1. **Cyware’s CyTaxii2**
2. **CYTAXII2, Cyware's TAXII 2.x Client**

**Introduction**

**CYTAXII2** is an **Open Source** offering from Cyware that provides developers with the support for interacting with the TAXII server using a Python library.

TAXII (Trusted Automated eXchange of Indicator Information) is a collection of specifications defining a set of services and message exchanges used for sharing cyber threat intelligence information between parties.

CYTAXII2 acts as a TAXII client that you can install as a Python [Pip] Library. It implements all TAXII services according to TAXII 2.X specifications such as Consume intel from sources such as Cyware Threat feeds or any other sources that send you Threat Intel in STIX format through TAXII protocol. (Poll) Contribute and send intel to a collection on a TAXII server. (Inbox)

**CYTAXII2 use cases**

Here are some ways to use Cyware’s CYTAXII2.

**Use CYTAXII2 with a Threat Intelligence Platform**

If you are an organization that uses a Threat Intelligence Platform such as CTIX, or any other TIP, you can use CYTAXII2 to perform some basic orchestration services to accelerate your incident detection and response capabilities.

You can fetch information from a collection in CTIX and orchestrate this information into a (Security Information and Event Management) SIEM tool to gain better insights from your collated logs.

You can fetch information from a collection in a TIP and orchestrate this information to an Endpoint Detection and Response (EDR) tool to successfully block malicious endpoints.

You can fetch information from a collection and put that information in a different collection on a new tool or a platform.

Use CYTAXII2 without a Threat Intelligence Platform

If you are a security analyst or a user and do not have access to technology such as a Threat Intelligence Platform (TIP), you can use CYTAXII2 to perform some basic TAXII server operations like consuming intel from a collection (poll) or contribute to intel into a collection (inbox).

**Use CYTAXII2 with Cyware and other STIX feeds**

You can use CYTAXII2 with Cyware’s open-source Cyware Threat Intelligence Feeds or any STIX threat intelligence feeds and fetch valuable intel or contribute your intel to a collection on the TAXII server.

Cyware Threat Intelligence Feeds provide you with valuable threat data from a wide range of open and trusted sources to deliver a consolidated stream of valuable and actionable threat intelligence.

When you sign up for Cyware Threat Intelligence feeds, you will receive TAXII server credentials that include user name, password, and TAXII Server URLs. You can use these credentials to access the threat intel data in Cyware Threat feeds. Along with the TAXII client (CYTAXII2) you now also have TAXII server credentials from Cyware.

**Key Takeaway**

In conclusion, CYTAXII2 is a unique and open-sourced TAXII 2.X client offering from Cyware that

* you can use in a multitude of ways,
* is developer-friendly,
* and delivers threat intel data in the universally accepted STIX 2.X format.

**Installation**

CYTAXII2 works as a Python library. Use the following procedure to install CyTaxii 2 and get started.

1. Install Python.

While CyTaxii2 works with all versions of python3 it is developed with python 3.8. You can install Python from [here](https://www.python.org/downloads/).

1. Install CYTAXII2.

You can now install the CYTAXII2 using the [Python Package Index](http://pypi.python.org/) (PyPI) command, pip as shown below.

**pip install cytaxii2**

1. Import the Cyware TAXII2 command library.

Use the following command to import the complete CYTAXII2 command library so that you can start performing basic STIX/ TAXII actions

**from cytaxii2 import cytaxii2**

1. Create a CYTAXII2 object with the TAXII server credentials. Here you are associating the CYTAXII2 object with a TAXII server. You will need the following information from the TAXII Server.

* **username**
* **password**
* **TAXII Server URL.**

Use the following commands to create an object with the TAXII URL, username, password, and TAXII version (2.0 or 2.1).

cytaxii\_object = cytaxii2.cytaxii2('taxii\_discovery\_url', 'taxii\_username', 'taxii\_password, version=taxii\_version)

**Use Cases**

**1. Taxii Discovery Request**

Perform a TAXII discovery request to check that your TAXII server URL credentials are valid. This request also returns some basic information about the TAXII Server such as title, description, contact, root URL, etc.

You can further use the Root URL received from this request to make a Root discovery request. This method does not need any arguments.

discovery\_response = cytaxii\_object.discovery\_request()

[Graphical user interface, text

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img1.png)

**2. Root Discovery Request**

Perform a root discovery request to get more information about the TAXII server. This method returns you details about the TAXII server such as TAXII version, content length, and organization-specific data. This method does not need any arguments.

root\_discovery\_response = cytaxii\_object.root\_discovery()

[Graphical user interface, text, application

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img5.png)

**3. Request the collection list**

In a TAXII server, the Cyber Threat Intelligence(CTI) data is grouped into **Collections**. Use the below command to request the list of STIX data collections that you have access to, from a TAXII server.

