ThreatQuotient

MITRE ATT&CK CAPEC CDF Guide

Version 1.0.1

March 07, 2022

ThreatQuotient
11400 Commerce Park Dr., Suite 200
Reston, VA 20191

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.

Versioning

- Current integration version: `1.0.1`
- Compatible with ThreatQ versions >= `4.35.0`

Introduction

The **MITRE ATT&CK CAPEC CDF** for ThreatQuotient enables the automatic ingestion of Common Attack Pattern Enumerations and Classifications distributed by MITRE.

The integration provides the following feeds:

- **MITRE ATT&CK CAPEC** - enables the automatic ingestion of Common Attack Pattern Enumerations and Classifications, distributed by MITRE.
- **Get CWE HTML (Supplemental)** - fetches the HTML page corresponding to the CWE ID in question.

The integration ingests the following object types into the ThreatQ platform:

- Attack Patterns
- Vulnerabilities

See the ThreatQ Mapping chapter for more details regarding feed ingestion.

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. If prompted, select the individual feeds to install and click **Install**. The feed will be added to the integrations page.

You will still need to configure and then enable the feed.

Configuration

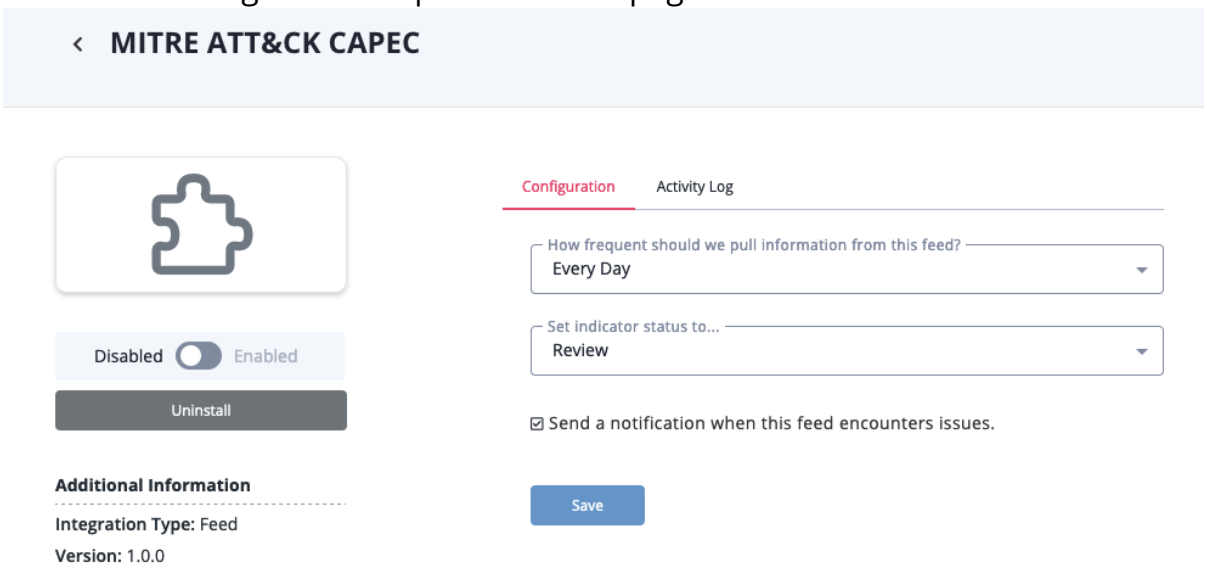
> ThreatQuotient does not issue API keys for third-party vendors. Contact the specific vendor to obtain API keys and other integration-related credentials.

To configure the integration:

1. Navigate to your integrations management page in ThreatQ.
2. Select the **OSINT** 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 to open its details page.



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

ThreatQ Mapping

MITRE ATT&CK CAPEC

`GET https://raw.githubusercontent.com/mitre/cti/master/capec/2.1/stix-capec.json`

