ThreatQuotient

Intel 471 Compromised Credentials CDF

Version 1.1.1

May 06, 2025

ThreatQuotient
20130 Lakeview Center Plaza Suite 400
Ashburn, VA 20147

ThreatQ Supported

Support

Email: support@threatq.com
Web: support.threatq.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: support@threatq.com
Support Web: https://support.threatq.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	1.1.1
Compatible with ThreatQ Versions	>= 5.6.0
Support Tier	ThreatQ Supported

Introduction

The Intel 471 Compromised Accounts CDF integration for ThreatQ allows you to track your organization's compromised credentials in order to efficiently mitigate any threats targeting your organization's employees.

The integration provides the following feed:

- **Intel 471 Compromised Credentials -** imports compromised credentials from Intel 471's API in order to track internal credentials that have been compromised.

The integration ingests the following system objects:

- Compromised Accounts
- Indicators
- Malware

Prerequisites

The integration requires the following:

- Your Intel 471 Titan Email Address.
- Your Intel 471 Titan API Key.
- Compromised Account custom object installed on the ThreatQ instance.

Compromised Account Custom Object

The integration requires the Compromised Account custom object.

Use the steps provided to install the Compromised Account custom object.

> ⚠ When installing the custom objects, be aware that any in-progress feed runs will be cancelled, and the API will be in maintenance mode.

ThreatQ V6 Steps

Use the following steps to install the custom object in ThreatQ v6:

1. Download the integration bundle from the ThreatQ Marketplace.
2. Unzip the bundle and locate the custom object files.

   > The custom object files will typically consist of a JSON definition file, install.sh script, and a images folder containing the svg icons.

3. SSH into your ThreatQ instance.
4. Navigate to the tmp folder:

   ```
   cd /var/lib/threatq/misc/
   ```

5. Upload the custom object files, including the images folder.

   The directory structure should be as the following:

   - misc
     - install.sh
     - <custom_object_name>.json
     - images (directory)
       - <custom_object_name>.svg
6. Run the following command:

   ```
   kubectl exec -it deployment/api-schedule-run -n threatq -- sh /var/
   lib/threatq/misc/install.sh /var/lib/threatq/misc
   ```

> The installation script will automatically put the application into maintenance mode, move the files to their required directories, install the custom object, update permissions, bring the application out of maintenance mode, and restart dynamo.

7. Delete the install.sh, definition json file, and images directory from the `misc` directory after the object has been installed as these files are no longer needed.

ThreatQ v5 Steps

1. Download the integration zip file from the ThreatQ Marketplace and unzip its contents.
2. SSH into your ThreatQ instance.
3. Navigate to tmp directory:

```
cd /tmp/
```

4. Create a new directory:

```
mkdir intel471_cdf
```

5. Upload the **account.json** and **install.sh** script into this new directory.
6. Create a new directory called **images** within the inte471_cdf directory.

```
mkdir images
```

7. Upload the account.svg.
8. Navigate to the **/tmp/intel471_cdf**.

   The directory should resemble the following:

   - tmp
     - **intel471_cdf**
       - **account.json**
       - **install.sh**
       - **images**
         - **account.svg**

9. Run the following command to ensure that you have the proper permissions to install the custom object:

   ```
   chmod +x install.sh
   ```

10. Run the following command:

    ```
    sudo ./install.sh
    ```

    > You must be in the directory level that houses the install.sh and json files when running this command.

    The installation script will automatically put the application into maintenance mode, move the files to their required directories, install the custom object, update permissions, bring the application out of maintenance mode, and restart dynamo.

11. Remove the temporary directory, after the custom object has been installed, as the files are no longer needed:

    ```
    rm -rf intel471_cdf
    ```

Installation

> ⚠ The CDF requires the installation of the Compromised Account custom object before installing the actual CDF. See the Compromised Account section of this guide for more details. The custom object must be installed prior to installing the CDF. Attempting to install the CDF without the custom object will cause the CDF install process to fail.

