

---

# OpenCTI client for Python

*Release 3.2.4*

Jun 03, 2020



---

## Contents:

---

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Using the helper functions . . . . .	3
<b>2</b>	<b>pycti</b>	<b>5</b>
2.1	Functions . . . . .	5
2.2	Classes . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>29</b>



The pycti library is designed to help OpenCTI users and developers to interact with the OpenCTI platform GraphQL API.

The Python library requires Python  $\geq 3$ .



# CHAPTER 1

---

## Getting Started

---

### 1.1 Installation

Please install the latest pycti version available from PyPI:

```
$ pip3 install pycti
```

### 1.2 Using the helper functions

The main class `OpenCTIApiClient` contains all what you need to interact with the platform, you just have to initialize it. The following example shows how you create an indicator in OpenCTI using the python library with TLP marking and OpenCTI compatible date format.

```
from dateutil.parser import parse
from pycti import OpenCTIApiClient
from stix2 import TLP_GREEN

# OpenCTI API client initialization
opencti_api_client = OpenCTIApiClient("https://myopencti.server", "mysupersecrettoken
↪")

# Define an OpenCTI compatible date
date = parse("2019-12-01").strftime("%Y-%m-%dT%H:%M:%SZ")

# Get the OpenCTI marking for stix2 TLP_GREEN
TLP_GREEN_CTI = opencti_api_client.marking_definition.read(id=TLP_GREEN["id"])

# Use the client to create an indicator in OpenCTI
indicator = opencti_api_client.indicator.create(
    name="C2 server of the new campaign",
    description="This is the C2 server of the campaign",
```

(continues on next page)

(continued from previous page)

```
pattern_type="stix",
indicator_pattern="[domain-name:value = 'www.5z8.info']",
main_observable_type="IPv4-Addr",
valid_from=date,
update=True,
markingDefinitions=[TLP_GREEN_CTI["id"]],
)
```



- *Functions*
- *Classes*

## 2.1 Functions

- `get_config_variable()`: [summary]

`pycti.get_config_variable(env_var, yaml_path, config={}, isNumber=False)`  
[summary]

### Parameters

- **env\_var** (str) – environnement variable name
- **yaml\_path** (list) – path to yaml config
- **config** (Dict) – client config dict, defaults to {}
- **isNumber** (Optional[bool]) – specify if the variable is a number, defaults to False

**Return type** Union[bool, int, None, str]

## 2.2 Classes

- *OpenCTIApiClient*: Main API client for OpenCTI
- *OpenCTIApiConnector*: OpenCTIApiConnector
- *OpenCTIApiJob*: OpenCTIApiJob
- *ConnectorType*: An enumeration.

- *OpenCTIConnector*: Main class for OpenCTI connector
- *OpenCTIConnectorHelper*: Python API for OpenCTI connector
- *Tag*: Undocumented.
- *MarkingDefinition*: Undocumented.
- *ExternalReference*: Undocumented.
- *KillChainPhase*: Undocumented.
- *StixEntity*: Undocumented.
- *StixDomainEntity*: Undocumented.
- *StixObservable*: Undocumented.
- *StixRelation*: Undocumented.
- *StixSighting*: Undocumented.
- *StixObservableRelation*: Undocumented.
- *Identity*: Undocumented.
- *ThreatActor*: Main ThreatActor class for OpenCTI
- *IntrusionSet*: Undocumented.
- *Campaign*: Undocumented.
- *Incident*: Undocumented.
- *Malware*: Undocumented.
- *Tool*: Undocumented.
- *Vulnerability*: Undocumented.
- *AttackPattern*: Undocumented.
- *CourseOfAction*: Undocumented.
- *Report*: Undocumented.
- *Note*: Undocumented.
- *Opinion*: Undocumented.
- *Indicator*: Main Indicator class for OpenCTI
- *OpenCTIStix2*: Python API for Stix2 in OpenCTI
- *ObservableTypes*: These are the possible values for OpenCTI's observable types.
- *CustomProperties*: These are the custom properties used by OpenCTI.

