrowdStrike host group name",
7       "format": "hostGroupName",
8       "title": "Host Group",
9       "x-cs-pivot": {
10        "entity": "devices.groups.name"
11      }
12    }
13  },
14  "required": [
15    "hostGroup"
16  ]
17 }
```

Then, you can add several conditions to the workflow, creating different conditions by setting **Parameter** to **Host Group** and selecting different host groups from the populated list.

The screenshot shows the 'Condition' configuration panel. It has a 'Parameter' dropdown set to 'Host Group', an 'Operator' dropdown set to 'is equal to', and a 'Value' dropdown. The 'Value' dropdown is open, showing a list of host groups: Host Group 1, Host Group 2, Host Group 3, Host Group 4, and Host Group 5.

In addition, whenever someone runs the workflow, they are prompted to provide the required input, which they then select from the list.

The screenshot shows a dialog box titled 'Execute on demand workflow now'. It contains the text 'My on-demand workflow' and a link that says 'Show workflow preview'.



Example: List hosts when adding an action based on an event query

Assume you have a workflow action based on a custom event query that has an argument based on CrowdStrike hosts. If the action's event query defines a parameter using ?hosts to allow user input for the host and the input schema has x-cs-pivot defined as in the example, the workflow author can choose a host name from a dropdown list when adding the action based on the event query to a workflow. The list shows all the hosts in the CID. For more info about event queries, see [Event queries, or saved searches, as workflow actions \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c022d93e\]](#).

```
{
  "type": "object",
  "$schema": "https://json-schema.org/draft-07/schema",
  "required": [
    "hosts"
  ],
  "properties": {
    "hosts": {
      "type": "string",
      "title": "Hosts",
      "x-cs-pivot": {
        "entity": "devices.hostname.raw"
      }
    }
  },
  "description": "Generated request schema"
}
```

Require any of a set of possible values

If a user must enter any of the required fields, use anyOf in your input JSON schema.

**Note:** This feature is only available when you are editing the JSON schema directly. It isn't supported using the form-based schema builder.

In this example focusing on the anyOf portion, if the user enters any of the required fields, the entry is accepted:

```
"anyOf": [
  {
    "required": [
      "HostNames"
    ]
  },
  {
    "required": [
      "tags"
    ]
  },
  {
    "required": [
      "HostGroups"
    ]
  },
  {
    "required": [
      "aids"
    ]
  }
]
```

In the context of the full input schema, the anyOf portion is at the end of the schema:

```
{
  "properties": {
    "HostGroups": {
      "items": {
        "properties": {
          "HostGroup": {
            "type": "string",
            "title": "Host group",
            "format": "hostGroupName",
            "x-cs-pivot": {
              "entity": "devices.groups.name"
            }
          }
        }
      }
    },
    "type": "object"
  },
  "anyOf": [
    {
      "required": [
        "HostNames"
      ]
    },
    {
      "required": [
        "tags"
      ]
    },
    {
      "required": [
        "HostGroups"
      ]
    },
    {
      "required": [
        "aids"
      ]
    }
  ]
}
```

```

    },
    "type": "array"
  },
  "HostNames": {
    "items": {
      "properties": {
        "Hostname": {
          "type": "string",
          "title": "Host name",
          "format": "hostname",
          "x-cs-pivot": {
            "entity": "devices.hostname.raw"
          }
        }
      }
    },
    "type": "object"
  },
  "type": "array"
},
"aids": {
  "items": {
    "properties": {
      "aid": {
        "pattern": "^[A-Fa-f0-9]{32}$",
        "type": "string",
        "title": "Inputted Agent Id",
        "format": "aid"
      }
    }
  },
  "type": "object"
},
"type": "array",
"title": "Agent Ids"
},
"tags": {
  "items": {
    "properties": {
      "tag": {
        "type": "string",
        "title": "Grouping tag",
        "x-cs-pivot": {
          "entity": "devices.tags"
        }
      }
    }
  },
  "type": "object"
},
"type": "array"
}
},
"type": "object",
"anyOf": [
  {
    "required": [
      "HostNames"
    ]
  },
  {
    "required": [
      "tags"
    ]
  },
  {
    "required": [
      "HostGroups"
    ]
  },
  {
    "required": [
      "aids"
    ]
  }
]
}
}

```


## Manage on-demand workflows that are used as actions

When you create a workflow that includes an on-demand trigger, Fusion SOAR automatically creates an action based on that workflow. You can then use that action in other workflows to start the original workflow. These dependencies can affect how you edit and delete the original workflow.

### Edit an on-demand workflow that is used as an action



1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).



2. Find the workflow to edit and click **Open**  for that workflow.
3. Select **Edit workflow**.  
If any other workflows use the action based on the on-demand workflow, a list of those workflows appears.
4. Choose one option:
  - If you're not going to edit the trigger, click **Proceed**.
  - If you are going to edit the trigger, in the list of workflows, click **duplicate**.  
The workflow is duplicated and shown in the current browser tab. Edit the duplicate workflow.
5. Complete your workflow edits and save the changes.

## Delete an on-demand workflow that is used as an action

If the action based on the on-demand workflow is used in any other workflows, you must remove that action from the other workflows before you can delete the on-demand workflow.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).
2. Find the workflow to delete and click **Open**  for that workflow.
3. Select **Delete workflow**.  
If any other workflows use the action based on the on-demand workflow, a list of those workflows appears.  
Remove the action from one of those workflows:
  - a. Click a workflow name in the list.  
The workflow opens in a new tab.
  - b. Click **Edit**.
  - c. Hover over the action to delete and click **Delete** .  
A warning appears.
  - d. Click **Delete**.
  - e. Click **Save and exit**.
  - f. Add a comment to describe the change to the workflow for its version history.
  - g. Click **Update workflow**.  
The **All workflows** tab opens again. You can close this browser tab.
  - h. Go back to the original browser tab and click **Go back**.  
The **All workflows** tab opens again.
  - i. Continue to remove the action from other workflows by repeating steps 2 and 3.  
When the action is no longer in any workflow, finding the on-demand workflow and clicking **Delete workflow** deletes the on-demand workflow.

## Create and manage actions based on event queries

For more info about event queries, including example use cases and limitations, see

[Event queries, or saved searches, as workflow actions \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c022d93e\]](#).

For info about related subscriptions, roles, and data sources, see

[Requirements \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#8d760c7\]](#).




For more info about working with these actions, see these topics:

- [Create an event query as an action when you create or edit a workflow \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#v3e2636a\]](#)
- [Use an existing event query as an action \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#t494c7ec\]](#)
- [Edit an action based on an event query \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ye9dafa6\]](#)
- [Remove an action based on an event query from a workflow \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#x3ccddb8\]](#)
- [Delete an action based on an event query \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r3d8c3d2\]](#)
- [Add search results as an attachment \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#j4551f7d\]](#)
- [Create and manage NG-SIEM lookup files \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#w3c18ea6\]](#)
- [Verify the output fields of an event query action \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z1898279\]](#)  
[Verify the event query action output is nonempty before acting on it \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r587e1f4\]](#)
- [Loop through the results of an event query \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o6bc1b01\]](#)
- [Customize an event query action's input and output \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u3794cfd\]](#)
  - [Example: Add a description, a default value, and validation to fields in the input schema \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p2e195be\]](#)
  - [Example: Trim an action's output \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#tf086e05\]](#)

## Create an event query as an action when you create or edit a workflow

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#).

2. Click **Create workflow**, select how to create the workflow, and continue to the workflow canvas—or click the workflow where you want to add the action. If you're editing a workflow, click **Edit**.

3. Hover over the trigger, condition, or action where you want to add an action and then, to expand the menu, click , , or .

4. Select **Action** .

5. In the **Add action** panel, click **Create event query**.  
The **Query builder** appears.

6. Configure the query.

a. Set **Data view** to the data source to query. The available sources depend on your subscriptions and role. For more info, see [Requirements](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#f8d760c7]. Here are the options:

- All: All event data in Falcon, Forensics, IT Automation, and XDR
- Falcon: Endpoint event data and sensor events
- Forensics: Triage data collected by Falcon Forensics
- IT Automation: Event data generated by Falcon for IT
- XDR: XDR event data generated by CrowdStrike and by integrated third parties

b. Enter a name for the query in **Query name**.

c. Optional. Enter a description for the query to help you and others later understand the purpose of the query.

d. Optional. To provide test data to run the query against, click **Upload test data**.

If you're creating an event query action for data that doesn't yet exist, you can upload test data in a JSON file—up to 20,000 records—to verify the query and create the action.

The file you upload must consist of data that uses one of these formats:

- One pair of brackets around one or more pairs of braces:

```
[  
  { },  
  { },  
  { }  
]
```



- One or more pairs of braces, separated by new lines:

```
{ }  
{ }  
{ }
```



Whenever you upload data, click **Run** to refresh query results.

e. Enter your query and run it, modifying it and running it until you get the desired results.

For info about the query language, see

[Get Started with CrowdStrike Query Language](#) [documentation/category/nbbb7a91/event-investigation/get-started-with-crowdstrike-query-language]

**Important:** Schema validation returns an error if certain special characters in the event query output, such as @ and #, are used for property names. If you see this error, create a new field in the query using the assignment operator. For example: myTime := @timestamp. Then, reference the new field so that the schema uses the new field as the property name.

**Tip:** To format query results as a table before sending them in an email or other notification-based action, use the `table()` function. For more info, see [table\(\)](#) [https://library.humio.com/data-analysis/functions-table.html].

f. Click **Continue**.


A new window opens with these tabs: **Query**, **Input schema**, and **Output schema**.

Fusion SOAR generates the schemas based on the query and its results.

**Important:** Be sure that the output includes all the fields you need, especially fields you plan to use in conditions or as action input. Also, fields marked as **required** must be in the output. The action fails with a schema validation issue if any of the required fields are not in the output. Only mark a field as required if you can guarantee the field is always in the output. If the field is dynamically populated, consider leaving all fields as optional and using a condition to check that the field has a value before using the field, as suggested in a later step.

For more info about these schemas, see

[Customize an event query action's input and output](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u3794cfd].

g. Optional. To adjust either of the schemas to better define the requirements you want for the query's input or output, click **Input schema** or **Output schema** and then . When done, click **Save changes**.

**Note:** For the input schema, you can only edit the metadata. You can't edit the schema.

h. Click **Add to workflow**.

By default, the action goes in the **Other (Custom, Foundry, etc.)** group. Later, if you want to add the action to another workflow, search for the action by name or browse for it in this group.

7. Optional. Actions based on event queries place their search results in their output for use with, for example, the **Send email** action and Jira ticket creation. The results are always available in JSON. If you would also like the results in CSV, in the **Add action** panel, select **Export to CSV**. When you choose this option, you can select the exact fields to include in the result using the **CSV header fields to export** field.

**Tip:** Large results might fail a workflow. Selecting the **Only return results as files** option might help avoid those failures. However, the **Full search results** and **Event query results** fields accessed by iterating over the results using a workflow loop are not accessible when using this option.

For more info, see [Add search results as an attachment](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#j4551f7d].





8. Click **Next**.

9. Before you do the rest of the work, make sure you've created your notification channels and events.

9. Define or edit the rest of the workflow as needed, resolve any warnings or errors, and save it.

**Important:** Create a condition to confirm the query found matches and that any required fields are populated. See [Verify the event query action output is nonempty before acting on it](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r587e1f4].

## Use an existing event query as an action

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].
2. If you're creating a workflow, click **Create workflow**, select how to create the workflow, and continue to the workflow canvas. If you're editing a workflow, click the workflow you want to edit to include the action and then click **Edit**.
3. Hover over the trigger, condition, or action where you want to add an action and then, to expand the menu, click , , or .
4. Select **Action** .
5. In the **Add action** panel, find the action based on the query. You can search for it by name or browse for it in the **Other (Custom, Foundry, etc.)** group. In addition, all actions based on event queries include Event Query at the beginning. Because of that convention, you can search for Event Query to find all the actions based on event queries.
6. Select the action.
7. Click **Next**.
8. Create a condition to confirm the query found matches.

Be sure that the output includes all the fields you need, especially fields you plan to use in conditions or as action input. See [Verify the output fields of an event query action](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z1898279].

Also, fields marked as required must be in the output. The action fails with a schema validation issue if any of the required fields are not in the output. Only mark a field as required if you can guarantee the field is always in the output. If the field is dynamically populated, consider leaving all fields as optional and using a condition to check that the field has a value before using the field. See [Verify the event query action output is nonempty before acting on it](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r587e1f4].
9. If the **Issues** panel opens, resolve any warnings or errors you see.
10. Click **Save and exit**.
11. Add a comment to describe the change to the workflow for its version history.
12. Click **Update workflow**.

## Edit an action based on an event query

When you update an action based on a query, you update its query configuration to change its data source, name, the query itself, or its schemas.



If an action is used in multiple workflows, edits you make for one workflow might not be suitable for another. The steps below guide you through this scenario.

To edit an action based on a query:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion] and click a workflow that uses the action.
2. In the workflow, click **Edit**.
3. Click the action to edit.

The **Action** panel shows the action.
4. Click **Manage event query**.

The event query manager opens.

**Note:** By clicking **Back**  instead of **Manage event query**, you can replace the action with another action.
5. Click **Edit** .
- Important:** If other workflows use the action, a dialog provides the names of those workflows. If you click **Proceed**, you can still edit the query configuration. If your changes break any of the affected workflows, you are prompted after you click **Continue** in the next step to cancel, remove the dependencies, or save the changes in a duplicate.
6. Edit the query configuration.

Change one or more of the configuration settings:

  - Set **Data view** to a different data source to query. The available sources depend on your subscriptions and role. For more info, see [Requirements](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#l8d760c7]. Here are the options:
    - All: All event data in Falcon, Forensics, IT Automation, and XDR
    - Falcon: Endpoint event data and sensor events
    - Forensics: Triage data collected by Falcon Forensics
    - IT Automation: Event data generated by Falcon for IT
    - XDR: XDR event data generated by CrowdStrike and by integrated third parties
  - Change the name for the query in **Query name**.
  - Optional. Enter or change a description for the query to help you and others later understand the purpose of the query.
  - Optional. To provide test data to run the query against, click **Upload test data**.

If you're creating an event query action for data that doesn't yet exist, you can upload test data in a JSON file—up to 20,000 records—to verify the query and create the action.

The file you upload must consist of data that uses one of these formats:

    - One pair of brackets around one or more pairs of braces:

• One pair of brackets around one or more pairs of braces:

