Assignment 3: Measuring the Global DNS Infrastructure (24pts)

Due on Monday, February 20th 11:59pm

1 Introduction

There are three goals of this assignment. The first is to understand different types of DNS queries and responses and be able to conduct active measurement of the global DNS infrastructure. The second is to be able to map IP addresses to their corresponding Autonomous Systems (ASes) in the Internet routing system. The third is to be able to jointly look at the first two goals and conduct analysis. The assignment aims to work through how answering real-world security questions may need a wide range of comprehensive data sets. Specifically, you will complete the following objectives:

(1) Parse the CAIDA Routeviews Prefix2AS dataset

(2) Use the ZDNS tool to evaluate `.gov` domains

(3) Use existing CAIDA datasets to analyze `.gov` domains.

1.1 Due Date

Monday, February 20th 11:59pm

1.2 Submission Instructions

There will be 4 tasks in this assignment. You will complete task 1 in a `python` file, and submit `pa3q1.py` to the **PA3 Autograder** on Gradescope. Your scores will be immediately available upon submission and you have unlimited submission attempts until due date. The remaining 3 tasks will in the form of a written homework, where you will put the results in a PDF and submit to the **PA3 Analysis** on Gradescope.

2 Mapping IP addresses to AS Numbers

For various topology-related projects, we need a mapping from an IP address to the Autonomous System (AS) that owns that IP address. The most common approach to map IP addresses to ASes is to use BGP table dumps from public sources like Routeviews and RIPE, and then perform a longest-prefix match on the set of prefixes. We are currently using one routing table from Routeviews (RV2) to map IP addresses to ASes.

2.1 CAIDA Routeviews Prefix2AS

The CAIDA Routeviews Prefix2AS dataset contains IP prefixes announced in BGP and their origin ASes.

2.1.1 Data Access

The CAIDA Routeviews Prefix2AS dataset can be downloaded at: `https://www.caida.org/catalog/datasets/routeviews-prefix2as/`

In the webpage from the link above, first read through "Note on Multi-origin ASes (MOASes)", then scroll down to the bottom of the page and click the link in the "Data Access" section. Then you will be redirected the data directory page with a list of directories. Click on the `Routeviews-prefix2as` directory and continue to enter the `2023/` and `02/` directory, and then you will see a list of prefix2as files. You will download `Routeviews-rv2-20230201-1200.pfx2as.gz` to complete the tasks in the next section.

For your convenience, the file is also provided here: `https://cseweb.ucsd.edu/classes/wi23/cse291-e/pa3/routeviews-rv2-20230201-1200.pfx2as.gz`

2.1.2 Data Format

The file format is line-oriented, with one prefix-AS mapping per line. The tab-separated fields are

* IP prefix
* prefix length
* AS

The AS can be a single AS number, an AS set (e.g. `{32,54}`), or a multi-origin AS (e.g. `10_20`). You should treat a multi-origin AS entry as separate BGP announcements. For example, the following announcement

```
IP Prefix   prefix length   AS
1.0.0.0     24              7377_195
```

should be interpreted as

```
1.0.0.0   24   7377
1.0.0.0   24   195
```

For this assignment, you can ignore *AS set* entries. An example *AS set* entry is below:

```
IP Prefix   prefix length   AS
2.0.0.0     16              {123, 456}
```

2.1.3 Data Processing

You will need to maintain a list of IP prefixes and their origin ASes. For instance, in Python, you should parse the multi-origin AS example above into the following dictionary entry:

```
{"1.0.0.0/24": [7377, 195]}
```

2.2 Longest Prefix Matching

Refer to the lecture slides on Jan 30 to review the mechanism for longest prefix matching. Longest prefix matching is usually implemented using a Trie data structure.

2.2.1 Existing Libraries

There are existing python libraries that implements longest prefix matching. You can install and import them into your python script so you don't have to re-invent the wheel. The autograder will have the following 2 libraries installed:

- `py-radix`: Docs and Install: `https://pypi.org/project/py-radix/`
- `pytricia`: Docs: `https://github.com/jsommers/pytricia`
  Install: `https://pypi.org/project/pytricia/`

2.2.2 Example

For example, after parsing the CAIDA Routeviews Prefix2AS Dataset, imagine you obtain the following prefix2as mapping:

```
{
    "1.0.0.0/24": [7377, 195],
    "1.0.0.0/16": [3356]
}
```

In this case, the IP address `1.0.0.0` should map to the ASes `7377, 195`. The IP address `1.0.4.0` should map to the AS `3356`.

