Silobreaker Platform

Playbooks App Documentation v1.0

Overview

The purpose of this document is to provide readers with an understanding of the Silobreaker Platform playbooks app for ThreatConnect.

The following topics are covered:

About Silobreaker

Silobreaker is a SaaS platform that allows users to monitor and research a wide variety of different topics, threats and incidents. Data from approximately a million openly available sources, as well as deep & dark web content (provided by partners), is regularly indexed and made available in the platform for users to query, visualize and export.

The data in question is primarily textual and “unstructured”, comprising news, blogs, reports, feeds, forum posts and tweets. It is not restricted to cybersecurity-related topics but also supports other use-cases, including reputational risk, business risk and physical security.

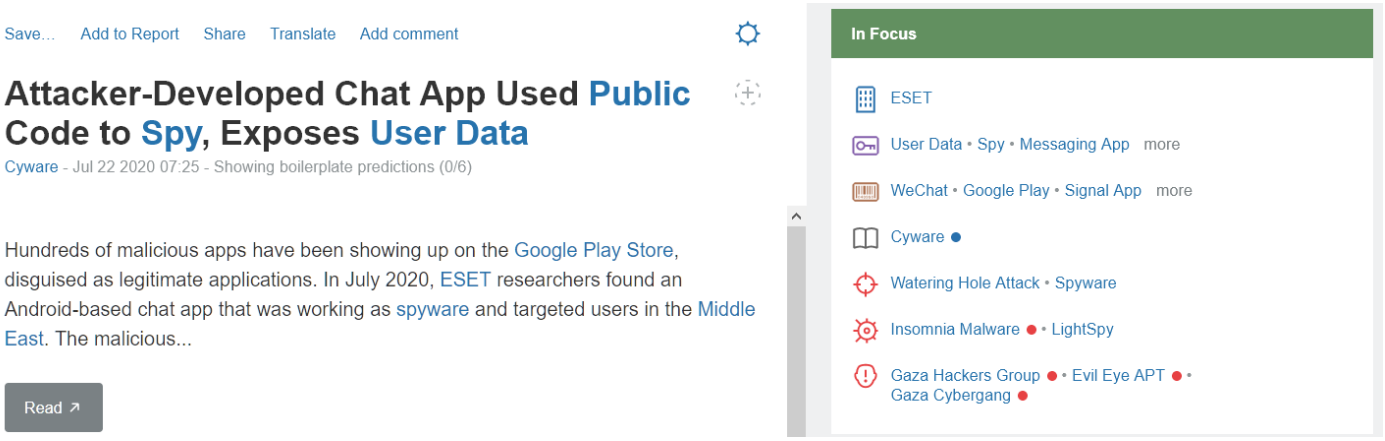
In addition to displaying information in terms of documents, Silobreaker also organizes data through the use of “entities”. These resemble ThreatConnect’s Indicators, but include people, places, companies, malware, threat actors, vulnerabilities and many other types. Users are encouraged to leverage entities when making searches and can use various tools to request both documents and entities back as results.

App Functionality Overview

The Silobreaker Platform app allows users to pull content from Silobreaker's API and work with it in ThreatConnect's playbooks.

To do this, the app makes several useful endpoints available to playbook owners. These let users bring documents and entities from Silobreaker into a playbook, and pass those results downstream to other apps. The app always returns content in the form of a JSON, which contains different fields depending on the endpoint being used to perform the query, and whether the result set consists of documents or entities. A brief description of the difference between documents and entities is helpful to understanding the app's output.

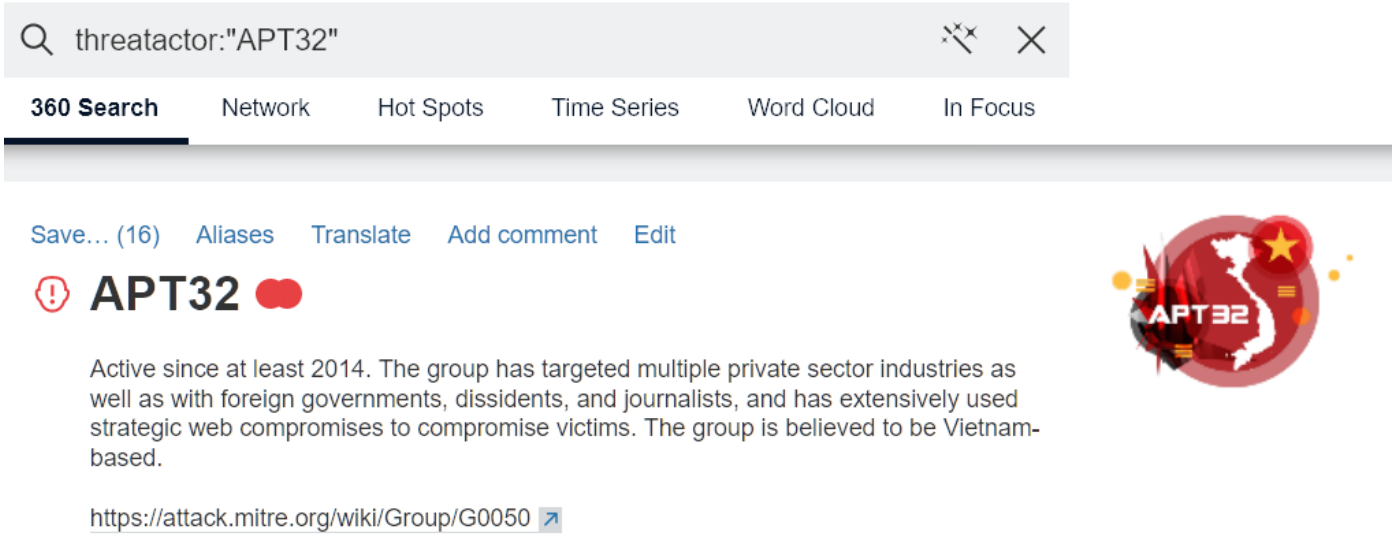
What is a document?



A document in Silobreaker, and a number of entities from within the same document (full document not shown).

A document is a piece of textual content from a specific source that contains attributes, such as a publication date and source URL. It could be a news article, a blog post, forum content or a PDF a user has uploaded. When a user makes a search, they will often want to get results back as documents. Silobreaker contains millions of documents in different languages and from different sources.

What is an entity?



A single entity in Silobreaker

An entity is a predefined search term that has been given a type and may contain a number of aliases.

For example, the advanced persistent threat known as APT32 is an entity of type “threatactor”, with several aliases, including Ocean Lotus. The IP address 123.123.123.123 is also an entity – it has the type “ipv4”, and no other aliases.

Silobreaker uses entities to bring structure to documents. If a Silobreaker user wants to know which malware is associated with APT32, they could do this by asking for malware entities that exist within the documents that mention the APT32 threat actor.

