Cofense Triage Connector Implementation Guide

Version 1.0.0

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.

Copyright © 2019 ThreatQuotient, Inc.

All rights reserved. This document and the software product it describes are licensed for use under a software license agreement. Reproduction or printing of this document is permitted in accordance with the license agreement.

Last Updated: Tuesday, August 13, 2019

Contents

Versioning

- Current integration version: `1.0.0`
- Supported on ThreatQ versions: `4.21.1` or higher

Introduction

Cofense Triage, a phishing-specific incident response platform, helps stop active phishing attacks in progress. By leveraging real-time, internally reported attack intelligence from conditioned users, Cofense Triage makes it easy to stop phishing attacks in progress by eliminating the noise of the abuse mailbox, automating standard responses, and orchestrating across other security systems to quickly respond to and eliminate phishing threats.

Installation

Complete the following steps to install the connector:

1. Login to https://download.threatq.com/integrations/.
2. Download the **cofense_triage.yaml** file.
3. From the ThreatQ user interface, select the **Settings icon > Incoming Feeds**.
4. Click **Add New Feed**.
5. In the Add New Feed dialog box, complete one of the following actions:
    - Drag and drop the yaml file into the dialog box.
    - **Click to browse** to the yaml file and select it.

   The connector installs as a feed on **Commercial** tab.

6. Under Cofense Triage, click **Feed Settings**.
7. The feed provides the following configuration parameters:
   - **Email address**: The email address used for authentication
   - **API Key**: The Cofense Triage token
   - **Threat Level**: Filter the response based on the threat (All, Malicious, Suspicious, Benign).
   - **Malicious**: Status custom mapping for Malicious indicators
   - **Suspicious**: Status custom mapping for Suspicious indicators
   - **Benign**: Status custom mapping for Benign indicators
8. Click the toggle button next to Cofense Triage to enable the feed.
9. Click **Save Changes**.

ThreatQ Mapping

Cofense provides an API that users can use to programmatically extract data from Cofense Triage in JSON format. The response contains a list of indicators and for each indicator a call to the API is made in order to load detailed information about the associated report.

To set up the API, the user generates the HTTP Authorization token needed to gain access to the API.

The request will contain the following parameters:

- level - the user can select the threat level to filter on: Malicious, Suspicious, or Benign.
- start_date - will be in iso format datetime (UTC) and will cause the API to only include indicators that were created starting this date; this value will be automatically set to the datetime of the last feed run; the default frequency is 24h, so

the value supplied to the API will be the time at which the feed begins execution minus 24 hours.

- end_date - will be in iso format datetime (UTC) and will cause the API to only include indicators that were created up to this date.

Endpoints:

- GET /triage_threat_indicators - This endpoint fetches the subjects, senders, domains, URLs, or MD5 or SHA256 hashes that operators identified in Cofense Triage as threat indicators within a specified timeframe. If no parameters are specified, fetches all identified threat indicators.

  Example:

```
{
    "id": 15,
    "created_at": "2019-07-02T12:47:16.307Z",
    "operator_id": 9,
    "report_id": 5826,
    "threat_level": "Malicious",
    "threat_key": "SHA256",
    "threat_value": "1e2c4ac7be08888c72c953adaeb79254e7e9b
                     821988bfdad5d75d75b2467def1"
}
```

- GET /reports/{report_id} - This endpoint fetches a single report that matches the specified report ID.