2.3 Task 1 - Autograder: Parse the Prefix2AS (5pts)

Your first task is to parse the Prefix2AS dataset and provide a list of ASes for any given IP address. Your code should do the following:

(1) Read an input file `input.txt`, which contains a **single** IP address. The autograder will place the input file in the same directory it places your `python` file. Example `input.txt`:

```
8.8.8.8
```

(2) Parse the Prefix2AS dataset and obtain a prefix-to-AS mapping.

(3) Match the IP address to the longest prefix and find its corresponding AS(es) in this mapping.

(3) Print the AS numbers, each separated by a newline, into `output.txt`. The ordering of AS numbers does not matter. The autograder expects the output file to be in the same directory as your `python` file. Example `output.txt`:

```
15169
```

You will complete this task in `pa3q1.py`. Your code should run using the following command:

```
python3 pa3q1.py
```

3 ZDNS

You will use `zdns` which is a command line utility for high-speed DNS lookups. We will use `zdns` to learn practical aspects of DNS.

3.1 Installation

First, download and install `go` on your machine. Instructions: `https://go.dev/doc/install`. Then install `zdns`. Instructions: `https://github.com/zmap/zdns`. You can use the following commands:

```
git clone https://github.com/zmap/zdns.git
cd zdns
go build
```

The 'go build' command should create a zdns binary in the folder.

3.2 Run `zdns`

Run the following command to test `zdns`:

```
echo caida.org | ./zdns A --iterative
```

This command runs the `A` query for the domain `caida.org`. You should see the output similar to the following in your command line:


```
{"data":{"answers":[{"answer":"192.172.226.78","class":"IN","name":"caida.o
rg","ttl":14400,"type":"A"}],"protocol":"udp","resolver":"128.54.16.2:53"},
"name":"caida.org","status":"NOERROR","timestamp":"2023-01-12T20:18:02-08:0
0"}
```


3.3 Task 2: Run DNS queries on `.gov` domains (6pts)

This task is a written task. You will submit a pdf of your results and answers to Gradescope. You will answer the following questions:

(1) Get a list of .gov domains. You can find an up to date list of .gov domains maintained by CISA here: `https://github.com/cisagov/dotgov-data/blob/main/current-full.csv`. Prepare a list of .gov domains as input to `zdns`. Answer the following question: **How many .gov domains are in the list? (1pt)**

(2) Run NS, A, AAAA queries using zdns for all .gov domains. **For each query, how many domains returned NOERROR, NXDOMAIN, or some other RCODE? (5pts)** Put the numbers in a table similar to the following:

RCODE	**NS**	**A**	**AAAA**
NOERROR			
NXDOMAIN			
SERVFAIL			
...			
TOTAL			

3.4 Task 3: Cross-reference with prefix2as and AS Organization (7pts)

Cross-Reference the IP addresses with prefix2as to find their origin AS(es) and their registration countries. Recall the AS Organization dataset from the Assignment 2, find the field in the data file that contains the country code of the organization.

You will use the January 2023 snapshot of the AS Organization file. The file is provided here: `https://cseweb.ucsd.edu/classes/wi23/cse291-e/pa3/20230101.as-org2info.txt.gz`.

Answer the following questions:

(1) **How many ASNs originate each .gov domain's NS, A, and AAAA records? Draw a Cumulative Distribution Function (CDF) plot showing those numbers for each type of record. (5pts)** You will need to further resolve the domain names in the NS records to IP addresses.

(2) **Are any ASNs geolocated outside the US? If so, which ones? (2pts)**

3.5 Task 4: Cross-reference with AS Relationship and AS Organization (6pts)

To be reachable on the global Internet, each domain depends on the AS(es) originating the IP addresses for hostnames in the domain (A and AAAA records) and the authoritative nameserve that provides mappings from hostnames to IP addresses. Those ASes, in turn, may have dependecies on upstream providers, i.e., those from whom they buy transit. For each AS you found in Task 3, use the most recent AS relationships data to identify the upstream ASNs that the AS relies on for transit.

You will use the January 2023 AS Relationship snapshot. The file is provided here: `https://cseweb.ucsd.edu/classes/wi23/cse291-e/pa3/20230101.as-rel.txt.bz2`

Answer the following questions:

(1) **How many different upstream ASNs does each domain depend on? (4pts)** Think of some ways that you can present your answer – draw a CDF? Table? Histogram?

(2) **Are any of these upstream ASNs geolocated outside the US? If so, which ones? (2pts)**