There are millions of distinct entities in Silobreaker, covering nearly 30 different types including:

- People, places and products
- Malware, attacktypes and threatactors
- Companies, vulnerabilities and BIN numbers
- Email addresses, domains and hashes

For more information on entities, please refer to the Silobreaker platform documentation.

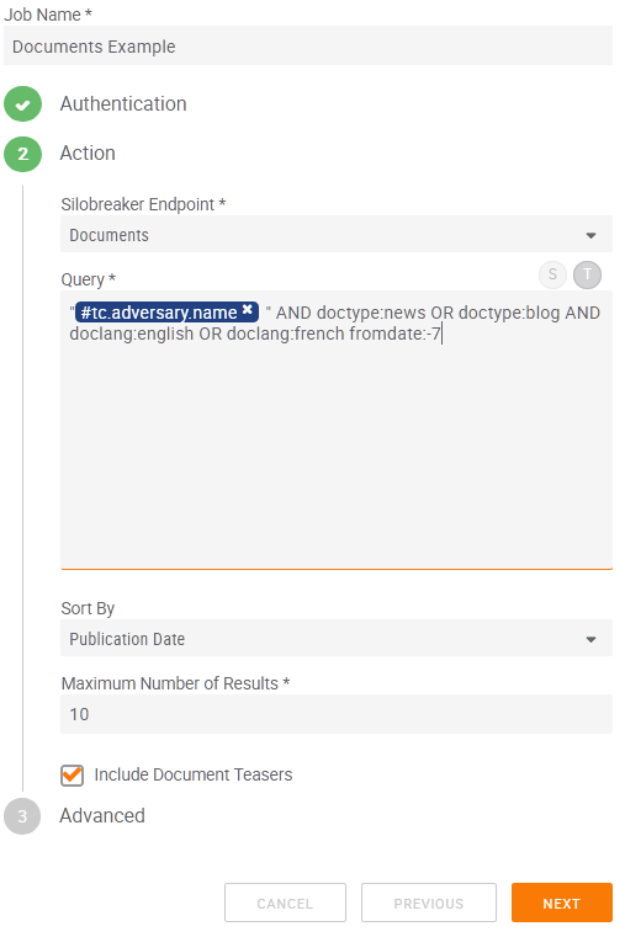
Endpoints

To use the Silobreaker Platform app, some familiarity with Silobreaker’s search syntax is recommended. Any syntax used in Silobreaker can also be used in the app. Search syntax should be placed in the `Query` or `Term Query` portion of the app’s Action step, depending on the selected endpoint.

The available endpoints for this app are described below.

Documents

The **Documents** endpoint returns documents based on a query supplied by the user. Options are available to sort, restrict or add more context to results.



Query

In this case, we are finding and returning news, reports or blog posts, written in English or French and published in the last 7 days, if they mention the selected `tc adversary`. The adversary is enclosed in vertical double quotes, because it is likely to contain more than one word, e.g. "Lazarus Group" - see Troubleshooting.

Sort By

Sorts results by publication date, or by relevance. Relevance is based on a variety of factors, including whether the query terms appear in the document’s title, and how unusual they are across Silobreaker’s document set.

Maximum Number of Results

Limits the maximum number of results to the number supplied.

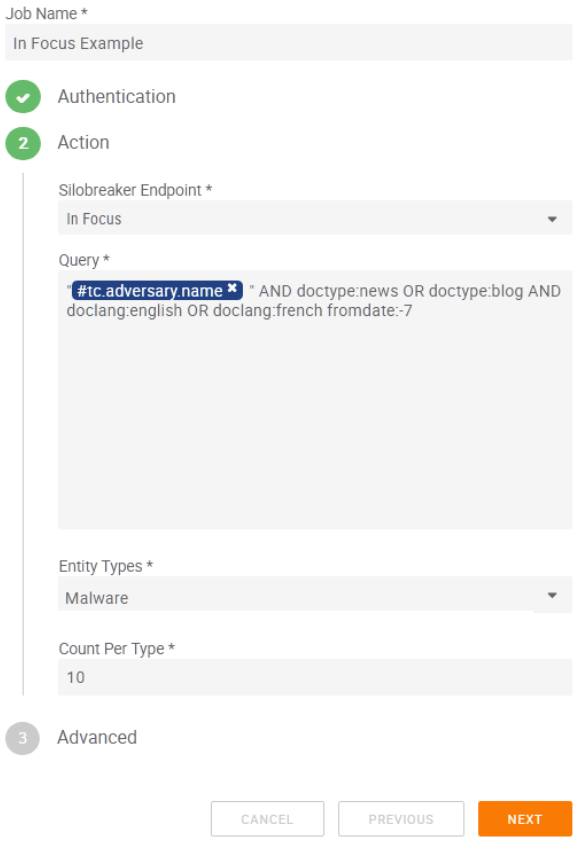
Include Document Teasers

Whether or not to include a small piece of content from the start of each document in the results.

For more information on searching, please see the corresponding documentation.

In Focus

The **In Focus** endpoint returns entities, sorted by relevance and entity type, based on a query supplied by the user. Options are available to select the entities returned by their type and count.



Query

The previous query is used again, however, instead of returning documents, up to 10 entities of the malware type will be extracted from this document set and returned.

The purpose of this endpoint is to answer questions like: “Which malware is associated with the supplied adversary?” In Focus can return the malware entities, but not the documents in which the malware is reported.

Entity Types

Selects the types of entities that should be returned from the document set created by the query. Multiple types can be chosen.

Count Per Type

How many entities of each type should be returned. In this case, up to 10 malware entities will be included in the result set.

For more information on In Focus, please see the corresponding documentation.

Heat

The **Heat** endpoint is similar to the **In Focus** endpoint, in that it will return entities rather than documents. However, while the latter is intended to return smaller amounts of entities that are sorted by relevance, Heat can return up to 20,000 entities, sorted by two metrics: volume and heat. Options are available to limit the number of entities returned, and to discard entities if they have failed to receive a hit in the past 30 days.

Query

The app is performing the same search as that in the previous examples.

Term Query

The `Term Query` field includes a second search; like the In Focus endpoint, this specifies which entities should be returned from the document set produced by the `Query`. This term query will return entities of type ipv4 or hash.

