Splunk® Enterprise Security REST API Reference 6.6.2

Generated: 10/21/2021 11:47 am

Table of Contents

Overview **1**
The Splunk Enterprise Security API 1

Threat Intelligence endpoints **2**
Threat Intelligence API reference 2

Notable Event endpoints **12**
Notable Event API reference 12

Analytic Story endpoints **14**
Analytic Story API reference 14

Overview

The Splunk Enterprise Security API

Splunk Enterprise Security offers a set of REST API endpoints that you can use to interact with the Splunk Enterprise Security frameworks programmatically or from Splunk search. These API endpoints are for Splunk Enterprise Security admins and for developers who are building integration applications for use with Splunk Enterprise Security.

The Splunk Enterprise Security REST API provides methods for accessing selected features in the Enterprise Security framework. The API follows the principles of Representational State Transfer (REST).

There are REST API access and usage differences between Splunk Cloud Platform and Splunk Enterprise. If you are using Splunk Cloud Platform, see Using the REST API with Splunk Cloud Platform in *REST API Tutorials*.

Navigate to specific endpoints and review available REST operations.

URI	Summary	GET	PUT	POST	DEL
Threat Intelligence endpoints					
/data/threat_intel/upload	Upload a threat intelligence file in STIX, IOC, or CSV format.				
/services/data/threat_intel/item/{threat_intel_collection}	Create or list rows in a threat intelligence collection.				
/services/data/threat_intel/item/{threat_intel_collection}/{item_key}	List, update, or delete a row in a threat intelligence collection.				
Notable Event endpoints					
/services/notable_update	Modify notable events.				
Analytic Story endpoints					
/services/analyticstories/configs/{stanza_type}	Acts as a proxy to configs/conf-analyticstories, with validation.				
/services/analyticstories/configs/{stanza_type}/{name}	Acts as a proxy to configs/conf-analyticstories/{stanza_name}.				
/services/analyticstories/configs/{stanza_type}/{name}/acl	Returns ACL information.				
/services/analyticstories/configs/{stanza_type}/{name}/move	Moves stanzas to other apps.				
/services/analyticstories/configs/_reload	Reloads data for the endpoint.				
/services/analyticstories/schemas/{version}	Reloads data for the endpoint.				
/services/analyticstories/batch	Takes a JSON array conforming to the analytic story JSON schema and saves it in proper format into analyticstories.conf.				

Threat Intelligence endpoints

Threat Intelligence API reference

Access the Threat Intelligence framework in Splunk Enterprise Security.

The Threat Intelligence framework is a mechanism for consuming and managing threat feeds, detecting threats, and alerting. For more information about working with the framework, see Threat Intelligence framework in Splunk ES.

Usage details

Authentication and Authorization

Username and password authentication is required for access to endpoints and REST operations. You must have the `edit_threat_intel_collections` capability to use the threat intelligence endpoints.

Alternatively, you can use token authentication. See Set up authentication with tokens in the Splunk Enterprise *Securing the Splunk Platform* manual.

Username and password authentication is used in the examples that follow.

Splunk Cloud Platform URL for REST API access

Splunk Cloud Platform has a different host and management port syntax than Splunk Enterprise. Depending on your deployment type, use one of the following options to access REST API resources.

Splunk Cloud Platform deployments

```
https://<deployment-name>.cloud.splunk.com:8089
```

To get the required credentials, submit a support case on the Support Portal. After installing the credentials, use the following URL.

```
https://input-<deployment-name>.cloud.splunk.com:8089
```

See Using the REST API in Splunk Cloud Platform in the the *Splunk REST API Tutorials* for more information.

/services/data/threat_intel/upload

Upload a threat intelligence file in STIX, IOC, or CSV format.

The REST API endpoint only uploads the threat intelligence file. To configure threat intelligence, you must define the `[threatlist]` stanza in the `inputs.conf` configuration file. For more information on configuring threat intelligence, see Configure Global Threat List Settings.

Syntax

```
https://<host>:<mPort>/services/data/threat_intel/upload
```

Usage details
For details of how Splunk Enterprise Security processes threat intelligence files, see Intelligence framework in Splunk ES.

POST

Upload a file.

Request parameters

Field	Type	Default	Description
content required	String		The threat Intelligence file content encoded in base64 format.
filename required	String		The threat intelligence file name, with extension. Format: `__threat_<name>.dmy`
overwrite	Boolean	true	If set to true and a file with this name already exists, the API overwrites the file and reports success. If set to false and a file with this name already exists, the API returns an error.
max_size	bytes	524288000 or 500 MB	The maximum size of the file that is uploaded.

Data payload
A JSON-encoded string with the arguments.

Response

The endpoint returns a success or failure message, with HTTP status codes.

Example request

```
curl -ku admin:changeme https://localhost:8089/services/data/threat_intel/upload -d '{"filename":
"__threat_example.dmy", "overwrite": true, "content": "dGhpcyBpcyBhIHRlc3Q="}' -H "Content-Type:
application/json" -X POST
```

Example response

```
{"message": "File uploaded successfully.", "success": true, "internal_status": 0}
```

/services/data/threat_intel/item/{threat_intel_collection}

Perform CRUD operations on an existing threat intelligence collection.

Syntax

```
https://<host>:<mPort>/services/data/threat_intel/item/{threat_intel_collection}
```

Usage details
The collection name must be one of the following:

- ip_intel
- file_intel
- user_intel

- http_intel
- email_intel
- service_intel
- process_intel
- registry_intel
- certificate_intel

Some methods require the `_key` field. To find the key for a row in a collection, run a search using the `inputlookup` command and the relevant `threat_intel_collection` and use eval to display the `_key` field for each row. For example:

```
| inputlookup ip_intel | eval item_key=_key
```

GET

List one or more rows from a collection.

Request parameters

Field	Type	Default	Description
item required	JSON encoded string		The field/value pairs to match. For example, `{'ip':'41.41.41.41'}` returns rows that have `ip=41.41.41.41` and `{'file_name':'threat.trt', 'file_extension':'trt'}` returns all the rows that match `file_name=threat.trt` and `file_extension=trt`.

Data payload

The item field must contain at least one of the important fields for the threat collection to use as search criteria, in a JSON encoded string.

