Splunk® Security Essentials
Develop Custom Content in Splunk Security Essentials 3.6.0

Generated: 10/31/2022 5:33 am

Table of Contents

Get started integrating custom content

Get started integrating custom content in Splunk Security Essentials

This manual is for partners or internal users who are developing on the Splunk Security Essentials (SSE) platform.

As a third-party developer, you can develop, customize, and extend the capabilities of SSE to help users find and deploy appropriate security procedures. By building add-ons and enabling them for users, you can integrate custom content with Splunk Security Essentials to help users analyze that content against MITRE ATT&CK frameworks and track and report their successes.

Use this documentation to help you do the following tasks:

- Add third-party content to Splunk Security Essentials
- Configure content using the ShowcaseInfo.json schema in Splunk Security Essentials
- Author simple and full-feature content on Splunk Security Essentials
- Use the Splunk Security Essentials file directory

Add third-party content

Third-party content integration in Splunk Security Essentials

Third-party developers can integrate and publish their own custom content into Splunk Security Essentials (SSE) environments. After you integrate custom content for users, users can analyze that content in the same way as the other content that ships with SSE.

SSE also supports the Splunk partner ecosystem. If you are a commercial security partner or an open-source provider, you can add your content into SSE. Users can then track the content they have and showcase how the content helps them meet their needs.

Add content in SSE

To add content in SSE, complete the following steps:

1. Convert the content into the SSE format.
2. Post the content for download.
3. Add the content to Splunkbase.

Convert content to the SSE format

To convert content that exists as active searches in the savedsearches.conf file into the SSE format, see Configure Splunk Security Essentials in the *Install and Configure Splunk Security Essentials* manual. If the content exists in a different repository, such as a database, you can create custom code that converts the content into the SSE format.

Post content for download

SSE downloads new content through the browser. There are no requirements for proxy configurations within Splunk Enterprise. Every time a page loads, a user's browser performs a lookup using external_content_lookup to determine when content was last updated. If more than one day has passed since the last lookup, Splunk Enterprise updates the content automatically.

If build_url and build_field are configured, SSE performs a GET request of build_url, expecting a JSON object, and looks for the buildnum. If the buildnum is not equal to the last buildnum received, the download proceeds. If no build_url exists, SSE still downloads all the content. To download content, SSE performs a GET request from the content_download_url and expects all the content to be contained within a JSON object.

If you store the content in GitHub, use the raw URL format. If you store the content in S3, configure the bucket to allow Cross-Origin Resource Sharing (CORS). For additional information about configuring CORS in S3, see How Do I Allow Cross-Domain Resource Sharing with CORS or this example in StackOverflow.

Add content to Splunkbase

To direct SSE to your add-on, create a stanza in the essentials_updates.conf file in the add-on.

Here is an example of what the configuration in Default/essentials_updates.conf looks like:

```
[ButtercupLabs]
channel=ButtercupLabs
name=Buttercup Labs
description=Buttercup Labs produces quality security analytics run through Splunk. Although Buttercup Labs
sells a commercial threat analytics app, the company has also released community content. All that content
is available through Splunk security Essentials.
type=app
app_context=Splunk_Security_Essentials
content_download_url=https://go.splunksecurityessentials.com/myContentLocation
```

Follow these guidelines when you create the stanza and the channel:

- Make sure the stanza name is unique to your organization.
- Make sure nothing else references the stanza name.
- Don't use spaces in either the stanza name or the channel name.
- Consider matching the stanza name with the channel name to help keep them organized.

The channel is configured on the backend and doesn't affect users.

The name of the stanza appears in many places throughout the app. It appears in filters on the **Security Content** page as well as when users view the content. The description only appears in the app configuration, where users can enable and disable different content sources.

Make sure the following is also true:

- The type must be an app.
- The app_context must be Splunk_Security_Essentials.
- The content_download_url must be the URL to where users can download the app.

When you test in your own environment, restart Splunk Enterprise after making any changes to the essentials_updates.conf file so that Splunk Enterprise rereads that file.

Create an entry in metadata/default.meta within your environment. By default, Splunk Enterprise doesn't share configurations to all systems in the app, but default.meta allows you to set configurations.

Metadata/default.meta (added to the bottom of the configuration):

```
[essentials_updates]
export = system
```

Add custom fields to Splunk Security Essentials

As a third-party developer, it might be helpful to provide users with details about your company. You can create custom fields, such as name and description, so that users can see that content. Here is an example of added custom fields and how they appear in the code:

```
{
    "company_description": "Buttercup Labs is the premier distributor of Pony-related security analytics. We
have been protecting organizations from bad ponies for over ten years now.\n\\n\\n\nEnjoy our freely
available content for detecting bad ponies in your environment, and reach out to us for a demo or trial
license of our premium Pony Detection app!\n\\n\\n\nHave you successfully found bad ponies in your own
environment? Buttercup Labs is hiring! We are a wholly owns subsidiary of Buttercup Games.",
    "company_link": "http://buttercupgames.com/",
    "company_logo":
"https://image.slidesharecdn.com/splunklivesfhowtoalignyourdailysplunkactivitiesbreakoutsession-160317192319/95/how
-to-align-your-daily-splunk-activities-breakout-session-23-638.jpg?cb=1458242654",
    "company_logo_height": 250,
    "company_logo_width": 444,
    "company_name": "Buttercup Labs"
}
```

Use included hard-coded fields

Splunk Security Essentials (SSE) includes the following hard-coded fields:

- Search
- Known False Positives
- How to Implement
- How to Respond
- Help
- Company Information

All of these fields appear as dedicated accordions on the **Custom Content** page. If you need additional categories, you can define those categories in additional_context, which is an array of objects. Each object shows up as a new accordion in the UI and contains up to five of the following fields:

Field name	Description
title	This field is text-only and provides the name of the accordion. If it's empty, it defaults to "Additional Context."
open_panel	This field can be true or false. The accordion is open by default.
detail	This field is for an optional markdown text block.
link	This field is for an optional URL users can follow to learn more. If it's filled, a button appears with the label **Learn More...**.
search_label	This field is text-only and provides the label for the search immediately before the pre. If it's empty, it defaults to "Search."
search_lang	This field is for an optional definition of what language the code is in. If you're using SPL, you can leave this field blank or provide the information as conf if you want to print file options. If you need to print options like Python, the default languages for highlight.js are supported. Valid options include properties, Python, Rust, Less, Perl, diff, SCSS, Bash, Shell, Makefile, JSON, INI, HTTP, CoffeeScript, CSS, Objective C, Ruby, YAML, Java, SQL, Apache, Kotlin, XML, Markdown, Swift, Plaintext, TypeScript, NGINX, Go, Javascript, PHP, CS, Lua, and CPP.
search	This field is for SPL or any other raw code.

The following code is an example of how to implement the search field:

```
{
    "additional_context": [
        {
            "search": "index=* sourcetype=ponies",
            "open_panel": true,
            "title": "Additional Potential Search",
            "link": "https://www.splunk.com/blog/2016/09/28/buttercup-games.html",
            "detail": "### Background\n* You may wish to look at all data regarding ponies.\n* Here you will
find all of the detail produced by Buttercup Labs Pony Monitoring"
        },
        {
            "title": "Conf file for Adding a New SSE Channel",
            "search": "[ButtercupLabs]\nchannel=ButtercupLabs\nname=Buttercup Labs\ndescription=Buttercup
Labs produces high quality security analytics run through Splunk Enterprise. While Buttercup sells a
commercial threat analytics app, they have also released a variety of community content. All is available
through Splunk Security
Essentials.\ntype=app\napp_context=Splunk_Security_Essentials\ncontent_download_url=https:/
/go.splunksecurityessentials.com/myContentLocation",
            "detail": "Adding this file and restarting Splunk will register a new content source into SSE.",
            "search_lang": "conf",
            "search_label": "Example Configuration",
            "open_panel": true
        }
    ]
}
```

If you're using SPL, an Open In Search button appears below the following:

```
if(additional_context[num].search){
    output += $("<div>").append($("<pre>").attr("class", "search").append($("<code>").attr("class",
lang).text(obj.search))).html()
}
```

Best practices for integrating content with Splunk Security Essentials

The following methods can help you seamlessly integrate custom content as a third-party developer.