Using Lists

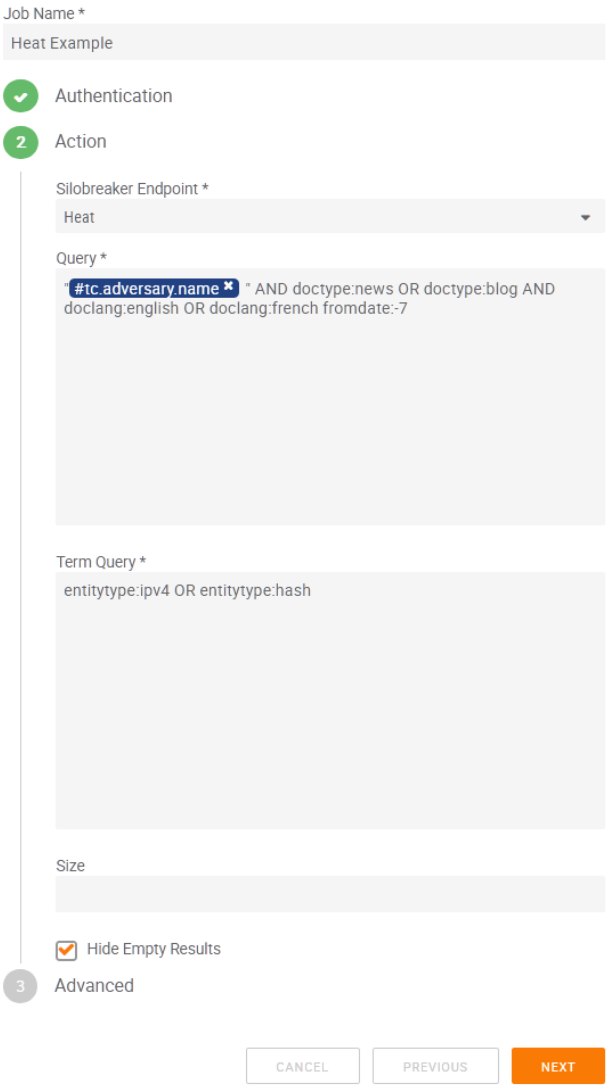
The `Term Query` field also supports the use of lists. Lists are customisable selections of entities created within the Silobreaker platform.

In this case, they would be useful if we wanted to know whether there was a relationship between our tc adversary and a known set of IPs or hashes.

You can find out more about lists in the lists documentation.

Size

How many entities will be returned. The default and maximum, visible here as an empty selection, is 20,000.



Hide Empty Results

Selecting this option means that only entities mentioned in documents published within the last 30 days will be included in the results.

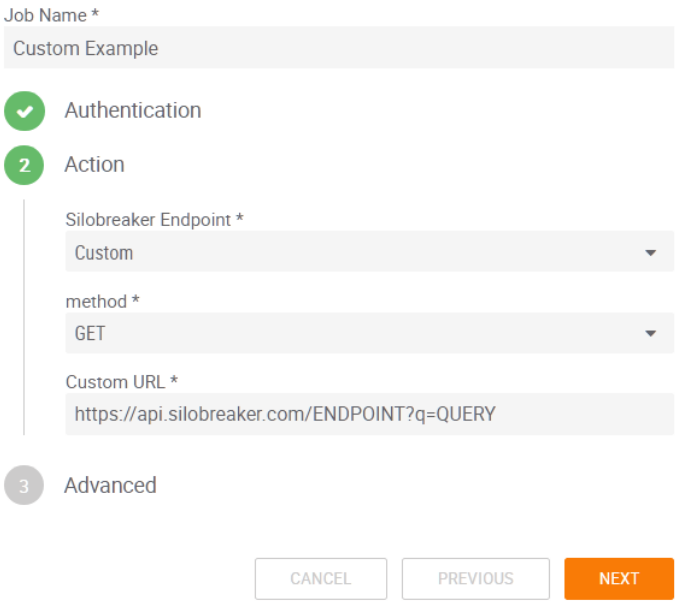
Heat results are based on two metrics, Volume and Heat, both of which are visible in the result JSON.

- **Volume** corresponds to how many distinct documents contain the entity over 1 day or 7 days.
- **Heat** corresponds to the "unusualness" of the entity within the document set from which it originates – also over a 1 day or 7-day period. The Heat metric is generated with reference to the average mentions an entity receives over a 30-day period.

For more information on Heat, see the corresponding documentation.

Custom

The Custom endpoint supports requirements that can't be met using Documents, In Focus or Heat. Any Silobreaker API URL can be supplied to the `Custom URL` field, allowing results to be retrieved from other Silobreaker endpoints.



Method

The method by which the query should be sent. Some endpoints allow both GET and POST methods.

Custom URL

The URL of the endpoint, including the query and any parameters for GET requests. An example request with placeholders has been used to show how a GET query to a Silobreaker endpoint might be constructed.

Note that this form will change depending on the endpoint and request method.

Body

Not shown here, the `Body` field should contain the request body for POST requests.

For more information on available endpoints, see the Silobreaker API documentation.

Requirements

The requirements for using this app are:

- A Silobreaker account
- A Silobreaker API key & shared key, supplied by a Silobreaker account manager

Installation

For installation instructions, refer to the ThreatConnect System Administration Guide (Install an App). For more information, contact your ThreatConnect Customer Success representatives.

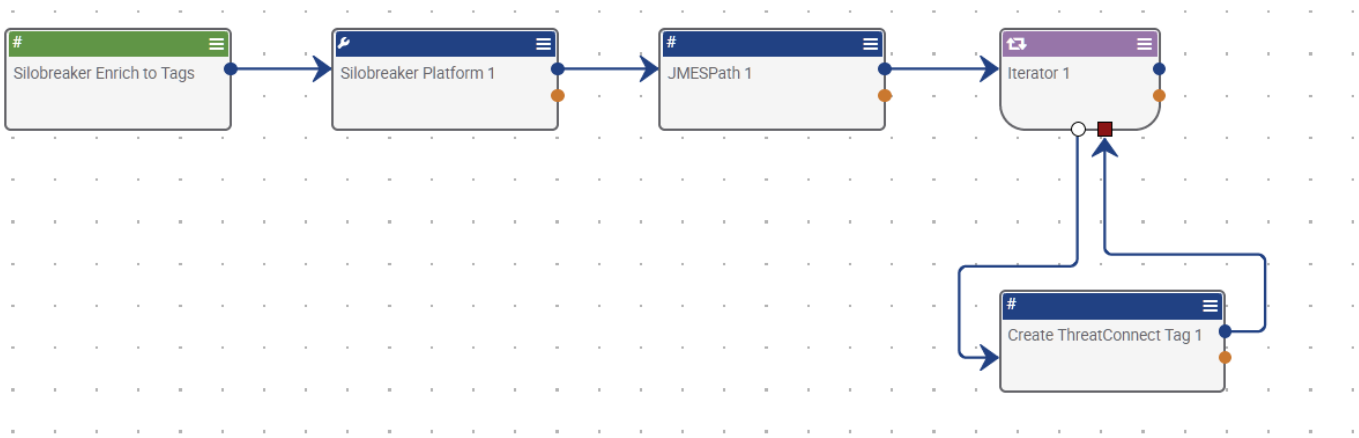
Practical Examples

Although many different workflows can be accomplished with the app, the quick and effective enrichment of ThreatConnect objects such as Groups and Indicators is a common use-case. This example will illustrate a few ways of addressing this use-case.

Indicator Enrichment with Tags

We can use Silobreaker entities to add additional context to ThreatConnect indicators by using tags.