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 zip file.
3. Extract the integration files and install the Compromised Account custom object if you have not done so already.
4. Navigate to the integrations management page on your ThreatQ instance.
5. Click on the **Add New Integration** button.
6. Upload the integration yaml file using one of the following methods:
    - Drag and drop the file into the dialog box
    - Select **Click to Browse** to locate the file on your local machine
7. Select the individual feeds to install, when prompted and click **Install**.

    > 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.

The feed(s) 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:

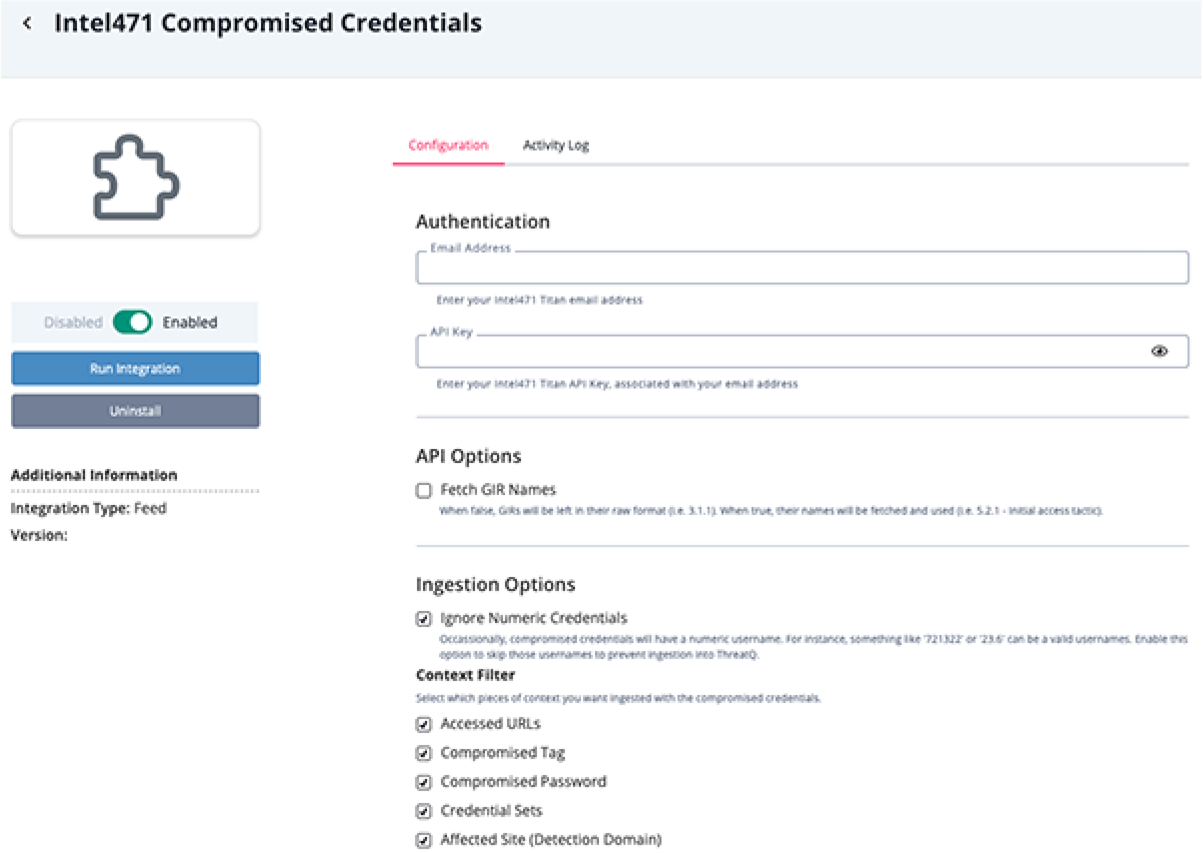
1. Navigate to your integrations management page in ThreatQ.
2. Select the **Commercial** option from the *Category* dropdown (optional).