Add a configuration to avoid a Splunk Enterprise restart after installation

During development, you might need to restart Splunk Enterprise to make the system recognize changes made to the configuration. However, users who install your app don't need to restart Splunk Enterprise. To avoid restarting Splunk Enterprise, add this configuration to app.conf in the default folder of your app:

```
[triggers]
reload.essentials_updates = simple
```

Ship the content through your app

If you want to ship the content through your app, rather than provide auto-updates, use this simplified method. For this, you don't need a web server. Instead, you can ship a JSON file through your app using the appserver/static folder.

For this method, follow these steps:

1. Create an appserver/static folder, if one doesn't exist already, and put the JSON file in there.
2. In the default/essentials_updates.conf file, start the line with "SPLUNKD" so that SSE searches for local updates.
3. Apply the default.meta and app.conf settings.

This is an example of what the default/essentials_updates.conf file should look like:

```
[ButtercupOfflineContent]
channel=ButtercupOfflineContent
order=10
name=Buttercup Offline Content
type=app
app_context=Splunk_Security_Essentials
content_download_url=SPLUNKD/static/app/Buttercup_App/mycontent.json
```

> You can choose any name for your app because there isn't a naming scheme. To help users find the app on Splunkbase, include "Security Essentials" in the title.

Verify your configuration

View your content by installing the app on your local machine. After you install it, run the following search to see important details about your content:

```
| sseanalytics include_json=true | search channel=ButtercupLabs | table id summaries *
```

Populate the data inventory

One of the main configuration interfaces in SSE is the Data Inventory configuration. By default, custom content doesn't appear there. In the default case, map your data_source_categories field to standard Data Source Categories (DSCs)â   for example, DS0001MAIL-ET02Receive.

To have your custom content appear as a vendor-specific-data-source category, include this code at the end of your configuration:

```
{
    "create_data_inventory": true
}
```

If that code is present, SSE creates a custom DSC with the data you've provided in the company_* fields.

When pullJSON.py pulls in the data_inventory.json, the python script inspects the custom_content kvstore collection, which is on the back end of the partner framework. That script includes any detections with a create_data_inventory=true configuration and processes the first piece of content found using this logic:

```
dscid = "VendorSpecific-" + row['channel']
baseSearch = "index=NOTAPPLICABLE TERM(No baseSearch Provided)"
legacyName = "Unknown Channel: " + row["channel"]
shortUnifiedName = "Unknown Channel: " + row["channel"]
description = "No Description Provided"
commonProductNames = []
if "company_description" in customJSON:
    description = customJSON['company_description']
if "company_name" in customJSON:
    legacyName = customJSON['company_name']
    shortUnifiedName = customJSON['company_name']
    commonProductNames.append(customJSON['company_name'])
if "company_base_spl" in customJSON:
    baseSearch = customJSON['company_base_spl']

data_inventory["VendorSpecific"]["eventtypes"][dscid] = {
    "baseSearch": baseSearch,
    "legacy_name": legacyName,
    "short_unified_name": shortUnifiedName,
    "description": description,
    "name": legacyName,
    "common_product_names": commonProductNames
}
```

As you work through the steps in the process of creating and bundling content into SSE, be aware of some limitations:

1. Because this process only ships JSON files into SSE, you must host any logo online.
2. Avoid excessive text in your content. This process requires SSE to download the content in SSE every time it refreshes, so large amounts of content can slow SSE, even though Splunk compresses the content by default.
3. In this SSE interface, adding demo data or different versions of searches is unsupported.

Secure app and clean data

Splunk Enterprise cleans all data provided through the partner framework to ensure the safety of users' machines. The cleaning configuration is hard coded into generateShowcaseinfo.py, the primary REST endpoint that powers SSE.

You can only pull content from the following fields in the partner content framework using the corresponding data-cleansing methods.

Field	Data type
bookmark_notes	string
bookmark_status	string
bookmark_status_display	string
bookmark_user	string
datasource	string
create_data_inventory	boolean
datasources	string
name	string
inSplunk	string
journey	string
usecase	string
highlight	string
alertvolume	string
severity	string
category	string
description	string

Field	Data type
domain	string
gdpr	string
gdprtext	string
hasSearch	string
mitre	string
released	string
killchain	SPLEase
searchkeywords	string
advancedtags	string
advancedtags	string
printable_image	string
icon	string
company_logo	string
company_logo_width	string
company_logo_height	string
company_name	string
company_description	string
company_link	string
dashboard	string
relevance	string
help	string
howToImplement	string
knownFP	string
operationalize	string
search	spl
data_source_categories	string
mitre_technique	string
mitre_tactic	string
open_search_panel	boolean
additional_context	array
additional_context.title	string
additional_context.search_label	string
additional_context.detail	string
additional_context.link	string

Field	Data type
additional_context.search_lang	string
additional_context.search	spl
additional_context.open_panel	boolean

Depending on the data type you work with, SSE uses one of the following cleaning methods:

- String
- Number
- Boolean
- Array
- Search Processing Language (SPL)

String

BeautifulSoup (bs4) in Python strips HTML:

```
obj[field] = BeautifulSoup(obj[field], "lxml").text
```

Number

BeautifulSoup (bs4) in Python strips HTML:

```
obj[field] = BeautifulSoup(obj[field], "lxml").text
```

Boolean

Use this statement to check for a boolean data type and delete if not:

```
if not isinstance(obj[field], bool):
    debug.append({"status": "WARN", "msg": "clean_content, deleting field because it's not actually a bool",
"path": path, "field": field, "value": obj[field]})
    del obj[field]
```

Array

Make sure each row in the array is an object. Use this statement to recursively call the clean_content function.

```
if key_checking[path + field] == "array":
    for i in list(range(0, len(obj[field]) )):
        if isinstance(obj[field][i], object):
            obj[field][i] = clean_content(obj[field][i], key_checking, path="")
```

SPL

Pass input into SSE through jQuery and into a pre-tag vsing .text(), which ensures that no code is executed:

Troubleshoot data-cleaning issues

In any case where you need to troubleshoot ShowcaseInfo, check the debug output for logs using the JavaScript console. Any time a field is deleted from SSE, it logs the event in the debug output:

```
require(['json!' + $C['SPLUNKD_PATH'] + '/services/SSEShowcaseInfo?bust=' + Math.random()],
function(showcase){
    window.debug_showcase = showcase;
    console.log("Got Showcase", showcase.debug);
})
```

Configure content using the ShowcaseInfo.json schema

Use the schemas in Splunk Security Essentials

Splunk Security Essentials (SSE) requires a standard format for all content in the app. You can find the formatting in the following locations:

Location	Format
Main	ShowcaseInfo.json
Native SSE	showcase_simple_search.json showcase_first_seen_demo.json showcase_standard_deviation.json

Most use cases only require ShowcaseInfo.json. For example, in partner integrations, configurations automatically merge into ShowcaseInfo.json, and you don't need to use any of the search-builder files.

Splunk Security Essentials uses the following two schemas:

- ShowcaseInfo.json. To reference the ShowcaseInfo.json schema and see examples of it in the code, see ShowcaseInfo.json schema.
- Search builder content. To reference the search builder content schema and see examples of it in the code, see Search builder.

ShowcaseInfo.json schema

The following table is a point-in-time reference of the fields in the ShowcaseInfo.json file.

JSON field name	Descriptiive field	Relevance	Description	Examples
name	Name	All	The name of the example (e.g., "New Local Admin Account.") Include a maximum of 150 characters and avoid punctuation if possible.	Basic Brute Force Detection
description	Description	All	The primary description of your custom content. Include 250â 300 characters.	
highlight	Featured	All	Whether content should appear on the main page.	No
advanced tags	Advanced Tags	All	Optional tags in the Advanced filter.	Cool Search
alertvolume	Alert Volume	All	Categorize the volume of the search.	Medium
bookmark_status	Bookmarked Status	All	The method SSE uses to track content.	Bookmarked

JSON field name	Descriptiive field	Relevance	Description	Examples
category	Category	All	A user-defined field.	IAM Analytics
data_source_categories	Data Source Categories	All	The data-source category ID that maps to data_inventory.json.	DS0003Authentication-ET02Failure
domain	Security Domain	All	The security domain. Options are "Access," "Network," "Endpoint," "Threat," and "Other."	Access
hasSearch	Has Search	All	Whether the detection includes the related search string.	Yes
icon	Icon	All	The icon that appears wherever the icon tile appears, using BuildTile.js.	
journey	Journey	All	The stage of the journey at which the content appears.	Stage_1
killchain	Kill Chain Phase	All	The phase of the kill chain.	Actions on Objectives
mitre_tactic	MITRE ATT&CK and Pre-ATT&CK Tactics	All	The pipe-delimited list of MITRE ATT&CK Tactic IDs.	TA0006
mitre_technique	MITRE ATT&CK and Pre-ATT&CK Techniques	All	The pipe-delimited list of the MITRE ATT&CK Technique IDs.	T1110
released	Released	All	The release number.	3.0.0
searchkeywords	Search Keywords	All	The automatically indexed search keywords, which include the description, title, category, use case, response, implementation, and help, as well as known false positives. To add highly weighted custom keywords, include them as space separated values.	login log in logon log on sign
severity	Severity	All	The severity of the event. This value does doesn't surface in the UI but is available as an enrichment field via the sseanalytics command.	Medium