```
[
  { },
  { },
  { }
]
```

• One or more pairs of braces, separated by new lines:

```
{ }
{ }
{ }
```

Whenever you upload data, click **Run** to refresh query results.

- Change the query and run it, modifying it and running it until you get the desired results.

For info about the query language, see

[Get Started with CrowdStrike Query Language \[documentation/category/nbbb7a91/event-investigation/get-started-with-crowdstrike-query-language\]](#)

**Important:** Schema validation returns an error if certain special characters in the event query output, such as @ and #, are used for property names. If you see this error, create a new field in the query using the assignment operator. For example: myTime := @timestamp. Then, reference the new field so that the schema uses the new field as the property name.

**Tip:** To format query results as a table before sending them in an email or other notification-based action, use the table() function. For more info, see tableQ [https://library.humio.com/data-analysis/functions-table.html].

- Click **Continue**.

A new window opens with these tabs: **Query**, **Input schema**, and **Output schema**.

Fusion SOAR generates the schemas based on the query and its results.

**Important:** Be sure that the output includes all the fields you need, especially fields you plan to use in conditions or as action input. Also, fields marked as required must be in the output. The action fails with a schema validation issue if any of the required fields are not in the output. Only mark a field as required if you can guarantee the field is always in the output. If the field is dynamically populated, consider leaving all fields as optional and using a condition to check that the field has a value before using the field, as suggested in a later step.

For more info about these schemas, see

[Customize an event query action's input and output \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u3794cfd\]](#).

- Optional. To adjust either of the schemas to better define the requirements you want for the query's input or output, click **Input schema** or **Output schema** and make your changes. When done, click **Save changes**.

**Note:** For the input schema, you can only edit the metadata. You can't edit the schema.

7. Optional. Actions based on event queries place their search results in their output for use with, for example, the **Send email** action and Jira ticket creation. The results are always available in JSON. If you would also like the results in CSV, in the **Action** panel, select **Export to CSV**. When you choose this option, you can select the exact fields to include in the result using the **CSV header fields to export** field.

**Tip:** Large results might fail a workflow. Selecting the **Only return results as files** option might help avoid those failures. However, the **Full search results** and **Event query results** fields accessed by iterating over the results using a workflow loop are not accessible when using this option.

For more info, see [Add search results as an attachment \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#i4551f7d\]](#).

8. Click **Next**.

9. If you don't already have a condition to confirm the query found matches and that any required fields are populated, create one. See [Verify the event query action output is nonempty before acting on it \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r587e1f4\]](#)

10. If the **Issues** panel opens, resolve any warnings or errors you see.

11. Click **Save and exit**.

12. Add a comment to describe the change to the workflow for its version history.

13. Click **Update workflow**.

By default, the action goes in the **Other (Custom, Foundry, etc.)** group. Later, if you want to add the action to another workflow, search for the action by name or browse for it in this group.

## Remove an action based on an event query from a workflow


You can remove an action from a workflow—leaving the action available to other workflows.

If you prefer to delete an action so that it's no longer available, see

[Delete an action based on an event query \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r3d8c3d2\]](#).

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#) and click the workflow with the action to remove.

2. In the workflow, click **Edit**.

3. Hover over the action to delete and click **Delete** .

A warning appears highlighting any nodes affected by the deletion.

4. Click **Delete**.

5. Click **Save and exit**.

6. Add a comment to describe the change to the workflow for its version history.

7. Click **Update workflow**.





**Tip:** Remove any conditions used to confirm that the query found matches. These conditions check Event count.

## Delete an action based on an event query



You can delete an action so that it is no longer available to any workflow. If an action is used in multiple workflows, you must first remove the action from all workflows that use the action.

If you only want to remove an action from a workflow without deleting the action, see

[Remove an action based on an event query from a workflow \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#x3ccddb8\]](#).

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).
2. Use the **Actions** filter to show only the workflows that use the action you want to delete.
3. For each workflow with the action, click Open  and then select **Edit workflow** and remove the action from that workflow:
  - a. Hover over the action to delete and click **Delete**  .  
A warning appears.
  - b. Click **Delete**.
  - c. Click **Save and exit**.
  - d. Add a comment to describe the change to the workflow for its version history.
  - e. Click **Update workflow**.

With the action no longer in any workflow, you can delete the action.

4. To delete the action:
  - a. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#).
  - b. Click **Create workflow**.  
You only need this workflow temporarily. You will discard the workflow after you delete the action.
  - c. Start to define a workflow so that you can add an action.
  - d. In the **Add next** panel the workflow canvas, click **Action** .
  - e. Find the action to delete and select it.
  - f. Click **Manage event query**.  
The event query manager opens.
  - g. Click **Delete** .
5. You can now discard this temporary workflow.  
Click **All workflows** and then **Discard workflow**.

**Tip:** Remove any conditions used to confirm that the query found matches. These conditions check Event count.

## Add search results as an attachment

With some plugins, such as Jira and ServiceNow, you can add attachments to the tickets.

Event queries place their search results in their output.

If you're using either of these plugins, you can attach the results to your tickets.

For actions created before October 1, 2024, the results are in the File info field of their output.

For actions created after October 1, 2024, the results are in the new JSON file field of their output. For these actions, you can also get the results in the CSV format, which is then available in the new CSV file field in the action's output. To get the CSV format, after you add an action to a workflow, select the **Export to CSV** option in the **Action** panel. When you select this option, the **CSV header fields to export** field opens. You can use this field to select the exact CSV header fields you want in the CSV file. If you don't select any fields in **CSV header fields to export**, the CSV file contains all of the header fields.

**Tip:** Large results might fail a workflow. Selecting the **Only return results as files** option might help avoid those failures. However, the **Full search results** and **Event query results** fields accessed by iterating over the results using a workflow loop are not accessible when using this option.

If an action based on an event query was created after October 1, 2024, the **Export to CSV** option is also available when editing that action.

## Create and manage NG-SIEM lookup files

Use Fusion SOAR to create new lookup files and overwrite existing lookup files from a workflow. For example, you can retrieve data from an API call formatted as CSV or JSON and create a lookup file from it.

For info about the roles needed to create and overwrite lookup files, see

[Requirements \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#18d760c7\]](#).

These are the related actions:

- **Get lookup file metadata**  
Use this action to collect info to check whether a file exists before you use the create or overwrite actions.  
Conditions that use this metadata can use builtin parameters or CEL expressions that you define. For more info about CEL, see [CEL expressions \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#1e35e6c4\]](#).
- **Create lookup file**
- **Overwrite lookup file**

**Note:** Fusion SOAR does not create or overwrite more than 5 files within a 30-second period.

Fusion SOAR only shows lookup files created or overwritten in Fusion SOAR. However, Next-Gen SIEM shows its lookup files and any lookup files you create in Fusion SOAR.

**Tip:** To see how to work with lookup files in your workflows, see the [Introduction to Lookup file actions](#) playbook. The playbook shows how to check whether a file exists and then either creates a file or overwrites the existing file, in either CSV or JSON format.

When you create or overwrite a lookup file, you configure these items:

- Repository or view**  
Select the repository or view that corresponds to the repository or view that the related query action is using.
- Name**  
The name must end with `.csv` or `.json`.  
Also, the name must be unique within the repository or view you selected.  
To ensure uniqueness, you can insert a variable, such as `Workflow_execution_ID`, in the name.  
The name must be at least 5 characters but not more than 100 characters.
- Content type**  
Define the lookup file by selecting a file created by a previous action or by entering plain text inline.  
The accepted formats are CSV and JSON.  
The maximum size file that you can upload is 10 MB.  
The maximum amount of text that you can enter inline is 900 KB.  
The JSON formats accepted are the same as the JSON formats accepted by Falcon LogScale. Here are examples of those formats:

Object-based example

```
{
  "1": { "name": "chr" },
  "2": { "name": "krab" },
  "4": { "name": "pmm" },
  "7": { "name": "mgr" }
}
```

Array-based example

```
[
  { "userid": "1", "name": "chr" },
  { "userid": "2", "name": "krab" },
  { "userid": "4", "name": "pmm" },
  { "userid": "7", "name": "mgr" }
]
```

**Example**

Assume we have log entries that contain status codes for a web server. These codes are in the field named `status`. To provide a name field to match names to the status codes, we could upload a lookup file named `status_codes.csv` with content that corresponds to this table:

code	name
200	OK
400	Bad Request
401	Unauthorized
500	Internal Server Error

To use that lookup file in a query, the query would then include a like line this one:

```
groupby([status])
| match(file="status_codes.csv", column=code, field=status, include=name)
```

When the query is run, the results might look something like this:

status	_count	name
200	777	OK
400	1	Bad Request
500	10	Internal Server Error

**Verify the output fields of an event query action**

Make sure your event query action provides the output you expect. Verifying the output is particularly important when you plan to use it in a condition or loop or as input to another action.

**Note:** The output is in JSON and optionally CSV. These techniques are for the JSON output of the action. For more info about the CSV option, see [Add search results as an attachment \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#j4551f7d\]](#).

You can check the output at various times:

- When you're creating an action based on an event query, after you click **Continue**, click **Output schema**.
- When you're editing an action based on an event query, click **Output schema**.
- Whenever you're curious about an action's output:

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#) and click a workflow that uses the action.
2. In the workflow, click **Edit**.
3. Click the action to show the **Action** panel.
4. Click **View schema** and then **Output schema**.

For more info about JSON schema, see

[Manage action input, action output, and on-demand triggers \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d\]](#).

## Verify the event query action output is nonempty before acting on it

Every event query action includes an `Event count` field. This field indicates how many events the query found.

Before acting on the output of an event query action, verify the query found at least one event by setting a condition on the `Event count` field.

In the condition, use these settings:

**Parameter:** Event count

**Operator:** is greater than or equal to

**Value:** 1

The following image shows these settings.

## Loop through the results of an event query

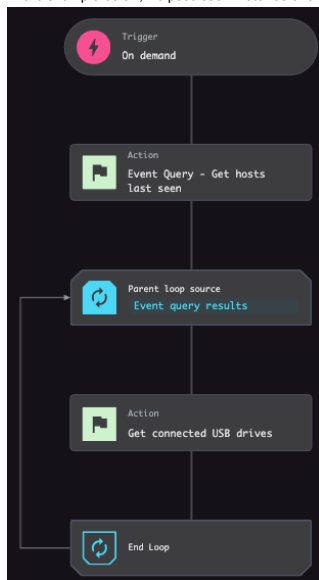
To build conditions or actions based on the results of an event query, you must add a loop to iterate through the search results. For info about looping, see

[Looping in workflows \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#k9bd6faf\]](#).

For info about how to find fields to loop over in the search results, see

[Verify the output fields of an event query action \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z1898279\]](#).

In the example below, we pass each instance of the `AID` returned in the event query results to the `Get connected USB drives` action's `Device ID` field.



## Customize an event query action's input and output

Fusion SOAR generates schemas for the action's input and output based on the query and its results. You can modify these schemas to better suit your needs. Use the input schema to define requirements for your action's input and to help users of the action be successful when using the action. For example, a schema can validate input or help the user understand the variables to pass when using the action. Similarly, the output schema helps the user of the action understand what output to expect from the action.

[Example: Add a description, a default value, and validation to fields in the input schema \[/documentation/page/dc4f8c45/workflows-falcon-](#)

[fusion-1692362310390.669#p2e195be\]](#)

- [Example: Trim an action's output \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#tf086e05\]](#)

## Example: Add a description, a default value, and validation to fields in the input schema

Consider this example query:

```
#event_simpleName = SensorHeartbeat
```



```
//Convert timestamp to epoch time
|formatTime(format="%Q", field=@timestamp, as="epoc_timestamp")
//Compare today and last time AgentOnline event was received
| last_seen:= ((now() - epoc_timestamp)/86400000)
| last_seen:=math:ceil(last_seen)
| last_seen >?last_seen
| table([aid, ComputerName, last_seen])
```

This query returns a table of all the hosts that were last seen more than the specified number of days ago—based on the ?last\_seen syntax. Because Fusion SOAR derives input and output schemas from the search query and search results, you can modify those schemas to be more specific. In this example, the input schema is the last\_seen field, which is an integer type.

Here's the input schema that Fusion SOAR derived:

To customize the properties of the field, use the JSON Schema editor. In this case, we are going to make these changes:

- Default value: Change from 2 to 7
- Description: Add a prompt for those who use this action, Input the number of days
- Minimum number of days: Prevent values less than 1
- Maximum number of days: Prevent values greater than 90

For more info about JSON schema, see

[Manage action input, action output, and on-demand triggers \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d\]](#).

Here's the modified input schema:

For the output, the output schema includes the column headers from the table produced by the query: aid, last\_seen, and ComputerName. Certain Fusion SOAR actions require some input fields to be in certain formats. For this schema, we change the format for the aid field to **Sensor ID**. Then the action's output can be used as input to actions that require a format of **Sensor ID**, such as Contain Device, Get Device Details, and Add Device to Watchlist.

Here's the output schema that Fusion SOAR derived:



Get Last Seen Hosts

Date & time modified

Modified by

March 26, 2024 12:46 PM

Query

Input schema

Output schema

Generate schema

JSON Schema editor

```

root  object
├── aid  string
├── last_seen  Integer
└── ComputerName  string

```

Cancel

Add to workflow

Here's the modified output schema:

Get Last Seen Hosts

Date & time modified

Modified by

March 26, 2024 12:46 PM

Query

Input schema

Output schema

Generate schema

JSON Schema editor

```

root  object
├── aid  string (Sensor ID)
├── last_seen  Integer
└── ComputerName  string

```

aid / string

Name

aid

Type

string

Constant value

Default

Enum

Description

Format

Sensor ID

Maximum length

Minimum length

Pattern

Title

Aid

Required

☒

To edit an action's schemas, edit the action as explained in

[Edit an action based on an event query](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ye9dafa6].

### Example: Trim an action's output

To limit the output of a query, you can use filters in the query itself. If you don't use filters though, Fusion SOAR generates the output schema using all the fields returned in the query. You can then trim fields that aren't useful to you from the output schema. To trim the schema, use either the JSON schema form builder or click **JSON Schema editor**.

**Tip:** If you opted to also export the results in CSV and want to trim the CSV output, list the exact fields to include as explained in [Add search results as an attachment](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#j4551f7d].

Consider this example query:

#event\_simpleName = ActiveDirectoryAccountPasswordUpdate

The query returns ActiveDirectoryAccountPasswordUpdate events. Fusion SOAR produces the following output schema:

Query

Input schema

Output schema

Generate schema

JSON Schema editor

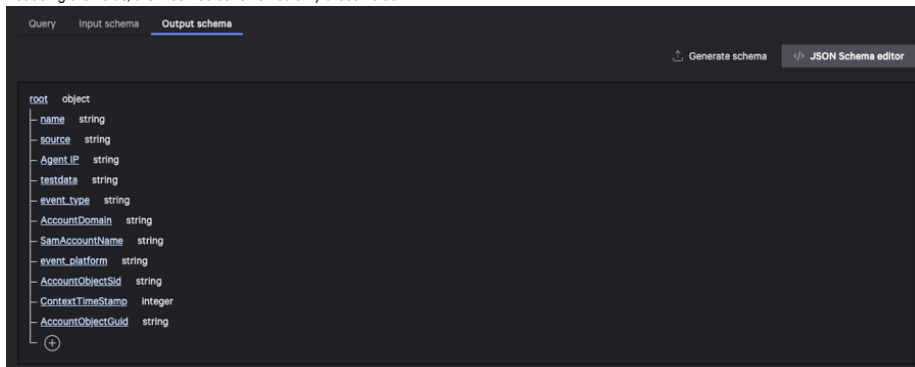
```

