yeti Documentation

Release 2.0a5

The TAXII Team

Jul 25, 2017

Contents

yeti is an example TAXII server that can be deployed as a prototype / test bed. YETI can be run locally or behind a webserver.

Contents:

CHAPTER 1

Getting Started

This page gives an introduction to **YETI** and how to use it. Please note that this page is being actively worked on and feedback is welcome (taxii@mitre.org).

Note that all of the TAXII functionality in YETI is powered by django-taxii-services. If you're a developer looking to use YETI to implement TAXII, then what you really want is django-taxii-services. If not, then that distinction may not be important.

Dependencies

In order to use YETI, you must first install YETI's dependencies (listed as of 10/24/2014):

- Django 1.7 or higher
- django-solo
- libtaxii 1.1.103 or higher
- taxii_services (aka django-taxii-services) 0.2 or higher
- python-dateutil
- pyOpenSSL

Local YETI Deployment

These instructions tell you how to get YETI up and running on your local machine.

1. Install the dependencies
2. Get YETI using one of two methods:

1. Clone the YETI repostory (`git clone https://github.com/TAXIIProject/yeti.git`) (Requires git)

2. Download and unpack the latest release (2.0a3 at the time of this writing): https://github.com/TAXIIProject/yeti/releases
3. Open a command prompt and navigate to the folder you extracted YETI into
4. Set up the database: `python manage.py syncdb`

1. This sets up the database, and only needs to be run once
2. Say "yes" to "Create a superuser?" - Supply your own credentials

4. Run the server: `python manage.py runserver --insecure 0.0.0.0:8080`

Now you have YETI running. You can play with it by:

- Pointing your browser at `http://localhost:8080/`
- Or go to `http://localhost:8080/admin/` to log into the admin interface
- Run any of the scripts in yeti/scripts. e.g.,
- `scripts\yeti_collection_information_client.bat`
- `scripts\yeti_discovery_client.bat`
- If you have a proxy, you'll have to specify it by appending `--proxy http://proxy.example.com:80` to the script e.g., `scripts\yeti_discovery_client.bat --proxy http://proxy.example.com:80`
- Run any libtaxii script from a command prompt:
- `discovery_client --host localhost --port 8080 [--proxy <proxy_address>]`

Testing with YETI

YETI comes packaged with some initial data (fixtures, in Django-speak) that allows a tester to stand up YETI and get going with a minimal amount of effort. YETI and django-taxii-services are evolving quickly, so if you have something you'd like to see, let us know!

YETI comes pre-packaged with:

- An Inbox Service that supports TAXII 1.0 and TAXII 1.1 at http://localhost:8080/services/inbox/
- A Poll Service that supports TAXII 1.0 and TAXII 1.1 at http://localhost:8080/services/poll/
- A Collection Management Service that supports TAXII 1.0 and TAXII 1.1 at http://localhost:8080/services/collection-management/
- A Discovery Service that Supports TAXII 1.0 and TAXII 1.1 at http://localhost:8080/services/discovery/

If you are testing with YETI, be sure to remember that YETI and django-taxii-services are written by humans, and humans can make mistakes. We can also document things poorly. If you see something that doesn't look right, if you have a question about how something works, or you have a bug report to file, please contact us using GitHub, the TAXII Discussion List, or taxii@mitre.org. We are always happy to help.

Exchanging Threat Information

This section describes how to use TAXII and YETI to exchange threat information. If you're looking for a primer on TAXII, please visit http://taxiiproject.github.io/getting-started/intro/. To exchange threat information using YETI, you can make use of the Inbox Service and Poll Service. The Inbox Service lets you push threat information to YETI and the Poll Service lets you pull threat information from YETI. In this section both automated and manual options are provided for pushing and pulling threat information to/from YETI.

Note: Examples in this section assume you are running YETI on localhost:8080. If you are not, replace the examples with the hostname / port number your YETI instance is running on

Getting data into YETI (Pushing) - You have two options to get data into YETI: use a TAXII Client or use the YETI UI.

1. Use a TAXII Client - You can use any TAXII Inbox Client to push data into YETI. If you want to use libtaxii, you can use the following command: `inbox_client --host localhost --port 8080`. Out of the box, `inbox_client` sets the `Destination_Collection_Name` parameter of the Inbox Message to `default` and sends a default STIX XML document.

2. Use the YETI UI - Navigate to the Content Block admin (this will be located at: http://localhost:8080/admin/taxii_services/contentblock/).

   1. Click the `Add Content Block +` button in the top right.
   2. Paste the content into the `Content` field, and set the `Content Binding And Subtype` field appropriately.
   3. Navigate to the Data Collection you want this content to appear in using the Data Collection Admin.
   4. In the `Content Blocks` field, add the Content Block you just added.

In all cases, all Content Blocks in the system can always be viewed by navigating to the Content Block admin page. Content Blocks must be explicitly added to a specific Data Collection - either through the `Destination_Collection_Name` parameter of an Inbox Message or through the UI - or else they won't be available for polling.

Getting data out of YETI (Pulling) - You have two options to get data out of YETI: use a TAXII Client or use the YETI UI.

1. Use a TAXII Client - You can use any TAXII Poll Client to get data out of YETI. If you want to use libtaxii, you can use the following command: `poll_client --host localhost --port 8080`. Out of the box, `poll_client` requests information from the `default` Data Collection.

2. Use the YETI UI - Navigate to the Content Block admin (this will be located at: http://localhost:8080/admin/taxii_services/contentblock/).

   1. Click the Content Block you'd like to get information from
   2. Copy/Paste the Content to wherever you'd like. Framed threat information makes a great gift for your significant other!

CHAPTER 2

Indices and tables

- genindex
- modindex
- search