   > 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
Email Address	Enter your Intel471 Titan email address.
API Key	Enter your Intel471 Titan API Key associated with the email address provided above.
Fetch Accessed URLs	Enable this option to have the feed fetch the accessed URLs for each compromised credential. This parameter is disable by default. ⚠ Enabling this parameter will result in one API call per credential.
Fetch GIR Names	Enable this parameter to fetch the GIR's name (example: 5.2.1 - initial Access Tactic). Disable this parameter to fetch the GIR's raw format (example: 3.1.1). This parameter is enabled by default.
Ignore Numeric Credentials	Enable this parameter to skip ingestion of compromised credentials with usernames consist solely of numeric values such as `721322`. This parameter is disabled by default.
Context Filter	Select which pieces of context to ingest with the compromised credentials. Options include:

PARAMETER	DESCRIPTION
	◦ Accessed URLs (default) ◦ Compromised Tag (default) ◦ Compromised Password ◦ Credential Sets (default) ◦ Affected Site (Detection Domain) (default) ◦ Credential Domain ◦ Affiliations (default) ◦ Password Strength ◦ Intel Requirements (default) ◦ Associated Adversary (default) ◦ Associated Malware Family (default) ◦ Intel471 Titan Link
Ingest Accessed URLs As	Select which entity type to ingest accessed URLs as. Options include: ◦ Indicators ◦ Attributes This parameter will only be accessible if you have selected the `Accessed URLs` option for the **Context Filter** parameter.
Accessed URL Status	Select the status that should be applied to the accessed URLs. The default status is `Review`. This parameter will only be accessible if you have selected the `Indicators` option for the **Ingest Accessed URLs As** parameter.
Disable Proxies	Enable this option if the feed should not honor proxies set in the ThreatQ UI.
Enable SSL Verification	Enable this parameter for the feed to validate the host-provided SSL certificate. This option is enabled by default.



Intel471 Compromised Credentials

Additional Information

Integration Type: Feed

Version:

Authentication

Email Address

Enter your Intel471 Titan email address

API Key

Enter your Intel471 Titan API Key, associated with your email address

API Options

☐ Fetch GIR Names

When false, GIRs will be left in their raw format (i.e. 3.1.1). When true, their names will be fetched and used (i.e. 5.2.1 - Initial access tactic).

Ingestion Options

☑ Ignore Numeric Credentials

Occasionally, compromised credentials will have a numeric username. For instance, something like '721322' or '23.6' can be a valid usernames. Enable this option to skip those usernames to prevent ingestion into ThreatQ.

Context Filter

Select which pieces of context you want ingested with the compromised credentials.

☑ Accessed URLs

☑ Compromised Tag

☑ Compromised Password

☑ Credential Sets

☑ Affected Site (Detection Domain)

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 Compromised Credentials

The Intel 471 Compromised Credentials feed imports compromised credentials from Intel 471's API in order to track internal credentials that have been compromised.

`GET https://api.intel471.com/v1/credentials/occurrences/stream`

Sample Response:

```
{
    "credential_occurrences_total_count": 123,
    "credential_occurrences": [
      {
        "uid": "c70ec3ffc8b095f685d52ec7ebcb3874",
        "data": {
          "file_path": "/Private Collection/Private combos/
result.txt_div_84_DupDel.txt",
          "accessed_url": "https://accounts.zoho.com/signin",
          "credential": {
            "uid": "08c9a1e7e811617a79290108cdc23b36",
            "credential_login": "john.smith@test-domain.com",
            "credential_domain": "test-domain.com",
            "detection_domain": "test-domain.com",
            "affiliations": [
              "my_employees"
            ],
            "password": {
              "complexity": {
                "lowercase": 15,
                "uppercase": 0,
                "numbers": 17,
                "symbols": 0,
                "punctuation_marks": 0,
                "separators": 0,
                "other": 0,
                "length": 32,
                "score": 0.9525726035123216,
                "weakness": 0.09375,
                "entropy": 121.83535750584332
              },
              "strength": "excellent",
              "id": "81105f09",
              "password_plain": "bad_pswrd"
            }
          },
          "credential_set": {
```

```
          "uid": "13951971fbce4bd11dc1eb13f04da669",
          "name": "Infostealer Collection"
        },
        "detected_malware": {
          "family": "[Raccoon Stealer] - v1.0 Golden Master Release"
        }
      },
      "classification": {
        "intel_requirements": [
          "2.1.1.1",
          "2.2.1",
          "2.2.2"
        ]
      },
      "last_updated": 1583241868411,
      "activity": {
        "first": 1569271060000,
        "last": 1569271060000
      }
    }
  ]
}
```

ThreatQuotient provides the following default mapping for this feed:

> The mapping for this feed is based on each item within the `.credential_occurrences` array.