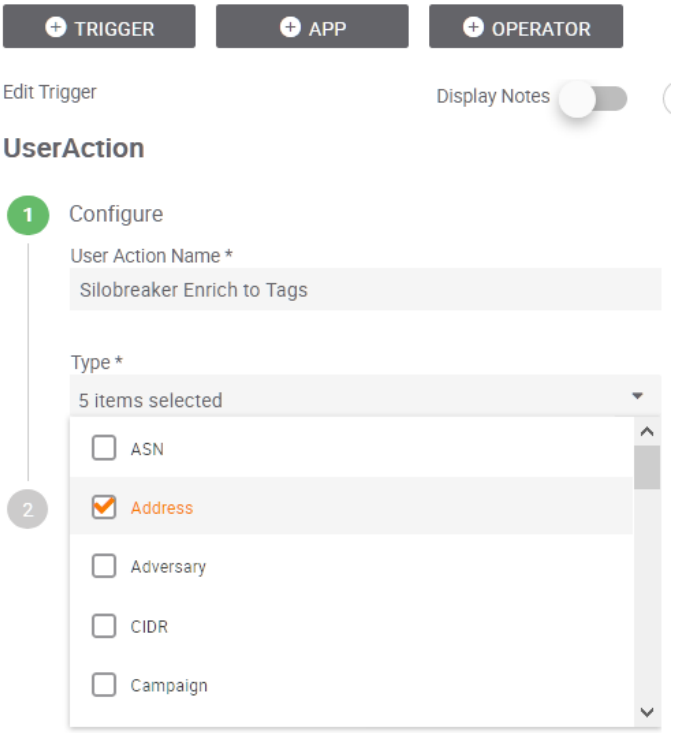
Using the playbook below, we can search for an indicator in Silobreaker and request entities back using the **In Focus** endpoint, then parse the JSON output to get the entity names, iterate through names to transform them into tags and finally attach those tags to the indicator we started with.

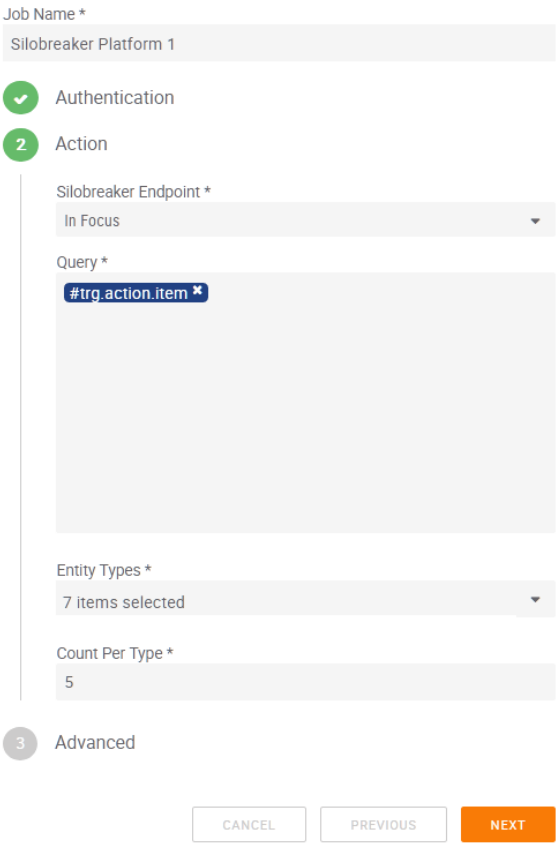


Playbook Configuration

1. TRIGGER: Silobreaker Enrich to Tags

- The UserAction trigger will allow the playbook to run on any of the selected indicators or groups.
- A number of indicator types are selected as options for the trigger action. EmailAddress, Address and URL are some of these.
- No response body has been added.





2. APP: Silobreaker Platform 1

- The `Authentication` fields have been filled in with an API key and shared key.
- The `In Focus` endpoint has been selected for the action.
- The `Query` is set to the tc item on which the trigger will run.
- The following seven `Entity Types` have been chosen: Attacktype, Threatactor, Vulnerability, Malware, Hash, Ipv4, Country. These are the types that can be returned by the app for this query.
- With the `Count Per Type` set at 5, up to five entities of each type can be returned, for a maximum total of 35 entities.

```
{
"ExecuteTimeMs": 24.349199295043945,
"Items": [
{
"EntityReference": "spear-phishing-11_130863708",
"DocumentFrequency": 96541,
"Id": "11_130863708",
"Description": "Spear Phishing",
"Type": "AttackType",
"LocalizedType": "AttackType",
"Relevance": 1.52105546
},
{
"EntityReference": "phishing-11_2278266",
"DocumentFrequency": 131419,
"Id": "11_2278266",
"Description": "Phishing",
"Type": "AttackType",
"LocalizedType": "AttackType",
"Relevance": 0.042234093
},
{
"EntityReference": "azorult-stealer-11_1297240144",
"DocumentFrequency": 4722,
"Id": "11_1297240144",
"Description": "AZORult Stealer",
"Type": "Malware",
"LocalizedType": "Malware",
"Relevance": 39.5533028
},
{
"EntityReference": "153920100-11_1520864045",
"DocumentFrequency": 538,
"Id": "11_1520864045",
"Description": "153.92.0.100",
"Type": "IPv4",
"LocalizedType": "IPv4",
"Relevance": 74.7611847
},
```


2.5 JSON response (not in diagram)

- This is a partial example of the JSON response from our app query. You can view this by looking at the app's output after it has run.

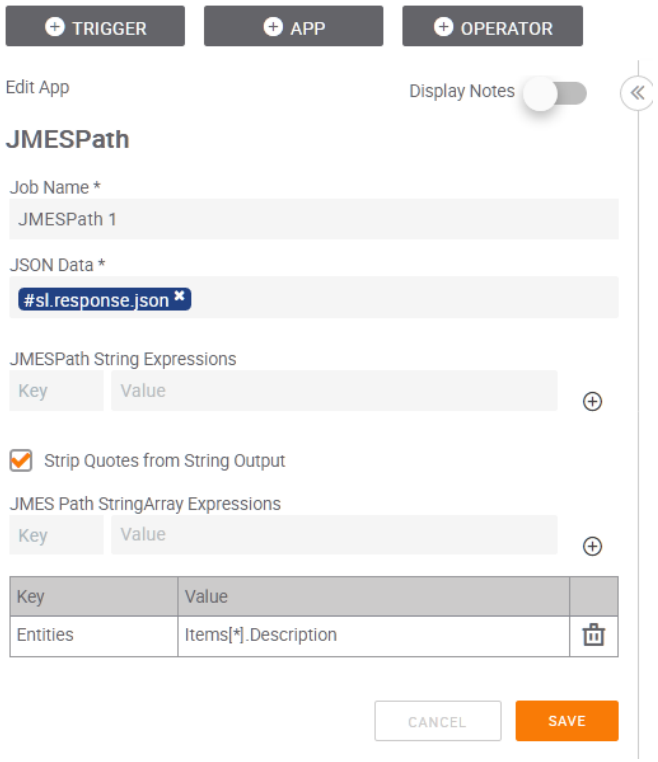
- This response contains a selection of the entities returned by the app.

