ThreatQuotient

A Securonix Company

Flashpoint CFM CDF

Version 1.0.0

August 03, 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	1.0.0
Compatible with ThreatQ Versions	>= 5.12.1
Support Tier	ThreatQ Supported

Introduction

The Flashpoint Card Fraud Mitigation (CFM) integration imports compromised payment card intelligence from the Flashpoint Fraud API into ThreatQ. The integration supports ingestion of both full-card and partial-card records, enabling organizations to identify payment cards exposed across illicit card shops and threat actor communities on the dark web. By bringing this intelligence into ThreatQ, security and fraud teams can investigate exposed payment cards, correlate them with other threat data, and respond more quickly to potential fraud. Full-card and partial-card records are processed separately to preserve the unique data and context provided by each record type.

The integration provides the following feed:

- **Flashpoint CFM - Monitored Cards** - ingests compromised card records from the Flashpoint Fraud API.

The integration ingests the following ThreatQ objects:

- Compromised Cards (custom object)
- Compromised Card Attributes

Prerequisites

The following is required to install and run the integration:

- A Flashpoint API key with access to the Flashpoint Fraud (Card Fraud Mitigation) API.
- The Compromised Card custom object must be installed on your ThreatQ instance.

Compromised Card Custom Object

The integration requires that the compromised card custom object be installed on your instance prior to installing the integration.

Use the steps provided to install the 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. Set your install pathway environment variable. This command will retrieve the install pathway from your configuration file and set it as variable for use during this installation process.

   ```
   INSTALL_CONF="/etc/threatq/platform/install.conf"

   if [ -f "$INSTALL_CONF" ]; then source "$INSTALL_CONF"

   fi
   ```

```
MISC_DIR="${INSTALL_BASE_PATH:-/var/lib/threatq}/misc"
```

5. Navigate to the tmp folder using the environment variable:

```
cd $MISC_DIR
```

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

   The directory structure should resemble the following:

   - install.sh
   - <custom_object_name>.json
   - images (directory)
     - <custom_object_name>.svg

7. 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.

8. Delete the install.sh, definition json file, and images directory from step 6 after the object has been installed as these files are no longer needed.

ThreatQ v5 Steps

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 /tmp/
```

5. Create a new directory for the custom object files:

```
mkdir <integration_name>
```

6. Upload the custom object files, including the images folder, to the new directory.
7. Navigate to the integration name directory if you have not done so already.

   The directory structure should be as the following:

   - tmp
     - <integration_name>
       - install.sh
       - <custom_object_name>.json
       - images (directory)
         - <custom_object_name>.svg
8. Run the following command to ensure you have the proper permissions to install the custom object:

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

9. Run the install script:

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

   ⚠ You must be in the directory that houses the install.sh and json file 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.

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

```
rm -rf <integration_name>
```

Installation

> The integration requires that the Compromised Card custom object be installed on your ThreatQ instance prior to installing the CDF. Failure to install the custom object will result in the CDF installation process failing.

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 and install the required custom object if you have not done so already.
4. Click on the **Add New Integration** button.
5. 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

   > 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 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
API Key	Enter a Flashpoint API key with permission to access the **Fraud (Card Fraud Mitigation)** API.
Export Type	Select the record types to ingest. Options include: ◦ **Full Cards** imports complete compromised card records. ◦ **Partial Cards** imports `partial_card*` records and generates a synthetic **Compromised Card** value because the full card number is not available. You may select one or both options.
Filter by Monitored BIN Assets	Enable this parameter to ingest only records associated with BIN assets currently monitored by your organization in Flashpoint. When enabled, the feed sends `include_bin_assets=true` to the Flashpoint API. If your organization has no monitored BIN assets configured, Flashpoint may return a `404` response, which the feed treats as an empty result.

PARAMETER	DESCRIPTION
BIN Filter (Optional)	Enter one BIN value per line to limit ingestion to specific BINs. The specified values are submitted to the Flashpoint API using the `include.bin` filter. Leave this field blank to apply no explicit BIN filter. When neither this parameter nor **Filter by Monitored BIN Assets** is configured, the feed ingests all fraud records available to the API key within the configured date window. If both parameters are configured, both filters are included in the request sent to Flashpoint.
Context Filter	Select the contextual fields to ingest as attributes for each compromised card object created in ThreatQ. Only the selected fields are imported. Options include: ◦ Account Number *(default)* ◦ BIN *(default)* ◦ CVV *(default)* ◦ Last 4 Digits *(default)* ◦ Shop Name *(default)* ◦ Release Name *(default)* ◦ Sale Price ◦ Site Actor *(default)* ◦ Site Actor Alias ◦ Source Type *(default)* ◦ Data Type ◦ Breach Title ◦ Last Observed At *(default)* ◦ First Observed At ◦ Indexed At ◦ Expiration *(default)*

PARAMETER	DESCRIPTION
	◦ Owner First Name ◦ Owner Last Name ◦ Owner Full Name ◦ Owner Middle Initial ◦ Owner Middle Name ◦ Owner Phone Number ◦ Owner Email ◦ Owner City ◦ Owner Region ◦ Owner Country ◦ Owner ZIP Code ◦ Flashpoint Link ◦ Flashpoint Document ID

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

Flashpoint CFM - Monitored Cards

The Flashpoint CFM - Monitored Cards feed imports compromised payment card records from the Flashpoint Card Fraud Mitigation (CFM) service into ThreatQ. The feed can ingest both full and partial card records, providing organizations with visibility into exposed payment cards discovered across illicit card shops and threat actor communities to support fraud monitoring and response.

