ThreatQuotient

A Securonix Company

Intel 471 Alerts CDF

Version 2.0.0

June 23, 2026

ThreatQuotient
20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

ThreatQ Supported

Support
Email: tq-support@securonix.com
Web: https://ts.securonix.com
Phone: 703.574.9893

Contents

Warning and Disclaimer

ThreatQuotient, Inc. provides this document “as is”, without representation or warranty of any kind, express or implied, including without limitation any warranty concerning the accuracy, adequacy, or completeness of such information contained herein. ThreatQuotient, Inc. does not assume responsibility for the use or inability to use the software product as a result of providing this information.

Support

This integration is designated as **ThreatQ Supported**.

Support Email: tq-support@securonix.com
Support Web: https://ts.securonix.com
Support Phone: 703.574.9893

Integrations/apps/add-ons designated as **ThreatQ Supported** are fully supported by ThreatQuotient's Customer Support team.

ThreatQuotient strives to ensure all ThreatQ Supported integrations will work with the current version of ThreatQuotient software at the time of initial publishing. This applies for both Hosted instance and Non-Hosted instance customers.

> ⚠ ThreatQuotient does not provide support or maintenance for integrations, apps, or add-ons published by any party other than ThreatQuotient, including third-party developers.

Integration Details

ThreatQuotient provides the following details for this integration:

Current Integration Version	2.0.0
Compatible with ThreatQ Versions	>= 5.29.4
Support Tier	ThreatQ Supported

Introduction

The Intel 471 Alerts CDF enables ThreatQ users to ingest watcher alerts and related context from the Intel 471 platform as Alert events. The integration retrieves alert metadata, source information, status details, watcher identifiers, and highlighted content associated with monitored activity. To provide additional context, the integration includes a supplemental feed that resolves watcher group IDs into human-readable watcher group names, which can be applied as event attributes and tags within ThreatQ.

The integration includes the following feeds:

- **Intel 471 Alerts** - returns a list of Alerts and related information.
    - **Intel471 Get Watcher Group Name (supplemental) -** returns threat data using the `.alerts[].watcherGroupUid` from the Intel471 Alerts feed as the groupId parameter.

The integration ingests the event type system objects.

Prerequisites

The integrations requires the following:

- Intel 471 Client ID and Client Secret.

Installation

Perform the following steps to install the integration:

> The same steps can be used to upgrade the integration to a new version.

1. Log into https://marketplace.threatq.com/.
2. Locate and download the integration file.
3. Navigate to the integrations management page on your ThreatQ instance.
4. Click on the **Add New Integration** button.
5. Upload the integration file using one of the following methods:
   - Drag and drop the file into the dialog box
   - Select **Click to Browse** to locate the integration file on your local machine

   > ThreatQ will inform you if the feed already exists on the platform and will require user confirmation before proceeding. ThreatQ will also inform you if the new version of the feed contains changes to the user configuration. The new user configurations will overwrite the existing ones for the feed and will require user confirmation before proceeding.
6. The feed will be added to the integrations page. You will still need to configure and then enable the feed.

Configuration

ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other integration-related credentials.

To configure the integration:

1. Navigate to your integrations management page in ThreatQ.
2. Select the **Commercial** option from the *Category* dropdown (optional).

   If you are installing the integration for the first time, it will be located under the **Disabled** tab.

3. Click on the integration entry to open its details page.
4. Enter the following parameters under the **Configuration** tab:

PARAMETER	DESCRIPTION
Client ID	Enter your Intel 471 Client ID.
Client Secret	Enter your Intel 471 Client Secret.
Enable SSL Certificate Verification	Enable this parameter if the feed should validate the host-provided SSL certificate.
Disable Proxies	Enable this parameter if the feed should not honor proxies set in the ThreatQ UI.
Count	The maximum number of records to retrieve from the provider per request. The value range is 0-1000. The default setting is 10.
Watcher Group Filter	Enter an optional line-separated list of watcher group UIDs to ingest. If no values are provided, alerts from all watcher groups will be ingested.