JSON field name	Descriptiive field	Relevance	Description	Examples
SPLEase	SPL Ease	All	The level of expertise to use SPL for your content.	Basic
usecase	Use Case	All	The high-level description of the use case for your content.	Security Monitoring
additional_context	Array of Custom Panels	All	The option to include custom panels.	
additional_context	Custom Panel: Detail	All	Optional text block. Supports Markdown.	
additional_context.link	Custom Panel: Link	All	An optional URL that allows users to navigate to a helpful website. If defined, a **Learn Moreâ** ¦ button appears.	Learn moreâ ¦
additional_context.open_panel	Open Custom Panel	All	Whether the custom panel is open by default.	
additional_context.search	Custom Panel: Code Block Contents	All	A display of raw code, like SPL. Code highlighting appears automatically.	
additional_context.search_label	Custom Panel: Code Block Title	All	A text-only label for the search, directly before the pre. If undefined, displays "Search."	Example
additional_context.search_lang	Custom Panel: Code Block Language	All	An optional tag to identify the language of the code on display. If undefined, identifies SPL. The default languages for highlight.js are supported.	conf
additional_context.title	Custom Panel: Title	All	A text-only label that names a custom panel. If undefined, displays "Additional Context."	Config settings
help	Help	All	A field that allows users to seek help. Supports Markdown.	
howToImplement	How to Implement	All	Optional text that describes to how implement a detection. Supports Markdown.	
knownFP	Known False Positives	All	Optional text that describes the known false positives that are created in a search. Supports Markdown.	
operationalize	How to Respond	All		

JSON field name	Descriptiive field	Relevance	Description	Examples
			Optional text that describes how to respond to a search. Supports Markdown.	
printable_image	Printable Image URL	All	Optional screenshot showing demo results when creating a PDF export.	
relevance	Security Impact	All	Text describing why your content is important. Supports Markdown.	
open_search_panel	Open Search Panel	All	Whether the search panel is open by default. If you provide a search string, this setting defaults to true.	Text String: "False"
search	Search String	All	Optional search string.	
search_name	Saved Search Name	All	The saved search name available in Correlation Search Introspection. Maps to the stanza name in savedsearches.conf.	ESCU - Abnormally High AWS Instances Terminated By User - Rule
company_description	Company Description	Partners	Description of your company. Supports Markdown.	Your Short Company Description
company_link	Company Link	Partners	An optional URL that allows users to navigate to a your company's website. If defined, a **Learn More** button appears.	https://www.splunk.com
company_logo	Company Logo	Partners	The URL and dimensions to an image of your company's logo. The dimensions must be less than 250 px in height and 500 px in width.	https://â ¦/yourlogo.png
company_logo_height	Company Logo: Height	Partners	The height of your company's logo in pixels.	200
company_logo_width	Company Logo: Width	Partners	The width of your company's logo in pixels.	400
company_name	Company Name	Partners	The name of your company.	Company Name
create_data_inventory	Create data Inventory	Partners	If defined as "True," creates an entry in the Data Inventory and overwrites existing data_source_categories.	Obj: True

JSON field name	Descriptiive field	Relevance	Description	Examples
inSplunk	Solved in Splunk	Partners	Marks functionality that exists in an environment but outside Splunk.	Yes
escu_cis	ESCU - CIS Mapping	ESCU Authors	ES Content Update Specific: The CIS Mapping	CIS 13
escu_data_source	ESCU - Data Source	ESCU Authors	ES Content Update Specific: The Data Sources	AWS CloudTrail Logs
escu_nist	ESCU - NIST Mapping	ESCU Authors	ES Content Update Specific: The NIST Mapping	DE.DP
escu_providing_technologies	ESCU - Providing Technologies	ESCU Authors	ES Content Update Specific: The technologies that enable a detection	AWS
story	ESCU - Story ID	ESCU Authors	ES Content Update Specific: Story ID	2e8948a5-5239-406b-b56b-6c50f1268af3
examples	Array of Examples for Detection	SSE Authors	An added object to the examples array. One adds for each kind of search (Live Data, Demo Data, Accelerated Data, etc.).	
examples.label	Name for Example	SSE Authors	The name of the example.	Live Data
examples.name	ID in Search Builder JSON	SSE Authors	The machine-readable name of the content. Don't include spaces or special characters. Custom content is overwritten by the channel definition.	Splunk_Security_Essentials
dashboard	Dashboard	SSE Authors	The name of a dashboard users should go to after they select its name. Append any fields to the name.	showcase_simple_search?ml_toolkit.dataset=Basic Brute Force - Demo
displayapp	Display App	SSE Authors	The name of the content. Custom content is overwritten by the channel definition.	Splunk Security Essentials
visualizations	Array of Visualization Objects	SSE Authors	The native Splunk Viz or screenshot visualizations you want to include in SSE.	
visualizations.basesearch	Base Search	SSE Authors	If defined, the value that initializes the a base search and post-process	

JSON field name	Descriptiive field	Relevance	Description	Examples
			in visualizations.search. Define this setting when you have multiple panels with input from the same dataset so the same search doesn't run multiple times.	
visualizations.dashboard	Dashboard Name	SSE Authors	The visualization that appears near the dashboard name when using automatic dashboarding. You can have 10â 30 dashboards on each panel.	Essential Network Security
visualizations.description	Panel Description	SSE Authors	Optional description that appears when using automatic dashboarding.	Provides a running count of identified DNS connections over time.
visualizations.header	Dashboard Header	SSE Authors	The visualization that appears near the header when using automatic dashboarding. Use 1â 5 panels for the header.	DNS Traffic
visualizations.hideInSearchBuilder	Hide in Search Builder	SSE Authors	If defined as "True," a visualization appears in automatic dashboarding but not the search builder.	Obj: True
visualizations.panel	Panel ID	SSE Authors	The panel in which a visualization made in the search builder renders. The panel contains three columns and rows.	row1cell1
visualizations.path	Image URL	SSE Authors	The path to the the visualization for visualizations.type.	
visualizations.recommended	Is Base Search	SSE Authors	Whether to mark panels in the automatic dashboarding feature as *recommended*.	Obj: True
visualizations.search	Search String	SSE Authors	The search string for the visualization.	
visualizations.title	Panel Title	SSE Authors	The title of the visualization.	DNS Traffic Over Time
visualizations.type	Image or Visualization	SSE Authors	Whether a visualization is an image or a Splunk Viz.	Image
visualization.vizParameters	Parameters For Native Viz	SSE Authors	The parameters referenced when you	

JSON field name	Descriptiive field	Relevance	Description	Examples
			initialize Splunk Viz using SplunkJS.	
visualizations.vizType	Visualization Type	SSE Authors	What type a Splunk Viz is. Options are "ChartElement," "SingleElement," "MapElement," and "TableElement."	TableElement
anomalies	UBA: Array of ShowcaseIDs for UBA Anomalies	UBA Authors	UBA specific: list of SSE IDs for any UBA anomalies the threat sees. Each should resolve to athreat. Often, the ID matches regex TT\d*.	
contributes_to_threats	UBA: Array of ShowcaseIDs for UBA Threats	UBA Authors	UBA specific: array that contains objects for each detection related to anomaly types.	
detections.data_source	UBA: Array of DSC IDs	UBA Authors	UBA specific: A proper JSON array with DSC IDs that resolve to the data_inventory.json, just like the data_source_categories field, except the detections.data_source field is a true JSON rather than a pipe-delimited string.	DS009EndPointIntel-ET01ProcessLaunch
detections.descriptions	UBA: Description for the Detection	UBA Authors	UBA specific: Description of the detection related to an anomaly type.	
detections.id	UBA: ID for the Detection	UBA Authors	Not currently in use.	
detections.internal_id	UBA: Internal ID for the Detection	UBA Authors	Not currently in use.	
detections.name	UBA: Detection Name	UBA Authors	UBA specific: the name of the detection related to an anomaly type.	
internal_id	UBA: Internal ID	UBA Authors	Not currently in use.	
is_custom	UBA: Is a Custom Threat	UBA Authors	UBA specific: not currently in use, but displays whether a threat is a rule-based custom threat rather than an ML-detected threat.	Yes
long_description	UBA: Long Description	UBA Authors	UBA specific: A description field not subject to character	

JSON field name	Descriptiive field	Relevance	Description	Examples
			limits, like other description fields. Appears if a user selects an anomaly or threat.	