**class** `pycti.OpenCTIApiClient` (*url*, *token*, *log\_level*=*'info'*, *ssl\_verify*=*False*)  
Main API client for OpenCTI

### Parameters

- **url** (*str*) – OpenCTI API url
- **token** (*str*) – OpenCTI API token
- **log\_level** (*str*, *optional*) – log level for the client
- **ssl\_verify** (*bool*, *optional*) –

## Inheritance

OpenCTIApiClient

**fetch\_opencti\_file** (*fetch\_uri*, *binary=False*)

get file from the OpenCTI API

**Parameters**

- **fetch\_uri** (*str*) – download URI to use
- **binary** (*bool*, *optional*) – [description], defaults to False

**Returns** returns either the file content as text or bytes based on *binary*

**Return type** str or bytes

**get\_logs\_worker\_config** ()

get the logsWorkerConfig

return: the logsWorkerConfig rtype: dict

**get\_token** ()

Get the API token

**Returns** returns the configured API token

**Return type** str

**health\_check** ()

submit an example request to the OpenCTI API.

**Returns** returns *True* if the health check has been successful

**Return type** bool

**log** (*level*, *message*)

log a message with defined log level

**Parameters**

- **level** (*str*) – must be a valid logging log level (debug, info, warning, error)
- **message** (*str*) – the message to log

**not\_empty** (*value*)

check if a value is empty for str, list and int

**Parameters** **value** (*str or list or int or float or bool or datetime.date*) – value to check

**Returns** returns *True* if the value is one of the supported types and not empty

**Return type** bool

**process\_multiple** (*data*, *with\_pagination=False*)

processes data returned by the OpenCTI API with multiple entities

**Parameters**

- **data** (*dict*) – data to process
- **with\_pagination** – whether to use pagination with the API

**Return type** Union[dict, list]

**Returns** returns either a dict or list with the processes entities

**process\_multiple\_fields** (*data*)

processes data returned by the OpenCTI API with multiple fields

**Parameters** **data** (*dict*) – data to process

**Returns** returns the data dict with all fields processed

**Return type** dict

**process\_multiple\_ids** (*data*)

processes data returned by the OpenCTI API with multiple ids

**Parameters** **data** – data to process

**Return type** list

**Returns** returns a list of ids

**query** (*query*, *variables={}*)

submit a query to the OpenCTI GraphQL API

**Parameters**

- **query** (*str*) – GraphQL query string
- **variables** (*dict*, *optional*) – GraphQL query variables, defaults to {}

**Returns** returns the response json content

**Return type** Any

**resolve\_role** (*relation\_type*, *from\_type*, *to\_type*)

resolves the role for a specified entity

**Parameters**

- **relation\_type** (*str*) – input relation type
- **from\_type** (*str*) – entity type
- **to\_type** (*str*) – entity type

**Returns** returns the role mapping

**Return type** dict

**set\_token** (*token*)

set the request header with the specified token

**Parameters** **token** (*str*) – OpenCTI API token

**upload\_file** (*\*\*kwargs*)

upload a file to OpenCTI API

**Parameters** **\*\*kwargs** – arguments for file upload (required: *file\_name* and *data*)

**Returns** returns the query respons for the file upload

**Return type** dict

```
class pycti.OpenCTIApiConnector (api)
```

### Inheritance

OpenCTIApiConnector

```
list ()
```

list available connectors

**Returns** return dict with connectors

**Return type** dict

```
ping (connector_id, connector_state)
```

pings a connector by id and state

**Parameters**

- **connector\_id** (*str*) – the connectors id
- **connector\_state** (*Any*) – state for the connector

**Returns** the response pingConnector data dict

**Return type** dict

```
register (connector)
```

register a connector with OpenCTI

**Parameters** **connector** ([OpenCTIConnector](#)) – *OpenCTIConnector* connector object

**Returns** the response registerConnector data dict

**Return type** dict

```
class pycti.OpenCTIApiJob (api)
```

### Inheritance

OpenCTIApiJob

```
initiate_job (work_id)
```

initiate a job with the API

**Parameters** **work\_id** (*str*) – id for the job

**Returns** the id for the initiateJob

**Return type** str

**update\_job** (*job\_id*, *status*, *messages*)

update a job with the API

**Parameters**

- **job\_id** (*str*) – job id
- **status** (*str*) – job status
- **messages** (*list*) – job messages

