Assignment 7: Explore ROV deployment using BGP2GO and BGPStream (30pts)

Due on Wednesday March 22, 2023 11:59pm

1 Introduction

The goal of this assignment is to provide an introduction to working with raw BGP routing data, driven by the following research question:

Which autonomous systems (AS) perform Route Origin Validation (ROV)?

Researchers have recently tackled this measurement challenge in different ways[1][2][3][4][5]. This assignment considers a more tractable from of the problem:

Which collector peers forward ROA-invalid updates to the collector?

The following sections will guide you through the process of tackling this question, with a brief review of the BGP collector project data, how to parse MRT files, exploring MRT files using CAIDA's BGP2GO system and with BGPstream, and applying these skills to measuring a proxy metric of ROV deployment.

[1] "ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment", 2022. https://www.ndss-symposium.org/ndss-paper/auto-draft-183

[2] "Revisiting RPKI Route Origin Validation on the Data Plane", 2021. https://homepages.dcc.ufmg.br/~cunha/papers/rodday21tma-rov.pdf

[3] "RPKI is Coming of Age A Longitudinal Study of RPKI Deployment and Invalid Route Origins" (2019). https://dl.acm.org/doi/pdf/10.1145/3355369.3355596

[4] "BGP Beacons, Network Tomography, and Bayesian Computation to Locate Route Flap Damping", 2020. https://archive.psg.com/201029.imc-rfd.pdf

[5] "To Filter or not to Filter: Measuring the Benefits of Registering in the RPKI Today", 2021. https://people.csail.mit.edu/ctestart/publications/RPKIfiltering.pdf

1.1 Due Date

Wednesday March 22, 2023 11:59pm

1.2 Submission Instructions

You will answer the questions in a written PDF report and submit to Gradescope.

2 Where does BGP routing data come from?

A BGP collector collects updates from multiple ASes, called *collector peers*. For example, if we observe an update with the AS path *A B C D E* at a collector:

```
Collector <- A B C D E
```

then E is the origin AS of the update, which collector peer A forwards to the collector.

BGP collectors periodically dump routing information as so-called MRT files[6]. There are two types of MRT files:

- RIB: snapshot of collectors' routing table
- Update: multiple update messages received by collector

While a **RIB** contains a snapshot of the routing table, **update** files contain single changes (announcements and withdrawals) to that routing table. However, those changes are sometimes not reflected in subsequent RIBs. Thus some research questions require both types.

Routeviews[7] and RIPE RIS[8] are the two most popular *route collector projects*. Both archive and publish RIB and update MRT files from their many distributed collectors.

In the case of the Routeviews `route-views3` collector, you can see the RIB files at

http://routeviews.org/route-views3/bgpdata/2023.01/RIBS/

[6] https://ris.ripe.net/docs/20_raw_data_mrt.html
[7] http://archive.routeviews.org/
[8] https://ris.ripe.net/docs/10_routecollectors.html

and the corresponding update files at

http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/

Let us consider the following example URL:

http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2

This URL directly points to a compressed MRT update file: `updates.20230102.0230.bz2`.

Question 2.1: Imagine you want to automate the download process of MRT files. Given the above URL, what are the variable elements that you can change to point to other files?

Question 2.2: How many collectors does each collector project operate?

Question 2.3: What are the typical dump intervals of RIBs and updates for Routeviews and RIPE RIS?

Question 2.4: Briefly discuss the pros and cons of the chosen intervals, e.g., with respect to disk space (provider-side), processing time, or visibility (user-side).

3 Parsing single MRT files

Next, we cover various ways of retrieving and processing MRT files. We first look at the general data structure that these tools output. Consider following MRT update file:

http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2

For convenience, we provide an annotated HTML version of bgpreader's output of that MRT file: http://nids.caida.org:45000/cgi-bin/bgpreader.sh?http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2,100,1299

In the annotated HTML version, hover over the table header text, e.g., timestamp, to get more information on the output format.

Question 3.1: How much time elapsed from line 80 to line 120?

Question 3.2: In the listed updates, does AS 1299 send updates directly to a collector? Briefly justify your answer.

Before moving to the next section, familiarize yourself with the output format of your chosen tool. RIPE links to many command-line MRT file parsers with similar functions, usage, and output format: https://ris.ripe.net/docs/20_raw_data_mrt.html#tooling.

One tool to parse a single MRT file is *bgpkit*: https://bgpkit.com/parser.

```
bgpkit http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2
```

Another useful tool is bgpreader: https://bgpstream.caida.org/docs/tools/bgpreader (part of CAIDAs BGPStream https://bgpstream.caida.org/docs/install):

```
bgpreader -d singlefile -o \
upd-file=http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2
```

An older MRT-parsing tool is RIPE's *bgpdump*: https://github.com/RIPE-NCC/bgpdump:

```
wget http://routeviews.org/route-views3/bgpdata/2023.01/UPDATES/updates.20230102.0230.bz2
bgpdump -m updates.20230102.0230.bz2 | less
```

All the above tools support the `--help` switch for more usage information.