Unless noted otherwise, multi-value fields populate with pipes separating each element. For example, content mapped to Security Monitoring and Advanced Detection appears as follows::

```
{
    "usecase": "Security Monitoring|Advanced Threat Detection"
}
```

To find valid values, see the following example search:

```
| sseanalytics | fieldsummary | table field *count values
```

ShowcaseInfo.json examples

Review the following examples to learn more:

- Partner content
- Simple content
- Native content
- Product-specific content
- Visualizations
- Dashboard panels

Partner content

SSE expects to download a JSON file. Each top-level key must be the ID of the showcase:

```
{
    "buttercuplabs_detect_bad_ponies": {
        "name": "Detect Bad Ponies",
        "inSplunk": "yes",
        "journey": "Stage_1",
        "usecase": "Security Monitoring",
        "highlight": "Yes",
        "id": "buttercuplabs_detect_bad_ponies",
        "channel": "ButtercupLabs",
        "alertvolume": "Low",
        "severity": "Very High",
        "category": "Account Compromise",
        "description": "This detection uses advanced analytics to determine which ponies are not good.",
        "domain": "",
        "killchain": null,
        "SPLEase": "None",
        "searchkeywords": "",
        "advancedtags": "",
        "printable_image": "",
        "icon":
```

```
"https://static.mylogocloud.com/shop/store/20180213730/assets/items/largeimages/SPK0153.jpg",
        "company_logo":
"https://image.slidesharecdn.com/splunklivesfhowtoalignyourdailysplunkactivitiesbreakoutsession-160317192319/95/how
-to-align-your-daily-splunk-activities-breakout-session-23-638.jpg?cb=1458242654",
        "company_logo_width": "444",
        "company_logo_height": "250",
        "company_name": "Buttercup Labs",
        "company_description": "Buttercup Labs is the premier distributor of Pony-related security
analytics. We have been protecting organizations from bad ponies for over ten years now.\n\\n\\n\nEnjoy our
freely available content for detecting bad ponies in your environment, and reach out to us for a demo or
trial license of our premium Pony Detection app!\n\\n\\n\nHave you successfully found bad ponies in your own
environment? Buttercup Labs is hiring! We are a wholly owns subsidiary of Buttercup Games, and are a proudly
equal opportunity employer. We welcome ponies of all heights, colors, hoof styles, sexual orientations, and
neuro-diversities.",
        "company_link": "http://buttercupgames.com/",
        "dashboard": "showcase_custom?showcaseId=buttercuplabs_detect_bad_ponies",
        "relevance": "Placeholder Text",
        "help": "",
        "howToImplement": "Placeholder Text",
        "knownFP": "Placeholder Text",
        "operationalize": "Placeholder Text",
        "search": "index=buttercup sourcetype=content",
        "data_source_categories": "DS003Authentication-ET02Failure|DS003Authentication-ET01Success",
        "mitre_technique": "T1098|T1003",
        "mitre_tactic": "TA0006"
    }
}
```

Simple content

Simple content requires less configuration than other types of content:

```
{
    "phantom_ec2_instance_isolation": {
        "alertvolume": "Very Low",
        "app": "Splunk_Phantom",
        "category": "Account Compromise|IAM Analytics|Account Sharing|SaaS|Insider Threat",
        "dashboard": "showcase_phantom?showcaseId=phantom_ec2_instance_isolation",
        "data_source_categories": "VendorSpecific-aws-cloudtrail",
        "description": "Isolate an EC2 instance by changing its security group in order to protect it from
malicious traffic. This playbook can be started alone or used from another playbook after doing
investigation and notification.",
        "displayapp": "Splunk Phantom",
        "domain": "Access",
        "hasSearch": "No",
        "help": "Simply deploy Phantom and work with your technical team to deploy this.",
        "highlight": "Yes",
        "howToImplement": "This playbook can be triggered off of several example searches available in the
Security Essentials app to take immediate action to quarantine an instance when suspicious behavior has been
detected.",
        "icon": "phantom_logo.png",
        "includeSSE": "Yes",
        "journey": "Stage_5",
        "name": "EC2 Instance Isolation",
        "operationalize": "Depending on the use case, this playbook can be modified using several of the
available Phantom apps to increase the scope of the actions taken in this playbook. For example using the
AWS WAF or AWS IAM app, additional actions can be added based on the type of alert triggered.",
        "printable_image":
"https://raw.githubusercontent.com/phantomcyber/playbooks/4.5/ec2_instance_isolation.png",
        "released": "3.0.0",
        "relevance": "Compromised AWS credentials can allow a malicious actor access to currently running
instances and configurations as well as the ability to start new instances and services. By detecting
suspicious behavior early this playbook allows for a security team to react quickly and further investigate
any suspicious behavior.",
        "searchKeywords": "",
        "usecase": "Advanced Threat Detection|Insider Threat|SOC Automation",
        "visualizations": [
            {
                "panel": "row1cell1",
                "path":
"https://raw.githubusercontent.com/phantomcyber/playbooks/4.5/ec2_instance_isolation.png",
                "title": "EC2 Instance Isolation",
                "type": "image"
            }
        ]
    }
}
```

Native content

When developing native content for SSE, additional fields are available:

```
{
    "new_cloud_provider": {
        "SPLEase": "Medium",
        "alertvolume": "High",
        "app": "Splunk_Security_Essentials",
        "category": "Data Exfiltration|Insider Threat|Shadow IT",
        "dashboard": "showcase_first_seen_demo?ml_toolkit.dataset=New Cloud Provider for User - Demo",
        "data_source_categories": "DS005WebProxyRequest-ET01RequestedWebAppAware",
        "description": "<p>Detect a user who is accessing a cloud storage provider they've never used
before.</p>",
        "displayapp": "Splunk Security Essentials",
        "domain": "Network",
        "examples": [
            {
                "label": "Demo Data",
                "name": "New Cloud Provider for User - Demo"
            },
            {
                "label": "Live Data",
                "name": "New Cloud Provider for User - Live"
            },
            {
                "label": "Accelerated Data",
                "name": "New Cloud Provider for User - Accelerated"
            }
        ],
        "hasSearch": "Yes",
        "help": "<p>This example leverages the Detect New Values search assistant. Our dataset is an
anonymized collection of Palo Alto Networks events. For this analysis, we are effectively grouping by
username and app name after filtering for the category, which will give us a row for each username+appname
combination. We check if the first time that has occurred was in the last day.</p>",
        "highlight": "Yes",
        "howToImplement": "<p>Implementation of this example (or any of the First Time Seen examples) is
generally very simple. <ul><li>Validate that you have the right data onboarded, and that the fields you want
to monitor are properly extracted.</li><li>Save the search.</li></ul></p><p>For most environments, these
searches can be run once a day, often overnight, without worrying too much about a slow search. If you wish
to run this search more frequently, or if this search is too slow for your environment, we recommend
leveraging a lookup cache. For more on this, see the lookup cache dropdown below and select the sample item.
A window will pop up telling you more about this feature.</p>",
        "icon": "Core_Use_Case.png",
        "includeSSE": "Yes",
        "journey": "Stage_2",
        "killchain": "Actions on Objectives",
        "knownFP": "<p class=\"disclaimer\">This is a strictly behavioral search, so we define \"false
positive\" slightly differently. Every time this fires, it will accurately reflect the first occurrence in
the time period you're searching over (or for the lookup cache feature, the first occurrence over whatever
time period you built the lookup). But while there are really no \"false positives\" in a traditional sense,
there is definitely lots of noise.</p><p>You should not review these alerts directly (except for access to
extremely sensitive system), but instead use them for context, or to aggregate risk (as mentioned under How
To Respond).</p> ",
        "mitre_tactic": "TA0010|TA0011",
        "mitre_technique": "|T1048|T1102",
        "name": "New Cloud Provider for User",
        "operationalize": "<p>When this search returns values, validate whether the usage of this cloud
provider is permitted by your policy, and investigate to see what data is being stored there. Common
allowable scenarios can be uploading into a box folder provided by a vendor for secure support file upload,
which might be allowable, versus the backup of data to a personal Google drive account. Ultimately this
search will generate many shades of gray, so it's prudent to understand supporting information such as the
amount of data transmitted before reaching out to the employee or their manager to determine next
steps.</p>",
        "phantomPlaybooks": [
            "phantom_malicious_insider_containment",
            "phantom_prompt_and_block_domain"
        ],
        "printable_image":
"/static/app/Splunk_Security_Essentials/images/printable_demo_images/new_cloud_provider.png",
        "released": "2.2.0",
        "relevance": "Data exfiltration techniques vary across the world, but certainly a very common
approach taken in 2018 is to upload data to a non-corporate file storage solution. Tracking new file storage
solutions end up in your environment is a key capability to track where data flows in your organization
along with the adoption of Shadow IT.",
        "searchKeywords": "",
        "usecase": "Insider Threat|Security Monitoring"
    }
}
```

Product-specific content

Splunk's products have unique configurations that vary by product.

