CERT Australia CTI Toolkit Documentation

Release 0.1

CERT Australia, Australian Government

April 05, 2016

CONTENTS

This package contains cyber threat intelligence (CTI) tools created by CERT Australia.

Contents:

CHAPTER

ONE

INSTALLATION

This document describes how to install the CERT Australia CTI Toolkit.

Installation is streamlined using Python's setuptools. The following installation process has been tested on clean install of Ubuntu 14.04.

1. Install prerequisites required by setuptools and libtaxii:

```
    $ sudo apt-get install python-pip python-dev libxml2-dev libxslt1-dev libz-dev
```

2. Clone the cti-toolkit repository (prompts for github username and password):

```
    $ git clone https://github.com/certau/cti-toolkit.git
```

3. Run the setup.py script to build and install the tools (and pip dependencies):

```
    $ cd cti-toolkit
    $ sudo python setup.py install
```

That's it. You should now be able to run utilities, such as stixtransclient.py:

```
$ stixtransclient.py -h
```

1.1 Documentation

To build the documentation you need Sphinx:

```
$ sudo pip install Sphinx sphinxcontrib-napoleon
$ cd docs
$ make html
```

This will create an HTML version of the documentation in `docs/_build/html`.

CHAPTER TWO

CONFIGURATION

The `stixtransclient.py` utility can read its configuration parameters from the following locations:

- /etc/ctitoolkit.conf
- ~/.ctitoolkit
- a configuration file specified using the `--config` command line option
- as explicit command line parameters

If a configuration option is specified in more than one of the above locations the last one processed will take precedence. Options are processed in the order listed above.

Any options that can be specified on the command line can be specified in a configuration file.

2.1 `ctitoolkit.conf` examples

Some examples follow:

YETI:

```
# Connect to the CERT Australia taxii server
# Authenticate using certificate and user credentials
# Poll indicators from the 'advisories' collection
# Output data in Bro intel framework format
source: YETI
hostname: yeti.host.tld
cert: /path/cert.pem
key: /path/key.pem
username: _USER_
password: _PASSWORD_
collection: advisories
base_url: https://source.host.com/advisories/
ssl: true
taxii: true
bro: true
aus: true
```

SoltraEdge:

```
source: HAT
hostname: hailataxii.com
username: guest
password: guest
path: /taxii-data
```

```
collection: guest.dataForLast_7daysOnly
taxii: true
soltra: true
bro: true
```

FILE:

```
# Process an STIX file and output to MISP
source: FILE
file: /path/to/stix/file.xml
misp: true
misp_url:http://misp.host.tld
misp_key:keykeykeykeykeykeyke
```

CHAPTER

THREE

SCRIPTS

Currently the only script in the toolkit is `stixtransclient.py`.

Contents:

3.1 `stixtransclient.py`

Few systems can utilise indicators and observables when stored in STIX packages. CERT Australia has developed a utility (`stixtransclient.py`) that allows the atomic observables contained within a STIX package to be extracted and presented in either a text delimited format, or in the Bro Intel Framework format. The utility can also communicate with a MISP server and insert observables from a STIX package into a new MISP event.

3.1.1 Examples

Display summary statistics about the object types (observables) contained in a STIX package (file):

```
$ stixtransclient.py --file CA-TEST-STIX.xml --stats

++++++++++++++++++++++++++++++++++++++++
Summary statistics: CA-TEST-STIX (WHITE)
++++++++++++++++++++++++++++++++++++++++
Address observables:                  2
DomainName observables:               3
EmailMessage observables:             2
File observables:                     6
HTTPSession observables:              1
Mutex observables:                    3
SocketAddress observables:            1
URI observables:                      1
WinRegistryKey observables:           1
```