- Note that the response structure is specific to the **In Focus** endpoint we are using. Some sample responses for other endpoints are included in the appendix to aid with structuring expressions.

- A good field to use for creating tags is the Description, which is the name of an entity in Silobreaker. Using the JMESPATH app, we will now extract the Description from the JSON array.

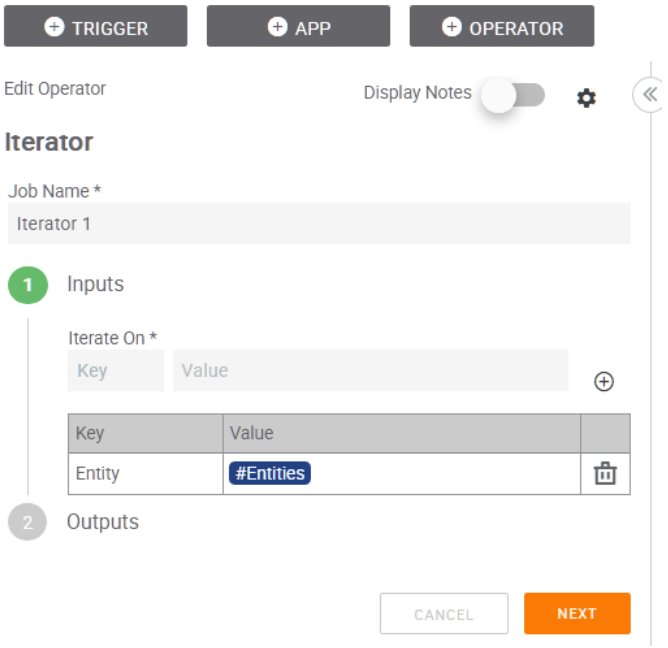
3. APP: JMESPath 1

- We now add the response to the JMESPath app as an input.
- Our JSON is a StringArray, from which we want descriptions. We will therefore ask for all Descriptions as an array, using the expression `Items[*].Description`
- We will label our key `Entities` and use these results in downstream apps.



4. OPERATOR: Iterator

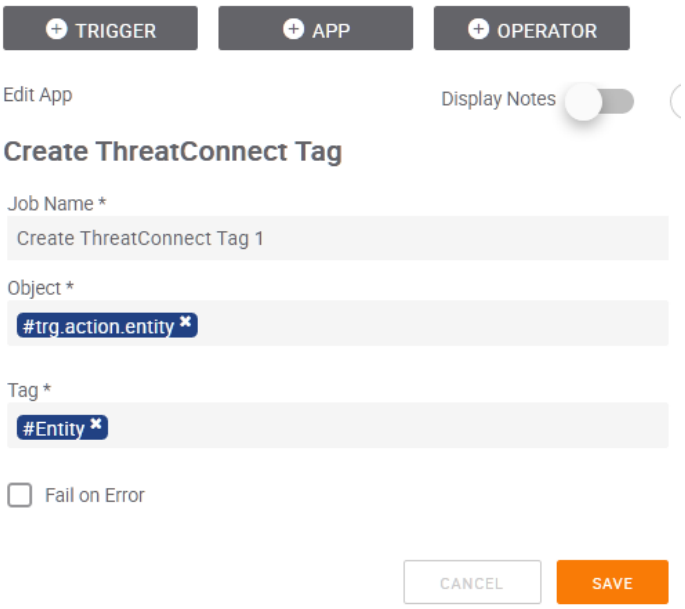
- We use the iterator to extract each description from the `#Entities` in the previous App.
- We'll define a new key, `Entity`, for this iterated value, and pass it to the next app, which will create the Tags.



5. APP: Create ThreatConnect Tag

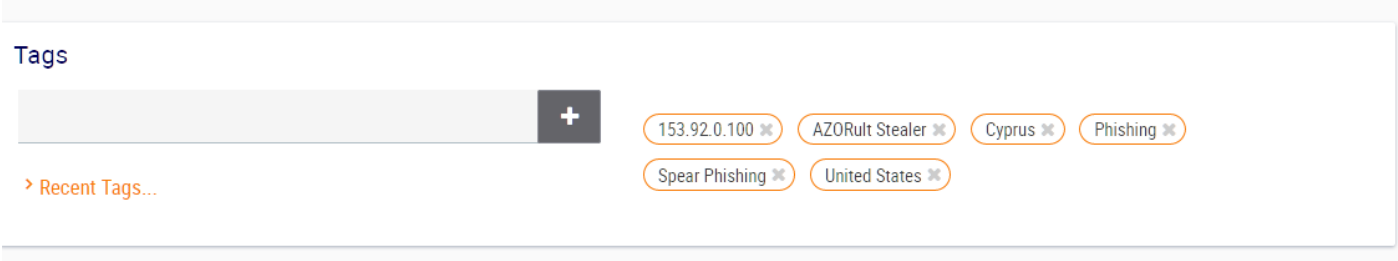
- The object we want to apply our tags to is the same one we triggered this playbook on, so it's set to `#trg.action.entity`

- The content of the tag will be the key `#Entity`, will contain each description from our JSON result.



6. Results

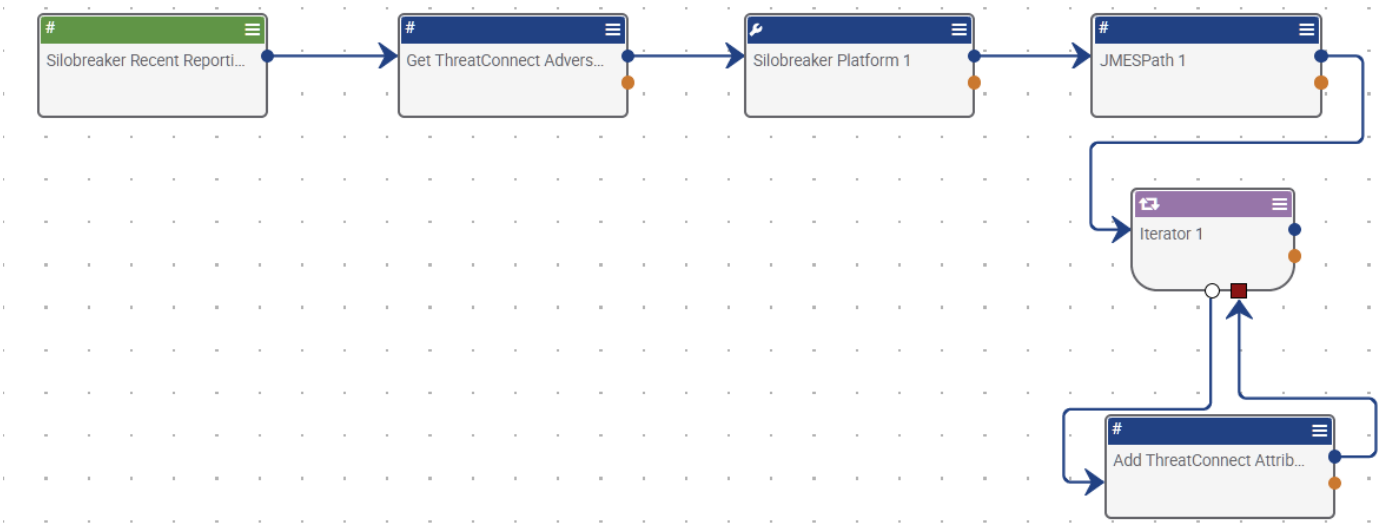
The tags below have been applied to our indicator. They include an IP address, two attack types, a known malware and two locations associated with the malicious activity.