Collection name	Important fields
user_intel	`user`
file_intel	`file_name`, `file_hash`
ip_intel	`ip`, `domain`
http_intel	`http_user_agent`, `ip`, `http_referrer`, `url`, `domain`
email_intel	`embedded_domain`, `src_user`, `subject`, `file_hash`, `file_name`, `embedded_ip`
service_intel	`service_file_hash`, `service_dll_file_hash`, `service_dll_file_name`, `service_file_name`, `service`
process_intel	`dest`, `src`, `process_file_name`, `process`
registry_intel	`registry_path`, `registry_value_name`, `registry_value_text`, `user`
certificate_intel	`certificate_issuer_common_name`, `certificate_subject_common_name`, `certificate_common_name`, `certificate_issuer_organization`, `certificate_subject_organization`, `certificate_serial`, `certificate_issuer`, `certificate_subject`, `certificate_issuer_unit`, `certificate_subject_unit`, `certificate_issuer_email`, `certificate_subject_email`, `ip`, `domain`

Response

If the request is successful, the endpoint returns all the rows that match the values supplied in the `item` parameter. If the request fails, the endpoint returns an error message.

Example 1: request
Retrieve a single intelligence item.

```
curl -k -u admin:pass https://localhost:8089/services/data/threat_intel/item/ip_intel  -d
item='{"ip":"10.10.1.1 "}'  -G -X  GET
```

Example 1: response
A single intelligence item is returned.

```
{"message": [{"_user": "nobody", "time": 1481566235.98246, "threat_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d|stix_file.xml.xml", "ip": "10.10.1.1", "_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d:fireeye:observable-00375905-04b4-4255-b453-2a5875c20b6d"}],
"status": true}
```

Example 2: request
Retrieve multiple intelligence items.

```
curl -k -u admin:pass https://localhost:8089/services/data/threat_intel/item/ip_intel -d
item='[{"ip":"5.5.5.5"},{"domain":"example.com"}]' -G -X GET
```

Example 2: response
Multiple intelligence items are returned.

```
{"message": [{"_user": "nobody", "time": 1481566235.98246, "threat_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d|stix_file.xml.xml", "ip": "5.5.5.5", "_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d:fireeye:observable-00375905-04b4-4255-b453-2a5875c20b5d"},
{"_user": "nobody", "time": 1481566235.98246, "threat_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491724d|stix_file.xml.xml", "domain": "example.com", "_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491724d:fireeye:observable-00375905-04b4-4255-b453-2a5875c20b6d"}],
"status": true}
```

PUT

Update one or more rows in a collection.

Request parameters

Field	Type	Default	Description
item required	JSON encoded string		The field/value pairs to update in one or more rows in a collection, accompanied by the `_key` for those rows. To update multiple rows with a single call, separate the JSON encoded strings with commas inside a set of brackets.
autofill_time_field optional	epoch	true	When set to "false", provides the current time for each row that you add to the KVStore collection. The value provided in this field is used for retention to phase out unused items instead of ingestion time. When set to "true" or if no value is provided, the time is automatically set for each row to the current time.

Data payload
The item field must contain at least one of the important fields for the threat collection being modified, in a JSON encoded string. The item field must also contain the `_key` value(s) of the rows you want to modify, in a JSON encoded string. You can update a single row or multiple rows.

Collection name	Important fields
user_intel	`user`

Collection name	Important fields
file_intel	file_name, file_hash
ip_intel	ip, domain
http_intel	http_user_agent, ip, http_referrer, url, domain
email_intel	embedded_domain, src_user, subject, file_hash, file_name, embedded_ip
service_intel	service_file_hash, service_dll_file_hash, service_dll_file_name, service_file_name, service
process_intel	dest, src, process_file_name, process
registry_intel	registry_path, registry_value_name, registry_value_text, user
certificate_intel	certificate_issuer_common_name, certificate_subject_common_name, certificate_common_name, certificate_issuer_organization, certificate_subject_organization, certificate_serial, certificate_issuer, certificate_subject, certificate_issuer_unit, certificate_subject_unit, certificate_issuer_email, certificate_subject_email, ip, domain

Response

The endpoint returns one of two responses:

```
{"status": true, "message": "Update operation successful."}

{"status": false, "message": <failure_cause>}
```

Example request

```
curl -k -u admin:pass https://localhost:8089/services/data/threat_intel/item/email_intel -d
item='[{"src_user": "user_new1", "subject":"click this new update", "time": 1620762180,
"_key":"126b0fd6d5c548fbb9a31107a4acddc1"}, {"src_user": "user_new2", "subject":"click this update",
"time": 1620762180, "_key":"46cdf4a3579244f4a409c69af02d5101"}]' -d autofill_time_field="false" -X PUT
```

Example response

```
{"status": true, "message": "Update operation successful."}
```

POST

Create one or more rows in a collection.

Request parameters

Field	Type	Default	Description
item required	JSON encoded string		The field/value pairs to insert as a new row in a collection. To create multiple rows with a single call, separate the JSON encoded strings with commas inside a set of brackets. Examples: `{'file_name':'threat.trt', 'file_extension':'trt'}` creates one new row that has `file_name=threat.trt` and `file_extension=trt`.

Field	Type	Default	Description
			`{'ip':'41.41.41.41'}` creates one new row with `ip=41.41.41.41`. `[{'ip':'41.41.41.41'}, {'ip':'10.10.10.10'}]` creates two new rows: one with `ip=41.41.41.41` and the other with `ip=10.10.10.10`. See the Data payload information for a list of important fields. The item must include at least one of the important fields for the relevant collection name, or the API returns an error.
autofill_time_field optional	epoch	true	When set to "false", provides the current time for each row that you add to the KVStore collection. The value provided in this field is used for retention to phase out unused items instead of ingestion time. When set to "true" or if no value is provided, the time is automatically set for each row to the current time.

Data payload

The item field must contain at least one of the important fields for the threat collection being modified, in a JSON encoded string.

Collection name	Important fields
user_intel	`user`
file_intel	`file_name`, `file_hash`
ip_intel	`ip`, `domain`
http_intel	`http_user_agent`, `ip`, `http_referrer`, `url`, `domain`
email_intel	`embedded_domain`, `src_user`, `subject`, `file_hash`, `file_name`, `embedded_ip`
service_intel	`service_file_hash`, `service_dll_file_hash`, `service_dll_file_name`, `service_file_name`, `service`
process_intel	`dest`, `src`, `process_file_name`, `process`
registry_intel	`registry_path`, `registry_value_name`, `registry_value_text`, `user`
certificate_intel	`certificate_issuer_common_name`, `certificate_subject_common_name`, `certificate_common_name`, `certificate_issuer_organization`, `certificate_subject_organization`, `certificate_serial`, `certificate_issuer`, `certificate_subject`, `certificate_issuer_unit`, `certificate_subject_unit`, `certificate_issuer_email`, `certificate_subject_email`, `ip`, `domain`

Response

The endpoint returns one of two responses:

```
{"status": true, "message": "Create operation successful."}

{"status": false, "message": <failure_cause>}
```

Example request

Create a single row.

```
curl -k -u admin:changeme https://localhost:8089/services/data/threat_intel/item/email_intel -d
item='{"src_user": "user_new", "subject":"click this", "time": 1620762180}' -d autofill_time_field="false"
```

Create two rows.

```
curl -k -u admin:pass https://localhost:8089/services/data/threat_intel/item/email_intel -d
item='[{"src_user": "user_new", "subject":"click this"},{"src_user": "user2_new", "subject":"click this"}]'
```

Example response

```
{"status": true, "message": "Create operation successful."}
```

DELETE

Delete one or more rows from a collection.