Splunk Enterprise Security

```
{
    "search_name": "Change - Abnormally High Number of Endpoint Changes By User - Rule",
}
```

Splunk Enterprise Security Content Update

```
{
    "escu_cis": "CIS 13",
    "escu_data_source": "AWS CloudTrail logs",
    "escu_nist": "DE.DP|DE.AE",
    "escu_providing_technologies": "AWS",
    "search_name": "ESCU - Abnormally High AWS Instances Terminated by User - Rule",
    "story": "2e8948a5-5239-406b-b56b-6c50f1268af3"
}
```

The Enterprise Security Content Updates (ESCU) Story ID in the story field references the story object in the Showcaseinfo.json file. The story object is a low-level field hosted at ShowcaseInfo['escu_stories']. This is the story detail referenced in the previous example code:

```
{
    "2e8948a5-5239-406b-b56b-6c50f1268af3": {
        "detections": [
            "detection_abnormally_high_instance_termination",
            "detection_ec2_instance_created_by_previously_unseen_user",
            "detection_aws_activity_in_new_region",
            "detection_abnormally_high_instances_launched"
        ],
        "escu_category": [
            "Cloud Security"
        ],
        "investigations": [
            "AWS Investigate User Activities By ARN",
            "Get EC2 Instance Details by instanceId",
            "Get Notable History",
            "Get Notable Info",
            "Get User Information from Identity Table",
            "Investigate AWS activities via region name",
            "Get EC2 Launch Details"
        ],
        "modification_date": "2018-02-09",
        "name": "Suspicious AWS EC2 Activities",
        "narrative": "AWS CloudTrail is an AWS service that helps you enable governance, compliance, and
risk auditing within your AWS account. Actions taken by a user, role, or an AWS service are recorded as
events in CloudTrail. It is crucial for a company to monitor events and actions taken in the AWS Console,
AWS command-line interface, and AWS SDKs and APIs to ensure that your EC2 instances are not vulnerable to
attacks. This Analytic Story identifies suspicious activities in your AWS EC2 instances and helps you
respond and investigate those activities.",
        "references": [
            "https://d0.awsstatic.com/whitepapers/aws-security-best-practices.pdf"
        ],
        "responses": [],
        "story_id": "2e8948a5-5239-406b-b56b-6c50f1268af3",
        "support": [
            "Previously Seen EC2 Launches By User",
            "Previously Seen AWS Regions"
        ]
    }
}
```

Splunk User Behavior Analytics anomalies

```
{
    "anomalies": [
        "AT11",
        "AT06"
    ],
    "internal_id": "Potential_Flight_Risk_Exfiltration",
    "long_description": "This threat brings to light users who are possible flight risks. However, in
addition to being a flight risk, this user has also performed actions that signal that it may be possible he
will exfiltrate data. Actions that lead to exfiltration of data in this instance are events such as:
suspicious data movement, upload of information to a cloud file system, or a DLP external alarm. ",
    "is_custom": "Yes",
}
```

Splunk User Behavior Analytics threats

```
{
    "anomalies": [
        "AT11",
        "AT06"
    ],
    "internal_id": "Potential_Flight_Risk_Exfiltration",
    "long_description": "This threat brings to light users who are possible flight risks. However, in
addition to being a flight risk, this user has also performed actions that signal that it may be possible he
will exfiltrate data. Actions that lead to exfiltration of data in this instance are events such as:
suspicious data movement, upload of information to a cloud file system, or a DLP external alarm. ",
    "is_custom": "Yes",
}
```

Visualizations

SSE's native search builders support custom visualizations.

Screenshots

```
{
    "visualizations": [
        {
            "panel": "row1cell1",
            "path":
"https://raw.githubusercontent.com/phantomcyber/playbooks/4.5/ec2_instance_isolation.png",
            "title": "EC2 Instance Isolation",
            "type": "image"
        }
    ]
}
```

Splunk dashboard panels

Because searches differ, define panel configurations like the one in this example in the search builder JSON file:

```
{
    "visualizations": [
        {
            "dashboard": "General Windows and Linux Posture",
            "header": "AV Data",
            "panel": "row1cell1",
            "search": "| `Load_Sample_Log_Data(Symantec Endpoint Protection Operations)`  |stats
max(eval(if(like(Event_Description, \"%LiveUpdate session ran successfully%\") , _time, null))) as
LatestUpdate max(_time) as LatestMessage max(eval(if(tag=\"error\", _time, null))) as LatestError by
Host_Name   | eval Up_To_Date = if( LatestUpdate < relative_time(LatestMessage, \"-3d\") OR LatestError >
LatestUpdate , \"No\", \"Yes\") | lookup gdpr_system_category host as Host_Name| search category=* | stats
count by Up_To_Date",
            "title": "Percentage of In-Scope Hosts with Up To Date AV",
            "type": "viz",
            "vizParameters": {
                "charting.chart": "pie",
                "link.exportResults.visible": "false",
                "link.inspectSearch.visible": "false",
                "link.openPivot.visible": "false",
                "link.openSearch.visible": "false",
                "link.visible": "false",
                "refresh.link.visible": "false",
                "resizable": true
            },
            "vizType": "ChartElement"
        },
        {
            "dashboard": "General Windows and Linux Posture",
            "header": "AV Data",
            "panel": "row1cell2",
            "search": "| `Load_Sample_Log_Data(Symantec Endpoint Protection Operations)`  |stats
max(eval(if(like(Event_Description, \"%LiveUpdate session ran successfully%\") , _time, null))) as
LatestUpdate max(_time) as LatestMessage max(eval(if(tag=\"error\", _time, null))) as LatestError by
Host_Name   | eval Up_To_Date = if( LatestUpdate < relative_time(LatestMessage, \"-3d\") OR LatestError >
LatestUpdate , \"No\", \"Yes\") | search Up_To_Date = Yes | lookup gdpr_system_category host as Host_Name|
search category=*| convert ctime(LatestUpdate) ctime(LatestMessage) ctime(LatestError) ",
            "title": "Hosts with Up To Date AV",
            "type": "viz",
            "vizParameters": {
                "link.exportResults.visible": "false",
                "link.inspectSearch.visible": "false",
                "link.openPivot.visible": "false",
                "link.openSearch.visible": "false",
                "refresh.link.visible": "false",
                "resizable": true
            },
            "vizType": "TableElement"
        }
    ]
}
```

Search builder

The search builder files are those used to display a file. SSE contains five search builder files:

Dashboard	Dashboard name	JSON file	Notes
showcase_simple_search	Simple Search	showcase_simple_search.json	For the majority of native SSE content, this search builder runs a search string without any additional SPL.
showcase_first_seen_demo	First Time Seen	showcase_first_seen_demo.json	The first time this search builder takes a base dataset and two fields used for splittingâ if that happens in the last 24 hoursâ the file applies logic for alerting. This search builder has high-scale versions, allowing users to cache the result to a lookup. A peer group option is available, also, allowing the search to find values that are new not only for the entity in question, but also for the peer group (or department, geo, etc.).
showcase_standard_deviation	Time Series Spike	showcase_standard_deviation.json	This search builder takes a base dataset that is already aggregated each day, a field representing a numerical

Dashboard	Dashboard name	JSON file	Notes
			measure, and a feature for the entity the system intends to analyze. This search builder then performs a standard-deviation analysis to find unusually high values.
showcase_phantom	Splunk Phantom	ShowcaseInfo	This search builder is used for Phantom content.
showcase_custom	Custom Content	ShowcaseInfo	This search builder is used for custom content and partner content.