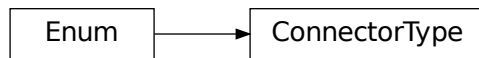
**Returns** the id for the updateJob

**Return type** str

**class** pycti.**ConnectorType**

An enumeration.

### Inheritance



**class** pycti.**OpenCTIConnector** (*connector\_id*, *connector\_name*, *connector\_type*, *scope*)

Main class for OpenCTI connector

**Parameters**

- **connector\_id** (*str*) – id for the connector (valid uuid4)
- **connector\_name** (*str*) – name for the connector
- **connector\_type** (*str*) – valid OpenCTI connector type (see *ConnectorType*)
- **scope** (*str*) – connector scope

**Raises** **ValueError** – if the connector type is not valid

### Inheritance



**to\_input()**  
connector input to use in API query

**Returns** dict with connector data

**Return type** dict

**class** pycti.**OpenCTIConnectorHelper**(*config*)  
Python API for OpenCTI connector

**Parameters** **config**(*dict*) – Dict standard config

## Inheritance

OpenCTIConnectorHelper

**static** **check\_max\_tlp**(*tlp, max\_tlp*)  
check the allowed TLP levels for a TLP string

**Parameters**

- **tlp**(*str*) – string for TLP level to check
- **max\_tlp**(*str*) – the highest allowed TLP level

**Returns** list of allowed TLP levels

**Return type** bool

**date\_now**()  
get the current date (UTC)

**Returns** current datetime for utc

**Return type** str

**get\_state**()  
get the connector state

**Returns** returns the current state of the connector if there is any

**Return type**

**listen**(*message\_callback*)  
listen for messages and register callback function

**Parameters** **message\_callback** (*Callable[[Dict], List[str]]*) – callback function to process messages

**Return type** None

**send\_stix2\_bundle**(*bundle, entities\_types=None, update=False, split=True*)  
send a stix2 bundle to the API

**Parameters**

- **bundle** – valid stix2 bundle

- **entities\_types** (*list, optional*) – list of entities, defaults to None
- **update** (*bool, optional*) – whether to update data in the database, defaults to False
- **split** (*bool, optional*) – whether to split the stix bundle before processing, defaults to True