Request parameters

Field	Type	Default	Description
item required	JSON encoded string		The `_key` values of the rows to delete from a threat collection.

Data payload

The item field must contain `_key` values of the rows you intend to delete in a threat collection, in a JSON encoded string. You can delete a single row or multiple rows.

Response

The endpoint returns one of two responses:

```
{"message": "Delete operation successful.", "status": true}
```

```
{"status": false, "message": <failure_cause>}
```

Example request

```
curl -k -u admin:changeme https://localhost:8089/services/data/threat_intel/item/email_intel -d
item='[{"_key":"2e58177235804a739d4d768c26077b24"},{"_key":"2e58177235804a739d4d768c26077bd4"}]' -G -X
DELETE
```

Example response

```
{"message": "Delete operation successful.", "status": true}
```

/services/data/threat_intel/item/{threat_intel_collection}/{item_key}

Perform read, update, and delete operations on a row of an existing threat intelligence collection.

Syntax

```
https://<host>:<mPort>/services/data/threat_intel/item/{threat_intel_collection}/{item_key}
```

Usage details

The `threat_intel_collection` must be one of the following:

- ip_intel
- file_intel
- user_intel
- http_intel
- email_intel
- service_intel
- process_intel
- registry_intel
- certificate_intel

The `item_key` must be a valid key. To find the key for a row in a collection, run a search using the `inputlookup` command and the relevant `threat_intel_collection` and use eval to display the `_key` field for each row. For example:

```
| inputlookup ip_intel | eval item_key=_key
```

GET

Access a row from a collection.

Request parameters
None

Data payload
None

Response

The endpoint returns one of two responses:

```
{"message":<The row of the collection you requested, as a JSON object>, "status": true}
```

```
{"message": "No matching entries found in kvstore.", "status": false}
```

Example request

```
curl -k -u admin:changeme
https://localhost:8089/services/data/threat_intel/item/ip_intel/fireeye:stix-b7b16e67-4292-46a3-ba64
-60c1a491723d:fireeye:observable-00375905-04b4-4255-b453-2a5875c20b6d
```

Example response

```
{"message": [{"_user": "nobody", "time": 1481566235.98246, "threat_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d|stix_file.xml.xml", "ip": "58.64.179.144", "_key":
"fireeye:stix-b7b16e67-4292-46a3-ba64-60c1a491723d:fireeye:observable-00375905-04b4-4255-b453-2a5875c20b6d"}],
"status": true}
```

PUT

Update a row in a collection.

Request parameters

Field	Type	Default	Description
item required	JSON encoded string		The field/value pairs in the row to be updated. For example, `{'ip':'41.41.41.41'}` updates the `ip` to `41.41.41.41`. See the Data payload information for a list of important fields. The item must include at least one of the important fields for the relevant collection name, or the API returns an error.
autofill_time_field optional	epoch	true	When set to "false", provides the current time for each row that you add to the KVStore collection. The value provided in this field is used for retention to phase out unused items instead of ingestion time. When set to "true" or if no value is provided, the time is automatically set for each row to the current time.

Data payload

The item field must contain at least one of the important fields for the threat collection being modified.

Collection name	Important fields
user_intel	`user`
file_intel	`file_name`, `file_hash`
ip_intel	`ip`, `domain`
http_intel	`http_user_agent`, `ip`, `http_referrer`, `url`, `domain`
email_intel	`embedded_domain`, `src_user`, `subject`, `file_hash`, `file_name`, `embedded_ip`
service_intel	`service_file_hash`, `service_dll_file_hash`, `service_dll_file_name`, `service_file_name`, `service`
process_intel	`dest`, `src`, `process_file_name`, `process`
registry_intel	`registry_path`, `registry_value_name`, `registry_value_text`, `user`
certificate_intel	`certificate_issuer_common_name`, `certificate_subject_common_name`, `certificate_common_name`, `certificate_issuer_organization`, `certificate_subject_organization`, `certificate_serial`, `certificate_issuer`, `certificate_subject`, `certificate_issuer_unit`, `certificate_subject_unit`, `certificate_issuer_email`, `certificate_subject_email`, `ip`, `domain`

Response

The endpoint returns one of two responses:

```
{"status": true, "message": "Update operation successful."}

{"status": false, "message": <failure_cause>}
```

Example request

```
curl -k -u admin:changeme
https://localhost:8089/services/data/threat_intel/item/email_intel/7bce624696ad4bc48cc781d8c8204c8f -d
item='{"src_user": "user_new1", "subject":"click this!", "time": 123123123}' -d autofill_time_field="false"
-X PUT
```

Example response

```
{"status": true, "message": "Update operation successful."}
```

POST

Create one or more rows in a collection.

Request parameters

Field	Type	Default	Description
autofill_time_field optional	boolean	true	When set to "false", provides a valid time for each row that you add to the KVStore collection. The value provided in this field is used for retention to phase out unused items instead of ingestion time. When set to "true" or if no value is provided, the time is automatically set for each row to the current time.

DELETE

Delete a row from a collection.

Usage details
The delete operation does not delete the row from the KV Store. Instead, the entry is disabled from participating in threat intelligence matching.

Request parameters
None

Data payload
None

Response

The endpoint returns one of two responses:

```
{"message": "Delete operation successful.", "status": true}

{"message": <failure_reason>, "status": false}
```

Example request

```
curl -k -u admin:changeme
https://localhost:8089/services/data/threat_intel/item/email_intel/af8e379623f643a3be149c014b977e6b -X
DELETE
```

Example response

```
{"message": "Delete operation successful.", "status": true}
```

Notable Event endpoints

Notable Event API reference

Access the Notable Event framework in Splunk Enterprise Security.

The Notable Event framework provides a way to identify noteworthy incidents from events and then manage the ownership, triage process, and state of those incidents. For more information about working with the framework, see Notable Event framework in Splunk ES on the Splunk developer portal.

There is no GET method for notable events in Splunk Enterprise Security. Instead, you can use Splunk search and macros to access the notables programmatically. See Using notable events in search on the Splunk developer portal.

Usage details

Authentication and Authorization

Username and password authentication is required for access to endpoints and REST operations. You must have the `edit_notable_events` capability to use the notable event endpoint.

Alternatively, you can use token authentication. See Set up authentication with tokens in the Splunk Enterprise *Securing the Splunk Platform* manual.

Username and password authentication is used in the examples that follow.

