# ThreatQuotient

## Volexity Threat Intelligence CDF

**Version 1.1.0**

May 13, 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** | 1.1.0 |
| **Compatible with ThreatQ Versions** | >= 6.5.0 |
| **Support Tier** | ThreatQ Supported |

# Introduction

The Volexity Threat Intelligence CDF enables users to ingest threat intelligence including, but not limited to, indicators, vulnerabilities, and YARA signatures from Volexity.

Volexity's solutions provide advanced analytics about the state of your devices and rapid insights into the risk those devices pose to your organization. These solutions are used by organizations across the globe including leading technology companies in Silicon Valley, the Fortune 500, and the largest government institutions.

The integration provides the following feeds:

- **Volexity - Entities** - ingests IOCs from Volexity.
- **Volexity - Tags** - ingests Adversaries, Campaigns, Vulnerabilities, Reports and Malware type objects from Volexity.
- **Volexity - Signatures** - ingests Yara Signatures from Volexity.
- **Volexity - Reports** - ingests Volexity Reports.

The integration ingests the following system object types:

- Adversaries
- Campaigns
- Indicators
  - Email Address
  - FQDN
  - MD5
  - SHA-1
  - SHA-256
- Malware
- Vulnerabilities
- Reports
- Signatures
  - Yara
  - Snort

# Prerequisites

The following is required to utilize the feeds provided by the integration:

- A Volexity Username.
- A Volexity API Key.

# 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 yaml file.
3. Navigate to the integrations management page on your ThreatQ instance.
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
6. 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).

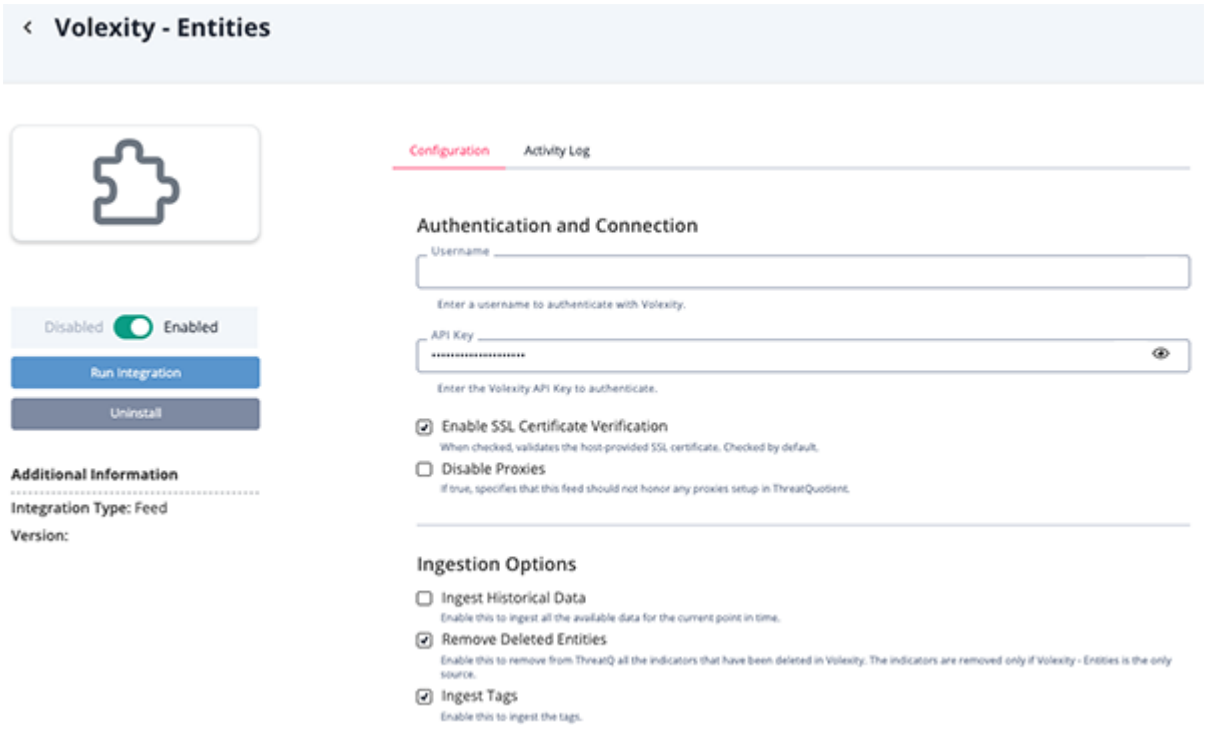
   > If you are installing the integration for the first time, it will be located under the **Disabled** tab.

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

## Volexity Entities Parameters

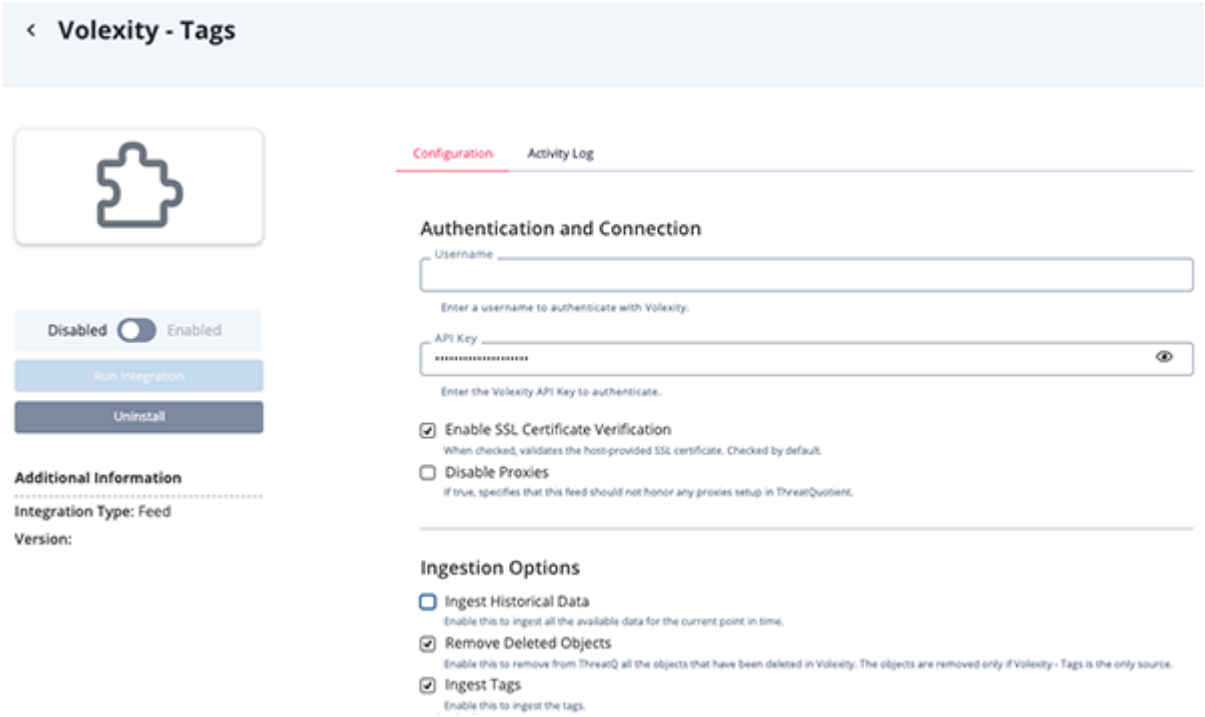
| PARAMETER | DESCRIPTION |
|---|---|
| **Username** | Your Volexity username. |
| **API Key** | Your Volexity API Key. |
| **Enable SSL Certificate Verification** | Enable this parameter if the feed should validate the host-provided SSL certificate. |
| **Disable Proxies** | Enable this parameter if the feed should not honor proxies set in the ThreatQ UI. |
| **Ingest Historical Data** | Enable this to ingest all the available data for the current point in time. This parameter is disabled by default. |
| **Remove Deleted Objects** | Enable this parameter to remove all indicators from ThreatQ that have been deleted in Volexity. This parameter is enabled by default. |