Add new reporting to adversary

We can add Silobreaker documents from sources, including news, blogs and forums to ThreatConnect adversaries, giving us an insight into recent activity.

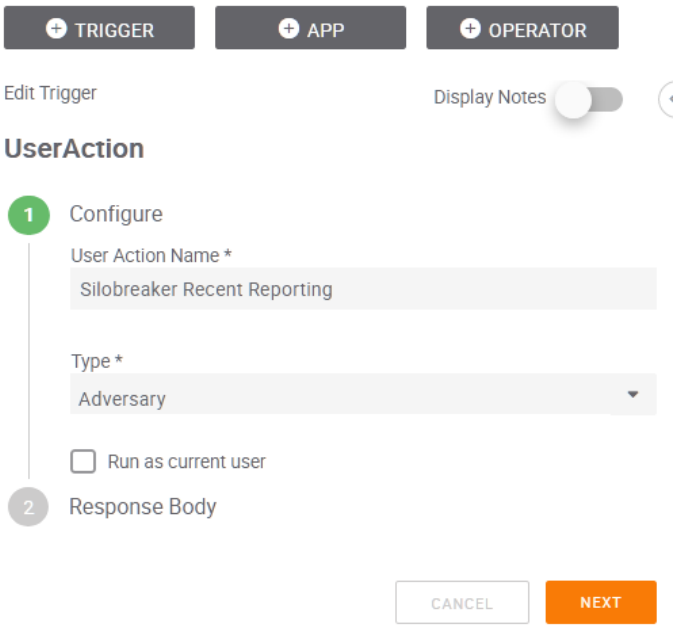
Using the playbook below, we can search for the name of an ThreatConnect adversary in Silobreaker and get reporting back using the `Documents` endpoint. We can extract the useful fields from these results, such as the title, publisher and URL, and then add them as context to the adversary we started with.



Playbook Configuration

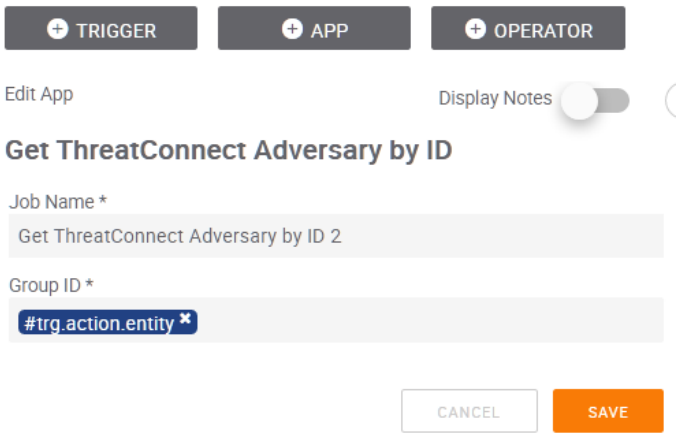
1. TRIGGER: Silobreaker Recent Reporting

- The UserAction trigger will allow the playbook to run on any of the selected indicators or groups.
- The `Adversary` type has been chosen for the trigger action.
- No response body has been added.



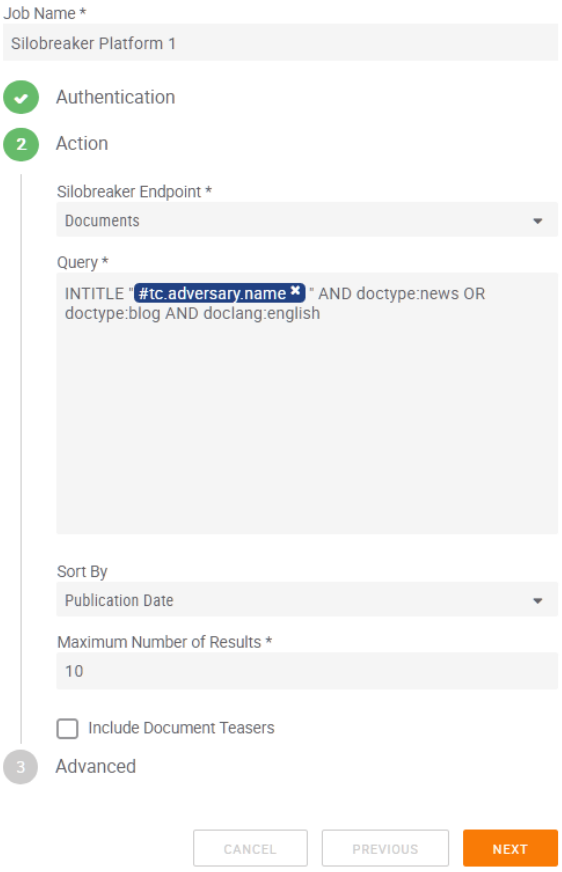
2. APP: Get ThreatConnect Adversary by ID

- We want to search the name of the adversary to Silobreaker, so we will extract it from the ID sent by the trigger.
- The downstream output will be the adversary name.



3. APP: Silobreaker Platform 1

- The `Authentication` fields have been filled in with an API key and shared key.
- The `Documents` endpoint has been selected for the action.
- The `Query` will search for the adversary name in the title of documents from the publishers of news or blogs, written in English.
- The syntax used in the query can be found in the Silobreaker documentation.
- The `Sort By` field is set to Publication Date, returning documents in order of recency.
- The `Maximum Number of Results` is set to 10, so only 10 documents will be returned from the app.



```
{
    "ResultCount": 100,
    "TotalCount": 164,
    "Description": "INTITLE Turla AND doctype:news OR doctype:blog AND
doclang:english",
    "Items": [
        {
            "FirstReported": "2020-12-07T19:24:00Z",
            "NoDocuments": 1,
            "PublicationDate": "2020-12-07T19:24:00Z",
            "CreatedDate": "2020-12-07T19:26:08.243Z",
            "Publisher": "Cyware",
            "PublisherType": "Public",
            "PublisherId": "4_2956147",
            "Language": "en",
            "SourceUrl": "https://cyware.com/news/turla-apt-active-again-with-
crutch-malware-toolkit-d10dd57e",
            "IndexOrder": 2265095888021672400,
            "ContributingUserId": 0,
            "SilobreakerUrl": "https://my.silobreaker.com/5_2275178961354883128",
            "Id": "5_2275178961354883128",
            "Description": "Turla APT Active Again with Crutch Malware Toolkit",
            "Type": "News",
            "LocalizedType": "News",
            "LastUpdated": "2020-12-07T19:24:00Z"
        },
```


3.5 JSON response (not in diagram)

- This is a partial example of the JSON response from our app query. You can view this by looking at the app's output after it has run.