PARAMETER	DESCRIPTION
Watcher Filter	Optional - enter a comma-separated or line-separated list of specific watcher IDs to ingest.
Status Filter	Optional - filter alerts by their current state. Options include: ◦ Generated ◦ Needs Action ◦ In Progress ◦ Completed ◦ False Positive
Include Trashed Alerts	Enable this parameter to include alerts that have been marked as `trash`.

5. Review any additional settings, make any changes if needed, and click on **Save**.
6. Click on the toggle switch, located above the *Additional Information* section, to enable it.

ThreatQ Mapping

Intel 471 Alerts

The Intel 471 Alerts endpoint returns a list of Alerts and related information.

`GET - https://api.intel471.cloud/integrations/watchers/v1/alerts/stream`

Sample Response:


```
{
  "alerts": [
    {
      "creation_ts": "2026-06-16T14:22:10Z",
      "highlights": [
        {
          "field_name": "message",
          "snippets": [
            "sample highlighted watcher match"
          ]
        }
      ],
      "id": 1001,
      "is_trashed": false,
      "links": {
        "verity_api": {
          "href": "https://api.intel471.cloud/integrations/watchers/v1/alerts/1001"
        },
        "verity_portal": {
          "href": "https://titan.intel471.com/alerts/1001"
        }
      },
      "source_id": "post--88124",
      "source_type": "forum_post",
      "status": "generated",
      "watcher_group_id": 2272,
      "watcher_id": 9941
    },
    {
      "creation_ts": "2026-06-16T14:25:18Z",
```

```
      "highlights": [
        {
          "field_name": "message.text",
          "snippets": [
            "example.com mentioned in monitored chat channel"
          ]
        }
      ],
      "id": 1002,
      "is_trashed": false,
      "links": {
        "verity_api": {
          "href": "https://api.intel471.cloud/integrations/watchers/
v1/alerts/1002"
        },
        "verity_portal": {
          "href": "https://titan.intel471.com/alerts/1002"
        }
      },
      "source_id": "msg--9911",
      "source_type": "instant_message",
      "status": "needs_action",
      "watcher_group_id": 2272,
      "watcher_id": 9942
    }
  ],
  "count": 2,
  "cursor_next": "cursor--next-sample"
}
```


An `Alert` event will be created for each item in `.alerts[]`.

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
`.alerts[].id`	`Event.Title`	`N/A`	`.alerts[].creation_ts`	`Intel 471 Alert 1001`	`Built as Intel 471 Alert {{id}}`
`N/A`	`Event.Type`	`Alert`	`.alerts[].creation_ts`	`Alert`	`Hardcoded event type`
`.alerts[].source_type, .alerts[].id`	`Event.Description`	`N/A`	`.alerts[].creation_ts`	`Intel 471 watcher alert 1001 from forum_post`	`Built from alert ID and source type`
`.alerts[].creation_ts`	`Event.Happened At`	`N/A`	`.alerts[].creation_ts`	`2026-06-16T14:22:10Z`	`Used for event time and attribute published date`

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.alerts[].id	Event.Attribute	UID	.alerts[].creation_ts	1001	Unique Intel 471 alert identifier
.alerts[].status	Event.Attribute	Status	.alerts[].creation_ts	generated	Allowed values include generated, needs_action, in_progress, completed, false_positive
.alerts[].watcher_group_id	Event.Attribute	Watcher Group UID	.alerts[].creation_ts	2272	Watcher group name is added only when the optional supplemental lookup returns a name
.alerts[].watcher_id	Event.Attribute	Watcher ID	.alerts[].creation_ts	9941	N/A
.alerts[].source_type	Event.Attribute	Source Type	.alerts[].creation_ts	forum_post	Source type from Intel 471
.alerts[].source_id	Event.Attribute	Source ID	.alerts[].creation_ts	post--88124	Source object identifier from Intel 471
.alerts[].is_trashed	Event.Attribute	Is Trashed	.alerts[].creation_ts	false	Boolean
.alerts[].highlights[].field_name/snippets	Event.Attribute	Highlights	.alerts[].creation_ts	message: sample highlighted watcher match	Highlight objects are rendered as field_name: snippet; string highlights are passed through
.alerts[].links.verity_portal.href	Event.Attribute	Portal URL	.alerts[].creation_ts	https://titan.intel471.com/alerts/...	Optional in the API response
.alerts[].links.verity_api.href	Event.Attribute	API URL	.alerts[].creation_ts	https://api.intel471.cloud/...	N/A

Get Watcher Group Name (Supplemental)

The Get Watcher Group Name supplemental feed returns threat data using the `.alerts[].watcherGroupUid` from the Intel471 Alerts feed as the groupId parameter.