root  object
├── id  string
├── aid  string
├── alp  string
├── cid  string
├── name  string
├── source  string
├── Agent IP  string
├── testdata  string
├── event_type  string
├── AccountDomain  string
├── SamAccountName  string
├── event_platform  string
├── AccountObjectSid  string
├── ContextTimeStamp  Integer
└── AccountObjectGuid  string

```

© 2024 Fusion SOAR. All rights reserved. | [Privacy Policy](#) | [Terms of Service](#)

Reducing the fields, the modified schema has only these fields:



To edit an action's schemas, edit the action as explained in

[Edit an action based on an event query](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ye9dafa6].

For more info about JSON schema, see

[Manage action input, action output, and on-demand triggers](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#y566df3d].

## Create an action in Fusion SOAR to perform an HTTP request

**Tip:** To invoke APIs for on-premises tools, see

[Create actions in Foundry that use APIs for on-premises tools](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d434618d].

Make on-demand API calls inside workflows using the Fusion SOAR **HTTP Request** action. When you use this action, you can take advantage of inline testing, dynamic variable injection, and conditional branching. This action is available directly within Fusion SOAR without requiring a Foundry app.

For info about the requirements to use this action, see [Requirements](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#l8d760c7].

In a workflow, at the point where you want to make an API call, add the **HTTP Request** action. Alternatively, in the **Add action** panel, instead of searching for the **HTTP Request** action, click the **Create HTTP request** shortcut shown after the **Search** field.

To configure the action, enter values for these fields:

- Authentication:

- When choosing an authentication option, you can create a new authentication, use an existing one, or use no authentication.
- When you create an authentication, that authentication is then available for subsequent actions.
- After you configure an authentication, you cannot change it.

**OAuth 2.0 limitations.** This action supports basic OAuth 2.0 client credentials authentication. However, these advanced OAuth flows and features are not supported:

- OAuth scopes (custom scope handling, such as Microsoft Graph)
- private\_key\_jwt authentication
- 3-legged OAuth (authorization code) flow
- AWS role-based authentication (STS/OIDC)
- Google service account authentication
- Custom grant types, including Zoom client credentials

- Request:

- Set the HTTP method, such as GET, POST, PUT, or DELETE, and enter the API endpoint URL.

**Tip:** You can insert variables in the URL so that when the workflow is run, the workflow prompts you to enter values for the variables. However, when testing, you are not prompted to enter values; the literal variable string is used.

- Body:

- Depending on the resource being requested, you might need to define the body to describe the resource being created or modified. To understand what is needed, see the documentation for the API you are using.
- For the request body, you can select **JSON**, **Plain text**, **CSV**, or **No request body**.

- Headers:

- Headers provide additional information about the resource being fetched or about the client making the request.
- You can enter variables by clicking {x}.

- Query:

- Depending on the resource being requested, you might need to define query parameters. To understand what is needed, see the documentation for the API you are using.
- These parameters are appended to the end of the request URL and appear after a question mark (?) with a key and value. Multiple query parameters are separated by ampersands (&).
- You can enter variables by clicking {x}.

- Response:

- Depending on the method, URL, and query parameters, the response from the API includes a header and a body. To understand the response, see the

documentation for the API you are using. The header contains information about the response, such as the HTTP response code, content-type, and character encoding. The body contains the content of the resource being requested.

- The response size can be up to 10 MB.
- To manage the data in the response, you can generate a schema from a sample response payload or you can manually define a schema based on the expected structure.

After you set up the request, verify that the request works and returns the expected response by clicking **Test**.

When you are satisfied that the request works as expected, click **Next** to add the action to the workflow.

**Note:** An action that makes an HTTP request times out if it does not get a response within 30 seconds.

To help you troubleshoot executions of workflows with this action, the execution log shows the response payload and the status code.

## Create actions in Foundry that use APIs for on-premises tools

**Tip:** For APIs that are not on-premises, use the **HTTP Request** action directly within Fusion SOAR. For more info, see [Create an action in Fusion SOAR to perform an HTTP request \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#u981b536\]](#).

If you use APIs for on-premises tools, you can create Fusion SOAR workflow actions that use these APIs.

For info about the requirements to create these actions, see [Requirements \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#l8d760c7\]](#).

### Important:

- Avoid running these actions in rapid succession: The requests come from the CrowdStrike cloud to your environment and will experience some latency.
- You can make up to 10 requests per minute per sensor in the host group.
- If the action does not get a response within 90 seconds, it times out.
- These actions use RTR sessions.
  - Configure a response policy with Real Time Response (RTR) enabled and assign it to the host group. Make sure the response policy has these settings enabled:

- Custom Scripts
- get
- put
- run
- put-and-run (Windows and macOS)

For more info, see [Configuring response policies \[/documentation/page/b8c1738c/real-time-response#nc81f726\]](#) and [Assigning a response policy to host groups \[/documentation/page/b8c1738c/real-time-response#o79ff398\]](#).

- If you have any workflows with an **Audit event > RTR Session** trigger, these actions will trigger those workflows. However, you can set a condition in these workflows to ignore the RTR sessions created by these actions. To set such a condition, set its fields as indicated here:
  - Set **Parameter** to **Connected From**
  - Set **Operator** to **is not equal to**
  - Set **Value** to **On-premises API call**

Then click **Add condition line** and set the fields as indicated here:

- Set **Parameter** to **User**
- Set **Operator** to **is not equal to**
- Set **Value** to **async-rtr@crowdstrike.com**

- For info about enabling multifactor authentication for workflows with these actions, see [Real Time Response \(RTR\) \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ua24dff0\]](#).

For on-premises Jira installations, we provide the **Atlassian Jira Data Center SOAR Actions** app in the CrowdStrike Store to help you get set up faster.

These sections explain how to set up an action for any on-premises API and for an on-premises Jira installation:

- [Create an action in Foundry for an on-premises API \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#f12d558e\]](#)
- [Create an action in Foundry for an on-premises Jira API \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#e60f83c5\]](#)

## Create an action in Foundry for an on-premises API

**Tip:** This topic addresses the general case of how to create an action that invokes the API for an on-premises tool. For info about setting up actions for Jira in particular, see [Create an action in Foundry for an on-premises Jira API \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#e60f83c5\]](#).

At a high-level, the process to create and use such an action that invokes an API has these steps:

1. Use Falcon Foundry to create an app with an API integration for an on-premises tool.
2. Make the app available to Falcon Fusion SOAR workflows as an action.
3. After the app is deployed, workflows can use the action to invoke the on-premises API.

**Important:** The response size can be up to 1 MB.

**Important:** The response size can be up to 1 MB.

**Note:** When adding an action for the API for an on-premises tool, the dropdown lists can take up to 30 seconds to populate with dynamic values because of the delay in getting the values from the on-premises tool.

For details about how to set up an API integration, see [API Integrations \[/documentation/page/te8afcf6/api-integration\]](#).

Follow those steps with these variations.

- When you set up the API integration profile, set **Host environment** to **On-Premises**.  
For more info, see [Step 2: Add an API integration to that app \[/documentation/page/te8afcf6/api-integration#tfe2be35\]](#).

- When you test the operation, set up the configuration with these values:

- **Host group**

Required. Given that the API might be handling sensitive data, define and use host groups to help secure the data.

**Important:** Dynamic host groups are not supported.

Use only static host groups--with explicit hostnames or IDs--that you know are secured, have limited access, and have both security and compliance controls in place to ensure proper monitoring.

Configure a response policy with Real Time Response (RTR) enabled and assign it to the host group. For more info, see

[Create actions in Foundry that use APIs for on-premises tools \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d434618d\]](#)

.

Also put firewall rules in place.

As a best practice, limit these host groups to at most 20 hosts.

For more info, see [Host and Host Group Management \[/documentation/page/f8a0f751/host-and-host-group-management\]](#) and

[Falcon Firewall Management \[/documentation/page/a6e15696/falcon-firewall-management\]](#).

- **Trust any certificate (insecure)**

Optional.

**Important:** Trusting any certificate, which disables certificate verification, has security risks. Only use this option when absolutely necessary.

If the server uses an untrusted certificate or performs SSL termination, an error can occur. If this configuration is expected for your environment, you can select this option to check whether the errors persist.

- **Proxy URL**

Optional. Enter the URL for a proxy server used in your environment. Here are the valid formats:

- http://<host>:<port>
- https://<host>:<port>

For more info, see [Step 6: Test the operation to verify that it works as expected \[/documentation/page/te8afcf6/api-integration#c22794f6\]](#).

- When you deploy the app, you are prompted for the host group name during configuration.

## Create an action in Foundry for an on-premises Jira API

We provide the **Atlassian Jira Data Center SOAR Actions** app in the CrowdStrike Store to help you get set up more quickly.

**Important:** The response size can be up to 1 MB.

**Note:** This app does not support file attachment or any other file operations.

1. Complete the app configuration for the **Atlassian Jira Data Center SOAR Actions** app in the CrowdStrike Store.

- Go to [CrowdStrike Store > CrowdStrike Store > All apps \[/store-v2\]](#).
- Find the **Atlassian Jira Data Center SOAR Actions** app.
- Click the app, click **Configure**, and click **Add configuration**.
- Set up the configuration with these values:

- **Name**

Required. Enter a name for the configuration. You'll use this name when setting up workflow actions.

- **Base URL**

Required. Enter the URL for your on-premises Jira installation.

- **API Key**

Required. Enter the Personal Access Token (PAT) for your JIRA installation.

- **Host group**

Required. Given that the API might be handling sensitive data, define and use host groups to help secure the data.

**Important:** Dynamic host groups are not supported.

Use only static host groups--with explicit hostnames or IDs--that you know are secured, have limited access, and have both security and compliance controls in place to ensure proper monitoring.

Configure a response policy with Real Time Response (RTR) enabled and assign it to the host group. For more info, see

[Create actions in Foundry that use APIs for on-premises tools \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#d434618d\]](#)

.

Also put firewall rules in place.

As a best practice, limit these host groups to at most 20 hosts.

For more info, see [Host and Host Group Management \[/documentation/page/f8a0f751/host-and-host-group-management\]](#) and

[Falcon Firewall Management \[/documentation/page/a6e15696/falcon-firewall-management\]](#).

- **Trust any certificate (insecure)**

Optional.

**Important:** Trusting any certificate, which disables certificate verification, has security risks. Only use this option when absolutely necessary.

If the server uses an untrusted certificate or performs SSL termination, an error can occur. If this configuration is expected for your environment, you can select this option to check whether the errors persist.

- **Proxy URL**

Optional. Enter the URL for a proxy server used in your environment. Here are the valid formats:

- `http://<host>:<port>`
- `https://<host>:<port>`

e. Click **Save configuration**.

2. In Fusion SOAR, use either of these actions—and set **Account** to the configuration you just created:

- Create Jira Data Center issue
- Create Jira Data Center comment

**Note:** When adding an action for the API for an on-premises tool, the dropdown lists can take up to 30 seconds to populate with dynamic values because of the delay in getting the values from the on-premises tool.

## Save data from a workflow

- [Save data using the "Write to log repo" action](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#e39145f2]
- [Save data using workflow output](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c0b19bdf]

## Save data using the "Write to log repo" action

Fusion SOAR can write data, such as output from an RTR script or an HTTP-based action, by using the **Write to log repo** action. The **Write to log repo** action saves output from either an **On demand** trigger or an action. You can then query the data through [Investigate > Search > Advanced event search](#) [/investigate/search] or use the data in other workflow actions.

For example, you could use this feature to write the output of an RTR script to a repository to access and search later.

To query the data interactively, go to [Investigate > Search > Advanced event search](#) [/investigate/search] and set the repo to Fusion by using this text as the first line in your query:

```
#repo = fusion
```

Then define and run your query. For more info, see [Advanced Event Search](#) [/documentation/page/ic5d7b7d/event-search-advanced].

Alternatively, later actions, either in the same workflow or other workflows, can use an event query with the **Data view** set to **All** to process that data. If you're querying the data within the same workflow and the data is not yet available, add a **Sleep** action. For more info, see [Event queries, or saved searches, as workflow actions](#) [/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c022d93e].

**Note:** There is a limit of 950 KB of data per **Write to log repo** action execution.

In Flight Control environments, workflows in a parent CID can write data into the parent CID's own data view but not to any of the data views in the child CIDs. However, workflows in a parent CID can query data in the data views in the child CIDs.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows](#) [/workflow/fusion].

2. Create or edit a workflow.

3. Set up the data to save. You have these options:

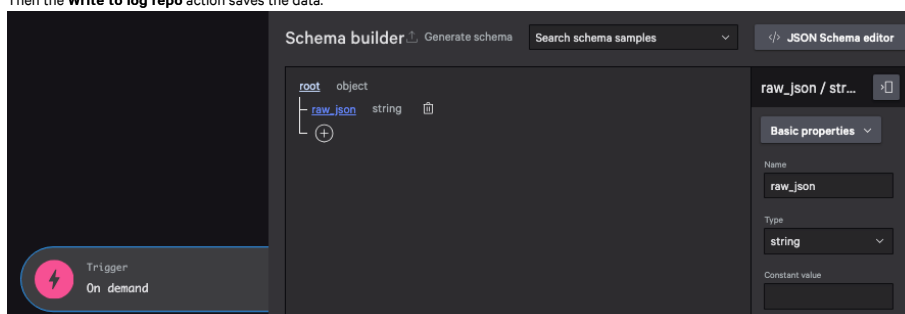
- The workflow uses an **On demand** trigger followed immediately by a **Write to log repo** action.  
You define the trigger's JSON schema to require input:

- In the schema builder, add a property, such as `raw_json`, of type string.
- Click the property to show its basic properties.
- Set **Format** to **Raw JSON string**.

When someone runs the workflow, they are prompted for the data. That person uses one of these formats:

- For one record:  
`{"foo": "bar"}`
- For multiple records:  
`[{"a": "b"}, {"c": "d"}]`

Then the **Write to log repo** action saves the data.



- The workflow uses any kind of trigger and includes a **Write to log repo** action.  
You define the action's input JSON schema using one or more of these fields. The combination of the data from all 3 fields is saved.
  - **Data to include**  
Click in this field and select one or more items, one at a time.  
To simplify querying in the repo, select **Remove action prefix**.
  - **Raw JSON data**  
Select an option. These options are only available if the trigger or previous actions have schemas that define a property of type string with **Format** set to **Raw JSON string**.
  - **Custom JSON data.**  
Enter your JSON directly in this field.

4. Define or edit the rest of the workflow as needed, resolve any warnings or errors, and save it.

## Save data using workflow output

You can configure workflows to save output for you to view in the Falcon console or to act on programmatically.

If you define workflow output for an on-demand workflow, any workflow that runs the on-demand workflow can then use the output of that on-demand workflow.

In addition, to sum up a workflow's results, you can specify an execution summary that includes variables.

**Tip:** This summary can be helpful in on-demand workflows so that someone considering using the workflow can quickly understand what the workflow provides.

When you create or edit a workflow, click **Workflow details**, followed by **Select output data**, and then select the output to include in the workflow executions.

This output is then available in the Execution log in the workflow's execution details.

To access that output, you have these options:

- Go to **Fusion SOAR > Fusion SOAR > Workflows** [/workflow/fusion], click the **Execution log** tab, click the workflow, go through the execution details—where you can view the output or download it manually.
- Download the execution details and its workflow output using APIs discussed in [Fusion SOAR Workflow APIs](#) [/documentation/page/z028de1a/fusion-workflow-apis].

## Retry a failed execution

Workflows typically fail because a host is offline or some other momentary connection issue. Under some conditions, Falcon retries failed actions up to a certain limit.

**Note:** A loop can run, or iterate, up to 100,000 times. If a loop attempts to iterate more than 100,000 times, both the loop and its workflow go into an error state. Such a workflow is not available to retry.

1. Go to **Fusion SOAR > Fusion SOAR > Workflows** [/workflow/fusion] and click the **Execution log** tab.
2. For a failed execution in the table, click **Open menu** ⋮ and select **Retry execution**.

**Tip:** You can also retry a failed execution by clicking the **Open menu** ⋮ in the **Execution details** panel.

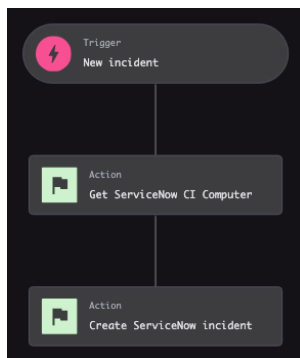
## Create ServiceNow incidents with host details

Automate ServiceNow incident generation with the ServiceNow ITSM SOAR Actions plugin. You can customize which data to include on the incident. For example, to add host identifiers, configure the **Get ServiceNow CI Computer** action to include the ServiceNow configuration item (CI) ID and CI name fields.

Insert sequential actions to retrieve CI details before setting up the **Create ServiceNow incident** action:

**Note:** In Flight Control environments, the CI must be in the parent CID.

1. When configuring actions for a workflow, search for and then select the **Get ServiceNow CI Computer** action.
2. From the **Account** dropdown, select the ServiceNow account to pull the CI details from. Set the **Hostname** dropdown. Then click **Next**. The action appears on the workflow canvas.
3. Hover over the action where you want to add the next action and then, to expand the menu, click ⌵ and then **Action** 📄.
4. In the **Add action** panel, search for and then select **Create ServiceNow incident**.
5. From the **Account** dropdown, select the same ServiceNow account as before.
6. Fill out the fields to include on the ServiceNow incident.
7. In the **Data to include** field, select the fields to add to the incident, including **ServiceNow CI ID** and **ServiceNow CI name**.
8. When you're finished, click **Next**.



## Create Jira notification actions

Configure workflows to create prefilled Jira tickets. You can customize Jira fields while setting up a notification action.

**Note:** For details about configuring Jira tickets for vulnerability management, see [Set up Jira workflows](#) [/documentation/page/e552ea54/vulnerability-management-ticketing-workflows#m6b619c6].

1. When configuring actions for a workflow, search for **Create Jira issue** and then expand the **Atlassian** list and select the **Create Jira issue** action.
2. From the **Account** dropdown list, select a Jira configuration.
3. From the **Project** dropdown list, select the project you want to assign the Jira ticket to.
4. Optional. Enter labels to attach to the Jira ticket.
5. Set a priority for the Jira ticket.

**Important:** The Priority dropdown list shows values for all Jira issue types. Select a priority that's available for the selected issue type or the workflow execution will fail. If the issue type isn't configured for priorities, select **None**.

6. From the **Issue type** field, select a Jira issue type.

7. Enter a description and summary for the Jira ticket.
8. In the **Data to include** field, select which info to include on the ticket.

**Note:** Available options depend on the trigger type.

9. When you're done, click **Next** and complete the workflow setup.

## Add sequential actions to Jira notification actions

Build complex workflows by adding sequential actions to an existing **Create Jira issue** action. For example, you can attach a file associated with an endpoint detection to a Jira ticket to give additional context to an issue.

### Add comments to Jira tickets

Add relevant comments to a Jira ticket generated within a workflow.

1. On the workflow canvas, find the **Create Jira issue** action and add an action after it.
2. From the action panel, find and select the **Add Jira comment** action.
3. From the **Account** dropdown list, select the same Jira configuration used in the **Create Jira issue** action.
4. From the **Issue** dropdown list, select **Jira issue ID** action.
5. In the **Body** field, enter text to include in the comment field on the Jira ticket.
6. In the **Data to include** field, select the Falcon data you want to include on the ticket.
7. When you're done, click **Next** and complete the workflow setup.

### Attach files from RTR actions to Jira tickets

For workflows triggered by endpoint or custom IOA detections, retrieve files associated with the detection and attach them to Jira tickets.

**Note:** You must have the correct [Real Time Responder role \[documentation/page/b8c1738c/real-time-response#nfa82ecc\]](#) to access RTR actions.

1. On the workflow canvas, find the **Create Jira issue** action and add an action after it.
2. From the action panel, find and select the Real Time Response **Get file** action.
3. Click **Next**.
4. After the **Get file** action, add another action.
5. From the action panel, find and select **Add Jira attachment**.
6. From the **Account** dropdown list, select the same Jira configuration used in the **Create Jira issue** action.
7. When you're done, click **Next** and complete the workflow setup.

### Create subtask Jira tickets in a workflow branch

Generate Jira tickets and related Jira subtask tickets within a single workflow branch.

1. On the workflow canvas, find the **Create Jira issue** action and add an action after it.
2. From the action panel, find and select the **Create Jira issue** action.
3. When setting up the Jira template, in the **Issue type** field, select **Subtask**.

**Important:** Selecting an issue type that is not **Subtask** causes the workflow to fail.

4. When you're done, click **Next** and complete the workflow setup.

**Note:** If a sequential **Create Jira issue** action creates a Jira subtask, any subsequent sequential actions that add Jira comments or files attach those items to the parent Jira ticket, not the subtask.

## Create third-party integration actions with prebuilt Foundry templates

Configure workflows to integrate with third-party security platforms through prebuilt Falcon Foundry app templates. Available integrations include threat intelligence, identity and access management, DevOps security, and email security. You can use the prebuilt integrations as-is or customize them to your needs.

To create an integration action, you deploy a Foundry app template and then release and install the app to make the custom action available in Fusion SOAR.

## Requirements

**Requires one or more of these subscriptions:**

- Next-Gen SIEM, or Falcon Complete powered by Next-Gen SIEM
- Falcon Foundry

**Note:** Customers with a Falcon Insight XDR or Falcon Prevent subscription can install one Foundry app per CID for no additional cost. Installing additional Foundry apps requires a Falcon Foundry subscription or a Falcon Next-Gen SIEM subscription. Contact your CrowdStrike sales representative for more info.

**Default roles:**

- App Developer plus Workflow Author



- Falcon Administrator

## Create a third-party integration action

- On the **SOAR Dashboard** page ([Next-Gen SIEM > Fusion SOAR > Dashboard \[/workflow/fusion/dashboard\]](#)), on the **Create an integration for custom Fusion actions in Foundry** banner, click **Visit page**.

**Tip:** You can also go to [Foundry > Foundry > Templates \[/foundry/app-templates\]](#).

The Foundry **Templates** page opens.

- Click **Deploy** for your selected template.

The message **Deployment in progress** appears.

When the app has been deployed, the **App overview** page opens.

**Note:** To edit, test, or delete operations, click [Edit app](#).

- Release the app.

- Click **Release**.
- In the **Commit release** dialog, for **Change type**, select **Major**.
- Add a description of the app in the **Release notes** field for this initial release, and click **Release**.  
The message **Releasing deployment** appears, followed by the message **Deployment released successfully**.

- Install the app.

- Click **View in app catalog**.  
Your app's catalog details page opens, showing the list of releases.
- Click **Install now**.
- Acknowledge the app permissions information by clicking **Accept and continue**.
- If required, enter configuration information.
- Click **Install app**.  
When your app has been installed, a notification appears. API operations in the app are now available as third-party integration actions.

- Add a third-party integration action to a workflow.



- Click **Fusion SOAR** in the installation message.