Display observable details in text (delimited) format:

```
$ stixtransclient.py --file CA-TEST-STIX.xml --text

# CA-TEST-STIX (TLP:WHITE)

# Address observables
# id|category|address
cert_au:Observable-fe5ddeac-f9b0-4488-9f89-bfbd9351efd4|ipv4-addr|158.164.39.51
cert_au:Observable-ccccceac-f9b0-4488-9f89-bfbd9351efd4|ipv4-addr|111.222.33.44

# DomainName observables
# id|domain|domain_condition
cert_au:Observable-6517027e-2cdb-47e8-b5c8-50c6044e42de|bad.domain.org|None
cert_au:Observable-c97cc016-24b6-4d02-afc2-308742c722dc|dnsupdate.dyn.net|None
cert_au:Observable-138a5be6-56b2-4d2d-af73-2d4865d6ff71|free.stuff.com|None

# EmailMessage observables
# id|fromaddr|fromaddr_condition|toaddr|toaddr_condition|subject|subject_condition|attachment_ref
cert_au:Observable-b6770e76-7f05-48cb-a3de-7ba5fece8751|sender@domain.tld|Equals|None|None|None|None
cert_au:Observable-31e5af27-2f71-4922-b49c-cfd3ddee2963|None|None|None|None|Important project details

# File observables
# id|file_name|file_name_condition|hash_type|hashes
cert_au:Observable-5d647351-f8cf-442f-9e5a-ba6967cccccc|filenameonly.doc|None|None|None
cert_au:Observable-5d647351-f8cf-442f-9e5a-ba6967c16301|project.doc|Equals|MD5|1111111111b42b57f51819
cert_au:Observable-cccccd51-a524-483f-8f17-2e8ff8474d80|None|None|MD5|cccccccccccccc33574c79829dc1cc
cert_au:Observable-84060d51-a524-483f-8f17-2e8ff8474d80|Execute_this.jar|Equals|MD5|11111111111111133
cert_au:Observable-3ad6c684-80aa-4d92-9fef-7a9f70ccba95|malware.exe|Equals|MD5|1111111111111111f2601
cert_au:Observable-7cb2ac9f-4cae-443f-905d-0b01cb1faedc|VPN.exe|Equals|SHA256|1111111111111119f16768
cert_au:Observable-7cb2ac9f-4cae-443f-905d-0b01cb1faedc|VPN.exe|Equals|SHA1|893fb19ac24eabf9b1fe1ddd
cert_au:Observable-7cb2ac9f-4cae-443f-905d-0b01cb1faedc|VPN.exe|Equals|MD5|11111111111111112977fa0588

# HTTPSession observables
# id|user_agent|user_agent_condition
cert_au:Observable-6a733d83-5d19-4d17-a51f-5bcb4ebc860a|Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537

# Mutex observables
# id|mutex|mutex_condition
NCCIC:Observable-01234567-6868-4ffd-babc-ba2ad0e34f43|WIN_ABCDEF|None
NCCIC:Observable-abcdef01-3363-4533-a77c-10d71c371282|MUTEX_0001|None
CCIRC-CCRIC:Observable-01234567-e44c-473a-85c6-fc6c2e781114|iurlkjashdk|Equals

# SocketAddress observables
# id|category|address|port_value|port_protocol
CCIRC-CCRIC:Observable-01234567-2823-4d6d-8d77-bae10ca5bd97|ipv4-addr|183.82.180.95|2665|TCP

# URI observables
# id|uri|uri_condition
cert_au:Observable-1a919136-ba69-4a28-9615-ad6ee37e88a5|http://host.domain.tld/path/file|None

# WinRegistryKey observables
# id|hive|hive_condition|key|key_condition|name|name_condition|data|data_condition
cert_au:Observable-d0f4708e-4f2b-49c9-bc31-29e7119844e5|HKEY_CURRENT_USER\Software|Equals|\Microsoft\
```