The value of `.alerts[].watcher_group_id` from the Intel 471 Alerts feed is normalized to `watcherGroupUid` and used as the `groupId` parameter.

`GET - https://api.intel471.cloud/integrations/watchers/v1/watcher-groups?watcher_group_id={groupId}`

Sample Response:

```
{
  "watchers_groups": [
    {
      "id": 2272,
      "name": "Corporate Domain Monitoring",
```

```
      "description": "This Intel 471 watcher group tracks configured
corporate domains.",
      "created_by": "Intel 471"
    }
  ]
}
```

ThreatQ provides the following default mapping for this feed:

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.watchers_groups[].name	tag.name	N/A	N/A	Corporate Domain Monitoring	Numeric watcher group IDs are mapped after using the data fetched from the Get Watcher Group Name supplemental feed. Each tag is trimmed to 50 chars
.watchers_groups[].name	event.attribute	Watcher Group Name	.alerts[].creation_ts	Corporate Domain Monitoring	Numeric watcher group IDs are mapped after using the data fetched from the Get Watcher Group Name supplemental feed

Average Feed Run

Object counts and Feed runtime are supplied as generalities only - objects returned by a provider can differ based on credential configurations and Feed runtime may vary based on system resources and load.

METRIC	RESULT
Run Time	2 minutes
Events	10
Event Attributes	45

Change Log

- **Version 2.0.0**
  - Updated the feed to utilize the latest Verity Intel 471 integration APIs for all HTTP-based data retrieval.
  - Added the `User-Agent: threatq-alerts-feed-2.0.0` header and enhanced authentication handling with explicit responses for HTTP 401 (Unauthorized) and 403 (Forbidden) errors.
  - Implemented comprehensive HTTP status code handling to improve error reporting and feed reliability.
  - Removed duplicate Breach Alerts ingestion functionality that was already available through existing integration capabilities.
  - Updated the authentication method to use Client ID and Secret.
  - The **Count** configuration parameter now supports up to 1000 records per request.
  - Added the following new configuration parameters:
    - **Enable SSL Certificate Verification**
    - **Disable Proxies**
    - **Watcher Filter**
    - **Status Filter**
    - **Include Trashed Alerts**
  - Removed the **Ingest CVEs** As configuration parameter.
  - Updated the minimum ThreatQ version to 5.29.4.
- **Version 1.2.5**
  - Added a new configuration parameter:
    - **Watcher Group Filter** - filter alert ingestion by watcher group UID.
- **Version 1.2.4**
  - Resolved an issue where users would encounter a `Cannot parse argument of type None` error message.
- **Version 1.2.3**
  - Resolved a filtering issue where users would encounter an `Error applying filter` message.

  - The Ingest CVEs As is now set to Vulnerabilities by default.
  - Resolved an issue where certain event attributes were not mapped correctly.
- **Version 1.2.2**
  - Resolved a parsing attribute issue for events.
- **Version 1.2.1**
  - Fixed a `Get Report by ID` supplemental feed indicator ingestion bug.
  - Added the ability to parse CVEs from CVE Report Alerts description.
- **Version 1.2.0**
  - Fixed an issue with Spot Reports Events when the event did not have relationships.
  - Fixed an indicator bug where the relationship between the report and indicator was not created if the indicator was ingested into the ThreatQ platform by another feed.
- **Version 1.1.0**
  - Updated the integration to ingest more data about events and related items.
  - Added new configuration option: `Ingest CVEs As` . See the Configuration chapter for more information.
- **Version 1.0.0**
  - Initial Release