```
POST https://api.flashpoint.io/sources/v2/fraud
```

Sample Response:


```
{
  "items": [
    {
      "id": "0a1b2c3d4e5f60718293a4b5c6d7e8f9",
      "author": "darkvendor",
      "author_alias": ["darkvendor"],
      "bin": "123456",
      "card": {
        "cvv": "123",
        "expiration": "11/2030",
        "number": "1234567890123456"
      },
      "last4": "3456",
      "site": "web.telegram.org",
      "account_number": "123456789012345",
      "site_source_uri": "web.telegram.org",
      "created_at": "2021-07-23T08:14:02.512Z",
      "date": "2021-07-23T08:14:02Z",
      "first_observed_at": "2021-07-23T08:14:02.512Z",
      "indexed_at": "2021-07-23T09:01:55.004Z",
      "last_observed_at": "2021-08-20T00:00:00.000Z",
      "original_id": "src-0a1b2c3d",
      "source_type": "chat",
      "sort_date": "2021-07-23T08:14:02Z",
      "type": "full_card"
    },
    {
      "id": "VT715m6iUyCgx_gVx6UtSA",
```

```
      "author": "briansclub",
      "bin": "477597",
      "card": {
        "expiration": "02/30"
      },
      "cardholder": {
        "location": {
          "country": "United Kingdom",
          "region": "-",
          "zip_code": "PH***"
        },
        "name": {
          "first": "GARY  ***",
          "full_name": "GARY  ***"
        }
      },
      "date": "2026-07-27T17:28:14Z",
      "prices": [34.65],
      "release": {
        "id": "7ZqUzpSEVeaJ0IjIo84Pvg",
        "name": "0727_GB_US_IP"
      },
      "site": "Brian's Club",
      "source_type": "shop",
      "type": "partial_card_cvv"
    }
  ],
  "size": 2,
  "total": { "value": 2, "relation": "=" }
}
```


The following mapping describes how fields from each object in the Flashpoint Fraud API response's **items** array are imported into ThreatQ. The **Published Date** for all ingested objects and attributes is derived from the `.date` field, with `.created_at` used as a fallback when `.date` is not available.

For full-card records, the ThreatQ **Compromised Card** object value is populated from `.card.number`. For partial-card records, the object value is constructed using the available BIN, last four digits, expiration month, expiration year, and CVV. If Flashpoint masks the last four digits, the feed substitutes `****` and appends the record's `.id` to ensure the resulting object value remains unique. For example: `477597-****|02|30|VT715m6iUyCgx_gVx6UtSA`.

The mapping table lists all fields supported by the integration. Not every ingested record contains every supported field, and attributes configured through the **Context Filter** are created only when the corresponding option is selected and the source field contains a value. As a result, some attribute types may not appear under **Manage Columns** until at least one value for that attribute has been ingested.

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.card.number	Card.Value	N/A	.date	1234567890123456	Full cards
.bin, .last4, .card.expiration, .card.cvv, .id	Card.Value	N/A	.date	477597-****|02|30|VT715...	Partial cards
.account_number	Card.Attribute	Account Number	.date	123456789012345	User-Configurable
.bin	Card.Attribute	BIN	.date	123456	User-Configurable
.card.cvv	Card.Attribute	CVV	.date	123	User-Configurable
.last4	Card.Attribute	Last 4 Digits	.date	7890	User-Configurable
.site	Card.Attribute	Shop Name	.date	web.telegram.org	User-Configurable
.release.name	Card.Attribute	Release Name	.date	Ugly Duckling	User-Configurable
.prices	Card.Attribute	Sale Price	.date	19.8	User-Configurable
.author	Card.Attribute	Site Actor	.date	darkvendor	User-Configurable
.author_alias	Card.Attribute	Site Actor Alias	.date	dv_old_handle	User-Configurable
.source_type	Card.Attribute	Source Type	.date	chat	User-Configurable