Splunk Cloud Platform URL for REST API access

Splunk Cloud Platform has a different host and management port syntax than Splunk Enterprise. Depending on your deployment type, use one of the following options to access REST API resources.

Splunk Cloud Platform deployments
Use the following URL for single-instance deployments.

```
https://<deployment-name>.cloud.splunk.com:8089
```

Use the following URL for clustered deployments. If necessary, submit a support case to open port 8089 on your deployment.

```
https://<deployment-name>.splunkcloud.com:8089
```

To get the required credentials, submit a support case on the Support Portal. After installing the credentials, use the following URL.

```
https://input-<deployment-name>.cloud.splunk.com:8089
```

See Using the REST API in Splunk Cloud Platform in the the *Splunk REST API Tutorials* for more information.

/services/notable_update

Edit all notable events that match one or more ruleUIDs, or edit all notable events that match a search.

Syntax

```
https://<host>:<mPort>/services/notable_update
```
POST

Update the status, urgency, owner, or comment of one or more notable events.

Request parameters

An argument string must include at least one of the following arguments: comment, status, urgency, newOwner. It also must include either a searchID or one or more ruleUIDs.

Field	Description
comment	A description of the change or some information about the notable events.
status	A status ID matching a status in `reviewstatuses.conf`. Only required if you are changing the status of the event.
urgency	An urgency. Only required if you are changing the urgency of the event.
newOwner	An owner. Only required if reassigning the event.
ruleUIDs	A list of notable event IDs. Must be provided if a searchID is not provided. Include multiples of this attribute to edit multiple events. For example, `ruleUIDs=29439FBC-FFCB-45FF-93C2-420202012E1E@@notable@@75ecb6a3938d114b29c095f7ee9278b0&ruleUIDs=F93A9857-59D8-4AEB-AD97-4F182E0C959E@@notable@@1363d37ec74a79d00e22af26bfe0718b`.
searchID	An ID of a search. All of the events associated with this search will be modified unless a list of ruleUIDs are provided that limit the scope to a subset of the results.
disposition	An ID for a disposition that matches a disposition in the `reviewstatuses.conf` configuration file. Required only if you are changing the disposition of the event.

Response

A success or failure message.

Example request

```
curl -u admin:changeme -k https://localhost:8089/services/notable_update --data
"ruleUIDs=29439FBC-FFCB-45FF-93C2-420202012E1E@@notable@@75ecb6a3938d114b29c095f7ee9278b0&comment=Just
adding a comment&status=5&urgency=high&newOwner=analyst_name"
```
Example response

```
{"message":"1 event updated successfully","failure_count":0,"success":true,"details":{},"success_count":1}
```

Analytic Story endpoints

Analytic Story API reference

Access the Analytic Story framework in Splunk Enterprise Security.

The Analytic Story framework is a mechanism for proxy access to the stories in the Splunk Enterprise Security Content Update (ESCU). The ESCU analytic story content is available directly in Splunk ES through the use case library.

Usage details

Authentication and authorization

Username and password authentication is required for access to endpoints and REST operations. You must have the `edit_analyticstories` capability to use the analyticstories endpoint.

Alternatively, you can use token authentication. See Set up authentication with tokens in the Splunk Enterprise *Securing the Splunk Platform* manual.

Username and password authentication is used in the examples that follow.

Splunk Cloud Platform URL for REST API access

Splunk Cloud Platform has a different host and management port syntax than Splunk Enterprise. Depending on your deployment type, use one of the following options to access REST API resources.

Splunk Cloud Platform deployments'
Use the following URL for single-instance deployments.

```
https://<deployment-name>.cloud.splunk.com:8089
```

Use the following URL for clustered deployments. If necessary, submit a support case to open port 8089 on your deployment.

```
https://<deployment-name>.splunkcloud.com:8089
```

To get the required credentials, submit a support case on the Support Portal. After installing the credentials, use the following URL.

```
https://input-<deployment-name>.cloud.splunk.com:8089
```

See Using the REST API in Splunk Cloud Platform in the the *Splunk REST API Tutorials* for more information.

Common return format

For success and error responses, the general format follows.

Success
The general response for successes, such as `200/201`, follows:

```
{
  "entry": [
    {JSON object 1},
    ...
  ],
  "paging": {
    "offset": <number>,
    "perPage": <number>,
    "total": <number>
  }
}
```
where each object in the `entry` array would be specified per endpoint.

Error
The general response for errors, such as `4xx/5xx`, follows:

```
{
  "messages": [
    {
      "type": "ERROR",
      "text": "<error message>"
    },
    ...
  ]
}
```

/services/analyticstories/configs/{stanza_type}

Acts as a proxy to configs/conf-analyticstories, with validation.

Syntax

```
https://<host>:<mPort>/services/analyticstories/configs/{stanza_type}
```
Usage details
The {stanza_type} must be one of the following:

- analytic_story
- analytic_story_category
- savedsearch

GET

Return all a list of stanzas of the give stanza_type, or all stanzas if stanza_type is "all".

Example request
Retrieve a list of all analytic stories.

```
curl -k -u admin:pass https://localhost:8089/services/analyticstories/configs/analytic_story -G -X GET
```
Example response