  Example:

```
{
    "id": 5826,
    "cluster_id": 3079,
    "reporter_id": 3312,
    "primary_recipe_id": null,
    "recipe_name": null,
    "processing_operator_id": null,
    "created_at": "2019-05-17T19:54:02.421Z",
    "updated_at": "2019-06-13T16:13:49.372Z",
    "reported_at": "2019-05-17T11:37:52.000Z",
    "processed_at": null,
    "report_subject": "NEW ORDER",
    "report_headers": "Date: Fri, 17 May 2019 19:54:02
          +0000\r\nMessage-ID: \u003c5cdf
          115a4b2ac_658b2af8ffdcb3386206b
          @ip-10-132-9-188.ec2.internal.mail\
          u003e\r\nSubject: NEW ORDER\r\nMime-
          Version: 1.0\r\nContent-Type:
          multipart/mixed;\r\n boundary=\"--==
          _mimepart_5cdf1159f09a4_658b2af8ff
          dcb338619b\";\r\n charset=UTF-8\r\
          nContent-Transfer-Encoding: 7bit",
    "report_body": "Good day\n\n\nPlease arrange to provide
    the best offer for below attached Purchase
        Order\nThe requirement for our green field
        project in Berghofen,Dortmund.\nKindly get
        back to us\n\n \n\n\n1) Proforma invoice
        with bank details\n\n2) Delivery date \n\n3)
    FOB/CIF Port\n\n \n\n \n \nRegards,\n\nkahn
        Gotze\nSales \u0026 Services Assistant\n",
    "md5": "2b2b8f5d82e04225c9c7987417d8cae7",
    "sha256": "812e3d517611176ff99d09d7a7723489b16d4a91499cb
              2beb02ecf396ca520b2",
    "category_id": null,
    "match_priority": 5,
    "tags": [],
    "reporter_phishme_reports_count": 0,
    "suspect_received_at": "2019-05-17T11:38:08.000Z",
    "suspect_from_address": null,
    "email_attachments": [
      {
        "id": 10420,
        "report_id": 5824,
        "decoded_filename": "ORDER#t571BA80.rar",
        "content_type": "application/octet-stream; name=ORDER
                        #t571BA80.rar",
        "size_in_bytes": 219777,
        "email_attachment_payload": {
          "id": 5818,
          "md5": "e74c45a697651f3942f86fc5fce009df",
          "sha256": "1e2c4ac7be08888c72c953adaeb79254e7e9b
                    821988bfdad5d75d75b2467def1",
          "mime_type": "application/x-rar; charset=binary"
        }
      }
    ],
    "email_urls": [],
    "rules": [
      {
        "id": 3114,
        "name": "PM_Rar_with_exe",
        "reports_count": 407,
        "active": true,
        "created_at": "2019-04-11T16:53:54.183Z",
        "updated_at": "2019-04-11T16:53:54.183Z",
        "priority": 5,
        "author_name": "PhishMe"
      }
    ]
}
```

ThreatQ provides the following default mapping for the feed:

Cofense Key	ThreatQ Entity	ThreatQ Name	Examples	Notes
Indicator				
threat_value	indicator.value		1e2c4ac7be08888c72c953adaeb 79254e7e9b821988bfdd75d75b2467def1	
threat_key	indicator.type		SHA256	see mapping below
threat_level	indicator.status		Malicious	see mapping below
created_at	indicator.published		2019-07-02T12:47:16.307Z	

Cofense Key	ThreatQ Entity	ThreatQ Name	Examples	Notes
	_at			
id	indicator.attribute	ID	15	
operator_id	indicator.attribute	Operator ID	9	
report_id	indicator.attribute	Report ID	5826	
Event				
report_subject	event.title		PO NO.AWJCC-18-1120	
event.type	Phishing			
created_at	event.published_at		2019-05-17T19:53:29.071Z	
reported_at	event.happened_at		2019-05-15T16:39:49.000Z	
ID	event.attribute	ID	5802	

Cofense Key	ThreatQ Entity	ThreatQ Name	Examples	Notes
cluster_id	event.attribute	Cluster ID		
reporter_id	event.attribute	Reporter ID	495	
location	event.attribute	Location	Processed	
primary_recipe_id	event.attribute	Primary Recipient ID		
recipe_name	event.attribute	Recipient Name		
processing_operator_id	event.attribute	Processing Operator ID		
updated_at	event.attribute	Updated At	2019-05-22T20:46:51.788Z	
processed_at	event.attribute	Processed At	2019-05-22T20:46:51.366Z	
report_headers	event.attribute	Report Headers	Date...	

Cofense Key	ThreatQ Entity	ThreatQ Name	Examples	Notes
report_body	description	Report Body	Hello \nPlease refer attached purchase...	
email_attachments[] .mime_type	event. attribute	Mime Type		
rules[].name, rules[].author_name	event. attribute	Rule Name, Author		
Related Indicator				
md5	indicator. value		1603df775fe5448806627d8e8c8dab35	
indicator.type		MD5		
created_at	indicator. published_at		2019-07-02T12:47:16.307Z	
Related Indicator				

Cofense Key	ThreatQ Entity	ThreatQ Name	Examples	Notes
sha256	indicator.value		6ff04a4a594af8545e5c0b662611b41c90f8a34c99f3236d9afb0629ea2bfbcc	
indicator.type		SHA-256		
created_at	indicator.published_at		2019-07-02T12:47:16.307Z	

The mapping between the indicator types in Cofense Triage and ThreatQ is as follows:

Cofense Triage	ThreatQ
Sender	Email Address
Subject	Email Subject
Domain	FQDN
MD5	MD5
SHA256	SHA-256
URL	URL

The user can enter custom mapping between the status of the indicator and threat type. The default mappings are:

Cofense Triage	ThreatQ
Malicious	Active
Suspicious	Review
Benign	Whitelisted

If the user enters an incorrect status, it will default to Active.