- On the **SOAR Dashboard** page ([Next-Gen SIEM > Fusion SOAR > Dashboard \[/workflow/fusion/dashboard\]](#)), create a new workflow or edit an existing workflow. For info, see [Manage workflows \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#bb70fd80\]](#).
- Add a third-party integration action to the workflow. To locate the action, search using any of these names:
  - The app name
  - The name of the API integration configured in the Foundry app
  - The name of the API operation configured in the Foundry app
- Create a condition to confirm the API operations output.  
Be sure the output includes all the fields you need, especially fields you plan to use in conditions or as input to other actions.  
Also, fields marked as required must be in the output. The action fails with a schema validation issue if any of the required fields are not in the output.  
In Foundry, only mark a field as required if you can guarantee the field is always in the output. If the field is dynamically populated, consider leaving all fields as optional and then using a condition in Fusion SOAR to check that the field has a value before using the field.

For more info about adding an action to a workflow, see [Workflow actions \[/documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#p7d19355\]](#).

For more info about Foundry apps, see [Falcon Foundry \[/documentation/category/c3d64B8e/falcon-foundry\]](#).

## Edit a third-party integration action

To edit a third-party integration action, you must edit its source Foundry app. You can add, edit, or remove API operations. You can also add other app capabilities.

- Go to [Foundry > Foundry > App manager \[/foundry/app-manager\]](#).
- Search for the app by its name or description.
- When you locate the app, from its **Open menu** , select **Edit app**.
- Click the arrow on the **Integrations** panel.
- In the **Operations** list, find the operation used by your workflow action, and from its **Open menu** , select **Edit operation**.
- Modify the operation as required. For info about the fields, see [Create an API integration using the UI \[/documentation/page/te8afcf6/api-integration#r0a1f109\]](#).

**Tip:** If the action is failing with a schema validation issue because required fields that are dynamically populated are not always in the output, consider leaving all fields as optional and then using a condition in Fusion SOAR to check that the field has a value before using the field.

- Optional. Test your changes to the operation.
  - Click **Test** in the side menu.
  - Click **Create** to create a temporary configuration, enter required values, and click **Create Configuration**.
  - Enter values for any required parameters, and click **Test operation**. A status of **Operation succeeded (200)** indicates a successful test.
- Optional. Add other capabilities to the app. For info, see [App Capabilities \[/documentation/category/u0daabab/falcon-foundry/app-capabilities\]](#).
- Click **Done** when finished.

10. Deploy the app to save your changes. For more info, see [Deploy an App \[documentation/page/bfd46a1c/deploy-an-app\]](#).
11. Release the app. For more info, see [Release an App \[documentation/page/z19188c4/release-an-app\]](#).
12. If the app was previously installed, and you marked the release as **Major**, update it in the app catalog. For more info, see [Update an app version \[documentation/page/ha2dd02e/app-administration#p7504831\]](#).

## Edit a workflow's CIDs in a Flight Control environment

If you have a workflow created in a Flight Control environment, the workflow can apply to multiple CIDs. To change the workflow's CIDs, edit the workflow and click **Edit CID selection**.

For info about how to make other changes to the workflow, see [Edit a workflow element \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#jb86e89d\]](#).

## Review the Workflows Audit Log

Visit the **Audit Log** to review all changes made to your workflows.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[workflow/fusion\]](#) and click the **Audit log** tab.
  - Filter the list by **Workflow name**, **Action** (what was changed), or **Modified by** (who made the change).
  - Click a column header to sort the table.

## Monitor executions

### Fusion SOAR dashboard

Fusion SOAR provides a dashboard so you can see activity for the last 90 days at a glance.

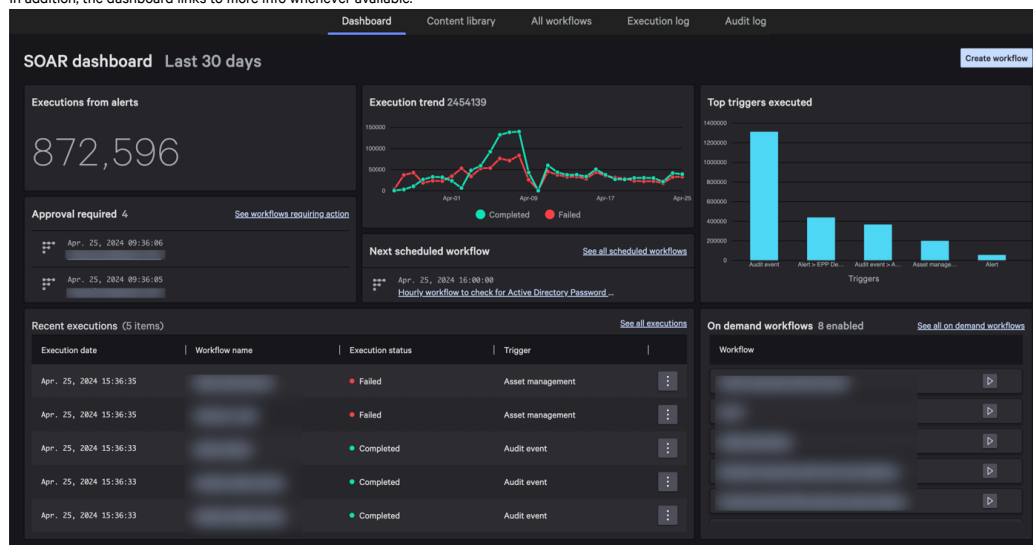
To see the dashboard, go to [Fusion SOAR > Fusion SOAR > Dashboard \[workflow/fusion/dashboard\]](#) or to [Next-Gen SIEM > Fusion SOAR > Dashboard \[workflow/fusion/dashboard\]](#).

**Note:** In addition to the Fusion SOAR dashboard, there's a Fusion SOAR execution dashboard built using Next-Gen SIEM's advanced queries and your workflow execution log data to give you additional information on your workflow activity. For more info, see [Fusion SOAR execution dashboard \[documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#r4c2b5df\]](#).

The dashboard provides this info for the last 90 days:

- **Executions from alerts:** A count of the alerts that Fusion SOAR has responded to
- **Execution trend:** Trend lines that show completed and failed workflow executions
- **Top triggers executed:** Bar chart of the triggers used most to start workflow executions
- **Approval required:** List of workflows that require approval
- **Next scheduled workflow:** The workflow scheduled to run next
- **Recent executions:** Table of the workflows most recently executed
- **On demand workflows:** List of workflows you can run on demand

In addition, the dashboard links to more info whenever available.



## View the execution log


Visit the **Execution log** to review every time your workflows have been triggered in the last 90 days.

**Note:** Workflows often execute multiple times for a single detection because they trigger off of specific attributes within the detection itself.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#) and click the **Execution log** tab.

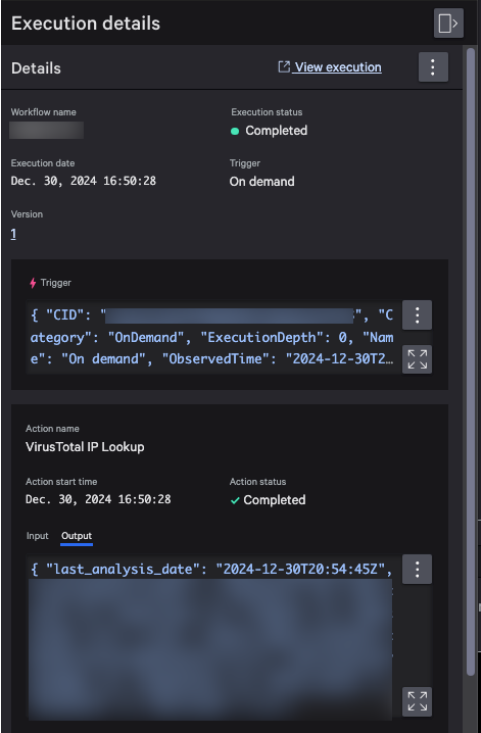
- Filter the list by **Workflow name**, **Execution date**, **Execution status**, **Trigger**, and **Action**.

**Note:** The Workflow name search supports wildcards. It doesn't return results for partial search terms. For example, `detectio` returns no results while `detectio*` returns results for both `detection` and `detections`.

- Click a column header to sort the table.
- Click **View execution details**  for any execution to open a quick view of information about the execution, including a link to the triggering event, the action taken, and the **Execution Status** (whether the workflow was fully completed).


## Review actions

The **Execution log** provides details about the actions taken in each workflow execution. When a workflow executes Real Time Response commands or initiates a VirusTotal hash lookup, the **Output** field shows what was returned, and the option to download is available for successful **Get file** actions.



## Review workflow executions

Examine the complete view of a workflow's executions to see which conditions were met each time it ran and where it might have failed.

1. Go to [Fusion SOAR > Fusion SOAR > Workflows \[/workflow/fusion\]](#) and click the **Execution log** tab.
2. For an execution in the table, click **Open menu**  for an execution and select **View execution**.
  - In the panel, switch between the workflow's execution records and see information about each execution, including a link to the triggering event and details about each action.
  - In the canvas, see a clear map of which conditions were met, which actions were taken, and where the workflow failed, if applicable.


When you have workflows with loops, you also see loop-specific info such as:

- Loop source type (the item being looped through)
- Status for the entire loop
- Status for each iteration
- Status for any nested loops
- For sequential loops, the reason the loop stopped

## Workflow execution status values

For workflows, the possible status values are shown in this table.

Status	Description
Completed	The workflow executed successfully and completed.
Failed	The workflow failed to execute properly.
In progress	The workflow is currently executing.
Action	The workflow is pending approval before it can start. The status is the result of an action that requests input.

<b>Action required</b>	<p>The status is the result of an action that requests input.</p> <p>You can grant approval in the execution log that shows all workflows by clicking <b>View execution details</b>  for the workflow and approving. Alternatively, you can grant approval in the workflow's execution log by clicking the action and approving.</p>
------------------------	--

### Action status values

For actions, the possible status values are shown in this table.

Status	Description
Completed	The action executed successfully and completed
Failed	The action failed to execute properly
In progress	The action is currently executing or being retried
Pending	The action status hasn't executed yet
Skipped	The action skipped execution

### Loop status values

For loops, possible status values are shown in this table.

Status	Description	Appearance in the canvas
Completed	The loop ran successfully	Green outline and a green check mark icon
Failed	The loop ran unsuccessfully If the action or condition within a nested loop has failed, then the parent loop also has a <b>Failed</b> status	Red outline and a red x icon
In progress	The loop is currently running	Gray outline and an in-progress icon
Pending	The loop didn't run or hasn't run yet Possible causes: <ul style="list-style-type: none"> <li>No loop source items to loop through</li> <li>The loop is on a branch skipped because a condition wasn't met</li> </ul>	Gray outline The details panel indicates no executions were performed
Skipped	The loop didn't run because a trigger wasn't activated or a condition wasn't met	Gray outline with an x icon

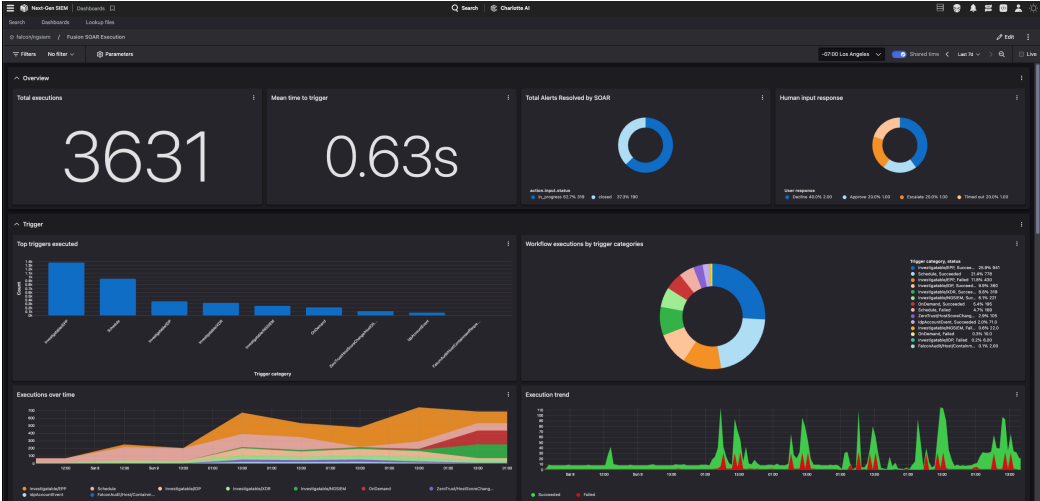
## Query workflow executions

You can query your Falcon Fusion SOAR workflow execution logs using Next-Gen SIEM's advanced event search. To get started, go to [Next-Gen SIEM > Log management > Advanced event search \[/investigate/search\]](#). You can use these queries to build charts and dashboards for the workflows you're most interested in monitoring.

Workflow execution log queries use the standard advanced event search query syntax plus a set of fields specific to workflow data.

## Fusion SOAR execution dashboard

The Fusion SOAR Execution dashboard uses Next-Gen SIEM's advanced queries to show your workflow execution log data in a variety of ways. This dashboard is available to all Falcon Insight LogScale/Next-Gen SIEM customers.



To see the dashboard, go to [Next-Gen SIEM > Log management > Dashboards \[/investigate/search/custom-dashboards\]](#), then search for Fusion SOAR Execution.

The charts in this dashboard include:

- **Execution trend:** An area chart with total counts of succeeded and failed executions over time
- **Executions over time:** An area chart with total counts of executions broken down by category
- **Human input response:** A pie chart with a breakdown of executions requiring a human response
- **Mean time to trigger:** The mean time it takes for in execution to trigger, in seconds
- **Recent executions:** A table of the most recent executions and their status
- **Top actions executing across all workflows:** A table of the top actions executed across all of your workflows
- **Top failing actions across all workflows:** A table of the top failing actions across all of your workflows
- **Top triggers executed:** A bar chart with total counts of the top 10 executed triggers
- **Total Alerts Resolved by SOAR:** A total count of the alerts resolved by SOAR
- **Total executions:** A total count of executions
- **Workflow action executions grouped by vendor and use case:** A pie chart of workflow action executions broken down by vendor and use case
- **Workflow executions by action categories:** A pie chart of workflow executions broken down by action category
- **Workflow executions by trigger categories:** A pie chart of workflow executions by trigger categories

## Data retention

Your workflow log data retention is based on your paid subscription retention period.

## Sample queries

These sample queries use the advanced event search CrowdStrike Query Language. For more info, see [Advanced Event Search \[documentation/page/ic5d7b7d/event-search-advanced\]](#).

### Top executed triggers

This query returns total counts of the most commonly executed triggers.