Schema

JSON field name	Descriptive field	Search builder	Description	Examples
label	Name	All	Matches examples.name from ShowcaseInfo.json and must match the ID.	Emails With Lookalike Domains - Demo
value	Search	All	The search string. Line breaks in the SPL are implemented via \n and the number of lines must match the length of the description array.	index=abc \n
description	Line-by-Line SPL Documentation	All	An array that contains the line-by-line SPL docs. The number of elements in the array must match the number of \n values plus one.	"Base Data", "Count them", "Filter"
Prereqs	Array of Pre-Requisites	All	An array that contains all of the prerequisite objects.	
prereqs.field	Pre-Req Field	All	The field that determines whether the pre-req is satisfied.	count
prereqs.greaterorequalto	Pre-Req Min Value	All	The value of the field in prereqs.field must be greater than or equal to.	1
prereqs.name	Pre-Req Name	All	The label provided to the user for this pre-req.	"Must have data"
prereqs.resolution	Pre-Req Resolution	All	Description of how the user can satisfy a pre-req.	Install an add-on.
prereqs.test	SPL for Pre-Req	All	The search that runs for pre-reqs.	
actions_createNotable	Create Risk	All	Whether a notable event is created. Triggers ES workflows.	1
actions_createRisk	Create Risk	All	Whether to create a risk object for a search.	1
actions_riskObject	Risk Object	All	The risk_object, as aligned with the Splunk Enterprise Security risk framework.	Computer_Name
actions_riskObjectScore	Risk Score	All	The risk_score, as aligned with the Splunk Enterprise Security risk framework.	60
actions_riskObjectType	Risk Object Type	All	The risk_object_type, aligning with the Splunk Enterprise Security risk framework.	system
actions_UBASeverity	UBA Severity	All	An indication that SSE should send an event to Splunk User Behavior Analytics with a severity.	5
visualizations	Array of Visualizations Objects	All	The native Splunk Viz or screenshot visualizations you want to include in SSE.	

JSON field name	Descriptive field	Search builder	Description	Examples
visualizations.basesearch	Base Search	All	If defined, the value that initializes the a base search and post-process in visualizations.search. Define this setting when you have multiple panels with input from the same dataset so the same search doesn't run multiple times.	
visualizations.dashboard	Dashboard Name	SSE Authors	The visualization that appears near the dashboard name when using automatic dashboarding. You can have 10â 30 dashboards on each panel.	Essential Network Security
visualizations.description	Panel Description	SSE Authors	Optional description that appears when using automatic dashboarding.	Provides a running count of identified DNS connections over time.
visualizations.header	Dashboard Header	SSE Authors	The visualization that appears near the header when using automatic dashboarding. Use 1â 5 panels for the header.	DNS Traffic
visualizations.hideInSearchBuilder	Hide in Search Builder	SSE Authors	If defined as "True," a visualization appears in automatic dashboarding but not the search builder.	Obj: True
visualizations.panel	Panel ID	SSE Authors	The panel in which a visualization made in the search builder renders. The panel contains three columns and rows.	row1cell1
visualizations.path	Image URL	SSE Authors	The path to the the visualization for visualizations.type.	
visualizations.recommended	Is Base Search	SSE Authors	Whether to mark panels in the automatic dashboarding feature as *recommended*.	Obj: True
visualizations.search	Search String	SSE Authors	The search string for the visualization.	
visualizations.title	Panel Title	SSE Authors	The title of the visualization.	DNS Traffic Over Time
visualizations.type	Image or Visualization	SSE Authors	Whether a visualization is an image or a Splunk Viz.	Image
visualization.vizParameters	Parameters For Native Viz	SSE Authors	The parameters referenced when you initialize Splunk Viz using SplunkJS.	
visualizations.vizType	Visualization Type	SSE Authors	What type a Splunk Viz is. Options are "ChartElement," "SingleElement," "MapElement," and "TableElement."	TableElement
OutlierPeerGroup	Peer Group	First Time Seen	The option to perform peer-group analysis. Disabled by default but can be enabled.	
outlierValueTracked1	Value #1	First Time Seen	First Time Seen splits by two values. Most often, the first value is the subject (e.g., user, dest, src, etc.).	user
outlierValueTracked2	Value #2	First Time Seen	First Time Seen splits by two values. Most often, the second value is the unusual attribute (e.g., geo, API, etc.).	eventName

JSON field name	Descriptive field	Search builder	Description	Examples
cardinalityTest	Cardinality Test	Time Series Spike	The SPL that test whether a split-by results in an excessive number of results.	
outlierSearchType	Search Type	Time Series Spike	Reserved for future use. Must be "Avg" currently.	Avg
outlierVariable	Numeric Measure	Time Series Spike	The value containing the count the system measures.	count
outlierVariableSubject	Subject	Time Series Spike	The subject or entity the system is monitoring. Options include user, src, dest, or any other Splunk field.	user
scaleFactor	# of StDevs	Time Series Spike	The number of standard deviations required for the system to consider a detection "abnormal" (z-score). Usually, use 3 for anomaly detection and 10 for high confidence.	6
windowSize	windowSize	Time Series Spike	Not currently in use. Value must be "0."	0

Search builder examples

Simple search

```
{
    "DynDNS - Live": {
        "actions_UBASeverity": 7,
        "actions_createRisk": 1,
        "actions_riskObject": "user",
        "actions_riskObjectScore": 30,
        "actions_riskObjectType": "user",
        "description": [
            "First we bring in our dataset of proxy logs.",
            "Because we are looking for dynamic dns providers, we're going to need to separate out
subdomains from the registered domain. URL Toolbox is just the tool for this job!",
            "Next we can use our lookup of ddns domains (see How to Implement). This will add a field called
inlist with the value \"true\" for any matches.",
            "And finally we can look for those records that are matches.",
            "With our dataset complete, we just need to arrange the fields to be useful."
        ],
        "label": "DynDNS - Live",
        "prereqs": [{
                "field": "count",
                "greaterorequalto": 1,
                "name": "Must have Proxy data",
                "resolution": "Proxy data can come in many forms, including from Palo Alto Networks and
other NGFWs, dedicated proxies like BlueCoat, or network monitoring tools like Splunk Stream or bro.",
                "test": "| metasearch earliest=-2h latest=now index=* sourcetype=pan:threat OR
(sourcetype=opsec URL Filtering) OR sourcetype=bluecoat:proxysg* OR sourcetype=websense* | head 100 | stats
count "
            },
            {
                "field": "count",
                "greaterorequalto": 1,
                "name": "Must have URL Toolbox Installed",
                "resolution": "The URL Toolbox app, written by Cedric Le Roux, provides effective URL
Parsing. Download <a href=\"https://splunkbase.splunk.com/app/2734/\">here</a>.",
                "test": "| rest /services/apps/local | search disabled=0 label=\"URL Toolbox\" | stats
count"
            }
        ],
        "value": "index=* sourcetype=pan:threat OR (tag=web tag=proxy) earliest=-20m@m earliest=-5m@m \n|
eval list=\"mozilla\" | `ut_parse_extended(url,list)`\n| lookup dynamic_dns_lookup domain as ut_domain
OUTPUT inlist\n| search inlist=true \n| table _time ut_domain inlist bytes* uri",
        "visualizations": [{
                "panel": "row1cell1",
                "header": "DNS Traffic",
                "dashboard": "Essential Network Security",
                "description": "Provides a running count of identified DNS connections over time",
                "search": "| `Load_Sample_Log_Data(Sample Firewall Data)` | search dest_port=53 OR app=dns
| timechart count",
                "title": "DNS Traffic Over Time",
                "type": "viz",
                "vizParameters": {
                    "charting.chart": "column"
                },
                "vizType": "ChartElement"
            },
            {
                "panel": "row1cell2",
                "header": "DNS Traffic",
                "dashboard": "Essential Network Security",
                "recommended": false,
                "description": "Shows the top destinations for DNS traffic. These should be your DNS
servers, or a standard upstream DNS server.",
                "basesearch": "index=* sourcetype=pan:threat OR (tag=web tag=proxy) (dest_port=53 OR
app=dns)",
                "search": "| `Load_Sample_Log_Data(Sample Firewall Data)` | search dest_port=53 OR app=dns
| stats count by dest_ip | sort - count",
                "title": "Likely Resolvers",
                "type": "viz",
                "vizParameters": {},
                "vizType": "TableElement"
            }
        ]
    }
}
```