Note: To unleash the full potential of BGP2GO, you should be relying on bgpreader. Please check documentation on how to install bgpreader on macOS[9] *or inside a docker container.*[10]

[9] https://bgpstream.caida.org/docs/install/bgpstream#osx
[10] https://bgpstream.caida.org/docs/tutorials/docker

4 Data exploration using BGP2GO

For the analysis, you will use special prefixes, so-called RPKI routing beacons.[11] Routing beacons are injected into the routing system to study their propagation behavior. This assignment focuses on beacons with different ROA validity states to observe their propagation.

RIPE RIS announces the following three beacons from the same location:

- 93.175.146.0/24 announced by AS12654 which is the *valid* origin as per ROA.
- 93.175.147.0/24 announced by AS196615 which is the *invalid* origin as per ROA.
- 84.205.83.0/24 is not a ROA protected prefix, i.e., *unknown*.

Those prefixes are constantly announced, i.e., they are never withdrawn by the origin AS.

In the following, you will explore those prefixes using BGP2GO, a web app with an interactive exploratory interface that indexes MRT update files based on metadata generated from those files. BGP2GO allows you to:

- study macroscopic trends of prefixes, ASNs, and communities appearing in update files.
- customize the set of update files for subsequent analysis, by selecting the date and the collectors.

The BGP2GO web application is accessible via http://nids.caida.org:44444/[12].

Note: BGP2GO is under development so please be patient during loading.

Question 4.1: Using BGP2GO, look up each of the three beacons and apply the following filters: years=2022, months=January, and collectors=route-views2 (or simply follow the footnotes: 93.175.146.0/24[13], 93.175.147.0/24[14], 84.205.83.0/24[15]).

BGP2GO displays the *overall* number of occurrences, files containing the occurrences, aggregate size of these files, and number of collectors tha t observed the specified identifier and range.

[11] https://ris.ripe.net/docs/30_routing_beacons.html#resource-certification-rpki-routing-beacons
[12] http://nids.caida.org:44444/
[13] http://nids.caida.org:44444/details?pre=93.175.146.0/24&years=2022&months=1&collectors=1
[14] http://nids.caida.org:44444/details?pre=93.175.147.0/24&years=2022&months=1&collectors=1
[15] http://nids.caida.org:44444/details?pre=84.205.83.0/24&years=2022&months=1&collectors=1

Upon applying the above filters, the plots in BGP2GO will switch to a daily granularity, which enables one to discern detailed pattern.

Question 4.2: Briefly describe differences and commonalities in occurrence patterns among these prefixes. Justify your answer using their ROA validity status.

Question 4.3: Given that the beacons are never withdrawn by the origin AS, why do you observe withdrawals? Briefly justify.

By applying filters, you can control the number of files and the aggregated file size for subsequent analysis in BGPStream (see *CURRENT* row).

For the ROA-invalid prefix 93.175.147.0/24:

Question 4.4: How many files are selected?

Question 4.5: What is the aggregated file size?

5 Data analysis using BGPStream

In the following, you will take the first step in working with raw BGP data. You will use BGPStream to stream MRT files that you previously selected via BGP2GO.

Question 5.1: While in BGP2GO, analyzing the ROA-invalid prefix 93.175.147.0/24, click the button "BGPStream". A dialog box will show up that will provide you with two equivalent methods to perform the analysis:

1. BGPStream in the command line using bgpreader:

```
bgpreader -d csvfile -o csv-file=\
"http://nids.caida.org:45000/cgi-bin/urls.sh?pre=93.175.147.0/24&years=2022&months=1&collectors=1" \
  2</dev/null | grep "93.175.147.0/24"
```

   This command instructs `bgpreader` to output all MRT update files that contain the ROA-invalid prefix, and filter (`grep`) for all respective announcements and withdrawals.

2. Python code using PyBGPStream (python API):

```
import pybgpstream

bgp2go_meta = 'http://nids.caida.org:45000/cgi-bin/urls.sh?'
bgp2go_meta += 'pre=93.175.147.0/24&years=2022&months=1&collectors=1'

stream = pybgpstream.BGPStream(data_interface="csvfile")
stream.set_data_interface_option("csvfile", "csv-file", bgp2go_meta)
stream.add_filter('prefix', '93.175.147.0/24')

for elem in stream:
    print(elem)
```

Question 5.2: How many lines contain the ROA-invalid prefix?

Note: Depending on your machine and your Internet connection, performing the analysis can take up to 30 minutes. Where possible, store intermediate results to avoid recomputation.

6 Measuring ROV deployment

Given the accumulated knowledge and assembled tools so far, you are going to analyze the forwarding behavior of collector peers regarding ROA-invalid prefixes, using raw BGP data.

Question 6.1: How many collector peers forward the ROA-invalid prefix 93.175.147.0/24?

Question 6.2: How has the number of collector peers forwarding ROA-invalid prefixes changed over the last ten years?

Question 6.3: When a collector peer forwards ROA-invalid prefixes to the collector, does it send that prefix to other AS neighbors, too? Briefly justify your answer.

Question 6.4: Can the AS path be used to determine whether ASes along the path are forwarding ROA-invalid prefixes? Briefly justify your answer.