```
#repo=fusion
| execution_log_type = summary AND execution_log_subtype = start
| rename(field="trigger.data.Trigger.Category", as="Trigger category")
| top(["Trigger category"])
| rename(field="_count", as="Count")
```

ResultsEvents

3614:3514:3614:3714:3814:3914:4014:4114:4214:4314:4414:4514:4614:4714:4814:49

Trigger category	Count
Investigatable/CWPP	90
ReconNotification/ReconNotificationCreation	50
Investigatable/EPP	33
Investigatable/XDR	32
FalconAudit/RTR/SessionStart	31
KubernetesAndContainers/ImageAssessment/Vulnerabilities	21
FalconAudit/Investigatable/Tag	18
CloudSecurityAssessment/Configuration	12
AssetManagement/NewUnmanagedUnsupportedAsset	7
Schedule	2

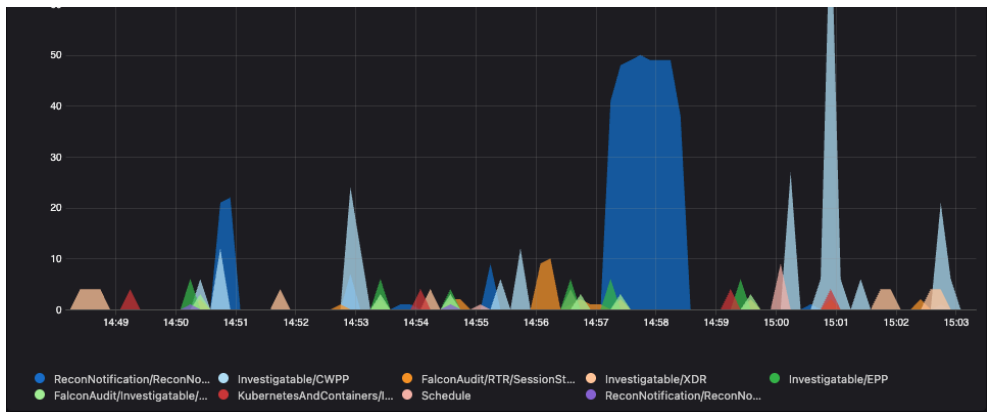
### Executions over time

This query returns a time chart that shows what trigger categories occur the most at specific times.

```
#repo=fusion
| "execution_log_type" = summary AND execution_log_subtype = "start"
| not "parent_execution_id" = * AND not root_execution_id = *
| timechart(span=1d, series=trigger.data.Trigger.Category)
```

ResultsDataEvents

9614:4914:5014:5114:5214:5314:5414:5514:5614:5714:5814:5915:0015:0115:0215:03



## Workflow executions by trigger category

This query returns total counts how many workflow executions there were for each trigger category and their status.