If the request is successful, the endpoint returns a JSON array of objects from the stanza_type. Each object is of format returned by "single entity" endpoint, {stanza_type}/{name}. If the request fails, the endpoint returns an error message.

```
{
        "paging": {
                "offset": 0,
                "perPage": 30,
                "total": 3
        },
        "entry": [{
                "acl": {
                        "can_share_app": true,
                        "modifiable": true,
                        "owner": "nobody",
                        "can_write": true,
                        "removable": false,
                        "sharing": "global",
                        "can_change_perms": true,
                        "can_share_user": false,
                        "can_list": true,
                        "perms": {
                                "write": ["*"],
                                "read": ["*"]
                        },
                        "app": "DA-ESS-AccessProtection",
                        "can_share_global": true
                },
                "name": "Access Protection",
                "stanza_type": "analytic_story",
                "content": {
                        "narrative": "Access protection&#151;selective restriction of who may access data
or other resources via authentication and authorization&#151;is arguably one of the most important
activities in network security. The first part of the equation, authentication, involves ensuring that the
user is who they purport to be. Next, an authorization process determines the level of privilege the user or
role possesses.\n\nAlthough implementing an end-to-end access-protection system is critical, no system is
foolproof, making ongoing automated monitoring critical. This use case is designed to help you detect
attackers who may be attempting to circumvent access-protection mechanisms in your environment. It includes
the following searches: \n\n1. **Excessive Failed Logins:** Detects excessive numbers of failed login
attempts (typical of a brute-force attack)\n\n1. **Inactive Account Usage:** Discovers previously inactive
accounts that are now being used.\n\n1. **Insecure Or Cleartext Authentication Detected:** Detects
authentication requests that transmit the password over the network as cleartext (unencrypted).\n\n1.
**Short-lived Account Detected:** Looks for accounts that were created and or deleted within the previous
four hours&#151;a possible red flag.",
                        "spec_version": 1,
                        "description": "Monitoring account activity and securing authentication are
critical to enterprise security. This use case includes searches that detect suspicious account activity and
alert you to the use of cleartext (non-encrypted) authentication protocols.",
                        "last_updated": "2018-09-13",
                        "version": 1.0,
                        "maintainers": [{
                                "email": "rvaldez@splunk.com",
                                "name": "Rico Valdez",
                                "company": "Splunk"
                        }],
                        "searches": ["Access - Excessive Failed Logins - Rule", "Access - Inactive Account
Usage - Rule", "Access - Insecure Or Cleartext Authentication - Rule", "Access - Short-lived Account
Detected - Rule"],
                        "category": "Best Practices",
                        "references":
["https://www.sans.org/media/critical-security-controls/critical-controls-poster-2016.pdf"]
                }
        }, {
                "acl": {
                        "can_share_app": true,
```

```
                        "modifiable": true,
                        "owner": "nobody",
                        "can_write": true,
                        "removable": false,
                        "sharing": "global",
                        "can_change_perms": true,
                        "can_share_user": false,
                        "can_list": true,
                        "perms": {
                                "write": ["*"],
                                "read": ["*"]
                        },
                        "app": "DA-ESS-EndpointProtection",
                        "can_share_global": true
                },
                "name": "Endpoint Protection",
                "stanza_type": "analytic_story",
                "content": {
                        "narrative": "Continuous advanced threat monitoring on the endpoint is one of the
most critical components of your organization's security strategy. Its focus is on early detection and
analysis of suspicious events that may indicate the presence of an internal or external threat in your
environment. \n\nThe time to get serious about advanced threat detection on your endpoints is after you have
implemented basic security measures related to your network perimeter, devices, and applications. The most
critical areas of focus include detection of anomalous and prohibited processes and services, suspicious
behavior involving user accounts, and malware outbreaks.\n\nThis use case includes searches to help with all
of these and more. They include: \n\n1. **Anomalous New Process:** This search alerts you when an unusual
number of hosts are detected running a new process.\n\n1. **Anomalous New Service:** Alerts when an
anomalous number of hosts are detected with a new service.\n\n1. **Prohibited Process Detected:** Detects
prohibited processes running in the environment.\n\n1. **Prohibited New Service:**  Detects prohibited
services running in the environment.\n\n1. **New User Account Created on Multiple Hosts:** This search
alerts you when it detects that numerous accounts are created for a username across multiple hosts. \n\n1.
**Outbreak Detected:** This search will give you a distinct count of the number of systems affected by the
malware event destination for the last 24 hours.",
                        "spec_version": 1,
                        "description": "Use the included searches to set up continuous monitoring for some
of the most common threats in your network.",
                        "last_updated": "2018-11-05",
                        "version": 1.0,
                        "maintainers": [{
                                "email": "rvaldez@splunk.com",
                                "name": "Rico Valdez",
                                "company": "Splunk"
                        }],
                        "searches": ["Endpoint - Anomalous New Processes - Rule", "Endpoint - Anomalous New
Services - Rule", "Endpoint - Anomalous User Account Creation - Rule", "Endpoint - Outbreak Observed -
Rule", "Endpoint - Prohibited Process Detection - Rule", "Endpoint - Prohibited Service Detection - Rule"],
                        "category": "Adversary Tactics",
                        "references": [""]
                }
        }, {
                "acl": {
                        "can_share_app": true,
                        "modifiable": true,
                        "owner": "nobody",
                        "can_write": true,
                        "removable": false,
                        "sharing": "global",
                        "can_change_perms": true,
                        "can_share_user": false,
                        "can_list": true,
                        "perms": {
                                "write": ["*"],
```

```
                                "read": ["*"]
                        },
                        "app": "DA-ESS-NetworkProtection",
                        "can_share_global": true
                },
                "name": "Network Protection",
                "stanza_type": "analytic_story",
                "content": {
                        "narrative": "Continuous network monitoring for advanced threats has become mission
critical in the context of today's constantly evolving threat landscape. What's more, given the potentially
devastating consequences of a breach, the necessity for strong defense, rapid detection, and the agility to
respond immediately to threats has never been higher.\n\nThis use case focuses on detecting anomalous
activity that may indicate that an adversary has penetrated your network. It monitors DNS and ports for
unusual activity and looks for vulnerability scanners.\n\nIt includes the following searches:\n\n1.
**Excessive DNS Failures:** This search alerts when a host is detected with a high number of failed DNS
requests. \n\n1. **Excessive DNS Queries:** Detect an excessive number of DNS queries from a host, which may
indicate that a threat actor is sending data to a malicious server.\n\n1. **Prohibited Port Activity
Detected:** Use of unapproved ports can help you monitor for the installation of new software (potentially
unapproved) or a successful compromise of a host (such as the presence of a backdoor or a system
communicating with a botnet). This search returns values where the destination port is > 0 and
`is_prohibited` is not \"false.\"\n\n1. **Vulnerability Scanner Detected (by events):** Detects possible
vulnerability scanners by monitoring for devices that have triggered a large number of unique events,
activity that is typical of a vulnerability scanner (because each vulnerability check tends to trigger a
unique event).\n\n1. **Vulnerability Scanner Detected (by targets):** This search detects possible
vulnerability scanners by monitoring for devices that have triggered events against a large number of unique
targets, which is typical for vulnerability scanners (they are scanning a network for vulnerable hosts).",
                        "spec_version": 1,
                        "description": "Detect anomalous activity that may indicate that an adversary has
penetrated your network. This use case monitors DNS and ports for unusual activity and looks for
vulnerability scanners.",
                        "last_updated": "2018-11-05",
                        "version": 1.0,
                        "maintainers": [{
                                "email": "rvaldez@splunk.com",
                                "name": "Rico Valdez",
                                "company": "Splunk"
                        }],
                        "searches": ["Network - Excessive DNS Failures - Rule", "Network - Excessive DNS
Queries - Rule", "Network - Unapproved Port Activity Detected - Rule", "Network - Vulnerability Scanner
Detection (by event) - Rule", "Network - Vulnerability Scanner Detection (by targets) - Rule"],
                        "category": "Best Practices",
                        "references":
["https://www.esecurityplanet.com/network-security/security-information-event-management-siem.html"]
                }
        }]
}
```


POST

Add a new stanza of the given stanza_type. Following splunkd common practice, PUT with `name` argument will also create a new stanza of that name.

Request string

An argument string must include the following fields: `name`, `stanza_type`, `content`.