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES	
`.data.credential.credential_login`	Compromised Account Value	N/A	N/A	`.activity.first`	N/A	N/A
`.data.detected_malware.family`	Malware Value	N/A	N/A	`.activity.first`	`Redline`	Configurable via user-field
`.data.credential_sets.name`	Adversary Name	N/A	N/A	`.activity.first`	N/A	Configurable via user-field; Extrapolated from credential sets if applicable
`.data.credential_sets.name`	Attribute	Credential Set	N/A	`.activity.first`	`General Infostealers`	Configurable via user-field
`.classification.intel_requirements[]`	Attribute	Intel Requirement	N/A	`.activity.first`	`5.2.6 - Credential access tactic`	Configurable via user-field
`.data.credential.affiliations[]`	Attribute	Affiliation	N/A	`.activity.first`	`my_customers`	Configurable via user-field
`.data.credential.detection_domain`	Attribute	Affected Site	N/A	`.activity.first`	`gmail.com`	Configurable via user-field
`.data.credential.detection_domain`	Attribute	Target Domain	N/A	`.activity.first`	`gmail.com`	Configurable via user-field; Applied to the related threat actors
`.data.credential.password.strength`	Attribute	Password Strength	N/A	`.activity.first`	`Weak`	Configurable via user-field
`.data.credential.credential_domain`	Attribute	Credential Domain	N/A	`.activity.first`	`company.com`	Configurable via user-field
`.data.credential.password.password_plain`	Attribute	Compromised Password	N/A	`.activity.first`	N/A	Configurable via user-field
`.data.file_path`	Attribute	File Path	N/A	`.activity.first`	N/A	Configurable via user-field
`.data.accessed_url`	Attribute	Accessed URL	N/A	`.activity.first`	N/A	Configurable via user-field; When user selects `Attributes` for the `Ingest Accessed URLs As` user-field
`.data.accessed_url`	Indicator Value	URL	N/A	`.activity.first`	N/A	Configurable via user-field; When user selects `Indicators` for

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES	
						the `Ingest Accessed URLs` As user-field

`.uid`	Attribute	Intel471 Titan Link	N/A	`.activity`
				`.first`
					N/A	Configurable via user-field; Concatenates the UID with the Intel471 Portal URL

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	1 minute
Compromised Accounts	1
Compromised Account Attributes	10
Indicators	1
Indicator Attributes	1
Malware	1

Change Log

- **Version 1.1.1**
  - Updated the integration to use the `/credentials/occurrences/steam` endpoint which supports Accessed URLs and removes the 1,000 credential return limit.
- **Version 1.1.0**
  - Updated the feed to use the `/credentials/occurrences` endpoint.
  - Updated the feed to optimize overall performance, ThreatQuotient integration standards, and to provide more granular control over the data ingested.
  - Added the following configuration parameters:
    - **Fetch Accessed URLs** - fetches the accessed URLs for each compromised credential.
    - **Fetch GIR Names** - ability to select whether to fetch the GIR name or raw value.
    - **Ignore Numeric Credentials** - ability to skip ingestion of compromised credentials with usernames that consist solely of numeric values.
    - **Context Filter** - allows you to select which pieces of context to ingest with the compromised credentials.
    - **Ingest Accessed URLs As** - allows you to determine whether to ingest accessed URLs as indicators or attributes.
    - **Accessed URL Status** - allows you to select the status that should be applied to the accessed URLs.
    - **Disable Proxies** - allows you to determine whether or not the feed honors the proxy configuration set in the ThreatQ UI.
    - **Enable SSL Verification** - allows you to determine if the feed should validate the host-provided SSL certificate.
  - Removed the **Ingest Compromised Passwords as Attributes** configuration parameter.
- **Version 1.0.0**
  - Initial release