**Raises** **ValueError** – if the bundle is empty

**Returns** list of bundles

**Return type** list

**set\_state** (*state*)

sets the connector state

**Parameters** **state** (*dict*) – state object

**Return type** None

**split\_stix2\_bundle** (*bundle*)

splits a valid stix2 bundle into a list of bundles

**Parameters** **bundle** – valid stix2 bundle

**Raises** **Exception** – if data is not valid JSON

**Returns** returns a list of bundles

**Return type** list

**static stix2\_create\_bundle** (*items*)

create a stix2 bundle with items

**Parameters** **items** – valid stix2 items

**Returns** JSON of the stix2 bundle

**Return type**

**static stix2\_deduplicate\_objects** (*items*)

deduplicate stix2 items

**Parameters** **items** – valid stix2 items

**Returns** de-duplicated list of items

**Return type** list

**stix2\_get\_embedded\_objects** (*item*)

gets created and marking refs for a stix2 item

**Parameters** **item** – valid stix2 item

**Returns** returns a dict of created\_by\_ref of object\_marking\_refs

**Return type** dict

**stix2\_get\_entity\_objects** (*entity*)

process a stix2 entity

**Parameters** **entity** – valid stix2 entity

**Returns** entity objects as list

**Return type** list



**stix2\_get\_relationship\_objects** (*relationship*)

get a list of relations for a stix2 relationship object

**Parameters** **relationship** – valid stix2 relationship

**Returns** list of relations objects

**Return type** list

**stix2\_get\_report\_objects** (*report*)

get a list of items for a stix2 report object

**Parameters** **report** – valid stix2 report object

**Returns** list of items for a stix2 report object

**Return type** list

**class** `pycti.Tag` (*opencti*)

### Inheritance

Tag

**class** `pycti.MarkingDefinition` (*opencti*)

### Inheritance

MarkingDefinition

**class** `pycti.ExternalReference` (*opencti*)

### Inheritance

ExternalReference

```
class pycti.KillChainPhase(opencti)
```

### Inheritance

KillChainPhase

```
class pycti.StixEntity(opencti)
```

### Inheritance

StixEntity

```
class pycti.StixDomainEntity(opencti, file)
```

### Inheritance

StixDomainEntity

```
class pycti.StixObservable(opencti)
```

### Inheritance

StixObservable

```
class pycti.StixRelation(opencti)
```

### Inheritance

StixRelation

```
class pycti.StixSighting(opencti)
```

### Inheritance

StixSighting

```
class pycti.StixObservableRelation(opencti)
```

### Inheritance

StixObservableRelation

```
class pycti.Identity(opencti)
```

## Inheritance

Identity

```
class pycti.ThreatActor (opentcti)
```

Main ThreatActor class for OpenCTI

**Parameters** *opentcti* – instance of OpenCTIApiClient

## Inheritance

ThreatActor

```
create (**kwargs)
```

Create a Threat-Actor object

The Threat-Actor entity will only be created if it doesn't exist. By setting *update* to *True* it acts like an upsert and updates fields of an existing Threat-Actor entity.

The create method accepts the following *\*\*kwargs*.

Note: *name* and *description* or *stix\_id\_key* is required.

### Parameters

- **id** (*str*) – (optional) OpenCTI *id* for the Threat-Actor
- **name** (*str*) – the name of the Threat-Actor
- **description** (*str*) – descriptive text
- **stix\_id\_key** (*str*) – stix2 id reference for the Threat-Actor entity
- **alias** (*list*) – (optional) list of alias names for the Threat-Actor
- **first\_seen** (*str*) – (optional) date in OpenCTI date format
- **last\_seen** (*str*) – (optional) date in OpenCTI date format
- **goal** (*str*) – (optional) describe the actors goal in text
- **sophistication** (*str*) – (optional) describe the actors sophistication in text
- **resource\_level** (*str*) – (optional) describe the actors resource\_level in text
- **primary\_motivation** (*str*) – (optional) describe the actors primary\_motivation in text

- **secondary\_motivation** (*str*) – (optional) describe the actors secondary\_motivation in text
- **personal\_motivation** (*str*) – (optional) describe the actors personal\_motivation in text
- **created** (*str*) – (optional) date in OpenCTI date format
- **modified** (*str*) – (optional) date in OpenCTI date format
- **createdByRef** (*str*) – (optional) id of the organization that created the knowledge
- **markingDefinitions** (*list*) – (optional) list of OpenCTI marking definition ids
- **tags** (*list*) – (optional) list of OpenCTI tag ids
- **update** (*bool*) – (optional) choose to updated an existing Threat-Actor entity, default *False*

**list** (*\*\*kwargs*)

List Threat-Actor objects

The list method accepts the following *\*\*kwargs*:

#### Parameters

- **filters** (*list*) – (optional) the filters to apply
- **search** (*str*) – (optional) a search keyword to apply for the listing
- **first** (*int*) – (optional) return the first n rows from the *after* ID or the beginning if not set
- **after** (*str*) – (optional) OpenCTI object ID of the first row for pagination
- **orderBy** (*str*) – (optional) the field to order the response on
- **orderMode** (*bool*) – (optional) either “*asc*” or “*desc*”
- **getAll** (*bool*) – (optional) switch to return all entries (be careful to use this without any other filters)
- **withPagination** (*bool*) – (optional) switch to use pagination

**Return type** *dict*

**read** (*\*\*kwargs*)

Read a Threat-Actor object

read can be either used with a known OpenCTI entity *id* or by using a valid filter to search and return a single Threat-Actor entity or None.