- This response contains a selection of the documents returned by the app.

- Note that the response structure is specific to the In Focus endpoint we are using. Some sample responses for other endpoints are included in the appendix to aid with structuring expressions.

- The field we will be extracting from the JSON are:

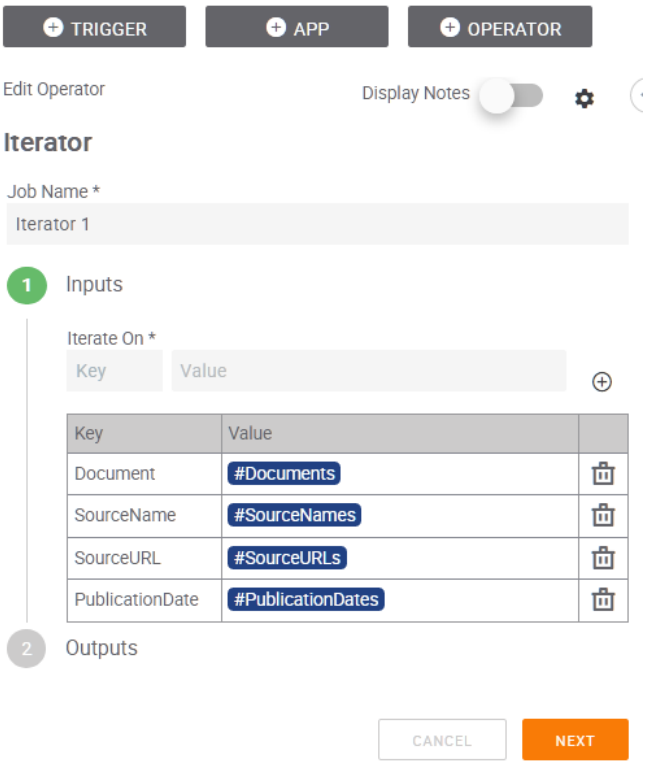
    - Description – the title of the document
    - SourceUrl – the url for the reporting source
    - Publisher – the name of the source
    - PublicationDate – the date and time when the document was published

4. APP: JMESPATH 1

- We now add the response to the JMESPath app as an input.
- Our JSON is a StringArray, from which we want descriptions. We will therefore ask for all the previously mentioned fields as an array, using the expressions:
    - Items[*].Description,
    - Items[*].Publisher
    - Items[*].SourceUrl,
    - Items[*].PublicationDate.
- The order in which these are extracted does not matter; they can be reordered in the final step when we add them to our adversary.
- We will create some appropriately named keys for these values, so we can use them downstream.

5. OPERATOR: Iterator

- We use the iterator to extract each result from the keys in the previous app.
- We'll define a new set of keys for these values, and pass them to the next app, which will add them as attributes to our adversary.
- The order of iteration does not matter.



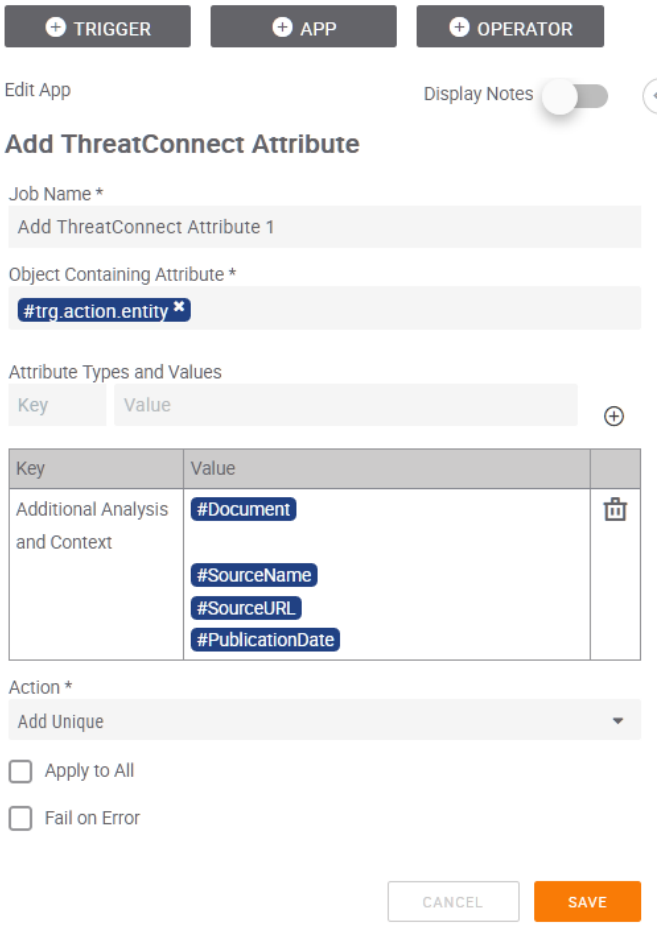
6. APP: Add ThreatConnect Attribute

- The final step is to apply the results to our adversary as attributes – the object to contain the attributes will be the entity on which the playbook was triggered `#trg.action.entity`

- The results will be applied as attributes in the category specified by the key; in this case, `Additional Analysis and Context`. Custom attribute categories can also be created within the ThreatConnect platform.