The `MITRE ATT&CK CAPEC CDF` for ThreatQuotient enables the automatic ingestion of Common Attack Pattern Enumerations and Classifications, distributed by MITRE. In addition, the feed brings in any related MITRE ATT&CK Patterns, related MITRE ATT&CK CWEs, and more.


```
{
    "objects": [
        {
            "definition_type": "statement",
            "id": "marking-definition--17d82bb2-eeeb-4898-bda5-3ddbcd2b799d",
            "definition": {
                "statement": "CAPEC is sponsored by US-CERT in the office of Cybersecurity and Communications at the
U.S. Department of Homeland Security. Copyright \u00a9 2007 - 2021, The MITRE Corporation. CAPEC and the CAPEC logo
are trademarks of The MITRE Corporation."
            },
            "type": "marking-definition",
            "created": "2021-01-08T19:55:06.45963Z"
        },
        {
            "name": "The MITRE Corporation",
            "identity_class": "organization",
            "id": "identity--e50ab59c-5c4f-4d40-bf6a-d58418d89bcd",
            "object_marking_refs": [
                "marking-definition--17d82bb2-eeeb-4898-bda5-3ddbcd2b799d"
            ],
            "type": "identity",
            "created": "2021-01-08T19:55:06.461Z",
            "modified": "2021-01-08T19:55:06.461Z"
        },
        {
            "id": "attack-pattern--92cdcd3d-d734-4442-afc3-4599f261498b",
            "name": "Accessing Functionality Not Properly Constrained by ACLs",
            "description": "In applications, particularly web applications, access to functionality is mitigated by
an authorization framework. This framework maps Access Control Lists (ACLs) to elements of the application's
functionality; particularly URL's for web apps. In the case that the administrator failed to specify an ACL for a
particular element, an attacker may be able to access it with impunity. An attacker with the ability to access
functionality not properly constrained by ACLs can obtain sensitive information and possibly compromise the entire
application. Such an attacker can access resources that must be available only to users at a higher privilege level,
can access management sections of the application, or can run queries for data that they otherwise not supposed to.",
            "created_by_ref": "identity--e50ab59c-5c4f-4d40-bf6a-d58418d89bcd",
            "object_marking_refs": [
                "marking-definition--17d82bb2-eeeb-4898-bda5-3ddbcd2b799d"
            ],
            "created": "2014-06-23T00:00:00.000Z",
            "modified": "2020-12-17T00:00:00.000Z",
            "external_references": [
                {
                    "source_name": "capec",
```

```
                    "url": "https://capec.mitre.org/data/definitions/1.html",
                    "external_id": "CAPEC-1"
                },
                {
                    "source_name": "cwe",
                    "url": "http://cwe.mitre.org/data/definitions/276.html",
                    "external_id": "CWE-276"
                },
                {
                    "source_name": "ATTACK",
                    "description": "Hijack Execution Flow: ServicesFile Permissions Weakness",
                    "url": "https://attack.mitre.org/wiki/Technique/T1574/010",
                    "external_id": "T1574.010"
                }
            ],
            "x_capec_likelihood_of_attack": "High",
            "x_capec_typical_severity": "High",
            "x_capec_prerequisites": [
                "The application must be navigable in a manner that associates elements (subsections) of the
application with ACLs.",
                "The various resources, or individual URLs, must be somehow discoverable by the attacker",
                "The administrator must have forgotten to associate an ACL or has associated an inappropriately
permissive ACL with a particular navigable resource."
            ],
            "x_capec_skills_required": {
                "Low": "In order to discover unrestricted resources, the attacker does not need special tools or
skills. They only have to observe the resources or access mechanisms invoked as each action is performed and then try
and access those access mechanisms directly."
            },
            "x_capec_resources_required": [
                "None: No specialized resources are required to execute this type of attack."
            ],
            "x_capec_consequences": {
                "Confidentiality": [
                    "Gain Privileges"
                ],
                "Access_Control": [
                    "Gain Privileges"
                ],
                "Authorization": [
                    "Gain Privileges"
                ]
            },
            "x_capec_abstraction": "Standard",
            "x_capec_example_instances": [
                "\n               <xhtml:p>Implementing the Model-View-Controller (MVC) within Java EE's Servlet
paradigm using a \"Single front controller\" pattern that demands that brokered HTTP requests be authenticated before
hand-offs to other Action Servlets.</xhtml:p>\n               <xhtml:p>If no security-constraint is placed on those
Action Servlets, such that positively no one can access them, the front controller can be subverted.</xhtml:p>\n
"
            ],
            "x_capec_execution_flow": "<h2> Execution Flow </h2><div><h3>Explore</h3><ol><li> <p> <b>Survey: </b>The
attacker surveys the target application, possibly as a valid and authenticated user</p></
li><table><tbody><tr><th>Techniques</th></tr><tr><td>Spidering web sites for all available links</td></
tr><tr><td>Brute force guessing of resource names</td></tr><tr><td>Brute force guessing of user names / credentials</
td></tr><tr><td>Brute force guessing of function names / actions</td></tr></tbody></table><li> <p> <b>Identify
Functionality: </b>At each step, the attacker notes the resource or functionality access mechanism invoked upon
performing specific actions</p></li><table><tbody><tr><th>Techniques</th></tr><tr><td>Use the web inventory of all
forms and inputs and apply attack data to those inputs.</td></tr><tr><td>Use a packet sniffer to capture and record
network traffic</td></tr><tr><td>Execute the software in a debugger and record API calls into the operating system or
important libraries. This might occur in an environment other than a production environment, in order to find
```

```
weaknesses that can be exploited in a production environment.</td></tr></tbody></table></ol></
div><div><h3>Experiment</h3><ol><li> <p> <b>Iterate over access capabilities: </b>Possibly as a valid user, the
attacker then tries to access each of the noted access mechanisms directly in order to perform functions not
constrained by the ACLs.</p></li><table><tbody><tr><th>Techniques</th></tr><tr><td>Fuzzing of API parameters (URL
parameters, OS API parameters, protocol parameters)</td></tr></tbody></table></ol></div>",
            "x_capec_status": "Draft",
            "x_capec_version": "3.4",
            "type": "attack-pattern"
        }
    ]
}
```