The list method accepts the following *\*\*kwargs*.

Note: either *id* or *filters* is required.

#### Parameters

- **id** (*str*) – the id of the Threat-Actor
- **filters** (*list*) – the filters to apply if no id provided

**Return type** *Optional[dict]*

**to\_stix2** (*\*\*kwargs*)

Returns a Stix2 object for a Threat-Actor id

Takes either an *id* or a Threat-Actor python object via *entity* and returns a stix2 representation of it.

The `to_stix2` method accepts the following `**kwargs`.

### Parameters

- **id** – (optional) *id* of the Threat-Actor you want to convert to stix2
- **mode** – (optional) either *simple* or *full*, default: *simple*
- **entity** – (optional) Threat-Actor object to convert

```
class pycti.IntrusionSet (opencti)
```

### Inheritance

IntrusionSet

```
class pycti.Campaign (opencti)
```

### Inheritance

Campaign

```
class pycti.Incident (opencti)
```

### Inheritance

Incident

```
class pycti.Malware (opencti)
```

### Inheritance

Malware

```
class pycti.Tool(opencti)
```

### Inheritance

Tool

```
class pycti.Vulnerability(opencti)
```

### Inheritance

Vulnerability

```
class pycti.AttackPattern(opencti)
```

### Inheritance

AttackPattern

```
class pycti.CourseOfAction(opencti)
```

### Inheritance

CourseOfAction

```
class pycti.Report(opencti)
```

### Inheritance

Report

```
class pycti.Note(opencti)
```

### Inheritance

Note

```
class pycti.Opinion(opencti)
```

### Inheritance

Opinion



```
class pycti.Indicator(openti)
```

Main Indicator class for OpenCTI

**Parameters** `openti` – instance of OpenCTIApiClient

## Inheritance

Indicator

```
list(**kwargs)
```

List Indicator objects

The list method accepts the following `**kwargs`:

### Parameters

- **filters** (*list*) – (optional) the filters to apply
- **search** (*str*) – (optional) a search keyword to apply for the listing
- **first** (*int*) – (optional) return the first n rows from the *after* ID or the beginning if not set
- **after** (*str*) – (optional) OpenCTI object ID of the first row for pagination
- **orderBy** (*str*) – (optional) the field to order the response on
- **orderMode** (*bool*) – (optional) either “*asc*” or “*desc*”
- **customAttributes** (*list*) – (optional) list of attributes keys to return
- **getAll** (*bool*) – (optional) switch to return all entries (be careful to use this without any other filters)
- **withPagination** (*bool*) – (optional) switch to use pagination

```
read(**kwargs)
```

Read an Indicator object

`read` can be either used with a known OpenCTI entity *id* or by using a valid filter to search and return a single Indicator entity or `None`.

The list method accepts the following `**kwargs`.

Note: either *id* or *filters* is required.

### Parameters

- **id** (*str*) – the id of the Threat-Actor
- **filters** (*list*) – the filters to apply if no id provided

```
class pycti.OpenCTIStix2(openti)
```

Python API for Stix2 in OpenCTI

**Parameters** `openti` – OpenCTI instance

## Inheritance

OpenCTIStix2

**check\_max\_marking\_definition** (*max\_marking\_definition\_entity*, *entity\_marking\_definitions*)  
checks if a list of marking definitions conforms with a given max level

**Parameters**

- **max\_marking\_definition\_entity** (*str*, *optional*) – the maximum allowed marking definition level
- **entity\_marking\_definitions** (*list*) – list of entities to check

**Returns** *True* if the list conforms with max marking definition

**Return type** *bool*

**convert\_markdown** (*text*)  
converts input text to markdown style code annotation

**Parameters** **text** (*str*) – input text

**Returns** sanitized text with markdown style code annotation

**Return type** *str*

**extract\_embedded\_relationships** (*stix\_object*, *types=None*)  
extracts embedded relationship objects from a stix2 entity