In addition to listing all the collections that you have access to, this method will also return information about each STIX collection such as

* Collection ID
* Title
* Description
* If you have read or write access to the collection.
* Format of data in the collection

You can use the information returned by this method to check for details such as your read or write access to this collection. To poll or receive information from a collection, you should have read access to the collection. To contribute or inbox to a collection, you should have write access to it. This method does not need any arguments.

collections = cytaxii\_object.collection\_request()

**[Text

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img6.png)**

**4. Request data about a collection**

Use the following command to get data about one particular collection in the TAXII Server. You have to pass the collection ID of that collection as a mandatory argument for this function.

This method will return the following information for one collection with your collection ID.

* Collection ID
* Title
* Description
* If you have read or write access to the collection.
* Format of data in the collection

collection\_data = cytaxii\_object.collection\_data\_request(collection\_id="collection\_id")

[Graphical user interface, text, website

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img4.png)

**5. Poll data from a TAXII server**

Use this method to poll for data from a TAXII server. You can poll for data present in one particular collection on the TAXII server. So, a Collection ID that is used to uniquely identify a collection is a mandatory argument for this request.

**Note**: To poll for data from a collection, you should have **READ** access to the collection.

Along with Collection ID, here are some other optional arguments that you can pass to this method. Use these optional arguments to refine or filter the threat intel data received from this collection.

* **collection\_id** - Enter the collection ID
* **added\_after** - Enter a DateTime string to get data after. If you don’t pass this argument, all data is returned without a date limitation.
* **data\_limit** - Enter the number of responses to return.
* **object\_id**: Enter the ID of the object.
* **object\_type**: Enter the type of object that you want to poll for.
* **next**: Enter a number like 2, 3, etc.

poll\_response = cytaxii\_object.poll\_request(collection\_id='collection\_id', added\_after=None, limit=None, object\_id=None)

[Text

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img3.png)

**5.1 Poll for data by an Object Type**

To poll for data in a collection by an object Type such as Malware, hash, vulnerability, indicator, etc, pass this as an argument in **object\_type** using the same function.

If you pass Malware as the object\_type, you will only see malware related data from the STIX collection.

**5.2 Poll for data by an object ID**

To poll for data in a collection by an object ID, pass this as an argument in **object\_type** using the same function.

If you pass indicator-67hjhkhk78-89hi-iiu9-kjj8-iuo8hgjggkg as the **object\_id**, you will see the data only for this ID.

**5.3 Poll for data after a given date**

To poll for data in a collection that is present after a particular date, pass **added\_after** as an argument. You will see the data from your date and the current date.

**5.4 Poll for additional data**

When data is huge and is paginated, use the **next** argument to view the additional data.

**5.5 Poll for a limited Data**

To view only limited data, pass **limit** as an argument and you will only see limited data fetched from your collection.

**6. Inbox STIX data to the TAXII server**

You can contribute and send STIX data to the TAXII server. Make sure that the data you send to the TAXII server is a valid JSON that has valid STIX data corresponding to the correct version of the TAXII server.

You can use the inbox request to send STIX data to the TAXII server. You have to send the data to a STIX collection on the TAXII server. So you have to pass the collection ID of the STIX collection as a mandatory argument to this method along with the STIX package.

**Note**: To inbox data into a collection, you should have **WRITE** access to the collection.

Arguments

* **collection\_id**: Enter the collection ID to send the data to the TAXII server.
* **stix\_bundle**: Enter the STIX bundle to send to the TAXII server.

See the following example for a valid STIX JSON format.

stix\_json = """

{

"objects": [

{

"id": "identity--a9e5f832-3bd2-41bb-b543-05341539c223",

"identity\_class": "organization",

"lang": "en",

"name": "CTIX",

"type": "identity",

"spec\_version": "2.1",

"created": "2021-06-02T05:58:46.325759Z",

"modified": "2021-06-02T05:58:46.325759Z"

},

{

"id": "marking-definition--f88d31f6-486f-44da-b317-01333bde0b82",

"created": "2017-01-20T00:00:00.000Z",

"definition\_type": "tlp",

"name": "TLP:AMBER",

"definition": {

"tlp": "amber"

},

"type": "marking-definition",

"spec\_version": "2.1"

},

{

"id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",

"created": "2017-01-20T00:00:00.000Z",

"definition\_type": "tlp",

"name": "TLP:GREEN",

"definition": {

"tlp": "green"

},

"type": "marking-definition",

"spec\_version": "2.1"

}

],

"more": true,

"next": "2"

}

"""

Here is the inbox request:

inbox\_response = cytaxii\_object.inbox\_request(collection\_id="collection\_id", stix\_bundle=stix\_json)

[Graphical user interface, text

Description automatically generated](https://github.com/cyware-labs/cytaxii2/blob/main/images/img2.png)

**7. Responses**

All the responses received from the methods in the CYTAXII2 library are pure JSON python dictionaries, and return data in the below format. You can modify or use them just as you do with any JSON object.

{

"response":"action\_response",

"status":"true/ false",

"status\_code":"status\_code\_given\_by\_server"

}

**Note**: You can use the example Jupyter notebook that are included in the CYTAXII2 repository as a reference towards getting started

**Glossary**

**STIX**

STIX (Structured Threat Information eXpression), is a standard language for describing cyber threat information.