ThreatQ provides the following default mapping for this feed:

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.objects[].external_id - .objects[].name	Attack Pattern.Value	N/A	.objects[].created	CAPEC-1 - Accessing Functionality Not Properly Constrained by ACLs	If `.objects[].external_references[].source_name` is `capec`
.objects[].external_references[]	Attack Pattern.Value	N/A	.objects[].created	T10003 - XXXXX	If `.objects[].external_references[].source_name` is `ATTACK`
.objects[].description	Attack Pattern.Description	N/A	.objects[].created	An attacker is able to cause a victim to load content into their web-browser	Description is concatenated with the `x_capec_execution_flow` value
.objects[].external_references[]	Vulnerability.Name	N/A	.objects[].created	CWE-833 - Deadlock	If `.objects[].external_references[].source_name` is `cwe`
.objects[].x_capec_likelihood_of_attack	Attack Pattern.Attribute	Likelihood of Attack	.objects[].created	High	N/A
.objects[].x_capec_typical_severity	Attack Pattern.Attribute	Typical Severity	.objects[].created	High	N/A
.objects[].x_capec_prerequisites	Attack Pattern.Attribute	Prerequisite	.objects[].created	The various resources, or individual URLs, must be somehow discoverable by the attacker	N/A
.objects[].x_capec_resources_required	Attack Pattern.Attribute	Resources Required	.objects[].created	None: No specialised resources are required to execute this type of attack.	N/A
.objects[].x_capec_abstraction	Attack Pattern.Attribute	Abstraction	.objects[].created	Standard	N/A
.objects[].x_capec_status	Attack Pattern.Attribute	Status	.objects[].created	Draft	N/A
.objects[].x_capec_version	Attack Pattern.Attribute	CAPEC Version	.objects[].created	3.4	N/A

FEED DATA PATH	THREATQ ENTITY	THREATQ OBJECT TYPE OR ATTRIBUTE KEY	PUBLISHED DATE	EXAMPLES	NOTES
.objects[].x_capec_skills_required	Attack Pattern.Attribute	Skills Required	.objects[].created	High: Detailed knowledge on scripting and SPI programming	N/A
.objects[].is_deprecated	Attack Pattern.Attribute	Is Deprecated	.objects[].created	1	Convert to `True` if value is `1`, othwewise won't be ingested
.objects[].x_capec_consequences	Attack Pattern.Attribute	.objects[].x_capec_consequences[].key	.objects[].created	Confidentiality: Read Data	Value Of: `.objects[].x_capec_consequences[key]`
.objects[].external_references[]	Attack Pattern.Attribute	External Reference	.objects[].created	https://www.zdnet.com/article/microsoft-worried-about--secured-core-pc/	Excluding Types: cwe, ATTACK, & capec

Get CWE HTML (Supplemental)

```
GET http://cwe.mitre.org/data/definitions/{{cwe_id}}.html
```

The `Get CWE HTML` Supplemental feed fetches the HTML page corresponding to the CWE ID in question. The CWE value/name is parsed out of the HTML response.

Average Feed Run

MITRE ATT&CK CAPEC

METRIC	RESULT
Run Time	2 minutes
Attack Patterns	722
Attack Pattern Attributes	5,989
Vulnerabilities	324

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.

Known Issues / Limitations

- The ingested CWEs do not come with context. To contextualize the CWEs, use the MITRE ATT&CK CWE feed from the Marketplace

Change Log

- **Version 1.0.1**
  - Fixed an issue where the user would encounter an error if CWE information was not provided.
- **Version 1.0.0**
  - Initial release