Netskope Source

The Netskope Source provides a secure endpoint to receive event data from the Netskope API. It securely stores the required authentication, scheduling, and state tracking information.

The following event types are collected:

- alert events from /alerts
- application
- audit
- infrastructure
- page

States

A Netskope Source tracks errors, reports its health, and start-up progress. You’re informed, in real-time, if the Source is having trouble connecting, if there’s an error requiring user action, or if it is healthy and collecting by utilizing Health Events.

A Netskope Source goes through the following states when created:

1. **Pending**: Once the Source is submitted it is validated, stored, and placed in a Pending state.
2. **Started**: A collection task is created on the Hosted Collector.
3. **Initialized**: The task configuration is complete in Sumo Logic.
4. **Authenticated**: The Source successfully authenticated with Netskope.
5. **Collecting**: The Source is actively collecting data from Netskope.

If the Source has any issues during any one of these states it is placed in an Error state.

When you delete the Source it is placed in a Stopping state, when it has successfully stopped it is deleted from your Hosted Collector.

On the Collection page, the Health and Status for Sources is displayed. Use Health Events to investigate issues with collection.

Hover your mouse over the status icon to view a tooltip with details on the detected issue.
Getting a token from Netskope Portal

Netskope REST APIs use an auth token to make authorized calls to the API. This section demonstrates how to obtain a token from the Netskope user interface (UI).

To obtain a Netskope auth token, do the following:

1. Login to Netskope as the Tenant Admin.
2. Go to the API portion of the Netskope, Settings > Tools > Rest API.
3. Copy the existing token to your clipboard, or you can generate a new token and copy that token.

Create a Netskope Source

When you create a Netskope Source, you add it to a Hosted Collector. Before creating the Source, identify the Hosted Collector you want to use or create a new Hosted Collector. For instructions, see Configure a Hosted Collector.

To configure a Netskope Source:

1. In the Sumo Logic web app, select Manage Data > Collection > Collection.
2. On the Collectors page, click Add Source next to a Hosted Collector.
3. Select Netskope.

4. Enter a Name for the Source. The description is optional.
5. (Optional) For **Source Category**, enter any string to tag the output collected from the Source. Category metadata is stored in a searchable field called `_sourceCategory`.

6. **Forward to SIEM**. Check the checkbox to forward your data to Cloud SIEM Enterprise. When configured with the **Forward to SIEM** option the following metadata fields are set:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_siemVendor</td>
<td>Netskope</td>
</tr>
<tr>
<td>_siemProduct</td>
<td>Security Cloud</td>
</tr>
<tr>
<td>_siemFormat</td>
<td>JSON</td>
</tr>
<tr>
<td>_siemEventID</td>
<td><code>&lt;eventType&gt;</code></td>
</tr>
</tbody>
</table>

Where `eventType` is one of the above event types with one exception. If the `eventType` contains logon/login or logoff/logout the `eventType` field will be the `eventType` value `-logon` or `-logoff` added respectively, such as, audit-logon or audit-logoff.

7. **Fields**. Click the **+Add** link to add custom log metadata **Fields**.
Define the fields you want to associate, each field needs a name (key) and value.

- A green circle with a check mark is shown when the field exists and is enabled in the Fields table schema.
- An orange triangle with an exclamation point is shown when the field doesn’t exist, or is disabled, in the Fields table schema. In this case, an option to automatically add or enable the nonexistent fields to the Fields table schema is provided. If a field is sent to Sumo that does not exist in the Fields schema or is disabled it is ignored, known as dropped.

8. Enter your Netskope customer specific Tenant ID, such as, https://<my-tenant-id>.goskope.com.

9. Enter the Netskope API Token you want to use to authenticate requests.

10. When you are finished configuring the Source click Submit.

Error types

When Sumo Logic detects an issue it is tracked by Health Events. The following table shows the three possible error types, the reason the error would occur, if the Source attempts to retry, and the name of the event log in the Health Event Index.

<table>
<thead>
<tr>
<th>Type</th>
<th>Reason</th>
<th>Retries</th>
<th>Retry Behavior</th>
<th>Health Event Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThirdPartyConfig</td>
<td>Normally due to an invalid configuration. You'll need to review your</td>
<td>No retries are attempted until the Source is updated.</td>
<td>Not applicable</td>
<td>ThirdPartyConfigError</td>
</tr>
<tr>
<td>ThirdPartyGeneric</td>
<td>Normally due to an error communicating with the third party service API.</td>
<td>Yes</td>
<td>The Source will retry for up to 90 minutes, after which retries will be attempted every 60 minutes.</td>
<td>ThirdPartyGenericError</td>
</tr>
<tr>
<td>FirstPartyGeneric</td>
<td>Normally due to an error communicating with the internal Sumo Logic API.</td>
<td>Yes</td>
<td>The Source will retry for up to 90 minutes, after which retries will be attempted every 60 minutes.</td>
<td>FirstPartyGenericError</td>
</tr>
</tbody>
</table>
JSON configuration

Sources can be configured using UTF-8 encoded JSON files with the Collector Management API. See how to use JSON to configure Sources for details.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Required?</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>JSON Object</td>
<td>Yes</td>
<td>Contains the configuration parameters for the Source.</td>
<td></td>
</tr>
<tr>
<td>schemaRef</td>
<td>JSON Object</td>
<td>Yes</td>
<td>Use {&quot;type&quot;:&quot;Netskope&quot;} for a Netskope Source.</td>
<td>not modifiable</td>
</tr>
<tr>
<td>sourceType</td>
<td>String</td>
<td>Yes</td>
<td>Use Universal for a Netskope Source.</td>
<td>not modifiable</td>
</tr>
</tbody>
</table>

The following table shows the config parameters for a Netskope Source.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Required?</th>
<th>Default</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Yes</td>
<td>null</td>
<td>Type a desired name of the Source. The name must be unique per Collector. This value is assigned to the metadata field _source.</td>
<td>modifiable</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>No</td>
<td>null</td>
<td>Type a description of the Source.</td>
<td>modifiable</td>
</tr>
<tr>
<td>category</td>
<td>String</td>
<td>No</td>
<td>null</td>
<td>Type a category of the source. This value is assigned to the metadata field _sourceCategory. See best practices for details.</td>
<td>modifiable</td>
</tr>
<tr>
<td>fields</td>
<td>JSON Object</td>
<td>No</td>
<td></td>
<td>JSON map of key-value fields (metadata) to apply to the Collector or Source.</td>
<td>modifiable</td>
</tr>
<tr>
<td>tenantID</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>Netskope customer specific Tenant ID, such as, https://&lt;my-tenant-id&gt;.goskope.com.</td>
<td>modifiable</td>
</tr>
<tr>
<td>apiToken</td>
<td>String</td>
<td>Yes</td>
<td></td>
<td>The Netskope API Token you want to use to authenticate requests.</td>
<td>modifiable</td>
</tr>
</tbody>
</table>

Netskope Source JSON example:

```json
{
  "api.version":"v1",
  "source":{
    "schemaRef":{
      "type":"Netskope"
```


```json
{
  "config": {
    "name": "Netskope",
    "description": "East field",
    "apiToken": "********",
    "tenantID": "https://id.goskope.com",
    "fields": {
      "_siemForward": false
    },
    "category": "eastTeamF"
  },
  "sourceType": "Universal"
}
```