| PARAMETER | DESCRIPTION |
| --- | --- |
| | The signatures are removed only if `Volexity - Entities` is the only source.<br><br>This parameter is only accessible if the Ingest Historical Data parameter is disabled. |
| **Ingest Tags** | Enable this parameter to ingest the tags. This parameter is enabled by default. |



# Volexity Tags Parameters

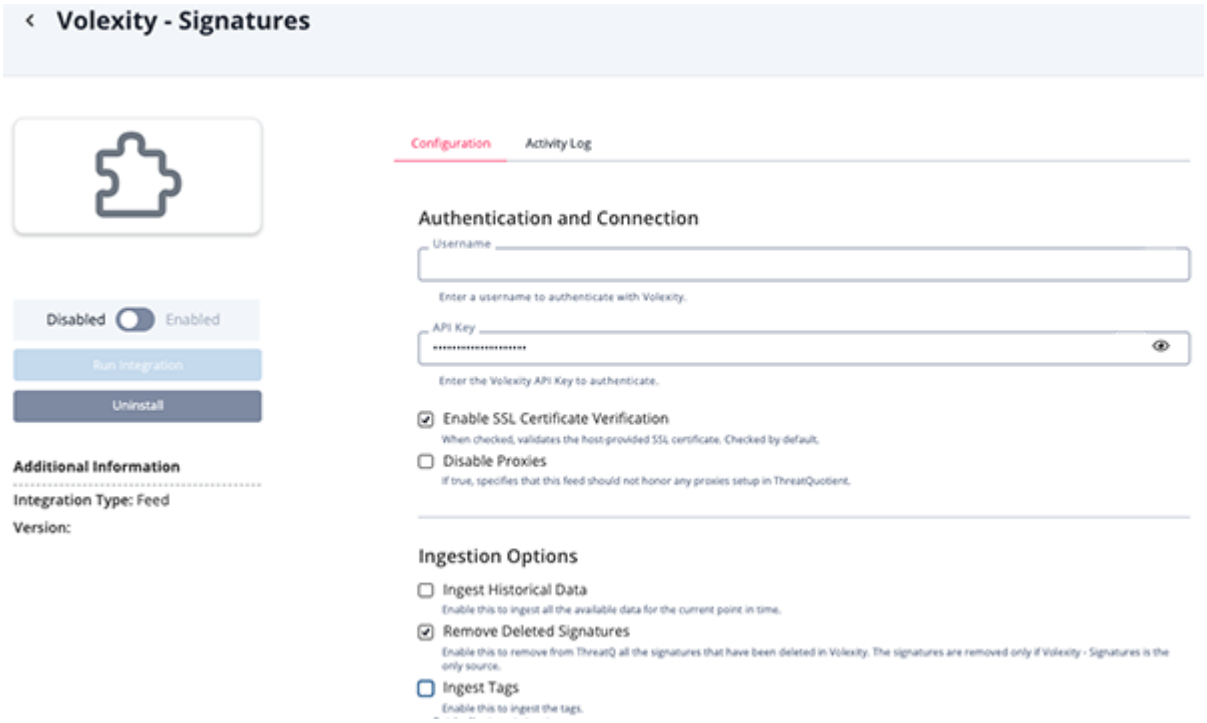
| PARAMETER | DESCRIPTION |
|---|---|
| **Username** | Your Volexity username. |
| **API Key** | Your Volexity API Key. |
| **Enable SSL Certificate Verification** | Enable this parameter if the feed should validate the host-provided SSL certificate. |
| **Disable Proxies** | Enable this parameter if the feed should not honor proxies set in the ThreatQ UI. |
| **Ingest Historical Data** | Enable this to ingest all the available data for the current point in time. This parameter is disabled by default. |
| **Remove Deleted Objects** | Enable this parameter to remove all objects from ThreatQ that have been deleted in Volexity. This parameter is enabled by default.<br><br>The signatures are removed only if `Volexity - Tags` is the only source.<br><br>This parameter is only accessible if the Ingest Historical Data parameter is disabled. |
| **Ingest Tags** | Enable this parameter to ingest the tags. This parameter is enabled by default. |




# Volexity Signatures Parameters

| PARAMETER | DESCRIPTION |
|---|---|
| **Username** | Your Volexity username. |
| **API Key** | Your Volexity API Key. |
| **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. |
| **Ingest Historical Data** | Enable this to ingest all the available data for the current point in time. This parameter is disabled by default. |
| **Remove Deleted Signatures** | Enable this parameter to remove all the signatures from ThreatQ that have been deleted in Volexity. This parameter is enabled by default. |

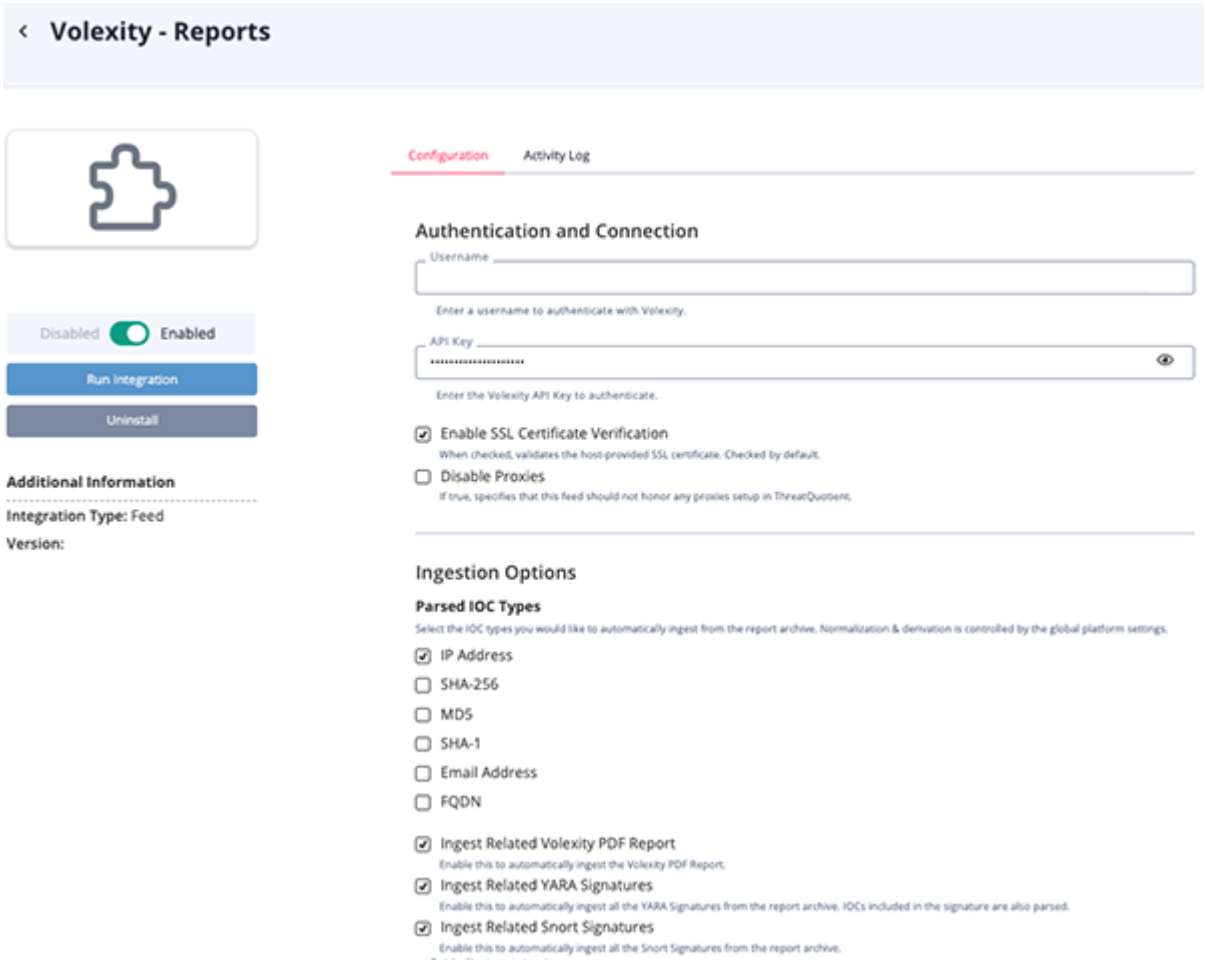
| PARAMETER | DESCRIPTION |
| --- | --- |
| | The signatures are removed only if `Volexity - Signatures` is the only source.<br><br>This parameter is only accessible if the Ingest Historical Data parameter is disabled. |
| **Ingest Tags** | Enable this parameter to ingest the tags. This parameter is enabled by default. |



# Volexity Reports Parameters

| PARAMETER | DESCRIPTION |
|---|---|
| **Username** | Your Volexity username. |
| **API Key** | Your Volexity API Key. |
| **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. |
| **Parsed IOC Types** | Select the IOC types to automatically ingest from the report archive.<br>◦ IP Address (Default)<br>◦ SHA-256 (Default)<br>◦ MD5<br>◦ SHA-1<br>◦ Email Address (Default)<br>◦ FQDN (Default)<br>Normalization and derivation is controlled by the global platform settings. |
| **Ingest Related Volexity PDF Report** | Enable this parameter to automatically ingest the Volexity PDF Report. This parameter is disabled by default. |
| **Ingest Related YARA Signatures** | Enable this parameter to automatically ingest all the YARA Signatures from the report archive. This parameter is disabled by default.<br>IOCs included in the signature are also parsed. |

| PARAMETER | DESCRIPTION |
| --- | --- |
| **Ingest Related Snort Signatures** | Enable this parameter to automatically ingest all the Snort Signatures from the report archive. This parameter is disabled by default. |




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

## Volexity - Entities

The Volexity - Entities feed feed ingests Volexity entities as ThreatQ Indicators. The feed gives the option to ingest all the data for the current point in time or to ingest only the data that was created or updated since the feed start date. The user configuration `Ingest Historical Data` is used to decide what data to ingest.

> If the `Ingest Historical Data` user configuration is enabled, the API value for `.action` is `deleted`.

### Ingest All Data

`GET https://intel-api.volexity.com/entities`

**Sample Parameters:**

```
{
  "limit": 500,
  "sort": "last_modified,entity_id"
}
```

**Sample Response:**

```
{
    "after": "2024-02-12T14:57:44Z,11db9815-fd7c-432e-9f49-2853ed18a296",
    "data": [
        {
            "attributes": {
                "md5": "c445d2c98d85325377a92342ee70b078",
                "sha1": "024767fa9c5678ed0f1dc12977562b7b050df324",
                "sha256":
"a1f9b76ddfdafc47d4a63a04313c577c0c2ffc6202083422b52a00803fd8193d",
                "size": -1
            },
            "created": "2020-07-09T12:02:11Z",
            "description": "Initial historical data import.",
            "entity_id": "0e85b534-182b-4f34-bc46-92548fc54520",
            "entity_type": "file",
            "is_indicator": true,
            "last_modified": "2021-09-10T11:00:02Z",
            "severity": "malicious",
            "tags": [
                "Unclassified"
            ],
            "value":
```

```
"a1f9b76ddfdafc47d4a63a04313c577c0c2ffc6202083422b52a00803fd8193d"
        },
        {
            "attributes": {
                "is_compromised": false
            },
            "created": "2020-04-22T00:00:00Z",
            "description": "Initial data import",
            "entity_id": "ffbe81cd-64cc-42d0-950d-0a1c750b5312",
            "entity_type": "hostname",
            "is_indicator": true,
            "last_modified": "2021-09-12T00:23:30Z",
            "severity": "malicious",
            "tags": [
                "APT",
                "OceanLotus"
            ],
            "value": "dmkatti.com"
        },
        {
            "attributes": {
                "asname": "AS-CHOOPA, US",
                "asnumber": 20473,
                "cc": "KR"
            },
            "created": "2022-01-17T11:14:08Z",
            "description": "SHADOWPAD IP discovered based on default response
data",
            "entity_id": "002b479b-160e-4d8a-8f1f-6ed307b2994f",
            "is_indicator": false,
            "last_modified": "2023-07-05T18:30:03Z",
            "severity": "malicious",
            "tags": [
                "APT",
                "ShadowPad"
            ],
            "entity_type": "ipaddress",
            "value": "158.247.227.6"
        },
        {
            "attributes": {},
            "created": "2022-06-23T06:54:17Z",
            "description": "Attacker controlled domain and sender associated
with Ukraine themed phishing campaign with a RAR file containing a malicious
LNK file that was observed 2022-06-22. The LNK file is designed to download
code from ustreamiptv.com that appears to be a variant of Nerbian RAT.",
            "entity_id": "82c692a3-e2aa-4de1-8ee1-08a0fb68eea0",
            "is_indicator": true,
            "last_modified": "2022-07-04T19:29:42Z",
            "severity": "malicious",
            "tags": [
```

```
                "NERBIANRAT",
                "CharcoalGrass"
            ],
            "entity_type": "emailsender",
            "value": "donate@donateforfuture.com"
        }
    ],
    "sort": "last_modified,entity_id"
}
```

## Ingest Only Updated/Created/Deleted Data

```
GET https://intel-api.volexity.com/feeds/entities
```

**Request Parameters**

```
{
  "listen": "false",
  "timestamp": "2025-04-20T00:00:00.000Z"
}
```

**Sample Response**

```
{"action":"created","data":{"attributes":
{"is_compromised":false},"created":"2025-04-28T01:32:40Z","description":"Remcos
discovered configured in malware sample:
99b39fdb3c9a16077c908bd0bda929cafaf93a2578d5cc43d7dc52c3de7ed50a","entity_id":"
5e122755-52ac-4c1c-8bd7-67d9cd994c70","is_indicator":true,"last_modified":"2025
-04-27T14:00:03Z","severity":"malicious","tags":
["REMCOS"],"entity_type":"hostname","value":"wealthybank.ddns.net"},"timestamp"
:"2025-04-27T14:00:03Z"}
{"action":"updated","data":{"attributes":{"asname":"DIGITALOCEAN-ASN,
US","asnumber":14061,"cc":"SG"},"created":"2024-10-27T04:00:18Z","description":
"Detects certificates generated using Metasploit's SSL generation algorithm.
This uses the Faker library wordlists.","entity_id":"ce98e33e-b439-460f-b6a2-
f1d06b0b0507","is_indicator":false,"last_modified":"2025-04-27T16:30:03Z","seve
rity":"malicious","tags":
["Metasploit"],"entity_type":"ipaddress","value":"157.245.144.118"},"timestamp"
:"2025-04-27T16:30:03Z"}
```

## Entities Mapping Table

ThreatQuotient provides the following default mapping for both request types for this feed:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `data[].entity_type` | Indicator.Type | SHA-256, FQDN, IP Address, Email Address | `data[].created` | `IP Address` | N/A |
| `data[].value` | Indicator.Value | N/A | `data[].created` | `158.247.227.6` | N/A |
| `data[].description` | Indicator.description | N/A | `data[].created` | `Initial historical data import.` | N/A |
| `data[].severity` | Indicator.Attribute | Severity | `data[].created` | `malicious` | Updatable |
| `data[].tags` | Indicator.Tags | N/A | `data[].created` | `NERBIANRAT` | N/A |
| `data[].attributes.asname` | Indicator.Attribute | ASN Name | `data[].created` | `AS-CHOOPA, US` | Only for IP Addresses |
| `data[].attributes.asnumber` | Indicator.Attribute | ASN Number | `data[].created` | `20473` | Only for IP Addresses |
| `data[].attributes.cc` | Indicator.Attribute | Country Code | `data[].created` | `KR` | Only for IP Addresses |
| `data[].attributes.is_compromised` | Indicator.Attribute | Is Compromised | `data[].created` | `False` | Only for FQDN, Updatable |
| `data[].attributes.md5` | Related.Indicator.Value | MD5 | `data[].created` | `c445d2c98d85325377a92342ee70b078` | Only for SHA-256 |
| `data[].attributes.sha1` | Related.Indicator.Value | SHA-1 | `data[].created` | `024767fa9c5678ed0f1dc12977562b7b050df324` | Only for SHA-256 |
| `data[].attributes.sha256` | Related.Indicator.Value | SHA-256 | `data[].created` | `a1f9b76ddfdafc47d4a63a04313c577c0c2ffc6202083422b52a00803fd8193d` | Only for SHA-256 |

# Volexity - Tags

The Volexity - Tags feed ingests Volexity tags as ThreatQ objects. The feed gives the option to ingest all the data for the current point in time or to ingest only the data that was created or updated since the feed start date. The user configuration `Ingest Historical Data` is used to decide what data to ingest.

> If the `Ingest Historical Data` user configuration is enabled, the API value for `.action` is `deleted`.

## Ingest All Data

`GET https://intel-api.volexity.com/tags`

**Request Parameters:**

```
{
  "limit": 500,
  "sort": "last_modified,tag_id"
}
```

**Sample Response:**

```
{
    "after": "2024-02-12T14:57:44Z,11db9815-fd7c-432e-9f49-2853ed18a296",
    "data": [
        {
            "aliases": [
                "NINEBLOG"
            ],
            "attributes": {
                "malware_type": "remote_access_trojan"
            },
            "created": "2021-09-10T10:27:19Z",
            "description": "The 9BLOG malware family has been in use for nearly
8 years with only minimal changes. It is a VBE (VBScript encoded) based malware
with a range of initial functionality, and is typically deploy by malicious
macro documents. It is used exclusively by the TEMP_JEWEL threat actor.",
            "hyperlinks": [
                "https://www.fireeye.com/blog/threat-research/2013/08/the-
curious-case-of-encoded-vb-scripts-apt-nineblog.html"
            ],
            "last_modified": "2021-09-10T10:27:19Z",
            "related_tags": [
                "DiamondGrass",
                "TIB-20200629"
            ],
            "tag_id": "04c65b36-02b1-430d-a47c-a1e260db39a6",
            "tag_name": "9BLOG",
```

```
            "tag_type": "malware_family"
        },
        {
            "aliases": [],
            "attributes": {},
            "created": "2021-09-10T10:27:20Z",
            "description": "This is a generic tag to label anything related to
advanced persistent threats (APT).",
            "hyperlinks": [
                "https://en.wikipedia.org/wiki/Advanced_persistent_threat"
            ],
            "last_modified": "2021-09-10T10:27:20Z",
            "related_tags": [
                "ACROCLOUD",
                "DARKDROP",
                "MAR-20240221",
                "MAR-20240708",
                "Nightdoor",
                "TIB-20240221",
                "TIB-20240412",
                "TIB-20240910",
                "TIB-20240919",
                "TIB-20241011",
                "TIB-20241113",
                "UTA0208",
                "UTA0233",
                "UTA0238",
                "UTA0240",
                "UTA0252",
                "UTA0304"
            ],
            "tag_id": "0934d98e-2ca7-448c-b7a7-8ab7da77fd7a",
            "tag_name": "APT",
            "tag_type": "informational"
        },
        {
            "aliases": [],
            "attributes": {},
            "created": "2021-09-10T10:27:39Z",
            "description": "DarkHalo was the initial campaign name given to the
infamous SolarWinds hack in 2021. A previously undocumented threat actor
implanted malware into legitimate third-party source code to infect the third-
party's customers. These Trojanized files that were distributed to customers
were used to perform reconaissance against the victims, allowing Dark Halo to
further infect machines they deemed worthy.",
            "hyperlinks": [
                "https://www.volexity.com/blog/2020/12/14/dark-halo-leverages-
solarwinds-compromise-to-breach-organizations/",
                "https://blogs.microsoft.com/on-the-issues/2021/05/27/nobelium-
cyberattack-nativezone-solarwinds/",
```

```
                "https://www.fireeye.com/blog/threat-research/2020/12/evasive-
attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-
backdoor.html"
            ],
            "last_modified": "2021-09-10 10:27:39+00:00",
            "related_tags": [
                "TIB-20201214"
            ],
            "tag_id": "5f933dfd-1d5f-40e9-b1e1-0377f2b96438",
            "tag_name": "DarkHalo",
            "tag_type": "campaign"
        },
        {
            "aliases": [],
            "attributes": {
                "affected_product": "Office"
            },
            "created": "2021-09-10T10:27:38Z",
            "description": "CVE-2021-40444 is a vulnerability for Microsoft
Word which was disclosed on the 7th September 2021. It relies on a
vulnerability in the MSHTML processing engine. Since it's release multiple
researchers have been able to reproduce the exploit and its likely that
malicious actors will quickly add it to their capabilities.",
            "hyperlinks": [
                "https://msrc.microsoft.com/update-guide/vulnerability/
CVE-2021-40444"
            ],
            "last_modified": "2021-09-22 11:11:50+00:00",
            "related_tags": [
                "TIB-20210908"
            ],
            "tag_id": "3d081dd1-c874-4eb9-aea5-3e07f0364f0a",
            "tag_name": "CVE-2021-40444",
            "tag_type": "exploit"
        },
        {
            "aliases": [],
            "attributes": {
                "actor_type": "unknown",
                "attacker_origin": "unknown",
                "target_countries": ["HK"]
            },
            "created": "2021-09-23T09:33:22Z",
            "description": "This tag suggests that the related entity is not
currently classified as either APT or Crime related.",
            "hyperlinks": [],
            "last_modified": "2021-10-04 16:08:11+00:00",
            "related_tags": [
                "CEFLOADER"
            ],
```

```
            "tag_id": "6705d522-64d7-415e-b14b-0a944b12e016",
            "tag_name": "Unclassified",
            "tag_type": "threat_actor"
        },
        {
            "aliases": [],
            "attributes": {
                "report_title": "TIB-20210218 | OceanLotus Creates Fake Tor
App"
            },
            "created": "2021-11-19T16:55:39Z",
            "description": "Volexity has continued to track the Vietnamese
threat group OceanLotus and identified a new fake website, set up by the group,
purporting to be created by the Tor Project. The website was designed to appear
as if it were providing access to a program called \"Torchat\" that could help
an individual chat freely without monitoring, supervision, or censorship. In
reality, the website was run by OceanLotus and was pushing malware for systems
running Windows, OSX, and Android. Volexity\u00e2\u20ac\u2122s previous
research identified fake websites distributing Windows and OSX malware in a
similar fashion. However, this is the first website Volexity has encountered
that added a mobile platform as one of the malicious download options alongside
the desktop platforms.",
            "hyperlinks": [],
            "last_modified": "2023-01-23 15:24:56+00:00",
            "related_tags": [
                "OceanLotus",
                "PHANTOMLANCE"
            ],
            "tag_id": "1e43d264-062e-46c3-915c-3f16099f6175",
            "tag_name": "TIB-20210218",
            "tag_type": "volexity_report"
        },
        {
            "aliases": [
                "SuperJump",
                "KMA VPN",
                "UTA0269",
                "SuperJumpers"
            ],
            "attributes": {},
            "created": "2024-12-23T16:36:18Z",
            "description": "Proxy network used by Chinese threat actors.
Reported by PWC who claim the network is small (<500 hosts) and comprised of
VPS and compromised pfSense devices. Initially observed by PWC in 2021.
Associated with exploitation of Citrix, Ivanti, Fortinet, OWA and Atlassian
servers.\n\nNetworking over a random port in the 10,000 - 20,000 range.",
            "hyperlinks": [],
            "last_modified": "2024-12-23 16:36:18+00:00",
            "related_tags": [
                "DiplomaticBamboo",
```

```
                "IcedBamboo",
                "Potential ORB Network IP",
                "SatelliteBamboo",
                "SleepyBamboo"
            ],
            "tag_id": "53bb5c72-df0f-413c-9102-4e8e3c224aa2",
            "tag_name": "RelayBulb0002",
            "tag_type": "orb"
        }
    ],
    "sort": "last_modified,tag_id"
}
```

## Ingest Only Updated/Created/Deleted Data

**Request Parameters:**

```
{
  "listen": "false",
  "timestamp": "2025-04-20T00:00:00.000Z"
}
```

**Sample Response:**


```
{"action":"created","data":{"aliases":[],"attributes":
{"actor_type":"apt","attacker_origin":"unknown","target_countries":
["US","LT","UA","RO","PL"]},"created":"2025-04-02T20:15:45Z","description":"Thr
eat actor conducting OAuth phishing using insiders.vscode.dev to retrieve the
subsequent auth code.","hyperlinks":
[],"tag_id":"9af6a414-29f4-4bf3-978c-81d221b0b676","last_modified":"2025-04-02T
20:15:45Z","tag_name":"UTA0352","related_tags":
["TIB-20250402","TIB-20250404"],"tag_type":"threat_actor"},"timestamp":"2025-04
-02T20:15:45Z"}
{"action":"updated","data":{"aliases":["Elfin","HOLMIUM","Refined
Kitten"],"attributes":
{"actor_type":"apt","attacker_origin":"IR","target_countries":
["KR","SA","US"]},"created":"2021-09-10T10:27:22Z","description":"UTA0101 is an
Iranian origin threat actor best known as APT33.","hyperlinks":["https://
symantec-enterprise-blogs.security.com/blogs/threat-intelligence/elfin-
indictments-iran-espionage","https://www.crowdstrike.com/blog/who-is-refined-
kitten/"],"tag_id":"56a09f4d-8fa7-4d91-
a6d5-2ddc073fd72e","last_modified":"2025-04-03T13:25:21Z","tag_name":"UTA0101",
"related_tags":
["PupyRAT","ScotchCypress"],"tag_type":"threat_actor"},"timestamp":"2025-04-03T
13:25:21Z"}
```

## Tags Mapping Table

ThreatQuotient provides the following default mapping for both request types for this feed:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `data[].tag_name` | Object.Value | Malware, Adversary, Campaign, Vulnerability | `data[].created` | `9BLOG` | Depending on the value in `tag_type` the object is set |
| `data[].description` | Object.description | N/A | `data[].created` | `The 9BLOG malware family has been in...` | N/A |
| `data[].related_tags[]` | Object.Tags | N/A | `data[].created` | `DiamondGrass` | N/A |
| `data[].hyperlinks[]` | Object.Attribute | Hyperlink | `data[].created` | `https://www.fireeye.com/blog/threat-resear...` | N/A |
| `data[].aliases[]` | Object.Attribute | Alias | `data[].created` | `NINEBLOG` | N/A |
| `data[].attributes.malware_type` | Malware.Attribute | Malware Type | `data[].created` | `remote_access_trojan` | Only for `"tag_type"=="malware_family"` |
| `data[].attributes.actor_type` | Adversary.Attribute | Actor Type | `data[].created` | `unknown` | Only for `"tag_type"=="threat_actor"` |
| `data[].attributes.attacker_origin` | Adversary.Attribute | Attacker Origin | `data[].created` | `unknown` | Only for `"tag_type"=="threat_actor"` |
| `data[].attributes.target_countries` | Adversary.Attribute | Target Country | `data[].created` | `HK` | Only for `"tag_type"=="threat_actor"` |
| `data[].attributes.affected_product` | Vulnerability.Attribute | Affected Product | `data[].created` | `Office` | Only for `"tag_type"=="exploit"` |

## Tag Type to ThreatQ Object Mapping

The following tables provides the Volexity Tag Type to ThreatQ Object Type mapping.

| TAG_TYPE | THREATQ OBJECT TYPE |
|---|---|
| `malware_family` | Malware |
| `informational` | Adversary |
| `threat_actor` | Adversary |
| `orb` | Adversary |
| `campaign` | Campaign |
| `exploit` | Vulnerability |
| `volexity_report` | Report |

# Volexity - Signatures

The Volexity - Signatures feed ingests Volexity Yara signatures as ThreatQ signatures. The feed gives the option to ingest all the data for the current point in time or to ingest only the data that was created or updated since the feed start date. The user configuration `Ingest Historical Data` is used to decide what data to ingest.

> If the `Ingest Historical Data` user configuration is enabled, the API value for `.action` is `deleted`.

## Ingest All Data

`GET https://intel-api.volexity.com/yara`

**Request Parameters:**

```
{
  "limit": 500,
  "sort": "last_modified,rule_id"
}
```

**Sample Response:**

```
{
    "data": [
        {
            "created": "2021-08-13T10:35:59Z",
            "rule_id": 5838,
            "last_modified": "2021-08-19T14:58:27Z",
            "name": "cf_office_macro_audiofile",
            "os": [
                "android",
                "darwin",
                "linux",
                "win"
            ],
            "os_arch": [
                "arm",
                "x86",
                "x64",
                "mips"
            ],
            "raw_rule": "rule cf_office_macro_audiofile: Unclassified\n{\n
meta:\n        author = \"threatintel@volexity.com\"\n        description =
\"Looks for macro content in MHT documents used by possible APT actor.\"\n
date = \"2021-08-13\"\n        hash1 =
\"507c319c6c0650497a297d9ca0d46ca0595c59f8aa466d599cc4839d9fe0c1bc\"\n
scan_context = \"file\"\n        last_modified =
\"2021-08-19T14:58:27.748156Z\"\n        license = \"Please see the license at
```

```
the head of this rules file for acceptable use. If you did not receive a
license please contact threatintel@volexity.com\"\n        rule_id = 5838\n
version = 3\n\n    strings:\n        $s1 = \"fileStream.Write
Chr( AscB( MidB( objHTTP\" ascii\n        $s2 = \"Call objHTTP.Open(\\\"POST\\
\", strURL, FALSE)\" ascii\n        $s3 = \".Calc()\" ascii\n\n    condition:\n
2 of them\n}",
            "scan_context": [
                "file"
            ],
            "tags": [
                "Unclassified"
            ]
        },
        {
            "created": "2021-08-13T10:31:42Z",
            "rule_id": 5837,
            "last_modified": "2021-08-19T14:58:28Z",
            "name": "general_office_mht_with_activemime",
            "os": [
                "android",
                "darwin",
                "linux",
                "win"
            ],
            "os_arch": [
                "arm",
                "x86",
                "x64",
                "mips"
            ],
            "raw_rule": "rule general_office_mht_with_activemime: General\n{\n
meta:\n        author = \"threatintel@volexity.com\"\n        description =
\"Looks for MHT documents exported from Word containing ActiveMime objects.\"\n
date = \"2021-08-13\"\n        hash1 =
\"507c319c6c0650497a297d9ca0d46ca0595c59f8aa466d599cc4839d9fe0c1bc\"\n
scan_context = \"file\"\n        last_modified =
\"2021-08-19T14:58:28.644144Z\"\n        license = \"Please see the license at
the head of this rules file for acceptable use. If you did not receive a
license please contact threatintel@volexity.com\"\n        rule_id = 5837\n
version = 3\n\n    strings:\n        $mht_header = \"MIME-Version:\"\n
$word = \"Generator content=3D\\\"Microsoft Word\"\n        $base64_hdr =
\"QWN0aXZlTWltZQAAAfAEAAAA\" ascii\n\n    condition:\n        $mht_header in
(0..32) and\n        $word and\n        $base64_hdr\n}",
            "scan_context": [
                "file"
            ],
            "tags": [
                "General"
            ]
        },
        {
```

```
            "created": "2021-08-11T11:07:26Z",
            "rule_id": 5830,
            "last_modified": "2021-08-19T14:58:34Z",
            "name": "apt_win_csloader_gonet_b",
            "os": [
                "android",
                "darwin",
                "linux",
                "win"
            ],
            "os_arch": [
                "arm",
                "x86",
                "x64",
                "mips"
            ],
            "raw_rule": "rule apt_win_csloader_gonet_b: Unclassified\n{\n
meta:\n        author = \"threatintel@volexity.com\"\n        description =
\"Detects a custom CobaltStrike loader.\"\n        date = \"2021-08-11\"\n
hash1 = \"595c15c61ddad76dfbca456a54ef499c1f86ed0d65b1d5a3dc10dfc5017430e3\"\n
scan_context = \"file\"\n        last_modified =
\"2021-08-19T14:58:34.551181Z\"\n        license = \"Please see the license at
the head of this rules file for acceptable use. If you did not receive a
license please contact threatintel@volexity.com\"\n        rule_id = 5830\n
version = 4\n\n    strings:\n        $mutex = \"!@#afssafasfasfs\" wide\n\n
$s_method1 = \"GetModuleHandle\" ascii fullword\n        $s_method2 =
\"LoadLibrary\" ascii fullword\n        $s_method3 = \"VirtualAlloc\" ascii
fullword\n        $s_method4 = \"GetProcAddress\" ascii fullword\n
$s_method5 = \"A_mutex\" ascii\n        $s_xor_key = {546388ff635174e963}\n
condition:\n        $mutex or\n        (\n            all of ($s*) and\n
filesize < 250KB\n        )\n}",
            "scan_context": [
                "file"
            ],
            "tags": [
                "Unclassified"
            ]
        }
    ],
    "after": "2021-08-19T14:58:34Z,5830",
    "sort": "last_modified,rule_id"
}
```


## Ingest Only Updated/Created/Deleted Data

**Request Parameters**


```
{
  "listen": "false",
```

```
  "timestamp": "2025-04-20T00:00:00.000Z"
}
```

**Sample Response**

```
{"action":"created","data":
{"created":"2022-10-03T12:58:42Z","rule_id":8422,"last_modified":"2022-10-03T12
:58:42Z","maximum_yara_ver":"4.5.2","minimum_yara_ver":"3.5.0","name":"apt_win_
unc_cn_delphi_loader","os":["android","darwin","linux","win"],"os_arch":
["arm","x86","x64","mips"],"raw_rule":"rule apt_win_unc_cn_delphi_loader:
Unclassified\n{\n    meta:\n        author = \"threatintel@volexity.com\"\n
description = \"Detects a Delphi loader used to load an unspecified .dat file.
\"\n        date = \"2022-10-03\"\n        hash1 =
\"df9a2471c23790a381e286bb96ea3401b94686b7ca067297a7920a76a7202112\"\n        /
/ Note that in the reference this file is listed as loading NBTScan but that
cannot\n        // be verified at this time.\n        reference = \"https://
symantec-enterprise-blogs.security.com/blogs/threat-intelligence/espionage-
asia-governments\"\n        scan_context = \"file,memory\"\n        severity =
\"critical\"\n        last_modified = \"2022-10-03T12:58:42Z\"\n        license
= \"Please see the license at the head of this rules file for acceptable use.
If you did not receive a license please contact threatintel@volexity.com\"\n
rule_id = 8422\n        version = 1\n\n    strings:\n        $w1 = \"fail to
open .dat file\" wide\n        $w2 = \"fail to get the file size\" wide\n
$w3 = \"fail to read file\" wide\n\n    condition:\n        2 of
them\n}","scan_context":["file","memory"],"tags":
["Unclassified"]},"timestamp":"2022-10-03T12:58:42Z"}
```

## Signatures Mapping Table

ThreatQuotient provides the following default mapping for both request types for this feed:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `data[].name` | Signature.Name | N/A | `data[].created` | `apt_win_csloader_gonet_b` | N/A |
| `data[].raw_rule` | Signature.Value | N/A | `data[].created` | `rule apt_win_csloader_gonet_b: Uncla...` | N/A |
| `data[].tags[]` | Signature.Tags | N/A | `data[].created` | `Unclassified` | N/A |
| `data[].scan_context[]` | Signature.Attribute | Scan Context | `data[].created` | `file` | N/A |
| `data[].os_arch[]` | Signature.Attribute | Operating System Architecture | `data[].created` | `arm` | N/A |
| `data[].os[]` | Signature.Attribute | Operating System | `data[].created` | `darwin` | N/A |

# Volexity - Reports

The Volexity - Reports feed ingests Volexity Reports. For each Volexity Report is created a ThreatQ Report Object.

The following objects can be related to the ThreatQ Report depending on user configurations:

`GET https://intel-api.volexity.com/reports`

- The PDF generated by Volexity for the Report - it is ingested as a ThreatQ File object if Ingest Related Volexity PDF Report is enabled.
- The indicators present in the PDF Report and in the YARA related rules - they are ingested according to the values enabled in Parsed IOC Types configuration.
- The YARA signatures referenced in the PDF Report - they are ingested if Ingest Related YARA Signatures is enabled.
- The Snort signatures referenced in the PDF Report - they are ingested if Ingest Related Snort Signatures is enabled.

**Request Parameters**

```
{
  "sort": "id",
  "modified":"2025-05-01T00:00:00Z"
}
```

**Sample Response**

```
{
  "data": [
    {
      "id": "kA1Pg0000000ZH7KAM",
      "attachments": [
        "rules.yar",
        "rules.rules",
        "iocs.csv"
      ],
      "last_modified": "2025-04-29T13:12:06Z",
      "report_title": "TIB-20240227 | MACMA-GIMMICK Fusion and RELOADEXT Chrome
Extension Analysis",
      "report_tag": "TIB-20240227",
      "topics": [
        "RELOADEXT",
        "StormBamboo",
        "China",
        "GIMMICK",
        "MACMA"
      ]
    }
  ],
  "sort": "id"
}
```

## Reports Mapping Table

ThreatQ provides the following default mapping for this feed:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `data[].report_title` | Report.Value | Report | N/A | `TIB-20240227 \| MACMA-GIMMICK Fusion and RELOADEXT Chrome Extension Analysis` | N/A |
| `data[].topics[]` | Report.Attribute | Topic | N/A | `RELOADEXT` | N/A |

## iocs.csv

CSV files having `iocs` or `indicators` in their name in `data[].attachments[]` are parsed to ingest the indicators according to `Parsed IOC Types` configuration

Example of `iocs.csv` file:

```
value,entity_type,description
singist.xyz,hostname,"Check-in domain for Charming Kitten maldoc"
bookreadersale.xyz,hostname,"C2 domain for PowerStar backdoor"
1thebstack1.xyz,hostname,"Related Charming Kitten infrastructure"
```

ThreatQ provides the following default mapping for CSV files:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `token(0)` | Indicator.Value | `token(1)` | N/A | `singist.xyz` | Type found using table `Volexity Entities Indicator Type Mapping` |
| `token(2)` | Indicator.Description | N/A | N/A | `Check-in domain for Charming Kitten maldoc` | N/A |

## rules.yar

YAR files from `data[].attachments[]` are parsed to ingest YARA signatures and indicators. Indicators are parsed according to `Parsed IOC Types` configuration

Example of `rules.yar` file:

```
rule apt_win_powerstar : CharmingKitten
{
    meta:
        author = "threatintel@volexity.com"
        description = "Custom PowerShell backdoor used by Charming Kitten."
        date = "2021-10-13"
        hash1 =
"de99c4fa14d99af791826a170b57a70b8265fee61c6b6278d3fe0aad98e85460"
        memory_suitable = 1
        severity = critical
    strings:
        $appname = "[AppProject.Program]::Main()" ascii wide // caller for C#
code
        $langfilters1 = "*shar*" ascii wide
        $langfilters2 = "*owers*" ascii wide
        $definitions1 = "[string]$language" ascii wide
        $definitions2 = "[string]$Command" ascii wide
        $definitions3 = "[string]$ThreadName" ascii wide
        $definitions4 = "[string]$StartStop" ascii wide
        $sess = "$session = $v + \";;\" + $env:COMPUTERNAME + $mac;" ascii wide

    condition:
        $appname or
        all of ($langfilters*) or
        all of ($definitions*) or
        $sess
}
```

ThreatQ provides the following default mapping for YAR files:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `*` | Signature.Value | YARA | `meta.date` | N/A | Entire text. |
| `rule * : {TAG}` | Signature.Tag | N/A | N/A | CharmingKitten | Value after `:` is added as tag. |
| `rule *` | Signature.Name | YARA | `meta.date` | `apt_win_powerstar` | N/A |
| `meta` | Signature.Attribute | All values before = sign except of `description`, `date`, `hash*`. Title cased. | `meta.date` | N/A | Attributes `Severity` and `Last Modified` are updatable if present. |
| `meta.description` | Signature.Description | N/A | N/A | `Custom PowerShell backdoor used by Charming Kitten.` | N/A |
| `meta.hash*` | Related Indicator.Value | Type determined based on length. | `meta.date` | `de99c4fa14d99af791826a170b57a70b8265fee61c6b6278d3fe0aad98e85460` | User-configurable. |

## rules.rules

Rules files from `data[].attachments[]` are parsed to ingest Snort signatures.

Example of `rules.rules` file:

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"Volex - PowerStar Sleep
response (NoComm) at 0"; http.response_body;content:"NoComm"; offset:0;
depth:6; isdataat:!1,relative; sid:2021101301; priority:2;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"VLX - CharmingCypress
meeting URI pattern - Nov 24"; http.method; content:"GET"; http.uri; content:"/
join-meeting?v="; startswith; sid:2024120601; rev:3; metadata:created_at
2024_12_06,updated_at 2024_12_06,signature_severity Major;)
```

ThreatQ provides the following default mapping for YAR files:

| FEED DATA PATH | THREATQ ENTITY | THREATQ OBJECT TYPE OR ATTRIBUTE KEY | PUBLISHED DATE | EXAMPLES | NOTES |
|---|---|---|---|---|---|
| `*` | Signature.Value | Snort | N/A | N/A | Entire text. |
| `msg` | Signature.Name | Snort | N/A | `Volex - PowerStar Sleep response (NoComm) at 0` | N/A |
| `metadata` | Signature.Attribute | All values before sign except of `tag` in the metadata section. | N/A | N/A | Attributes `Severity` and `Last Modified` are updatable if present. |

# 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.

## Volexity - Entities

Scenario - **Ingest Historical Data** parameter is enabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 2 hours 20 minutes |
| **Indicators** | 336,832 |
| **Indicator Attributes** | 852,270 |

Scenario - **Ingest Historical Data** parameter is disabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 2 minutes |
| **Indicators** | 500 |
| **Indicator Attributes** | 1,280 |

# Volexity - Tags

Scenario - **Ingest Historical Data** parameter is enabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 2 min |
| **Adversaries** | 365 |
| **Adversaries Attributes** | 3,221 |
| **Campaigns** | 24 |
| **Campaigns Attributes** | 28 |
| **Malware** | 781 |
| **Malware Attributes** | 1,994 |
| **Vulnerabilities** | 73 |
| **Vulnerabilities Attributes** | 186 |

Scenario - **Ingest Historical Data** parameter is disabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 1 min |
| **Adversaries** | 3 |
| **Adversaries Attributes** | 40 |

# Volexity - Signatures

Scenario - **Ingest Historical Data** parameter is enabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 2 minutes |
| **Signature** | 5,044 |
| **Signature Attributes** | 17,200 |

Scenario - **Ingest Historical Data** parameter is disabled.

| METRIC | RESULT |
|---|---|
| **Run Time** | 1 minute |
| **Signatures** | 6 |
| **Signature Attributes** | 18 |

# Volexity - Reports

| METRIC | RESULT |
|---|---|
| **Run Time** | 2 minutes |
| **Files** | 6 |
| **Indicators** | 100 |
| **Reports** | 6 |
| **Report Attributes** | 30 |
| **Signatures** | 10 |
| **Signature Attributes** | 40 |

# Change Log

- **Version 1.1.0**
  - Added the ability to choose whether to ingest historical or new data.
  - Added the ability to remove from ThreatQ objects deleted by Volexity.
  - Removed functionality to ingest reports from `Get Tags` feed.
  - Removed the following configuration parameters:
    - Objects per page
    - Number of Pages
  - **Volexity Entities** - added the following configuration parameters:
    - **Ingest Historical Data**: ingest all the available data for the current point in time.
    - **Ingest Tags**: ingest the tags.
    - **Remove Deleted Entities**: remove from ThreatQ all the indicators with a sole source of `Volexity - Entities` that have been deleted in Volexity.
  - **Volexity Tags** - added the following configuration parameters:
    - **Ingest Historical Data**: ingest all the available data for the current point in time.
    - **Ingest Tags**: ingest the tags.
    - **Remove Deleted Entities**: remove from ThreatQ all the objects with a sole source of `Volexity - Tags` that have been deleted in Volexity.
  - **Volexity Signatures** - added the following configuration parameters:
    - **Ingest Historical Data**: ingest all the available data for the current point in time.
    - **Ingest Tags**: ingest the tags.
    - **Remove Deleted Entities**: remove from ThreatQ all the indicators with a sole source of `Volexity - Signatures` that have been deleted in Volexity.
  - Added a new feed: **Volexity - Reports**.
- **Version 1.0.0**
  - Initial release