**Parameters**

- **stix\_object** – valid stix2 object
- **types** (*list*, *optional*) – list of stix2 types, defaults to None

**Returns** embedded relationships as dict

**Return type** *dict*

**filter\_objects** (*uuids*, *objects*)  
filters objects based on UUIDs

**Parameters**

- **uuids** (*list*) – list of UUIDs
- **objects** (*list*) – list of objects to filter

**Returns** list of filtered objects

**Return type** *list*

**format\_date** (*date*)  
converts multiple input date formats to OpenCTI style dates

**Parameters** **date** – input date

**Returns** OpenCTI style date

**Return type** datetime

**import\_bundle\_from\_file** (*file\_path*, *update=False*, *types=None*)  
import a stix2 bundle from a file

**Parameters**

- **file\_path** (*str*) – valid path to the file
- **update** (*bool*, *optional*) – whether to updated data in the database, defaults to False
- **types** (*list*, *optional*) – list of stix2 types, defaults to None

**Returns** list of imported stix2 objects

**Return type** List

**import\_bundle\_from\_json** (*json\_data*, *update=False*, *types=None*)  
import a stix2 bundle from JSON data

**Parameters**

- **json\_data** – JSON data
- **update** (*bool*, *optional*) – whether to updated data in the database, defaults to False
- **types** (*list*, *optional*) – list of stix2 types, defaults to None

**Returns** list of imported stix2 objects

**Return type** List

**import\_object** (*stix\_object*, *update=False*, *types=None*)  
import a stix2 object

**Parameters**

- **stix\_object** – valid stix2 object
- **update** (*bool*, *optional*) – whether to updated data in the database, defaults to False
- **types** (*list*, *optional*) – list of stix2 types, defaults to None

**Returns** list of imported stix2 objects

**Return type** list

**pick\_aliases** (*stix\_object*)  
check stix2 object for multiple aliases and return a list

**Parameters** **stix\_object** – valid stix2 object

**Returns** list of aliases

**Return type** list

**class** pycti.**ObservableTypes**

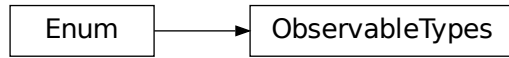
These are the possible values for OpenCTI's observable types.

Use in conjunction with the STIX custom property *x\_opencti\_observable\_type*.

ref: <https://github.com/OpenCTI-Platform/opencti/blob/8854c2576dc17da9da54e54b116779bd2131617c/opencti-front/src/private/components/report/ReportAddObservable.js>

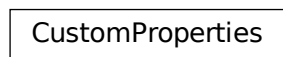
NOTE: should this be a mapping between the stix2 SDO objects (i.e. stix2/v20/sdo.py)?