```
#repo=fusion
| execution_log_type = summary AND execution_log_subtype = "end"
| rename(field="trigger.data.Trigger.Category", as="Trigger category")
| groupBy(["Trigger category", status])
```

Results		Events	
Trigger category		status	_count
AssetManagement/NewUnmanagedUnsupportedAsset		Succeeded	1
FalconAudit/Investigatable/Tag		Failed	8
FalconAudit/Investigatable/Tag		Succeeded	13
FalconAudit/RTR/SessionStart		Succeeded	42
Investigatable/CWPP		Succeeded	225
Investigatable/EPP		Failed	3
Investigatable/EPP		Succeeded	36
Investigatable/XDR		Succeeded	28
KubernetesAndContainers/ImageAssessment/Vulnerabilities		Failed	12
KubernetesAndContainers/ImageAssessment/Vulnerabilities		Succeeded	4
ReconNotification/ReconNotificationCreation		Succeeded	403
ReconNotification/ReconNotificationUpdate		Succeeded	1
Schedule		Failed	4
Schedule		Succeeded	6

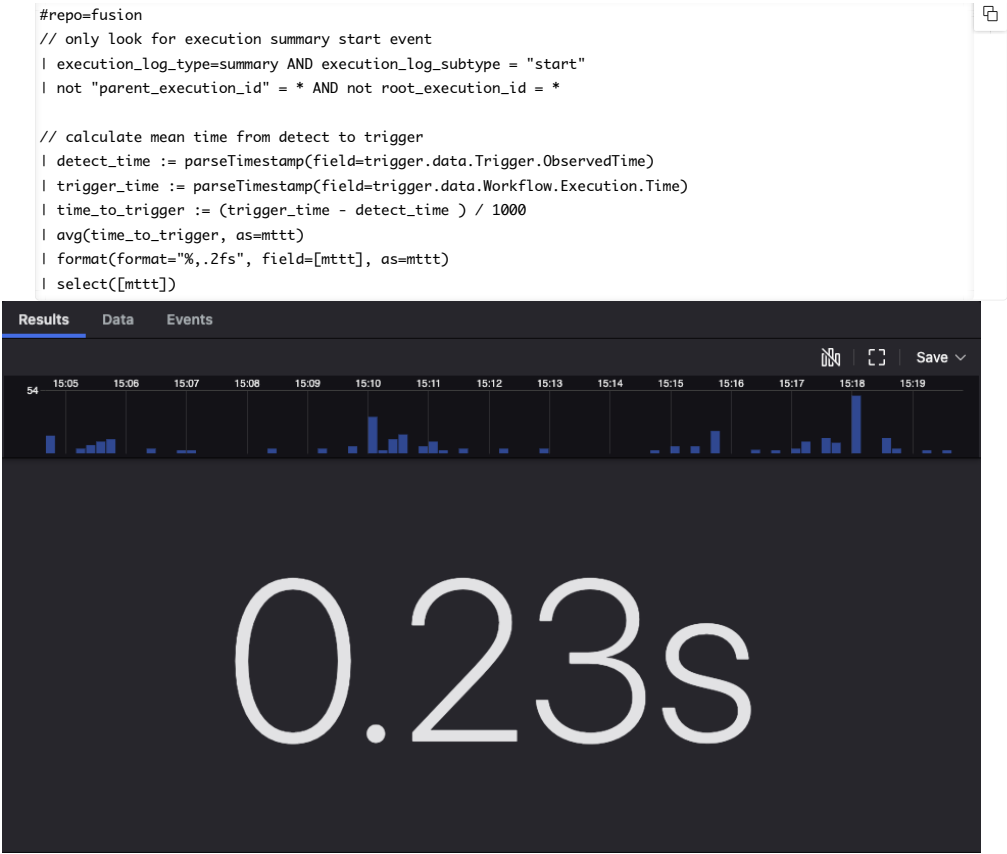
## Recent execution details and durations

This query returns details and durations of recent executions.

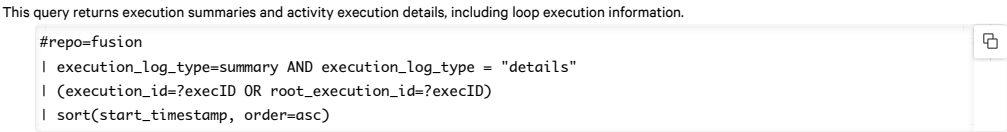
```
#repo=fusion
// only look for execution summary end event
| execution_log_type=summary AND execution_log_subtype = "end"
| not "parent_execution_id" = * AND not root_execution_id = *
| start := parseTimestamp(field="start_timestamp")
| end := parseTimestamp(field="end_timestamp")
| timeProcessed := end - start
| sort(start_timestamp, limit=200)
| formatDuration(timeProcessed, precision=2)
| format(format="%.19s", field=[start_timestamp], as=start_time)
| format(format="%.19s", field=[end_timestamp], as=end_time)
| rename(field="trigger.data.Trigger.ObservedTime", as="detect_time")
| rename(field="trigger.data.Trigger.Category", as="category")
| rename(field="trigger.data.Trigger.SourceEventID", as="source_event_id")
| rename(field="trigger.data.Trigger.SourceEventURL", as="source_event_url")
| select([definition_name, definition_id, definition_version, execution_id, category, status, detect_time,
start_time, end_time, timeProcessed, source_event_id, source_event_url])
| rename(timeProcessed, as=Duration)
```

## Meantime to trigger workflow

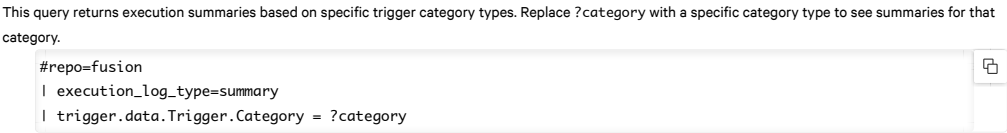
This query returns the average meantime it took to trigger workflows in the selected time period.



Execution summary, activity details, and loop info



Execution status based on specific events



Workflow execution fields query reference

Use these tables to find workflow execution fields that you can query.

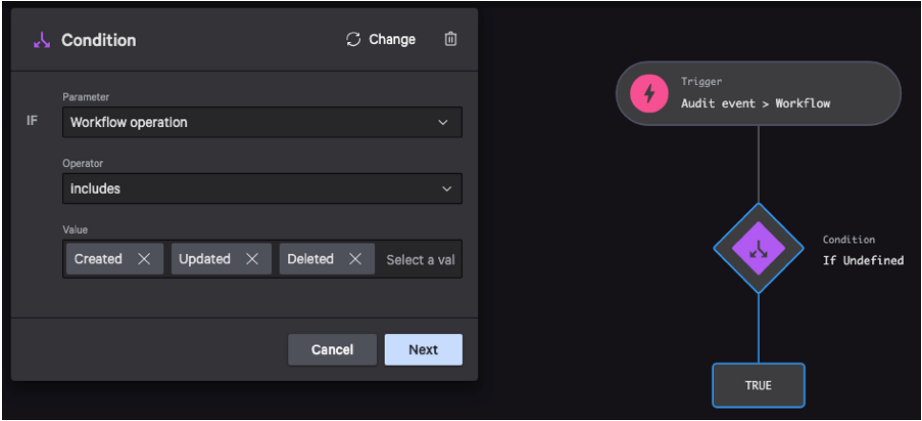
Workflow execution summary and activity fields

Workflow execution field	Description
action	Data object that returns activity detail results as JSON
cid	Customer ID
definition_id	Workflow definition ID
definition_name	Workflow definition name
definition_version	Workflow definition version
end_timestamp	Timestamp when the workflow execution ended. Example: 2024-12-11T22:21:27.691Z
execution_id	Execution summary ID
execution_log_subtype	Execution log subtype: start or loop_start
execution_log_type	Execution log type: summary or details
execution_log_version	The version of the execution log

parent_execution_id	Loop execution summary parent execution ID
root_execution_id	Loop execution summary root execution ID
start_timestamp	Timestamp when the workflow execution started. Example: 2024-12-11T22:21:11.196Z
status	Workflow execution status: Succeeded or Failed
trigger	Data object that returns trigger detail results as JSON.

## Monitor workflow creations, updates, and deletions

To get notifications about workflows being created, updated, or deleted, create a workflow to monitor these operations. In the workflow, use the **Audit event > Workflow** trigger. After the trigger, create a condition that checks whether the **Workflow operation** parameter includes any of the desired operations. Then choose a notification action, such as **Send email** or **Send Slack message**.



## CEL expressions

**Important:** This feature is available to all customers as a beta release. It includes CrowdStrike extensions, which are Common Expression Language (CEL) functions that start with the `cs.` prefix. CrowdStrike extensions are subject to change. During this beta release period, these changes might not be backward compatible. Any changes will be communicated. After general release, changes to CrowdStrike extensions will be backward compatible unless noted otherwise.

In condition nodes and **While** loop conditions, you can manipulate data and write more expressive conditions using Common Expression Language (CEL) functions, as defined in [google / cel-spec](https://github.com/google/cel-spec) [https://github.com/google/cel-spec].

CrowdStrike provides the standard functions as well as some extensions.

- Standard functions  
For info about the standard functions, see [Standard Definitions](https://github.com/google/cel-spec/blob/master/doc/langdef.md#standard-definitions) [https://github.com/google/cel-spec/blob/master/doc/langdef.md#standard-definitions].
- Extensions  
For info about the supported extensions, see [Strings](https://github.com/google/cel-go/blob/master/ext/README.md#strings) [https://github.com/google/cel-go/blob/master/ext/README.md#strings], [Math](https://github.com/google/cel-go/blob/master/ext/README.md#math) [https://github.com/google/cel-go/blob/master/ext/README.md#math], [Lists](https://github.com/google/cel-go/blob/master/ext/README.md#lists) [https://github.com/google/cel-go/blob/master/ext/README.md#lists], and [TwoVarComprehensions](https://github.com/google/cel-go/blob/master/ext/README.md#twovarcomprehensions) [https://github.com/google/cel-go/blob/master/ext/README.md#twovarcomprehensions].
- CrowdStrike extensions  
For info about these extensions, see the following tables.

CrowdStrike provides these extensions:

- [Base64 CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#sce7c520]
- [CSV CrowdStrike extension](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#e54fc75d]
- [Hash CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#na30ac40]
- [IP CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#h66c286e]
- [JSON CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#c968e2ae]
- [List CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#vc0e29c6]
- [Map CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#ld59ee29]
- [Math CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#b170f895]
- [Net CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#sef18e91]
- [String CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#j109ebe6]
- [Table CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#o8c03b04]
- [Timestamp CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#z4dc9688]
- [UUID CrowdStrike extensions](#) [documentation/page/dc4f8c45/workflows-falcon-fusion-1692362310390.669#q9487023]

### Base64 CrowdStrike extensions



Signature	Example	Example result	Description
<code>cs.base64.encode(&lt;string&gt;)</code>	<code>cs.base64.encode('hello')</code>	"aGVsbG8="	Base64 encodes the value
<code>cs.base64.decode(&lt;string&gt;)</code>	<code>cs.base64.decode('aGVsbG8=')</code>	"hello"	Base64 decodes the value

Example use cases:

- An API call needs to return base64-encoded data
- An API call requires data to be base64-encoded

## CSV CrowdStrike extension

Signature	Example	Example result	Description
<code>cs.csv.parse(&lt;string&gt;)</code>	<code>cs.csv.parse('name,age\nbob,22')</code>	<code>[["name", "age"], ["bob", "22"]]</code>	Parses a CSV string and returns a list of lists
<code>cs.csv.parseMaps(&lt;string&gt;)</code>	<code>cs.csv.parseMaps('name,age\nbob,22')</code>	<code>[{"name": "bob", "age": "22"}]</code>	Parses a CSV string and returns a list of maps

## Hash CrowdStrike extensions

Signature	Example	Example result
<code>cs.hash.md5(&lt;string&gt;)</code>	<code>cs.hash.md5('hello')</code>	"5d41402abc4b2a76b9719d911017c592"
<code>cs.hash.sha1(&lt;string&gt;)</code>	<code>cs.hash.sha1('hello')</code>	"aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d"
<code>cs.hash.sha256(&lt;string&gt;)</code>	<code>cs.hash.sha256('hello')</code>	"2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824"
<code>cs.hash.sha512(&lt;string&gt;)</code>	<code>cs.hash.sha512('hello')</code>	"9b71d224bd62f3785d96d46ad3ea3d73319bfbc2890caadae2dff72519673ca72323c3d99b"

## IP CrowdStrike extensions

Signature	Example	Example result	Description
<code>cs.ip.valid(&lt;string&gt;)</code>	<code>cs.ip.valid('4.4.4.4')</code>	true	Returns true if an IPv4 or IPv6 address
<code>cs.ip.isV4(&lt;string&gt;)</code>	<code>cs.ip.isV4('8.8.8.8')</code>	true	Returns true if an IPv4 address
<code>cs.ip.isV6(&lt;string&gt;)</code>	<code>cs.ip.isV6('2001:0db8:85a3:0000:0000:8a2e:0370:7334')</code>	true	Returns true if an IPv6 address
<code>cs.ip.isLoopback(&lt;string&gt;)</code>	<code>cs.ip.isLoopback('::1')</code> && <code>cs.ip.isLoopback('127.0.0.0')</code>	true	Returns true if an loopback IP address
<code>cs.ip.isPrivate(&lt;string&gt;)</code>	<code>cs.ip.isPrivate('fc00::')</code> && <code>cs.ip.isPrivate('10.255.0.0')</code>	true	Returns true if a private IP address
<code>cs.ip.inCIDR(&lt;string1&gt;, &lt;string2&gt;)</code>	<code>cs.ip.inCIDR('10.0.0.0', '10.0.0.0/8')</code>	true	Returns true if the IP address is in the provided CIDR

Example use case:

Check whether a string returned from an action is formatted as an IPv4 or IPv6 address before using it as an input for another action that requires a string formatted as an IPv4 or an IPv6 address

## JSON CrowdStrike extensions

Signature	Example	Description

cs.json.valid(<string>)	cs.json.valid('{ "hello": "world"}')	Returns a boolean indicating whether the string is valid JSON
cs.json.encode(<dyn>)	cs.json.encode(data)	Encodes a JSON object into a string
cs.json.pretty(<dyn>)	cs.json.pretty(data)	Encodes a JSON object into a pretty-printed string—that is, indented and with newlines
cs.json.decode(<string>)	cs.json.decode('{ "hello": "world"}')	Returns a JSON object decoded from a string

## List CrowdStrike extensions

Signature	Example	Example result	Description
cs.list.chunk(<list>, <size>)	cs.list.chunk(["hi", "ho", "he", "hu"], 2)	[[ "hi", "ho"], [ "he", "hu"]]	Chunks a list into sub-lists of the specified size
cs.list.shuffle(<list>)	cs.list.shuffle([1, 2, 3])	[3, 1, 2]	Returns a shuffled version of the provided list; the original list remains unchanged

## Map CrowdStrike extensions

Signature	Example	Example result	Description
cs.map.merge([<map1>, <map2>, ..., <mapN>])	cs.map.merge([{"hi": "ho"}, {"he": "be"}])	{ "hi": "ho", "he": "be" }	Merges 2 or more maps into a single map. The merge is not recursive.
cs.map.mergeDeep([<map1>, <map2>, ..., <mapN>])	cs.map.mergeDeep([{"a": 1, "b": {"c": 2, "d": 3}}, {"e": 4, "b": {"c": 5, "f": 6}}])	{ "a": 1, "b": {"c": 5, "d": 3, "f": 6}, "e": 4 }	Deep merges 2 or more maps into a single map. The merge is recursive.
cs.map.set(<map>, <key>, <value>)	cs.map.set({"hi": "ho", "a", "1"})	{ "hi": "ho", "a": "1" }	Returns a new map with the key inserted with the value. The original map remains unchanged.
cs.map.remove(<map>, <key>)	cs.map.remove({"a": {"b": 1, "c": 2}}, "a.b")	{ "a": {"c": 2} }	Returns a new map with the key removed. The original map remains unchanged.

## Math CrowdStrike extensions

Signature	Example	Example result	Description
cs.math.acos(<val>)	cs.math.acos(-1)	3.14159...	Returns the arc cosine of the provided value
cs.math.asin(<val>)	cs.math.asin(0.5)	0.52360...	Returns the arc sine of the provided value
cs.math.average(<list>)	cs.math.average([1, 2, 3])	2	Returns the average of all the elements in the list
cs.math.cos(<val>)	cs.math.cos(0.5)	0.87758...	Returns the cosine of the provided value
cs.math.random(<x>, <y>)	cs.math.random(5, 10)	7	Returns a random number between x (inclusive) and y (exclusive)
cs.math.sin(<val>)	cs.math.sin(0.5)	0.47943...	Returns the sine of the provided value
cs.math.sqrt(<val>)	cs.math.sqrt(9)	3	Returns the square root of the provided value
cs.math.sum(<list>)	cs.math.sum([1, 2, 3])	6	Returns the sum of all the elements in the list

## Net CrowdStrike extensions

Signature	Example	Example result	Description
cs.net.urlEncode(<string>)	cs.net.urlEncode('@crowdstrike.com more')	a%40crowdstrike.com+more	URL encodes a string
cs.net.urlDecode(<string>)	cs.net.urlDecode('a%40crowdstrike.com+more')	a@crowdstrike.com more	URL decodes a string
		<div> <pre>{   "  " }</pre> </div>	

cs.net.parseURL(<string>)	cs.net.parseURL('https://www.crowdstrike.com/search?id=foo&id=bar&baz=bip#123')	<pre> {   "scheme":   "https",   "host":   "www.crowdstrike.com",   "domain":   "crowdstrike.com",   "path":   "/",   "search":   "port": 80,   "query": {     "id":     ["foo",     "bar"],     "baz":     ["bip"]   },   "fragment":   "123",   "tld":   "com" }</pre>	Returns the URL parsed
cs.net.isURL(<string>)	cs.net.isURL('http://crowdstrike.com')	true	Returns true if a URL <b>Note:</b> Does not validate that the URL is reachable or resolvable
cs.net.isEmail(<string>)	cs.net.isEmail('a@crowdstrike.com')	true	Returns true if formatted as an email address
cs.net.htmlEncode(<string>)	cs.net.htmlEncode('<b>crowdstrike</b><p>yes</p>')	&lt;b&gt;crowdstrike&lt;/b&gt;&lt;p&gt;yes&lt;/p&gt;	HTML encodes a string
cs.net.htmlDecode(<string>)	cs.net.htmlDecode('&lt;b&gt;crowdstrike&lt;/b&gt;&lt;p&gt;yes&lt;/p&gt;')	<b>crowdstrike</b><p>yes</p>	HTML decodes a string

Example use case:

Parse a URL returned from an API call and check whether the domain is an internal domain

## String CrowdStrike extensions

Signature	Example	Example result	Description
cs.string.repeat(<string>, <count>)	cs.string.repeat('xyz', 2)	"xyzxyz"	Repeats a string <count> times. If <count> is less than 0, or the string length exceeds 100K characters, or if string length multiplied by <count> overflows, it returns an error.
cs.string.capitalize(<string>)	cs.string.capitalize('hello THERE')	"Hello there"	Capitalizes the first letter and lowercases the remaining letters.
cs.string.truncate(<string>, <length>)	cs.string.truncate("hello world", 5)	"he..."	Truncates a string to the specified length. If the string did not exceed the specified length, it is returned as is. Otherwise, the ellipses is placed where the truncation occurred. If the returned string includes the ellipses, the 3 characters in the ellipses are then included in the string length.
cs.string.ltrim(<string>)	cs.string.ltrim(" hello ")	"hello "	Trims the left-hand side of the string by removing all whitespace.
cs.string.rtrim(<string>)	cs.string.rtrim(" hello ")	" hello"	Trims the right-hand side of the string by removing all whitespace.
cs.string.find(<string>, <regex>)	cs.string.find("hello 123", '[0-9]+')	"123"	Returns the first substring that matches the provided regular expression.
cs.string.findAll(<string>, <regex>, <limit>) <limit> is optional	cs.string.findAll('hello 123 234', '[0-9]+') or cs.string.findAll('hello 123 234	["123", "234"]	Returns all substrings matching the provided regular expression, with an optional limit.

	456', '[0-9]+', 2)		
cs.string.replaceRegex(<string>, <regex>, <replacement>)	cs.string.replaceRegex('hello 123 234', '[0-9]+', 'wow')	"hello wow wow"	Replaces all substrings matching the provided regular expression, with a string literal replacement.

Table CrowdStrike extensions

Signature	Examples	Example result	Description
cs.table.ascii(<data>)	cs.table.ascii('["A","B"], [1,2]') or cs.table.ascii('{ "A": 1, "B": 2 }')		Returns an ASCII table as a string from the provided data. The <data> argument can be one of these formats: <ul style="list-style-type: none"><li>A JSON string that is a list of lists</li><li>A JSON string that is a list of maps</li><li>A list of lists</li><li>A list of maps</li></ul>
cs.table.html(<data>) or cs.table.html(<data>, <separator>, <style>)	cs.table.html('["A","B"], [1,2]') or cs.table.html('{ "A": 1, "B": 2 }') or cs.table.html('["A","B"], [1,2]', '.', "Pre")	<p>The first example shows the result for the first 2 example calls.</p> <p><b>Note:</b> For brevity, this examples uses snipped in place of actual styling.</p>  <p>This example shows the result for the example call that has the Pre style.</p> 	Returns an HTML table as a string from the provided data. The <data> argument can be one of these formats: <ul style="list-style-type: none"><li>A JSON string that is a list of lists</li><li>A JSON string that is a list of maps</li><li>A list of lists</li><li>A list of maps</li></ul> The <separator> parameter specifies the string to use when separating nested fields. The default is a dot (.). The <style> parameter specifies the table styling. These are the options: <ul style="list-style-type: none"><li>"Pre"</li><li>"None"</li></ul> "Pre" prints the table in ASCII style, but the table is wrapped with <pre></pre> tags. Use "Pre" to send HTML email messages to mail providers that do not support HTML style attributes. "None" removes all style attributes from the table. <b>Note:</b> When you use the <b>Send email</b> action with the output type set to HTML, use "Pre" with this extension to render the table correctly for the person viewing the email.
cs.table.markdown(<data>)	cs.table.markdown('["A","B"], [1,2]') or cs.table.markdown('{ "A": 1, "B": 2 }')		Returns a Markdown table as a string from the provided data. The <data> argument can be one of these formats: <ul style="list-style-type: none"><li>A JSON string that is a list of lists</li><li>A JSON string that is a list of maps</li><li>A list of lists</li><li>A list of maps</li></ul>

Timestamp CrowdStrike extensions

Signature	Example	Description
cs.timestamp.now()	cs.timestamp.now()	Returns the current time in UTC as a timestamp object
cs.timestamp.format(<timestamp>, <layout>)	cs.timestamp.format(data['Trigger.LastUpdated'], 'RFC822')	Formats a timestamp object into a string using the specified layout or format, such as "RFC3339" or a custom format like "Mon Jan 02 15:04:05 -0700 2006"

<code>cs.timestamp.parse(&lt;string&gt;, &lt;layout&gt;)</code>	<code>cs.timestamp.parse('2025-01-02 10:01:22', 'RFC3339')</code>	Returns a timestamp object by parsing a string according to the specified layout or format, such as "RFC3339" or a custom format like "Mon Jan 02 15:04:05 -0700 2006"
---	---	--

Example use case:

To see if more than 7 days, or 168 hours, have elapsed since an incident started, use this expression:

`cs.timestamp.now() - timestamp(data["Trigger.Category.Incident.StartTime"]) > duration('168h')`

📄

## UUID CrowdStrike extensions

Signature	Example	Description
<code>cs.uuid.canonical(&lt;string&gt;)</code>	<code>cs.uuid.canonical('EFF9770113E640FF89086F313E7CCA6B')</code>	Returns the UUID in its canonical form--that is, lowercase with hyphens. The result of the example is <code>eff97701-13e6-40ff-890b-6f313e7cca6b</code> .
<code>cs.uuid.new()</code>	<code>cs.uuid.new()</code>	Returns a UUID v4 as a string, such as <code>c72db0e4-e165-414e-9dae-32009a6fa97b</code> .
<code>cs.uuid.valid(&lt;string&gt;)</code>	<code>cs.uuid.valid('c72db0e4-e165-414e-9dae-32009a6fa97b')</code>	Returns whether a string is a valid UUID. <b>Note:</b> This returns true if it is a valid UUID, regardless of version, such as UUIDv4 vs. UUIDv7.

Example use case:

Check whether a string returned from an API call is formatted as a CrowdStrike sensor UUID