Splunk® Add-on Builder
Splunk Add-on Builder User Guide 4.1.2

Create alert actions

Generated: 2/15/2023 10:34 pm

Create alert actions

In **Create Alert Actions**, create and configure alert actions to make them available to your add-on's users. You can use Alert actions to define third-party integrations, add custom functionality, or support adaptive response actions for Splunk Enterprise Security.

After you add alert actions to your add-on, you can manage them from the Alert Action page under **Create Alert Actions**. Your alert actions also appear on the **Settings** > **Alert Actions** page in Splunk Web.

Create an alert action

1. On your add-on homepage, click the **Create Alert Actions** icon.
2. On the **Alert Actions** page, click **New Alert Action**.
   The **Create Alert Action** wizard starts.
3. On the **Alert Action Properties** page, enter the properties for this alert action:

   - Enter a name, label, and description for this alert action. The label is the friendly name that appears in Splunk Web.
   - Optionally, click **Upload my icon** to add an icon, such as a logo, to this alert action.
   - To create an adaptive response action, see Create an adaptive response action for Enterprise Security, below.

- Click **Next**.
- On the **Alert Action Inputs** tab, specify one or more input fields for this alert action.

  For each input, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.
  If certain permission is required to trigger the alerts, you can drag **Text field** and **Password** field on the Component Library, specify the properties and then add the account in the **Add-on Setup Parameters** page as the example below

- Optionally, click the **Add-on Setup Parameters** tab to define any parameters that are required for users to configure your add-on.

  - Select predefined options to prompt for account, proxy, or logging settings.
  - Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

    For more about setup pages, see Create a setup page.

- Click **Next**.
- On the **Alert Action Parameters** tab, use the Code Editor panel to edit the Python code to create your alert action.Find the comment `# TODO: Implement your alert action logic here`, which indicates where to add your code.
- The Python helper functions, which are dynamically generated as commented code, can help you as a guide to working with the input parameters you defined in the previous step.
- If you want to collect data using SSL certificate, use **send_http_request** function as follows.

```
import solnlib.splunkenv

#The cert file locates in <TA_folder>/default/example.pem

cert_file_path = solnlib.splunkenv.make_splunkhome_path(['<TA_folder>', 'default', 'example.pem'])
response = helper.send_http_request('https;//www.example.com/api', 'GET', verify=cert_file_path)
```

- helper.settings is a dict that contains information including SPL, Splunk environment configurations and alert configurations. If you want to use the information in helper.settings, use the following sample code as an example.

  ```
  Syntax:
  search_name = helper.settings.get('search_name')
  sid = helper.settings.get('sid')
  ```
- On the **Alert Action Parameters** tab, enter sample values for testing this alert action.
- If you created a setup page, on the **Add-on Setup Parameters** tab enter sample values to test the setup page.
- Click **Test** to test your code and the alert action configuration. The **Output** section indicates whether the command succeeded or failed.
- Click **Save**, then click **Finish**.

Create an adaptive response action for Enterprise Security

An adaptive response action is a type of alert action that is specifically created for Splunk Enterprise Security. An adaptive response action can be triggered from correlation searches or on an ad hoc basis when examining a notable event on the Incident Review dashboard.

Use the Splunk Add-on Builder to create an alert action that supports adaptive response. Adaptive response actions must conform to the Common Action Model, which is included with the Splunk Common Information Model add-on.

> **Note** The Splunk Common Information Model add-on version 4.6.0 or later is required by:
>
> · Add-on developers to create adaptive response actions
> · End users of your add-on to run adaptive response action
> Download the Splunk Common Information Model add-on from Splunkbase.

To create an adaptive response

1. On your add-on homepage, click the **Create Alert Actions** icon.
2. On the **Alert Actions** page, click **New Alert Action**.

   The **Create Alert Action** wizard starts.
3. On the **Alert Action Properties** page, enter the properties for this alert action:

   - Enter a name, label, and description for this alert action. The label is the friendly name that appears in Splunk Web.
   - Optionally, click **Upload my icon** to add an icon, such as a logo, to the alert action.

- Select **Support as an adaptive response action in Splunk Enterprise Security** and fill out the fields as follows:

  - For **Category**, enter the categories the action belongs to, such as "Information Gathering".
  - For **Task**, enter the functions performed by the action, such as "scan".
  - For **Subject**, enter the objects that the action's tasks can be performed on, for example, "endpoint.file".
  - For **Vendor**, enter the technology vendor that the action supports.
  - For **Product**, enter the products that the action supports.
  - For **Version**, enter the versions of the product that the action supports.

- Select **Support as an ad hoc action** if the action supports ad hoc invocation from the Actions menu on the Incident Review dashboard in Splunk Enterprise Security. For help determining whether your action supports ad hoc invocation, see Determine whether your action supports ad hoc invocation on the Splunk Developer Portal.