.type	Card.Attribute	Data Type	.date	full_card	User-Configurable
.breach_title	Card.Attribute	Breach Title	.date	Example Card Data Breach	User-Configurable
.last_observed_at	Card.Attribute	Last Observed At	.date	2026-07-23T12:00:00Z	User-Configurable
.first_observed_at	Card.Attribute	First Observed At	.date	2026-07-22T12:00:00Z	User-Configurable

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.indexed_at	Card.Attribute	Indexed At	.date	2026-07-23T12:05:00Z	User-Configurable
.card.expiration	Card.Attribute	Expiration	.date	11/2030	User-Configurable
.cardholder.name.first	Card.Attribute	Owner First Name	.date	john	User-Configurable
.cardholder.name.last	Card.Attribute	Owner Last Name	.date	smith	User-Configurable
.cardholder.name.full_name	Card.Attribute	Owner Full Name	.date	john d smith	User-Configurable
.cardholder.name.middle_initial	Card.Attribute	Owner Middle Initial	.date	d	User-Configurable
.cardholder.name.middle	Card.Attribute	Owner Middle Name	.date	daniel	User-Configurable
.cardholder.phone_number	Card.Attribute	Owner Phone Number	.date	+15551234567	User-Configurable
.cardholder.email	Card.Attribute	Owner Email	.date	john.smith@example.com	User-Configurable
.cardholder.location.city	Card.Attribute	Owner City	.date	halethorpe	User-Configurable
.cardholder.location.region	Card.Attribute	Owner Region	.date	md	User-Configurable
.cardholder.location.country	Card.Attribute	Owner Country	.date	usa	User-Configurable
.cardholder.location.zip_code	Card.Attribute	Owner Zip Code	.date	21227	User-Configurable
.site_source_uri	Card.Attribute	Flashpoint Link	.date	web.telegram.org	User-Configurable
.id	Card.Attribute	Flashpoint Document ID	.date	0a1b2c3d...	User-Configurable

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	35 minutes
Compromised Cards	6,174
Compromised Card Attributes	101,968

Known Issues / Limitations

- **Context Attribute Selection:** Selecting a large number of **Context Filter** options can significantly increase the number of attributes ingested for each compromised card. For optimal performance and data relevance, select only the context fields required by your organization.
- **Export Type Filtering:** The Flashpoint Fraud API returns both full-card and partial-card records in the same response. The feed filters the results after retrieval and ingests only the record types selected in the **Export Type** configuration.
- **Partial Card Data:** Flashpoint may mask the last four digits and CVV for partial-card records. When the last four digits are unavailable, the feed generates a unique placeholder value using the Flashpoint record identifier. This value is intended only to uniquely identify the record and does not represent a complete payment card number.
- **Incomplete Records:** Records that do not belong to the supported **full_card** or **partial_card** record families, or that do not contain the minimum data required to create a compromised card object, are skipped during ingestion.
- **Large Historical Imports:** The feed retrieves data using paginated API requests. Very large historical imports may exceed the Flashpoint API's pagination limits and result in incomplete retrieval. To ensure complete ingestion when importing historical data, reduce the feed run interval so each execution processes a smaller date range.
- **BIN Assets** - the **Filter by Monitored BIN Assets** parameter relies on BIN assets configured in your Flashpoint environment. If no monitored BIN assets are configured, Flashpoint may return a `404` response. In this case, the feed treats the response as an empty result set and no records are ingested.

Change Log

- **Version 1.0.0**
  - Initial release