### Inheritance



**class** `pycti.CustomProperties`  
These are the custom properties used by OpenCTI.

### Inheritance



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**p**

`pycti`, 5





## A

AttackPattern (class in pycti), 19

## C

Campaign (class in pycti), 18

check\_max\_marking\_definition()  
(pycti.OpenCTISix2 method), 22

check\_max\_tlp() (pycti.OpenCTIConnectorHelper  
static method), 11

ConnectorType (class in pycti), 10

convert\_markdown() (pycti.OpenCTISix2 method),  
22

CourseOfAction (class in pycti), 19

create() (pycti.ThreatActor method), 16

CustomProperties (class in pycti), 24

## D

date\_now() (pycti.OpenCTIConnectorHelper  
method), 11

## E

ExternalReference (class in pycti), 13

extract\_embedded\_relationships()  
(pycti.OpenCTISix2 method), 22

## F

fetch\_opencti\_file() (pycti.OpenCTIApiClient  
method), 7

filter\_objects() (pycti.OpenCTISix2 method), 22

format\_date() (pycti.OpenCTISix2 method), 22

## G

get\_config\_variable() (in module pycti), 5

get\_logs\_worker\_config()  
(pycti.OpenCTIApiClient method), 7

get\_state() (pycti.OpenCTIConnectorHelper  
method), 11

get\_token() (pycti.OpenCTIApiClient method), 7

## H

health\_check() (pycti.OpenCTIApiClient method),  
7

## I

Identity (class in pycti), 15

import\_bundle\_from\_file()  
(pycti.OpenCTISix2 method), 23

import\_bundle\_from\_json()  
(pycti.OpenCTISix2 method), 23

import\_object() (pycti.OpenCTISix2 method), 23

Incident (class in pycti), 18

Indicator (class in pycti), 20

initiate\_job() (pycti.OpenCTIApiJob method), 9

IntrusionSet (class in pycti), 18

## K

KillChainPhase (class in pycti), 14

## L

list() (pycti.Indicator method), 21

list() (pycti.OpenCTIApiConnector method), 9

list() (pycti.ThreatActor method), 17

listen() (pycti.OpenCTIConnectorHelper method),  
11

log() (pycti.OpenCTIApiClient method), 7

## M

Malware (class in pycti), 18

MarkingDefinition (class in pycti), 13

## N

not\_empty() (pycti.OpenCTIApiClient method), 7

Note (class in pycti), 20

## O

ObservableTypes (class in pycti), 23

OpenCTIApiClient (class in pycti), 6

OpenCTIApiConnector (class in pycti), 8

OpenCTIApiJob (*class in pycti*), 9  
OpenCTIConnector (*class in pycti*), 10  
OpenCTIConnectorHelper (*class in pycti*), 11  
OpenCTIStix2 (*class in pycti*), 21  
Opinion (*class in pycti*), 20

## P

pick\_aliases() (*pycti.OpenCTIStix2 method*), 23  
ping() (*pycti.OpenCTIApiConnector method*), 9  
process\_multiple() (*pycti.OpenCTIApiClient method*), 7  
process\_multiple\_fields() (*pycti.OpenCTIApiClient method*), 8  
process\_multiple\_ids() (*pycti.OpenCTIApiClient method*), 8  
pycti (*module*), 5

## Q

query() (*pycti.OpenCTIApiClient method*), 8

## R

read() (*pycti.Indicator method*), 21  
read() (*pycti.ThreatActor method*), 17  
register() (*pycti.OpenCTIApiConnector method*), 9  
Report (*class in pycti*), 20  
resolve\_role() (*pycti.OpenCTIApiClient method*), 8

## S

send\_stix2\_bundle() (*pycti.OpenCTIConnectorHelper method*), 11  
set\_state() (*pycti.OpenCTIConnectorHelper method*), 12  
set\_token() (*pycti.OpenCTIApiClient method*), 8  
split\_stix2\_bundle() (*pycti.OpenCTIConnectorHelper method*), 12  
stix2\_create\_bundle() (*pycti.OpenCTIConnectorHelper static method*), 12  
stix2\_deduplicate\_objects() (*pycti.OpenCTIConnectorHelper static method*), 12  
stix2\_get\_embedded\_objects() (*pycti.OpenCTIConnectorHelper method*), 12  
stix2\_get\_entity\_objects() (*pycti.OpenCTIConnectorHelper method*), 12  
stix2\_get\_relationship\_objects() (*pycti.OpenCTIConnectorHelper method*), 12

stix2\_get\_report\_objects() (*pycti.OpenCTIConnectorHelper method*), 13

StixDomainEntity (*class in pycti*), 14  
StixEntity (*class in pycti*), 14  
StixObservable (*class in pycti*), 14  
StixObservableRelation (*class in pycti*), 15  
StixRelation (*class in pycti*), 15  
StixSighting (*class in pycti*), 15

## T

Tag (*class in pycti*), 13  
ThreatActor (*class in pycti*), 16  
to\_input() (*pycti.OpenCTIConnector method*), 10  
to\_stix2() (*pycti.ThreatActor method*), 17  
Tool (*class in pycti*), 19

## U

update\_job() (*pycti.OpenCTIApiJob method*), 10  
upload\_file() (*pycti.OpenCTIApiClient method*), 8

## V

Vulnerability (*class in pycti*), 19