  Then, fill in the related fields as follows:

- Optionally, for **Custom drilldown**, enter a URL to a custom drilldown or view for the link that appears in the detailed view of a notable even on the Incident Review dashboard in Splunk Enterprise Security. If you don't specify a URL, the default URL runs a search for the result events created by this response action.

  To specify a target in an app outside Enterprise Security, use the format `../<app_context>/<viewname>?<additional drilldown parameters>`. For example:

  ```
  ../my_app/my_view?form.sid=$orig_sid$&form.rid=$orig_rid$
  ```

  To redirect to a custom view within Enterprise Security, use the format `/<viewname>?<additional drilldown parameters>`. For example:

  ```
  /my_view?form.sid=$orig_sid$&form.rid=$orig_rid$
  ```
- For **Sourcetype**, enter the source type to which to assign the events produced as a result of this response action.

- Click **Next**.
- On the **Alert Action Inputs** tab, specify one or more input fields required for configuring the alert action.

  For each input, drag a field from the Component Library to the center panel, then specify its properties in the Property Editor.

- Optionally, click the **Add-on Setup Parameters** tab to define any parameters that are required for users to configure your add-on.

  - Select predefined options to prompt for account, proxy, or logging settings.
  - Add custom inputs by dragging fields from the Component Library to the center panel panel. Select an input to specify its properties in the Property Editor.

    For more about setup pages, see Create a setup page.

- Click **Next**.
- On the **Alert Action Parameters** tab, use the Code Editor panel to edit the Python code to create your alert action. Find the comment `# TODO: Implement your alert action logic here`, to see where to add your code.
- Use the Python helper functions, which are dynamically generated as commented code, as a guide to working with the input parameters you defined in the previous step.
- For example Python code showing how to create an adaptive response, see Walkthrough: Create an ES adaptive response action on the Splunk Developer Portal.
- On the **Alert Action Parameters** tab, enter sample values for testing the alert action.
- If you created a setup page, on the **Add-on Setup Parameters** tab, enter sample values to test the setup page.
- Click **Test** to test your code and the alert action configuration. The **Output** section indicates whether the command succeeded or failed.
- Click **Save**, then click **Finish**.

You should also test and validate your response action in Enterprise Security. For more, see Validate your response action in Enterprise Security on the Splunk Developer Portal.

Pass values from setup parameters

When referring to setup parameters, include a namespace string before the parameter name to get the value from the parameter. For example, where `param_name` is the parameter name:

```
${__settings__.additional_parameters.param_name}
```

If you have a text input on your setup page that prompts the user for an API token, with the internal name "api_token", use the following format to pass the value of the API token to the REST call:

```
api-key=${__settings__.additional_parameters.api_token}
```

Do not set account, proxy, or logging fields directly. You can access the global account values as follows:

```
${global_account.username}
{{global_account.username}}

${global_account.password}
{{global_account.password}}
```

Read user credentials from multiple accounts

Alert actions do not support the **Global Account** input field. When you have alert actions that require different user credentials for different accounts, such as one account for production and one for development, set up your alert actions to read specific user credentials from the setup page.

The following workflow shows one way to read multiple credentials of a setup page.

1. Create an alert action.
2. When creating a setup page on the **Add-on Setup Page Parameters** tab in the **Create Alert Action** wizard, select **Add Global account settings** to prompt users to add credentials for one or more accounts.
3. When defining the input variables for the alert action, add a Text field that prompts for the username of an account. The value of this text field acts as a dictionary key to look up the corresponding credentials from the setup page.
4. When defining the code for the alert action, use the following code to retrieve a username and password from the setup page.

   Use the same internal name (replace "internal_name" below) that you used for the text field from the previous step:

   ```
   account = helper.get_user_credential(helper.get_arg("internal_name"))
   ```

   The data returned by the **helper.get_user_credential** function is a JSON dictionary, so retrieve the username and password from the `account` dictionary as follows:

```
username = account["username"]
password = account["password"]
```

- Save your alert action, then restart Splunk Enterprise.

For more, see the Python helper functions.

Test your alert action

Use the **sendalert** command using a hard-coded value to teat your alert action:

1. Build your alert action and save it.
2. Restart Splunk Enterprise.
3. Use **sendalert** command:

   ```
   index=_internal | head 1| eval fieldname="xyz" | sendalert myalertname param.abc="myvalue"
   ```

To pass search result values dynamically to different alert action parameters, use the `$result.fieldname$` format:

```
index=_internal | head 1| eval fieldname="xyz" | sendalert myalertname param.abc="$result.fieldname$"
```

For a list of possible alert action tokens, see Pass search result values to alert action tokens in the *Developing Views and Apps for Splunk Web* manual.

Learn more

For more information, see the following documentation:

- For creating adaptive response actions, see the Adaptive Response Framework on the Splunk Developer Portal
- For creating adaptive response actions, see Use the common action model to build custom alert actions in the *Common Information Model Add-on Manual*
- For installing the common action model, see Install the Splunk Common Information Model Add-on in the *Common Information Model Add-on Manual*