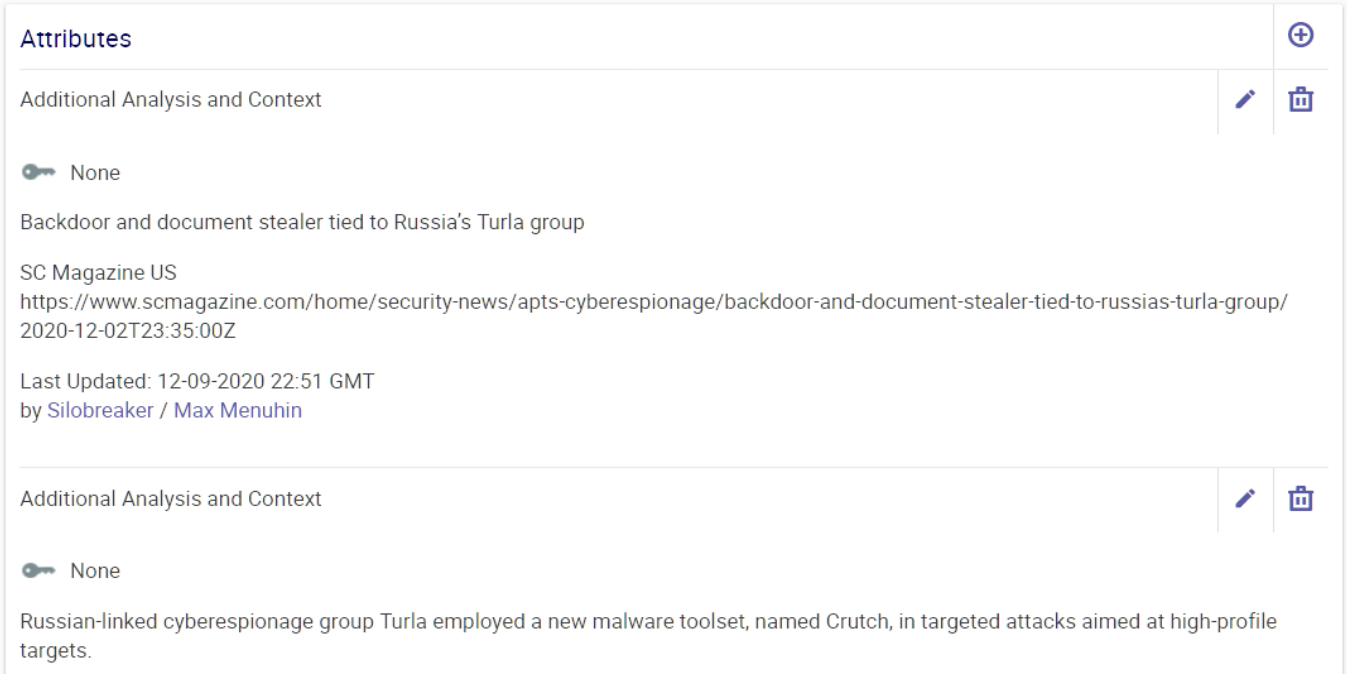
- The values are the keys from the previous step and will be applied in the order they are listed. The additional line break will also be reflected in the attribute text.

- The Action is set to `Add Unique`. This ensures that duplicate results created by running the app more than once will be ignored.



7. Results

The reporting below has been added to our adversary, Turla. Each attribute includes the name of the document, the publisher, the source URL and the time the document was published.



Troubleshooting

This section addresses some possible issues with the Silobreaker Platform app.

1. **When should I use double quotes for input data?**

   Silobreaker uses double straight quotes ("") to indicate that the input should be searched literally, and treated as a single search term. In situations where a search term includes a space, you should quote it in the query section of the app. Spaces are common in the names of adversaries and malware, such as `"Lazarus Group"` or `"Phoenix Keylogger"`. If you are supply the search term as a variable, you should quote the variable e.g. `"#tc.adversary.name "`

   When searching a single word or unbroken string, such an indicator, IP address or domain, these quotes are not usually necessary.

2. **I am seeing too many false positives in my results, how can I reduce them?**

While false positives are always a risk when dealing with OSINT, there are a number of ways to address them in the app.

- Use INTITLE

  By default, Silobreaker searches the title and body of every document for your search terms. However, prefixing your search terms with INTITLE will result in those terms being searched only in the title of documents. For example, `INTITLE "#tc.adversary.name " AND INTITLE entitytype:malware` will only find those documents that contain your adversary and any malware entity in the title.

- Use NEAR instead of AND

  AND will search the entire body of a document for the terms on either side of it. In contrast, NEAR will check whether the search terms provided are within approximately two paragraphs of each other.

- Reduce the document set

  Some document sets are not appropriate for certain queries. You can use **doctype** to restrict results to a type of document, or even set up a list of approved sources in your Silobreaker account, for use in the app. For guidance on this or other noise-reduction techniques, please contact support.

3. **Why am I seeing fewer results that expected?**

The more restrictive the query you are using, the fewer results you will see. If your app is intended to look up indicators, the best query will usually be the indicator itself with no additional filters e.g. `#trg.action.item`

Silobreaker has an entity system that attempts to find results for a given object in different languages, and using different aliasing. This entity system is not fully supported in the app, but tweaks to your playbooks may be possible if you contact support.

4. **I am seeing no results at all / I am receiving an error message**

   Silobreaker is not a TIP - you may not see results for indicator look-ups that you might expect to see in a TIP that consumes many indicator feeds.

   If you cannot find information that is commonly available on the open web, that you believe should be available through Silobreaker, or if you experience an error, please report this via our support email.

Support

Please convey any questions and support requests to your Silobreaker account manager, or contact premium-support@silobreaker.com

Appendix

Silobreaker's API documentation includes examples for all the endpoints included in this guide.

To aid in creating JMESPATH expressions, JSON responses for the app's main endpoints are included here.

Documents


```
{
  "ResultCount": 10,
  "TotalCount": 82,
  "Description": "intitle threatactor:turla* AND doclang:english AND doctype:news",
  "Items": [
    {
      "FirstReported": "2021-01-12T13:04:00Z",
      "NoDocuments": 1,
      "PublicationDate": "2021-01-12T13:04:00Z",
      "CreatedDate": "2021-01-12T13:07:23.853Z",
      "Publisher": "IT Security Guru",
      "PublisherType": "Public",
      "PublisherId": "4_102737",
      "Language": "en",
      "SourceUrl": "https://www.itsecurityguru.org/2021/01/12/potential-link-between-
solarwinds-and-turla-apt/?utm_source=rss&utm_medium=rss&utm_campaign=potential-link-
between-solarwinds-and-turla-apt",
      "IndexOrder": 2265095888044674800,
      "ContributingUserId": 0,
      "SilobreakerUrl": "https://my.silobreaker.com/5_2275541366400352552",
      "Provider": "Moreover10",
      "Id": "5_2275541366400352552",
      "Description": "Potential Link between SolarWinds and Turla APT",
      "Type": "News",
      "LocalizedType": "News",
      "LastUpdated": "2021-01-12T13:04:00Z"
    }
  ],
  "Source": "InMem"
}
```

In Focus


```
{
  "ExecuteTimeMs": 7.987299919128418,
  "Items": [
    {
      "EntityReference": "turla-apt-group-11_903805383",
      "DocumentFrequency": 1645,
      "Id": "11_903805383",
      "Description": "Turla APT Group",
      "Type": "ThreatActor",
      "LocalizedType": "ThreatActor",
      "Relevance": 13.625845
    }
  ],
  "Source": "InMem"
}
```


Heat


```
{
  "TotalCount": 288,
  "Description": "entitytype:threatactor",
  "Items": [
    {
      "EntityReference": "hafnium-group-11_866501475",
      "DocumentFrequency": 152,
      "Id": "11_866501475",
      "Description": "Hafnium Group",
      "Type": "ThreatActor",
      "LocalizedType": "ThreatActor",
      "Extras": {
        "DocVolume-1": {
          "Volume": 391
        },
        "DocVolume-7": {
          "Volume": 394
        },
        "DocVolume-30": {
          "Volume": 398
        },
        "Heat-1/30": {
          "Heat": 22.876447876447877
        },
        "Heat-7/30": {
          "Heat": 3.4914506343077774
        }
      }
    }
  ],
  "Source": "InMem"
}
```