toreator

Tor relay parser and UI

Installation guide

Libor Polčák

TARZAN project VI20172020062 document
Faculty of Information Technology, Brno University of Technology

Last change: April 24, 2020

toreator — Installation guide

Libor Polčák

Faculty of Information Technology, Brno University of Technology, e-mail:
polcak@fit.vutbr.cz

This software processes data about Tor network. The main source is Tor CollecTor[1], which is available for research purposes. The tool augment the data with data from MaxMind GeoLite2 data[2] and data fetched from public DNS.

CollecTor contains all regisered public nodes that were usable as entry nodes, exit nodes, and relays. Neither Tor bridges nor Tor users are stored in the database.

Conceptually, this tool was inspired by the service Exonerator[3] and Torstatus[4]. The database created by this tool differs from:

- *Exonerator* because it contains more details about the Tor nodes;
- *Torsatus* because it contains historical data.

This guide explains how to set up the tool to run your own instance of Toreator, the API, and GUI. You should set up your server so that the database is automatically updated. CollecTor updates once per hour. MaxMind updates once per week (ASN database) or month (location database).

1 Setting up dependencies

Download CollecTor database.

Register and download MaxMind GeoLite2 database.

Automatic updates of MaxMind GeoLite2 database.

2 Consensus parser

TorConsensus parser is a python program with the following dependencies:

- dev-lang/python3 (tested in python3.5, python3.6) https://www.python.org/downloads/source/
- dev-python/ipy: handling of IPv4 and IPv6 addresses and networks, https://github.com/haypo/python-ipy/wiki
- dev-python/python-dateutil: extensions to the standard Python datetime module, https://github.com/dateutil/dateutil/

[1] https://metrics.torproject.org/collector.html

[2] http://www.maxmind.com/

[3] https://exonerator.torproject.org/

[4] http://torstatus.blutmagie.de/

- dev-python/flask: a microframework used for the REST API (not necessary if you do not plan to run the API).
- Web server for running the REST API (not necessary if you do not plan to run the API), this guide expects apache2 with libapache2-mod-wsgi-py3 aka www-apache/mod_wsgi, https://github.com/GrahamDumpleton/mod_wsgi/.
- dev-python/beautifulsoup: used only for testing (not necessary for deployment only installations), https://www.crummy.com/software/BeautifulSoup/bs4/
- dev-python/flask-testing: used only for testing (not necessary for deployment only installations), https://github.com/jarus/flask-testing/

To configure Flask and Apache, see http://flask.pocoo.org/docs/1.0/deploying/#deployment.

Create /var/www/html/rest_tor_history.wsgi with the following content:

```
{{{
 import sys
 sys.path.insert(0, '/path/to/the/application/rest.py')

 from rest import app as application
}}}
```

Modify the Apache host as follows:

```
<VirtualHost *>
    ServerName example.com

    WSGIDaemonProcess {rest_tor_history} user={www-data} group={www-data} threads=5
    WSGIScriptAlias / /var/www/html/rest_tor_history.wsgi

    <Directory /var/www/html>
        WSGIProcessGroup rest_tor_history
        WSGIApplicationGroup %{GLOBAL}
        Order deny,allow
        Allow from all
    </Directory>
</VirtualHost>
```

Once ready, consensus_parser can be run in any directory. See the help for supported attributes.

3 Initial set up

3.1 Tor data

Download Tor descriptor database from https://collector.torproject.org/archive/relay-descriptors/consensuses/.

Unpack the archive.

```
for f in consensuses-*.tar.xz
do
  tar xJvf $f
done
```

Preprocess the archive:
./consensus_parser.py -c {descriptors_dir} -w {preprocessed_data_dir}
Once ready, keep the software running using the following command:

```
./consensus_parser.py -c {descriptors_dir} -w {preprocessed_data_dir} -U -a
  -m {youremail}
```

3.2 MaxMind data

Register an account for MaxMind GeoLite2 database at https://www.maxmind.com/en/accounts/current/people/current.

Add the following to your cron script:

```
cd {geolite_dir} && wget ’https://download.maxmind.com/app/geoip_download?
  edition_id=GeoLite2-ASN-CSV&license_key={MaxMind_key}&suffix=zip’
  -O GeoLite2-ASN-CSV.zip &&
 unzip GeoLite2-ASN-CSV.zip &&
 mv GeoLite2-ASN-CSV.zip archive/GeoLite2-ASN-CSV-‘date ’+\%Y-\%m-\%d’‘.zip &&
 {TorConsensusParser_dir}/preprocess_geolite.py
  --input ‘ls | grep ASN | tail -1‘
  --output /tmp/gl && mv /tmp/gl/* {geolite_preprocessed_dir}
[ "$(date ’+\%a’)" = "Wed" ] && cd {geolite_dir} &&
 wget ’https://download.maxmind.com/app/geoip_download?
  edition_id=GeoLite2-City-CSV&license_key=kymb1vDoa2j3&suffix=zip’
  -O GeoLite2-City-CSV.zip &&
 unzip GeoLite2-City-CSV.zip &&
 mv GeoLite2-City-CSV.zip archive/GeoLite2-City-CSV-‘date ’+\%Y-\%m-\%d’‘.zip &&
 {TorConsensusParser_dir}/preprocess_geolite.py
  --input ‘ls | grep City | tail -1‘
  --output /tmp/gl &&
  mv /tmp/gl/* {geolite_preprocessed_dir}
```

4 Toreator GUI

See the README.md for more information about this SW. Typically, you are going to use the Docker set up described below.

4.1 Toreator Proxy

Please instal Docker as the first step[5]. This project is only supported as a Docker image.

[5] https://docs.docker.com/install/

Once you install Docker, edit the docker-compose.yml volume section. Select a directory where to store the archives.

Then, installation of the applications is as simple as:

```
docker-compose build
docker-compose up -d
```

4.2 Toreator UI

Please instal Docker as the first step[6]. This project is only supported as a Docker image.

Once you install Docker, edit the docker-compose.yml volume section. Select a directory where to store the archives.

Then, installation of the applications is as simple as:

```
docker-compose build
docker-compose up -d
```

[6] https://docs.docker.com/install/