Display observables in the format used by the Bro Intelligence Framework (with a header row explaining columns):

```
$ stixtransclient.py --file CA-TEST-STIX.xml --bro --header

# indicator indicator_type  meta.source     meta.url        meta.do_notice  meta.if_in      meta.whit
158.164.39.51       Intel::ADDR     CERT-AU https://www.cert.gov.au/        T       -       -
111.222.33.44       Intel::ADDR     CERT-AU https://www.cert.gov.au/        T       -       -
bad.domain.org      Intel::DOMAIN   CERT-AU https://www.cert.gov.au/        T       -       -
dnsupdate.dyn.net   Intel::DOMAIN   CERT-AU https://www.cert.gov.au/        T       -       -
free.stuff.com      Intel::DOMAIN   CERT-AU https://www.cert.gov.au/        T       -       -
sender@domain.tld   Intel::EMAIL    CERT-AU https://www.cert.gov.au/        T       -       -
1111111111b42b57f518197d930471d9    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/        T
cccccccccccccc33574c79829dc1ccf    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/        T
1111111111111133574c79829dc1ccf    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/        T
1111111111111111f2601b4d21660fb    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/        T
```

```
1111111111111119f167683e164e795896be3be94de7f7103f67c6fde667bdf    Intel::FILE_HASH      CERT-AU h
893fb19ac24eabf9b1fe1ddd1111111111111111    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/
11111111111111112977fa0588bd504a    Intel::FILE_HASH        CERT-AU https://www.cert.gov.au/        T
Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.2309.372 Safari/537.3
183.82.180.95       Intel::ADDR     CCIRC   https://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/cc
host.domain.tld/path/file   Intel::URL      CERT-AU https://www.cert.gov.au/        T       -       -
```

3.1.2 Command line options (help)

The command line (and configuration) options for stixtransclient.py are displayed below:

```
$ stixtransclient.py -h

usage: stixtransclient.py [-h] [-c CONFIG] [-v] [-d]
                          (--file FILE [FILE ...] | --taxii)
                          (-s | -t | -b | -m | -x XML_OUTPUT) [-r]
                          [--hostname HOSTNAME] [--username USERNAME]
                          [--password PASSWORD] [--ssl] [--key KEY]
                          [--cert CERT] [--path PATH]
                          [--collection COLLECTION]
                          [--begin-timestamp BEGIN_TIMESTAMP]
                          [--end-timestamp END_TIMESTAMP]
                          [--subscription-id SUBSCRIPTION_ID]
                          [-f FIELD_SEPARATOR] [--header] [--title TITLE]
                          [--source SOURCE] [--bro-no-notice]
                          [--base-url BASE_URL] [--misp-url MISP_URL]
                          [--misp-key MISP_KEY]
                          [--misp-distribution MISP_DISTRIBUTION]
                          [--misp-threat MISP_THREAT]
                          [--misp-analysis MISP_ANALYSIS]
                          [--misp-info MISP_INFO] [--misp-published]

Utility to extract observables from local STIX files or a TAXII server. Args
that start with '--' (eg. -v) can also be set in a config file
(/etc/ctitoolkit.conf or ~/.ctitoolkit or specified via -c). The recognized
syntax for setting (key, value) pairs is based on the INI and YAML formats
(e.g. key=value or foo=TRUE). For full documentation of the differences from
the standards please refer to the ConfigArgParse documentation. If an arg is
specified in more than one place, then commandline values override config file
values which override defaults.

optional arguments:
  -h, --help            show this help message and exit

global arguments:
  -c CONFIG, --config CONFIG
                        configuration file to use
  -v, --verbose         verbose output
  -d, --debug           enable debug output

input (source) options:
  --file FILE [FILE ...]
                        obtain STIX packages from supplied files or
                        directories
  --taxii               poll TAXII server to obtain STIX packages

output (transform) options:
  -s, --stats           display summary statistics for each STIX package
  -t, --text            output observables in delimited text
  -b, --bro             output observables in Bro intel framework format
  -m, --misp            feed output to a MISP server
  -x XML_OUTPUT, --xml_output XML_OUTPUT
                        output XML STIX packages to the given directory (use
                        with --taxii)

file input arguments (use with --file):
  -r, --recurse         recurse subdirectories when processing files.

taxii input arguments (use with --taxii):
  --hostname HOSTNAME   hostname of TAXII server
  --username USERNAME   username for TAXII authentication
  --password PASSWORD   password for TAXII authentication
  --ssl                 use SSL to connect to TAXII server
  --key KEY             file containing PEM key for TAXII SSL authentication
  --cert CERT           file containing PEM certificate for TAXII SSL
                        authentication
  --path PATH           path on TAXII server for polling
  --collection COLLECTION
                        TAXII collection to poll
  --begin-timestamp BEGIN_TIMESTAMP
                        the begin timestamp (format: YYYY-MM-
                        DDTHH:MM:SS.ssssss+/-hh:mm) for the poll request
  --end-timestamp END_TIMESTAMP
                        the end timestamp (format: YYYY-MM-
                        DDTHH:MM:SS.ssssss+/-hh:mm) for the poll request
  --subscription-id SUBSCRIPTION_ID
                        a subscription ID for the poll request

other output options:
  -f FIELD_SEPARATOR, --field-separator FIELD_SEPARATOR
                        field delimiter character/string to use in text output
  --header              include header row for text output
  --title TITLE         title for package (if not included in STIX file)
  --source SOURCE       source of indicators - e.g. Hailataxii, CERT-AU
  --bro-no-notice       suppress Bro intel notice framework messages (use with
                        --bro)
  --base-url BASE_URL   base URL for indicator source - use with --bro or
                        --misp

misp output arguments (use with --misp):
  --misp-url MISP_URL   URL of MISP server
  --misp-key MISP_KEY   token for accessing MISP instance
  --misp-distribution MISP_DISTRIBUTION
                        MISP distribution group - default: 0 (your
                        organisation only)
  --misp-threat MISP_THREAT
                        MISP threat level - default: 4 (undefined)
  --misp-analysis MISP_ANALYSIS
                        MISP analysis phase - default: 0 (initial)
  --misp-info MISP_INFO
                        MISP event description - default: 'Automated STIX
                        ingest'
  --misp-published      set MISP published state to True
```