Field	Description
name	The name of the new stanza that you're creating. Special {stanza_type} "all" is not allowed. Required.
stanza_type	*analytic_story*

Field	Description
	For adding an analytic story, the `stanza_type` parameter is `analytic_story`. Required parameters for `analytic_story` are the following: `name`, `searches`, `spec_version`, `version`. *analytic_story_category* For adding additional info to an analytic story category, the `stanza_type` parameter is `analytic_story_category`. This is additional info for an analytic story category <name> where `name` is the value of `category`under analytic_story stanza. The required parameter for `analytic_story_category` is `name`. *savedsearch* For adding saved search metadata, the `stanza_type` parameter is `savedsearch`. This defines metadata for savedsearch named `name`. Required parameters for `analytic_story` are the following: `name`, `type`.
content	A JSON object containing the key/value pairs. For the parameters of each {stanza_type}, see Stanza_type Parameters.

Response

A success or failure message.

Example request

Add a new analytic story named "my_story" as follows:

```
curl -k -u admin:changeme https://localhost:8089/services/analyticstories/configs/analytic_story \
-d '{"name":"my_story","content":{"category":"Best Practices","description":"This is my own analytic
story","searches":["ESCU - Prohibited Software On Endpoint - Rule","ESCU - Detect malicious requests to
exploit JBoss servers - Rule"],"spec_version":1,"version":1}}'
```

Example response


```
{
        "entry": [{
                "stanza_type": "analytic_story",
                "acl": {
                        "can_share_global": true,
                        "can_change_perms": true,
                        "can_share_app": true,
                        "can_list": true,
                        "removable": true,
                        "modifiable": true,
                        "can_write": true,
                        "sharing": "app",
                        "owner": "admin",
                        "perms": {
                                "read": ["*"],
                                "write": ["admin", "power"]
                        },
                        "app": "search",
                        "can_share_user": true
                },
                "content": {
                        "spec_version": 1,
                        "searches": ["ESCU - Prohibited Software On Endpoint - Rule", "ESCU - Detect
malicious requests to exploit JBoss servers - Rule"],
                        "category": "Best Practices",
                        "description": "This is my own analytic story",
                        "version": 1
                },
                "name": "my_story"
        }],
        "paging": {
```

```
                "perPage": 30,
                "total": 1,
                "offset": 0
        }
}
```

/services/analyticstories/configs/{stanza_type}/{name}

Acts as a proxy to configs/conf-analyticstories/{stanza_name}. Accepts standard splunkd query terms.

Syntax

```
https://<host>:<mPort>/services/analyticstories/configs/{stanza_type}/{name}
```

Usage details

The {stanza_type} must be one of the following:

- analytic_story
- analytic_story_category
- savedsearch

POST

Update stanza {name} of the specified stanza_type. This updates corresponding fields specified in the REST call and leaves the other existing values intact.

Request string

An argument string must include the following fields: `name`, `stanza_type`, `content`.

Field	Description
name	The name of the new stanza that you're creating. Special {stanza_type} "all" is not allowed. Required.
stanza_type	*analytic_story* For adding an analytic story, the `stanza_type` parameter is `analytic_story`. Required parameters for `analytic_story` are the following: `name`, `searches`, `spec_version`, `version`. *analytic_story_category* For adding additional info to an analytic story category, the `stanza_type` parameter is `analytic_story_category`. This is additional info for an analytic story category <name> where `name` is the value of `category`under analytic_story stanza. The required parameter for `analytic_story_category` is `name`. *savedsearch* For adding saved search metadata, the `stanza_type` parameter is `savedsearch`. This defines metadata for savedsearch named `name`. Required parameters for `analytic_story` are the following: `name`, `type`.
content	A JSON object containing the key/value pairs. For the parameters of each {stanza_type}, see Stanza_type Parameters.

Example request

Update the analytic story named "Access Protection" with a new description.

Create a file called `update.json`, for example, with the following content:

```
{
```

```
  "content": {
    "last_updated": "2019-05-31",
    "description": "my new description",
    "version": 1
  }
}
```

Use the file to update the analytic story as follows:

```
curl -k -u admin:changeme
https://localhost:8089/services/analyticstories/configs/analytic_story/Access%20Protection -d @update.json |
python -m json.tool
```

Example response

If the request is successful, the endpoint returns a JSON object.


```
{
        "entry": [{
                "acl": {
                        "app": "search",
                        "can_change_perms": true,
                        "can_list": true,
                        "can_share_app": true,
                        "can_share_global": true,
                        "can_share_user": true,
                        "can_write": true,
                        "modifiable": true,
                        "owner": "admin",
                        "perms": {
                                "read": [
                                        "*"
                                ],
                                "write": [
                                        "admin",
                                        "power"
                                ]
                        },
                        "removable": true,
                        "sharing": "app"
                },
                "content": {
                        "category": "Best Practices",
                        "description": "my new description",
                        "last_updated": "2019-05-31",
                        "maintainers": [{
                                "company": "Splunk",
                                "email": "rvaldez@splunk.com",
                                "name": "Rico Valdez"
                        }],
                        "narrative": "Access protection&#151;selective restriction of who may access data
or other resources via authentication and authorization&#151;is arguably one of the most important
activities in network security. The first part of the equation, authentication, involves ensuring that the
user is who they purport to be. Next, an authorization process determines the level of privilege the user or
role possesses.\n\nAlthough implementing an end-to-end access-protection system is critical, no system is
foolproof, making ongoing automated monitoring critical. This use case is designed to help you detect
attackers who may be attempting to circumvent access-protection mechanisms in your environment. It includes
the following searches: \n\n1. **Excessive Failed Logins:** Detects excessive numbers of failed login
attempts (typical of a brute-force attack)\n\n1. **Inactive Account Usage:** Discovers previously inactive
accounts that are now being used.\n\n1. **Insecure Or Cleartext Authentication Detected:** Detects
authentication requests that transmit the password over the network as cleartext (unencrypted).\n\n1.
**Short-lived Account Detected:** Looks for accounts that were created and or deleted within the previous
```

```
four hours&amp;#151;a possible red flag.",
                        "references": [
                                "https://www.sans.org/media/critical-security
-controls/critical-controls-poster-2016.pdf"
                        ],
                        "searches": [
                                "Access - Excessive Failed Logins - Rule",
                                "Access - Inactive Account Usage - Rule",
                                "Access - Insecure Or Cleartext Authentication - Rule",
                                "Access - Short-lived Account Detected - Rule"
                        ],
                        "spec_version": 1,
                        "version": 1
                },
                "name": "Access Protection",
                "stanza_type": "analytic_story"
        }],
        "paging": {
                "offset": 0,
                "perPage": 30,
                "total": 1
        }
}
```


PUT

Replace stanza {name} of the specified {stanza_type}. This replaces the entire configuration of that stanza with the values in the REST call. If the REST call does not provide a value for a field, it is treated as unset.

Request string

An argument string must include the following fields: `name`, `stanza_type`, `content`.

Field	Description
name	The name of the stanza that you're updating. Special {stanza_type} "all" is not allowed. Required.
stanza_type	*analytic_story* For adding an analytic story, the `stanza_type` is `analytic_story`. Required parameters for `analytic_story` are the following: `name`, `searches`, `spec_version`, `version`. *analytic_story_category* For adding additional info to an analytic story category, the `stanza_type` is `analytic_story_category`. This is additional info for an analytic story category <name> where `name` is the value of `category`under analytic_story stanza. The required parameter for `analytic_story_category` is `name`. *savedsearch* For adding saved search metadata, the `stanza_type` is `savedsearch`. This defines metadata for savedsearch named `name`. Required parameters for `analytic_story` are the following: `name`, `type`.
content	A JSON object containing the key/value pairs. For the parameters of each {stanza_type}, see Stanza_type Parameters.

Example request

Update the savedsearch named "my_saved_search" with new parameters for type, spec_version, status, confidence, and asset_type.

```
curl -X PUT -u admin:changeme -k
https://localhost:8089/services/analyticstories/configs/savedsearch/my_saved_search \
-d '{"name":"my_saved_search","stanza_type":"savedsearch","content":{"type":"access","spec
_version":1,"status":"development","confidence":"medium","asset_type":"AWS"}}'
```

Example response

If the request is successful, the endpoint returns a JSON object.

```
{"paging": {"perPage": 30, "offset": 0, "total": 1}, "entry": [{"name": "my_saved_search", "content":
{"spec_version": 1, "asset_type": "AWS", "confidence": "medium", "status": "development", "type": "access"},
"stanza_type": "savedsearch", "acl": {"modifiable": true, "can_list": true, "owner": "admin", "can_write":
true, "app": "search", "can_share_app": true, "removable": true, "perms": {"write": ["admin", "power"],
"read": ["*"]}, "can_share_global": true, "can_change_perms": true, "can_share_user": true, "sharing":
"app"}}]}
```


GET

Return the stanza {name} of the specified stanza_type.

Example request
Retrieve the analytic story named "my_story."

```
curl -k -u admin:changeme https://localhost:8089/services/analyticstories/configs/analytic_story/my_story -G
-X GET
```

Example response
If the request is successful, the endpoint returns a JSON object.


```
{"entry":[{"acl":{"can_share
_global":true,"modifiable":true,"removable":true,"sharing":"global","app":"SplunkEnterpriseSecuritySuite","can
_share_app":true,"owner":"plainuser","can_list":true,"perms":{"read":["*"],"write":["*"]},"can_change
_perms":true,"can_share_user":true,"can_write":true},"content":{"descriptions":"asdf","spec_version":1},"stanza
_type":"analytic_story","name":"alala"}],"paging":{"offset":0,"perPage":30,"total":1}}
```


If the request fails, the endpoint returns an error message.


```
{"messages":[{"text":"version: u'B' is not of type u'integer'","type":"ERROR"},{"text":"maintainers:
u'bennay' is not of type u'array'","type":"ERROR"}]}
```


/services/analyticstories/configs/{stanza_type}/{name}/acl

Returns ACL information. The accepted parameters are the same as Splunk /acl endpoints. See Access Control List in the *REST API User Manual*.

Syntax

```
https://<host>:<mPort>/services/analyticstories/configs/{stanza_type}/{name}/acl
```

Usage details
The {stanza_type} must be one of the following:

- analytic_story
- analytic_story_category
- savedsearch

POST

Update ACL information.

Example request
Share the analytic story named "my other story" globally and change its permissions.

```
curl -k -u admin:changeme
```

```
https://localhost:8089/services/analyticstories/configs/analytic_story/my%20other%20story/acl \
        -d owner=plainuser \
        -d perms.read=* \
        -d sharing=global
```

Example response

If the request is successful, the endpoint returns a JSON object.

```
{"entry": [{"owner": "plainuser", "can_write": true, "perms": {"read": ["*"]}, "sharing": "global", "app":
"search", "modifiable": true, "can_share_app": true, "removable": true, "can_share_global": true,
"can_share_user": true, "can_change_perms": true, "can_list": true}], "paging": {"total": 1, "perPage": 30,
"offset": 0}}
```

GET

Return the ACL information.

Example request

Retrieve the ACL for the analytic story named "my_story."

```
curl -k -u admin:changeme
https://localhost:8089/services/analyticstories/configs/analytic_story/my_story/acl -G -X GET
```

Example response

If the request is successful, the endpoint returns a JSON object.

```
{"entry":[{"can_share
_global":true,"modifiable":true,"removable":true,"sharing":"global","app":"SplunkEnterpriseSecuritySuite","can
_share_app":true,"owner":"plainuser","can_list":true,"perms":{"read":["*"],"write":["*"]},"can_change
_perms":true,"can_share_user":true,"can_write":true}],"paging":{"offset":0,"perPage":30,"total":1}}
```

/services/analyticstories/configs/{stanza_type}/{name}/move

Moves stanzas to other apps. The accepted parameters are the same as Splunk /move endpoints. See Move an object to a different app in the *REST API Tutorials*.

Syntax

```
https://<host>:<mPort>/services/analyticstories/configs/{stanza_type}/{name}/move
```

Usage details

The {stanza_type} must be one of the following:

- analytic_story
- analytic_story_category
- savedsearch

POST

Move stanza to another app.

Example request

Make the analytic story named "my_story" available to all users and change the context to a different app.

```
curl -k -u admin:changeme
```

```
https://localhost:8089/services/analyticstories/configs/analytic_story/my_story/move  \
        -d user=nobody \
        -d app=otherapp
```

/services/analyticstories/configs/_reload

Reloads data for the endpoint.

Syntax

```
https://<host>:<mPort>/services/analyticstories/configs/_reload
```

GET

Append `_reload` to a URL to force the server to reload data for the endpoint.

Example request
Inform splunkd to reload analyticstories.conf, in case of manual edits to the file.

```
curl -k -u admin:changeme https://localhost:8089/services/analyticstories/configs/_reload  -G -X GET
```

/services/analyticstories/schemas/{version}

Return the JSON schema for the analytic stories. The schema is used for validation.

Syntax

```
https://<host>:<mPort>/services/analyticstories/schemas/{version}
```

Usage details
The last segment {version} would be either of the following:

- The letter `v` followed by spec version number, such as `v1`
- The word `latest` for the latest version

Frontend or users can use the returned schema to do validations.

GET

Return the JSON schema for version 1.

Example request
Return the JSON schema for version 1.

```
curl -k -u admin:changeme https://localhost:8089/services/analyticstories/schemas/v1 -G -X GET
```

/services/analyticstories/batch

Takes a JSON array conforming to the analytic story JSON schema and saves it in proper format into analyticstories.conf.

Syntax

```
https://<host>:<mPort>/services/analyticstories/batch
```

Usage details
The {stanza_type} must be one of the following:

- analytic_story
- analytic_story_category
- savedsearch

POST

Batch update or create multiple stanzas.

Example request
The POST input is a JSON array whose element is of format specified in "/services/analyticstories/configs/{stanza_type}"

Create a file called `batch.json`, for example, with the following content:

```
[{
  "name": "my other story",
  "stanza_type": "analytic_story",
  "content": {
    "category": "Best Practices",
    "description": "This is my own other analytic story",
    "searches": ["ESCU - Other Prohibited Software On Endpoint - Rule", "ESCU - Detect malicious requests to
exploit JBoss servers - Rule"],
    "spec_version": 1,
    "version": 1
  }
}, {
  "name": "my third story",
  "stanza_type": "analytic_story",
  "content": {
    "category": "Malware",
    "description": "Detect malware",
    "searches": ["ESCU - Detect malicious requests to exploit JBoss servers - Rule"],
    "spec_version": 1,
    "version": 1,
    "last_updated": "2019-06-02"
  }
}, {
  "name": "SpearPhishing",
  "stanza_type": "analytic_story_category",
  "content": {
    "icon": "spearphishing.png"
  }
}]
```

Use the file to update multiple analytic stories as follows:

```
curl -k -u admin:changeme https://localhost:8089/services/analyticstories/batch -d @batch.json | python -m
json.tool
```

Example response
If the request is successful, the endpoint returns the items created or updated.

```
{"entry":[{"acl":{"app":"search","can_change_perms":true,"can_list":true,"can_share_app":true,"can_share
_global":true,"can_share_user":true,"can
_write":true,"modifiable":true,"owner":"admin","perms":{"read":["*"],"write":["admin","power"]},"removable":tr
```

```
ue,"sharing":"app"},"content":{"category":"Best Practices","description":"This is my own other analytic
story","searches":["ESCU - Other Prohibited Software On Endpoint - Rule","ESCU - Detect malicious requests
to exploit JBoss servers - Rule"],"spec_version":1,"version":1},"name":"my other
story","stanza_type":"analytic_story"},{"acl":{"app":"search","can_change_perms":true,"can_list":true,"can_share
_app":true,"can_share_global":true,"can_share_user":true,"can
_write":true,"modifiable":true,"owner":"admin","perms":{"read":["*"],"write":["admin","power"]},"removable":tr
ue,"sharing":"app"},"content":{"category":"Malware","description":"Detect
malware","last_updated":"2019-06-02","searches":["ESCU - Detect malicious requests to exploit JBoss servers
- Rule"],"spec_version":1,"version":1},"name":"my third
story","stanza_type":"analytic_story"},{"acl":{"app":"search","can_change_perms":true,"can_list":true,"can_share
_app":true,"can_share_global":true,"can_share_user":true,"can
_write":true,"modifiable":true,"owner":"admin","perms":{"read":["*"],"write":["admin","power"]},"removable":tr
ue,"sharing":"app"},"content":{"icon":"spearphishing.png","spec_version":1},"name":"SpearPhishing","stanza
_type":"analytic_story_category"}]}
```


Stanza_type Parameters

The following parameters are used per stanza_type.

analytic_story parameters	description
name	Defines an analytic story of title `name`, such as analytic_story://<name>. Cannot be an empty string.
category	A string that best describes this type of analytic story, such as "Abuse" or "Compliance" or "Malware." If unset, it will be displayed as "Uncategorized". Optional.
description	A string explanation of why the story is useful or what the story includes.
last_updated	Update time of the analytic story in the format of ISO 8601 date format YYYY-MM-DD. Optional.

maintainers	A JSON array of the current maintainers of the analytic story. Defaults to an empty array. Optional.
	
	Format of each item:
	
	`{`
	`"company": "<string>",`
	`"email": "<string>",`
	`"name": "<string>"`
	`}`
	company (optional): Company associated with the person maintaining this analytic story
	email (required, valid RFC5322 addr-spec): Email address of the person maintaining this analytic story
	name (required, non-empty): Name of the person maintaining this analytic story

narrative	Long-form text that describes the analytic story use case and the rationale behind it, an overview of the included searches, and how to enable the story. Optional.
references	A JSON array of URLs that give information about the problem the story is addressing. Optional.
searches	A JSON array of searches used by the analytic story. Each string in the array refers to a unique savedsearch name. Required.
spec_version	Version of analytics spec used by current stanza, in positive integer number format. A larger number means a more recent update. Required.

analytic_story_category parameters	description
name	Additional info for analytic story category <name>, where <name> is the value of 'category' under analytic_story stanza. Cannot be an empty string.

analytic_story_category parameters	description
description	A string explanation of the category.
icon	An image file for the category. It should be the filename of the icon image located under <app>/appserver/static. Supported format is png. Optional.
spec_version	Version of analytics spec used by current stanza, in positive integer number format. A larger number means a more recent update. Required.

savedsearch parameters	description
name	Defines metadata for savedsearch named <name>. Cannot be an empty string.

annotations	A JSON object of metadata on the search, currently used to annotate a search with various industry standards and frameworks. Optional.
	
	The supported format and standards follow:
	
	`{`
	`  "cis20": ["<string>"],`
	`  "kill_chain_phases": ["<string>"],`
	`  "mitre_attack": ["<string>"],`
	`  "nist": ["<string>"]`
	`}`
	cis20: Critical security controls this search implements
	kill_chain_phases: Kill-chain phases to which the search applies
	mitre_attack: Techniques and tactics identified by the search
	nist: Controls the search implements

asset_type	Type of asset being investigated in string format. For example, AWS Instance. Optional.
confidence	Confidence that detected behavior is malicious. For instance, high, medium, or low. Optional.
earliest_time_offset	The number of seconds into the past from the event time the search should cover, in the format of a non-negative integer. Optional.
explanation	Detailed description of the SPL, written in a style that can be understood without deep technical knowledge. Optional.
how_to_implement	A description of how to put this search into effect, from what needs to be ingested, config files modified, and suggested per site modifications. Optional.
known_false_positives	Scenarios in which detected behavior is benign, coupled with suggestions on how to verify the behavior. Optional.
latest_time_offset	The number of seconds into the future from the event time the search should cover, in the format of a non-negative integer. Optional.
providing_technologies	External services, software, or hardware that would provide data this analytic story relies on, in the format of a JSON array of strings. Optional.
status	Current status of the search. For example: development, experimental, production. Optional.
type	Type of this search. Cannot be an empty string. Required.