**STIX Collection**

Cyber Threat Intelligence data is stored in the TAXII servers in the form of collections. A collection is a grouping of CTI data.

**Poll**

You can send a Poll request to fetch or pull CTI information from the TAXII servers.

**Inbox**

You can send an inbox request to put your information into a collection on the TAXII servers.

**Discovery**

Discovery enables you to learn what services an entity supports and how to interact with them.

**CYTAXII2, Cyware's TAXII 2.x Client**

cytaxii2 is a pip library [for python 3] by Cyware that can be used to connect to and, discover, get collections, poll and inbox data from and to a TAXII 2.x server. This supports stix/taxii 2, 2.1

This library supports python3 and can be installed by "pip3 install cytaxii2". After installation the library can be imported as follows:

In [ ]:

**from** **cytaxii2** **import** cytaxii2

After importing, we have to create an object with the TAXII URL, username and password and TAXII version [2.0/ 2.1], as follows:

In [ ]:

cytaxii\_object = cytaxii2.cytaxii2(discovery\_url='taxii\_discovery\_url',

username='taxii\_user\_name',

password='taxii\_password',

version=2.1)

After the object is created we can call the methods present in the library. Each of the methods are highlighted below.

The first method is where we can perform a TAXII discovery request and can be done as follows.

***Arguments***

None

In [ ]:

discovery\_response = cytaxii\_object.discovery\_request()

Next, we can also perform a root discovery, where we get information about the root server and can be done as follows.

***Arguments***

None

In [ ]:

root\_discovery\_response = cytaxii\_object.root\_discovery()

Next, we can perform a collection request, where we request all STIX data collections from the TAXII server. We must note in TAXII all data is held inside a collection and a list of collections can be requested as follows.

***Arguments***

None

In [ ]:

collections = cytaxii\_object.collection\_request()

Next, we can also get data about a particular TAXII collection URL and this can be done as follow.

***Arguments:***

* collection\_id: Enter the collection ID to get details of **[mandatory]**

In [ ]:

collection\_data = cytaxii\_object.collection\_data\_request(collection\_id="34e7e24a-1bad-48f1-b88d-bdfd5f99a6bd")

Next, we can get data from a TAXII collection. This is one of the most important use cases, where we can poll data from a TAXII server.

***Arguments:***

* collection\_id: Enter the collection ID to get details of **[mandatory]**
* added\_after: Enter a datetime string to parse. Data will be returned only after this date till current data. Defaults to all data
* limit: Enter the number of responses to return
* object\_id: Enter the object\_id. This param will limit the response to only this object in the response
* object\_type: Enter the object type to return
* next: If the data is paginated, the "next" param can be used to get additional data

In [ ]:

poll\_response = cytaxii\_object.poll\_request(collection\_id='34e7e24a-1bad-48f1-b88d-bdfd5f99a6bd', added\_after=**None**, limit=**None**, object\_id=**None**, next=**None**, object\_type=**None**)

Next, we can also send data AKA inbox STIX data to the TAXII server. To do this, we define the STIX package to send as seen below. We must note that we must send only valid json and STIX data which is respective to the TAXII server

In [ ]:

stix\_json = """

{

"objects": [

{

"id": "identity--a9e5f832-3bd2-41bb-b543-05341539c223",

"identity\_class": "organization",

"lang": "en",

"name": "CTIX",

"type": "identity",

"spec\_version": "2.1",

"created": "2021-06-02T05:58:46.325759Z",

"modified": "2021-06-02T05:58:46.325759Z"

},

{

"id": "marking-definition--f88d31f6-486f-44da-b317-01333bde0b82",

"created": "2017-01-20T00:00:00.000Z",

"definition\_type": "tlp",

"name": "TLP:AMBER",

"definition": {

"tlp": "amber"

},

"type": "marking-definition",

"spec\_version": "2.1"

},

{

"id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",

"created": "2017-01-20T00:00:00.000Z",

"definition\_type": "tlp",

"name": "TLP:GREEN",

"definition": {

"tlp": "green"

},

"type": "marking-definition",

"spec\_version": "2.1"

}

],

"more": true,

"next": "2"

}

"""

Lastly, we can inbox [send] the above defined STIX data to the TAXII server using the inbox request as seen below.

***Arguments***

* collection\_id: Enter the collection ID to get details of **[mandatory]**
* stix\_bundle: Enter the STIX bundle to send to the TAXII server

In [ ]:

inbox\_response = cytaxii\_object.inbox\_request(collection\_id="a3ca32aa-5441-423e-a6e9-a28f949e011b", stix\_bundle=stix\_json)

The responses can be viewed by printing the response below. We must note that the data returned is pure JSON with a structure like the follows:

{ response: response\_json, status: request\_status [True/ False], status\_code: response\_status\_code }

In [ ]:

print(inbox\_response)