CHAPTER

FOUR

API REFERENCE

Contents:

4.1 `certau.source` Module

Classes that provide a source of STIX packages.

These classes should implement the `next_stix_package()` method.

class `certau.source.`**`StixSource`**

A base class for sources of STIX packages.

`next_stix_package``()`

Return the next STIX package available from the source (or None).

class `certau.source.`**`StixFileSource`**`(`*files*, *recurse=False*`)`

Return STIX packages from a file or directory.

Parameters

- **`files`** – an array containing the names of one or more files or directories
- **`recurse`** – an optional boolean value (default False), which when set to True, will cause subdirectories to be searched recursively

class `certau.source.`**`SimpleTaxiiClient`**`(`*hostname*, *path*, *collection*, *use_ssl=False*, *username=None*, *password=None*, *key_file=None*, *cert_file=None*, *begin_ts=None*, *end_ts=None*, *subscription_id=None*`)`

A simple interface to the libtaxii libraries for polling a TAXII server.

The `certau.client.SimpleTaxiiClient` class provides a simple interface for polling a collection on a TAXII server and returning the response. It supports SSL (certificate-based) authentication in addition to a username and password.

Parameters

- **`hostname`** – the name of the TAXII server
- **`path`** – the URL path for the collection
- **`collection`** – the collection on the TAXII server to poll
- **`use_ssl`** – use SSL when connecting to the TAXII server
- **`username`** – a username for password-based authentication
- **`password`** – a password for password-based authentication

- **key_file** – a private key file for SSL certificate-based authentication
- **cert_file** – a certificate file for SSL certificate-based authentication
- **begin_ts** – a timestamp to describe the earliest content to be returned by the TAXII server
- **end_ts** – a timestamp to describe the most recent content to be returned by the TAXII server
- **subscription_id** – a subscription ID to include with the poll request

create_poll_request()

Create a poll request message using supplied parameters.

save_content_blocks(*directory*)

Save poll response content blocks to given directory.

send_poll_request()

Send the poll request to the TAXII server.

4.2 certau.transform Module

Classes for transforming STIX packages to various formats.

The base class *StixTransform* provides helper functions for processing STIXPackage elements.