First time seen

```
{
    "AWS New API Call Per Peer Group - Live": {
        "actions_UBASeverity": 7,
        "actions_createRisk": 1,
        "actions_riskObject": "user",
        "actions_riskObjectScore": 30,
        "actions_riskObjectType": "user",
        "description": [
            "First we bring in our basic dataset. In this case, AWS CloudTrail logs, filtered for individual
APIs that we want to pay close attention to."
        ],
        "label": "AWS New API Call Per Peer Group - Live",
        "outlierPeerGroup": "peer_group_for_git_use_case.csv",
        "outlierValueTracked1": "user",
        "outlierValueTracked2": "eventName",
        "prereqs": [{
            "field": "count",
            "greaterorequalto": 1,
            "name": "Must have AWS CloudTrail data",
            "resolution": "In order to run this search, you must have AWS CloudTrail data onboard. Visit the
<a href=\"/app/Splunk_Security_Essentials/data_source?technology=AWS%20CloudTrail\">data onboarding guide
for AWS CloudTrail in this app</a>, or browse to <a
href=\"https://splunkbase.splunk.com/app/1876/\">apps.splunk.com</a> for more information.",
            "test": "| tstats count where earliest=-2h latest=now index=* sourcetype=aws:cloudtrail"
        }],
        "value": "index=* sourcetype=aws:cloudtrail eventType=* NOT errorMessage=* NOT eventName=Describe*
NOT eventName=Get* NOT eventName=List*"
    }
}
```

Standard deviation

```
{
    "GCP APIs Called More Often Than Usual Per Account - Live": {
        "actions_UBASeverity": 2,
        "actions_createRisk": 1,
        "actions_riskObject": "data.protoPayload.authenticationInfo.principalEmail",
        "actions_riskObjectScore": 10,
        "actions_riskObjectType": "user",
        "cardinalityTest": "index=* sourcetype=google:gcp:pubsub:message \n| bucket _time span=1d | stats
dc(data.protoPayload.authenticationInfo.principalEmail) as count by _time",
        "description": [
            "First we bring in our basic GCP Audit logs, filtering out list activity.",
            "Bucket (aliased to bin) allows us to group events based on _time, effectively flattening the
actual _time value to the same day.",
            "Next we use stats to summarize the number of events per user per day."
        ],
        "label": "GCP APIs Called More Often Than Usual Per Account - Live",
        "outlierSearchType": "Avg",
        "outlierVariable": "count",
        "outlierVariableSubject": "data.protoPayload.authenticationInfo.principalEmail",
        "prereqs": [{
            "field": "count",
            "greaterorequalto": 1,
            "name": "Must have GCP Audit data",
            "resolution": "In order to run this search, you must have GCP Audit data onboard. Browse to <a
href=\"https://splunkbase.splunk.com/app/3088/\">apps.splunk.com</a> for more information.",
            "test": "| tstats count where earliest=-2h latest=now index=\"*\"
sourcetype=google:gcp:pubsub:message"
        }],
        "scaleFactor": 2,
        "value": "index=* sourcetype=google:gcp:pubsub:message NOT data.protoPayload.methodName=v1*.list*
 NOT data.protoPayload.methodName=storage*.list*  \n| bucket _time span=1d \n| stats count by
data.protoPayload.authenticationInfo.principalEmail _time",
        "windowSize": 0
    }
}
```

Enrich custom content using the ShowcaseInfo.json file

The ShowcaseInfo.json is the most important file in Splunk Security Essentials (SSE).

Architecture

Almost every SSE dashboard calls the REST endpoint /services/SSEShowcaseInfo or searches Splunk for | sseanalytics, which also calls the REST endpoint but on the back-end. For additional information, see the Splunk Security Essentials Schema topic.

Endpoint

web.conf

```
[expose:SSEShowcaseInfo]
methods = GET,POST
pattern = SSEShowcaseInfo
```

restmap.conf:

```
[script:sseshowcaseinfo]
match                     = /SSEShowcaseInfo
script                    = generateShowcaseInfo.py
scripttype                = persist
handler                   = generateShowcaseInfo.ShowcaseInfo
requireAuthentication     = true
output_modes              = json
passPayload               = true
passHttpHeaders           = true
passHttpCookies           = true
```

Call process

When you call generateShowcaseInfo.py, the call gathers configuration parameters from the URL and then follows this process to access the ShowcaseInfo.json file, add custom content, and enrich the content with files and lookups.

1. Initialize the service object used for API calls, pulling in the URL for splunkd in case the kvstore API fails and must perform a GET from the API directly.
2. Check the contents of essentials_updates.conf to determine which apps are enabled and which are disabled.
3. Check for MITRE ATT&CK updates. If more than one day has passed since the last check, the MITRE updates if an update is available by stashing the update in the kvstore, then the pullJSON, which allows kvstore to store large JSON files, renders the updated information.
4. Pull simple information with no modifications: bookmark, local_search_mappings, and data_source_check.
5. Pull information with slight modification: custom_content is cleaned as described in the Partner Integration topic. data_inventory_products are inserted into an array of scores per DSC so they can combine at the product level, where there might be multiple DSCs. The DSC-to-productId match is created.
6. Pull in core files: ShowcaseInfo.json and each search-builder file. If naming conflicts arise, SSE silently overrides.
7. Pull in supporting files: data_inventory.json as well as MITRE ATT&CK and MITRE Pre-ATT&CK (not grabbed directly because the pullJSON call always delivers some MITRE content)
8. Pull in custom content from the ShowcaseInfo.json as if from the JSON file itself.

9. Parse MITRE content into relevant objects, which relate to each other through relationship objects and relationship_type uses. All these objects have an ID, which maps to the source_ref and target_ref fields: x-mitre-tactic as tactics, attack-pattern as techniques, and intrusion-set as groups.
10. Clear out invalid characters.
11. Enrich and process.
12. Enrich with searches, looking for matches between information in the search builder JSON file and the ShowcaseInfo.json file. If a match appears, it's added as ShowcaseInfo['summaries']['my_summary_id']['examples'][NUM]['showcase'] = my_search_builder_obj.
13. Create local search mappings, checking each piece of content against the list of savedsearch.conf mappings in local_search_mappings and then setting search_title to match. A known limitation is that only one search can occur for each piece of SSE content. If you have multiple searches that are similar to the object you need to create custom content in SSE to complete mapping.
14. Enrich data availability, looking up each piece of content rendered in the Data Availability matrix to add the fields data_available and data_available_numeric.
15. Enrich MITRE variables already parsed with this information: the tactic and technique name, a technique description, the MITRE matrix it came from (i.e., ATT&CK or Pre-ATT&CK), the threat groups, and search keywords.
16. Clean up missing fields. This process can involve creating empty strings and changing strings to "None."
17. Exclude disabled channels by iterating through all the content in the ShowcaseInfo.json file while checking for channels in the exclusions list (channel_exclusion[â ¦]). If some content should be excluded, it's pulled. You can override this part of the process by including ignoreChannelExclusion=true in your request.
18. Export content to sse_content_exported to map savedsearch.conf names to the metadata in SSE. When you run ShowcaseInfo, it checks whether the content is correct in the configuration and fixes incorrect content. You can see the fields in that export in the fields = [â ¦] line. The fields are mirrored in collections.conf and transforms.conf.
19. Optionally minify ShowcaseInfo by adding fields=mini to the URL. Doing this strips all fields not specified in mini_fields, drastically reducing the amount of data transferred.
20. Return data in JSON format.

Error handling

To see errors from ShowcaseInfo, look for elements added into the SSE output: debug and throwError. The debug object contains a variety of debugging informationâ in particular, timing checks and exceptions. throwError denotes a critical error.

From the Security Contents page, the ShowcaseInfo response appears in the window object so you can open the JavaScript console and run this command:

```
console.log("throwError Status", ShowcaseInfo.throwError);
console.log("debug Status", ShowcaseInfo.debug);
```

This is an example of the debug status:

```
[
    "Stage -5 Time Check:9.05990600586e-06",
    "Stage -4 Time Check:8.20159912109e-05",
    "Stage -3 Time Check:9.20295715332e-05",
    "Stage -4 Time Check:0.000141143798828",
    {
        "localecheck": ""
    },
    "Not going cached! Prepare for a long ride.",
    "Stage -1 Time Check:0.000144004821777",
    {
        "channel_exclusion": {
            "mitrepreattack": false,
            "custom": false,
            "Splunk_App_for_Enterprise_Security": false,
            "mitreattack": false,
            "Enterprise_Security_Content_Update": false,
            "Splunk_Security_Essentials": false,
            "Splunk_User_Behavior_Analytics": false
        },
        "override": false,
        "msg": "Final Channel Exclusion"
    },
    "Stage 0 Time Check:0.106211185455",
    "Stage 1 Time Check:1.20219802856",
    "Stage 2 Time Check:1.27426409721",
    {
        "store": "bookmark",
        "message": "I got a kvstore request"
    },
    "Stage 3 Time Check:1.29301118851",
    {
        "store": "local_search_mappings",
        "message": "I got a kvstore request"
    },
    "... abridged ...",
    "Stage 25 Time Check:1.62445807457"
]
```

Access through JavaScript

From a Splunk dashboard in the browser, run this command to get the resulting object:

```
require(['json!' + $C['SPLUNKD_PATH'] + '/services/SSEShowcaseInfo?bust=' + Math.random()],
function(showcase){
    window.debug_showcase = showcase;
    console.log("Got Showcase", showcase);
})
```

Access through Splunk search

To see the output of all summaries, use the | sseanalytics command. That command breaks out key fields separately and converts inline-multivalue pipe-separated fields into native Splunk multi-value fields. You can also request full JSON output, identical to what appears when performing a REST call:

```
| sseanalytics include_json=true | search channel=ButtercupLabs | table id summaries *
```

Author simple and full-feature content

Author simple and full-feature content on Splunk Security Essentials

The process of adding content to Splunk Security Essentials (SSE) differs, depending on whether you're an end user, partner, or SSE author.

End users can follow a workflow to add content using the GUI itself. For details on that workflow, see Customize Splunk Security Essentials.

Partners can follow the partner-integration process. For details on that workflow, see Partner Integration topic.

SSE authors can perform the following steps to add custom content, depending on that custom content being simple or full-feature.

Author simple SSE content

Simple SSE content is custom content that doesn't completely utilize a search builder or have demo data. Some examples of simple content are content for Splunk Enterprise Security, Splunk Enterprise Security Content Update, Splunk User Behavior Analytics, and Splunk Phantom.

Deploying simple content is similar to the process for partner integration, but there are some differences. For more information on the partner integration process, see Partner Integration. To assess potential deployment differences, see Schema. Schema.

Add full-feature content

Complete the following two steps to author full-feature SSE custom content.

Configure the ShowcaseInfo.json file

The ShowcaseInfo.json file contains high-level information for SSE content and is the primary interface. Two configurations govern its relationship with the search builder:

- The dashboard tells the interface what dashboard to send users to when they select a link. The dashboard includes the search builder in its related URL.
- The examples object is a list of search-builder objects that exist for individual pieces of content. Most often, you at least have discrete objects for demo data, live data, and accelerated data, but you might need other objects, as well.

The search builder JSON files, such as showcase_simple_search.json, list searches, line-by-line SPL documentation, and other helpful information.

Define names and IDs

After configuring the JSON file, you must define the following four names and IDs:

- The ID in the ShowcaseInfo.json file is a summary object, which defines the keys for each showcase.

- The name in ShowcaseInfo.json that displays to the user. You can find the name using the ShowcaseInfo.json['summaries']['my_showcase_id']['name'] lookup.
- The search label is the name displayed on the search page.
- The search name is the internal ID for a search.

In the showcase examples object, the search name (ShowcaseInfo.json['summaries']['my_showcase_id']['examples'][0]['name'] appears. That name must exactly match the object name in the search builder JSON file and the search name (showcase_*.json['my_search_name']['label']. If those values don't match, no JavaScript executes when a user navigates to the dashboard, and no errors appear to alert you to the problem.

Example

The following example shows a configured ShowcaseInfo.json file and a search with properly defined names and IDs.

Configured ShowcaseInfo.json file

```
{
   "summaries": {
       "basic_brute_force": {
            "name": "Basic Brute Force Detection",
            "dashboard": "showcase_simple_search?ml_toolkit.dataset=Basic Brute Force - Demo",
            "examples": [
                {
                    "label": "Demo Data",
                    "name": "Basic Brute Force - Demo"
                },
                {
                    "label": "Live Data",
                    "name": "Basic Brute Force - Live"
                },
                {
                    "label": "Accelerated Data",
                    "name": "Basic Brute Force - Accelerated"
                }
            ],
       }
   }
}
```

Search with defined names and IDs

```
{
    "Basic Brute Force - Demo": {
        "label": "Basic Brute Force - Demo",
        "value": "... demo search ..."
    },
    "Basic Brute Force - Live": {
        "label": "Basic Brute Force - Live",
        "value": "... live search ..."
    },
    "Basic Brute Force - Accelerated": {
        "label": "Basic Brute Force - Accelerated",
        "value": "... accelerated search ..."
    },
}
```

Splunk Security Essentials file directory

Splunk Security Essentials file directory

Splunk Security Essentials (SSE) contains many key files that are shared across the app. Read the entries in the following table to understand what these files do at a high level. Review the code in the file itself for specifics.

File	Description
runPageScript.js	Defines the logic for what scripts launch JavaScript
generateShowcaseInfo.py and ShowcaseInfo.json	Contains the main configurations for Splunk Security Essentials
buildTile.js	Renders a tile for your content
system_config.js	Generates the configuration menu in Splunk Security Essentials
common_data_objects.js	Centralizes information that is standard across the app, for example bookmark status names and versus IDs
export_panel.js	Contains the main export modal, the CSV export logic, and the print-to-PDF logic
buildLilyXLSX.js	Controls XLSX export
manageSnapshots.js	Contains the dialog that handles all snapshots
processSummaryUI.js	Contains the actual display detail for rendering the print-to-PDF visuals. The code contains the same logic used in the search builders themselves to allow for displaying all accordions by default and removing links
pullJSON.py	A REST handler front end for pulling JSON files that allows you to swap content from the kvstore for the raw files, such as MITRE, or enrich content, such as custom_content into data_inventory.json
pullCSV.py	Allows you to send a GET request for a lookup in a `require()` statement
modal.js	The core code for generating modals in SSE
unattachedModal.js	Generates modals in SSE and copied from SA-devforall but only used on data_inventory.js
AlertModal.js	Used for the save search dialog
dashboard.js	This runs for every dashboard in the app. Many miscellaneous logic functions are contained in this file, such as collectDiag().
sendTelemetry.js	The wrapper for swa.js, which handles all telemetry for SSE
home.js	Handles the home page
intro_content.json	Stores all the logic for the guides
contents.js	This contains the original core code of SSE. This JavaScript file contains the logic for the Security Contents page.
data_inventory.js	This is the core file for the data_inventory dashboard. The file generates the display from data_inventory.json.
DrawDataInventoryProducts.js	Contains the UI elements for product configuration
data_inventory_introspection.js	Contains the introspection logic

File	Description
data_inventory.json	This contains the raw data inventory configuration. When grabbed through pullJSON, the script augments.
bookmarked_content.js	Contains the core logic for the Bookmarked Content dashboard
MapExistingSearchContent.js	Contains the logic for the correlation search introspection
showcase_simple_search.js, showcase_first_seen_demo.js, showcase_standard_deviation.js, showcase_phantom.js, and showcase_custom.js	The dedicated files for each of the standard search builders, each providing capabilities for specific types of searches
ProcessSummaryUI.js	Generates most of the display, allowing for equivalent displays across different apps
es_use_case.js	Renders content from Splunk Enterprise Security (ES), Splunk Enterprise Security Content Update (ESCU), and Splunk User Behavior Analytics (UBA), relying on these files: • es_use_case.xml • escu_use_case.xml • uba_use_case.xml.
data_source.js	Based on the SimpleXML Examples app, with some enhancements
securityjourney.js	Renders the Security Journey and contains custom JavaScript and CSS
highlight.pack.js	Allows syntax highlighting, particularly around custom and partner content, for line-by-line SPL
lunr.js	Displays the search engine used on contents.js and MapExistingSearchContent.js
showdown.js	Performs Markdown conversion for descriptive fields
FileSaver.js	Allows you to save a generated file with a particular filename
jszip	Allows you to generate ZIP files in JavaScript

See a full list of Custom search commands for SSE in the *Use Splunk Security Essentials* manual.