There are two broad types of transform currently supported:

1. Transforms to a text format (these transforms extend the *StixTextTransform* class):
   - *StixStatsTransform* - display statistics about a package
   - *StixCsvTransform* - display indicators in CSV format
   - *StixBroIntelTransform* - display indicators in the Bro Intel format
2. **Transforms that interact with a service:**
   - *StixMispTransform* - publish indicators to a MISP instance

class certau.transform.**StixTransform**(*package*)

Base class for transforming a STIX package to an alternate format.

This class provides helper functions for processing STIXPackage elements. This class should be extended by other classes that transform STIX packages into alternate formats.

The default constructor processes a STIX package to initialise self._observables, a dict keyed by object type. Each entry contains a list list of dict objects with three keys: ‘id’, ‘observable’, and ‘fields’, containing the observable ID, the Observable object itself, and extracted fields, respectively.

Parameters **package** – the STIX package to transform

OBJECT_FIELDS

a dict of supported Cybox object types and fields (‘properties’). The dictionary is keyed by Cybox object type string (see _observable_object_type()) with each entry containing a list of field names from that object that will be utilised during the transformation.

Field names may reference sub-objects using dot notation. For example the Cybox EmailMessage class contains a *header* field referring to an EmailHeader object which contains a *to* field. This field can be referenced using the notation *header.to*.

If OBJECT_FIELDS evaluates to False (e.g. empty dict()), it is assumed all object types are supported.

OBJECT_CONSTRAINTS

a `dict` of constraints on the supported object types based on 'categories' associated with that type. For example, the Cybox Address object uses the field *category* to distinguish between IPv4, IPv6 and even email addresses. Like OBJECT_FIELDS, the dictionary is keyed by object type. Each entry contains a dictionary keyed by field name, containing a list of values, or categories, (for that field name) that are supported by the transform.

Note. Does not support the expression of more complex constraints, for example combining different categories.

STRING_CONDITION_CONSTRAINT

a `list` of string condition values supported by the transform. For example, some transforms may not support 'FitsPattern' or 'StartsWith' string condition values. Use this to list the supported values. Note the values are strings, even 'None'.

classmethod `_observables_for_package`(*package*)

Extract observables from a STIX package.

Collects observables from a STIX package and groups them by object type. Only observables with an ID and containing a Cybox object are returned. Results are returned in a dictionary keyed by object type - see `_observable_object_type()`.

If OBJECT_FIELDS are specified only observables containing the object types listed will be returned, and only those with at least one of the listed fields containing a non-trivial value. OBJECT_CONSTRAINTS and STRING_CONDITION_CONSTRAINT are also applied.

If no OBJECT_FIELDS are specified no constraints are applied and all identified observables are returned.

Observables are sought from the following locations:

- the root of the STIX package
- within Indicator objects (where the indicators are in the package root)
- within ObservableComposition objects found in either of the two previous locations

Parameters **`package`** – a `STIXPackage` object

Returns a dictionary of valid observables, keyed by object type (See description above). May be empty.

Return type dict

`package_description`(*default=''*)

Retrieves the STIX package description (str) from the header.

`package_title`(*default=''*)

Retrieves the STIX package title (str) from the header.

`package_tlp`(*default='AMBER'*)

Retrieves the STIX package TLP (str) from the header.

class `certau.transform.`**`StixTextTransform`**(*package*, *separator='|'*, *include_header=True*, *header_prefix='#'*)

A transform for converting a STIX package to simple text.

This class and its subclasses implement the *`text()`* class method which returns a string representation of the STIX package. The entire text output may optionally be preceded by a header string. Typically, each line of the output will contain details for a particular Cybox observable. Output is grouped by observable type. Each group of observables (by type) may also contain an additional header string.

Parameters

- **package** – the STIX package to transform
- **separator** – the delimiter used in text output
- **include_header** – a boolean value indicating whether or not headers should be included in the output
- **header_prefix** – a string prepended to each header row

HEADER_LABELS
a list of field names that are printed by the `header()` function.

OBJECT_HEADER_LABELS
a dict, keyed by object type, containing field names associated with an object type. These are printed by the `header_for_object_type()` function.

header()
Returns a header string to display with transform.

header_for_object_type(*object_type*)
Returns a header string associated with an object type.

text()
Returns a string representation of the STIX package.

text_for_fields(*fields*, *object_type*)
Returns a string representing the given object fields.

text_for_object_type(*object_type*)
Returns a string representing observables of the given type.

text_for_observable(*observable*, *object_type*)
Returns a string representing the given observable.

class certau.transform.**StixStatsTransform**(*package*, *separator=’t’*, *include_header=True*, *header_prefix=’‘*, *pretty_text=True*)
Generate summary statistics for a STIX package.

Prints a count of the number of observables for each object type contained in the package.

Parameters

- **package** – the STIX package to process
- **separator** – a string separator used in the text output
- **include_header** – a boolean value that indicates whether or not header information should be included in the text output
- **header_prefix** – a string prepended to header lines in the output
- **pretty_text** – a boolean that indicates whether or not the text should be made pretty by aligning the columns in the text output

class certau.transform.**StixCsvTransform**(*package*, *separator=’|’*, *include_header=True*, *header_prefix=’#’*, *include_observable_id=True*, *include_condition=True*)
Generate a CSV formatted summary of observables from a STIX package.

This class can be used to generate a delimited text dump of the observable fields contained in a STIX package. Output is grouped by the object type contained in the observable.

Parameters

- **package** – the STIX package to process

- **`separator`** – a string separator used in the text output
- **`include_header`** – a boolean value that indicates whether or not header information should be included in the text output
- **`header_prefix`** – a string prepended to header lines in the output
- **`include_observable_id`** – a boolean value indicating whether or not the output should include the observable’s UUID
- **`include_condition`** – a boolean value indicating whether or not the output should include additional fields containing the Cybox string matching condition (which may be empty)

class `certau.transform.`**`StixBroIntelTransform`**(*package*, *separator=’t’*, *include_header=False*, *header_prefix=’#’*, *source=’UNKNOWN’*, *url=’‘*, *do_notice=’T’*)

Generate observable details for the Bro Intelligence Framework.

This class can be used to generate a list of indicators (observables) from a STIX package in a format suitable for importing into the Bro network-based intrusion detection system using its Intelligence Framework (see https://www.bro.org/sphinx-git/frameworks/intel.html).

Parameters

- **`package`** – the STIX package to process
- **`separator`** – a string separator used in the text output
- **`include_header`** – a boolean value that indicates whether or not header information should be included in the text output
- **`header_prefix`** – a string prepended to header lines in the output
- **`source`** – a value to include in the output metadata field ‘meta.source’
- **`url`** – a value to include in the output field metadata ‘meta.url’
- **`do_notice`** – a value to include in the output metadata field ‘meta.do_notice’, if set to ‘T’ a Bro notice will be raised by Bro on a match of this indicator

class `certau.transform.`**`StixMispTransform`**(*package*, *misp*, *distribution=0*, *threat_level=1*, *analysis=2*, *information=None*, *published=False*)

Insert data from a STIX package into a MISP event.

This class inserts data from a STIX package into MISP (the Malware Information Sharing Platform - see http://www.misp-project.org/). A PyMISP (https://github.com/CIRCL/PyMISP) object is passed to the constructor and used for communicating with the MISP host. The helper function *`get_misp_object()`* can be used to instantiate a PyMISP object.

Parameters

- **`package`** – the STIX package to process
- **`misp`** – the PyMISP object used to communicate with the MISP host
- **`distribution`** – the distribution setting for the MIST event (0-3)
- **`threat_level`** – the threat level setting for the MISP event (0-3)
- **`analysis`** – the analysis level setting for the MISP event (0-2)
- **`information`** – info field value (string) for the MISP event
- **`published`** – a boolean indicating whether the event has been published

static `get_misp_object` (*misp_url*, *misp_key*, *use_ssl=False*)

Returns a PyMISP object for communicating with a MISP host.

Parameters

- **`misp_url`** – URL for MISP API end-point
- **`misp_key`** – API key for accessing MISP API
- **`use_ssl`** – a boolean value indicating whether or not the connection should use HTTPS (instead of HTTP)

CHAPTER FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX