

Who is Using This IP Address?

A frequent question that faces network administrators or application developers consists in identifying *who is using a specific public IP address*. This information can be utilized for instance to perform user localization and enable location-based services or user access control. In this context, a main technical challenge is to associate the IP address or prefix with its corresponding Autonomous System (AS).

A typical method to identify the AS that announces a specific IP prefix is to use the whois protocol. A whois command is available on main OSES and enables to query the databases of regional registries such as ARIN, RIPE, LACNIC, ... A very interesting [article](#) provides tips for using the whois command. Here is a simple example that queries the whois.ripe.net server database in order to find the origin AS of the 148.60.0.0/16.

```
$ whois -h whois.ripe.net 148.60.0.0/16 | grep origin
origin:          AS2200
```

However things get complicated very rapidly since the route object information is not always provided or may be outdated. Trying for example to identify the AS announcing 203.178.141.194 (corresponding to the famous www.kame.net), no answer is obtained since the corresponding route object is not registered by WIDE.

```
$ whois -h whois.apnic.net 203.178.141.194 | grep origin
```

An alternative method for identifying the AS that announces a specific IP prefix consists in studying the BGP routing information. Typically, each BGP speaking router stores in a BGP table the routing announcements received for each prefix together with some protocol attributes such as the AS-PATH. This attribute contains the list of ASes traversed by the BGP announcement, with the first AS being the origin AS for the IP prefix! Therefore, the problem boils down to parsing the BGP routing information, matching the IP address or prefix, and then extracting the origin AS from the AS-PATH attribute. Such process is obviously optimal when the router has a global view of the Internet: this is the case for routers participating in the Default Free Zone (DFZ) where the BGP tables contain *all the prefixes* announced in the Internet. As of 2014, these routers have around 500 000 active BGP entries according to the latest statistics.



Figure 1. Active BGP entries

However, having access to a DFZ BGP router is not easy in practice. Alternatively, it is possible to find similar routing information on looking glasses or route servers that are made public by network operators (see for example a list of servers on www.routeserver.org). Such devices are originally deployed in order to contribute to the monitoring or the tracking of BGP anomalies in the Internet. Let us try for example to log on the Allstream route server in Canada and identify the origin AS of 148.60.0.0/16. The output of the `show ip bgp` command shows the AS path 15290 3356 1273 2200 in the BGP announcements. Therefore, the first AS, *i.e.*, AS 2200 is the origin AS of the studied prefix.

```
$ telnet route-server.east.bb.allstream.net
route-server.east>show ip bgp 148.60.0.0/16
BGP routing table entry for 148.60.0.0/16, version 270487514
Paths: (4 available, best #4, table default)
  Not advertised to any peer
  15290 3356 1273 2200
    199.212.162.69 from 199.212.162.69 (199.212.162.69)
      Origin IGP, localpref 100, valid, external
      Community: 15290:3356 15290:64995 15290:65050 15290:65506
  15290 3356 1273 2200
    199.212.162.68 from 199.212.162.68 (199.212.162.68)
      Origin IGP, localpref 100, valid, external
      Community: 15290:3356 15290:64995 15290:65050 15290:65506
  15290 3356 1273 2200
    199.212.162.66 from 199.212.162.66 (199.212.162.66)
      Origin IGP, localpref 100, valid, external
      Community: 15290:3356 15290:64995 15290:65050 15290:65506
  15290 3356 1273 2200
    199.212.162.67 from 199.212.162.67 (199.212.162.67)
      Origin IGP, localpref 100, valid, external, best
```

```
Community: 15290:3356 15290:64995 15290:65050 15290:65506
```

Despite its availability, this method remains cumbersome, especially if you want to quickly look up something or if you have a large number of prefixes that you want to analyse with a script. Fortunately, RIPE NCC and Team Cymru have already answered these questions: they provide solutions that combine the versatility of the whois protocol with the accuracy of the BGP information. In other words, you keep on using the legacy whois command but you get BGP-based results.

Team Cymru implements the `whois.cymru.com` server which provides the announcing AS number and name for any given IP prefix. The information in its database is based on the BGP feeds from 50+ BGP peers, and is updated at 4 hour intervals. Here is a simple example for using the cymru service:

```
$ whois -h whois.cymru.com 148.60.0.0/16
AS      | IP          | AS Name
2200    | 148.60.0.0 | FR-RENATER Reseau National de
telecommunications pour la Technologie
```

and another example that demonstrates the possibility of sending multiple prefixes in the same query:

```
$ whois -h whois.cymru.com 148.60.0.0/16 203.178.141.194
AS      | IP          | AS Name
2200    | 148.60.0.0 | FR-RENATER Reseau National de
telecommunications pour la Technologie
AS      | IP          | AS Name
2500    | 203.178.141.194 | WIDE-BB WIDE Project
```

RIPE NCC implements a similar whois service named RISwhois by providing a higher level view over the most recently collected set of routing tables from the Remote Route Collectors (RRCs) at different [locations](#) in the world. Given an IPv4 or IPv6 prefix, RISwhois will tell which prefixes and origin ASes on which RRCs match that particular IP.



As mentioned on the [Riswhois](#) website, BGP information is more accurate than that contained in the databases of the regional registries: 21% of a set of unique IPs were unmatched when using the routing registry vs. only 1% unmatched when using RIS data.

In the following, a simple example shows the output of a Riswhois query: `203.178.141.194` is originated by AS 2500 as seen by 16 RRCs.

```
$ whois -h riswhois.ripe.net 203.178.141.194
route:      203.178.128.0/17
origin:     AS2500
descr:      WIDE-BB WIDE Project
lastupd-frst: 2014-01-23 12:42Z 202.249.2.185@rrc06
lastupd-last: 2014-02-08 13:26Z 187.16.218.21@rrc15
seen-at:
rrc00, rrc01, rrc03, rrc04, rrc05, rrc06, rrc07, rrc10, rrc11, rrc12, rrc13, rrc14, rrc15
```

```
num-rispeers: 105  
source:      RISWHOIS
```



Due to BGP policies between ASes in the Internet, RRCs may receive different BGP information for the same IP prefix. Therefore, Riswhois provides multiple matchings for the IP prefix, as in the following example. In such cases, a longest prefix matching may help in choosing a single originating AS.

```
$ whois -h riswhois.ripe.net 217.70.184.1  
route:      217.0.0.0/8  
origin:     AS3303  
descr:     SWISSCOM Swisscom (Switzerland) Ltd  
lastupd-frst: 2014-01-30 00:20Z 217.29.66.120@rrc10  
lastupd-last: 2014-02-04 14:36Z 192.65.185.243@rrc04  
seen-at:    rrc04, rrc10  
num-rispeers: 2  
source:     RISWHOIS  
  
route:      217.70.176.0/20  
origin:     AS29169  
descr:     GANDI-AS Gandi SAS  
lastupd-frst: 2013-10-21 08:55Z 202.249.2.185@rrc06  
lastupd-last: 2014-02-08 13:28Z 187.16.218.21@rrc15  
seen-at:    rrc00, rrc01, rrc03, rrc04, rrc05, rrc06, rrc07, rrc10, rrc11, rrc12, rrc13, rrc14, rrc15  
num-rispeers: 111  
source:     RISWHOIS
```

When accessing the online servers (Riswhois, Cymru or legacy whois servers) is not possible or not recommended, an autonomous implementation of an equivalent server is possible. This can be the case when service availability constraints are very tight or when the Internet connection is not permanent. Here are some hints and recipes to implement a server that maps IP prefixes with AS numbers based on BGP information.

Start by downloading raw BGP data collected by RIPE NCC servers from <http://www.ripe.net/data-tools/stats/ris/ris-raw-data>. For example, the latest data file from RRC0 (Amsterdam) is always available on <http://data.ripe.net/rrc00/latest-bview.gz>.

```
$ wget http://data.ripe.net/rrc00/latest-bview.gz
```

As raw data is written in MRT format, you need to install [bgpdump](#) in order to parse it easily.

```
$ zcat latest-bview.gz | bgpdump -m - > latest-bview-parsed.txt
```

Here is a typical line in the dumped file, where you can see the prefix 148.60.0.0/16 and the AS-PATH ending by the originating AS 2200.

```
TABLE_DUMP2 | 1389513606 | B | 85.132.60.10 | 29049 | 148.60.0.0/16 | 29049 1273  
2200 | IGP | 85  
.132.60.10|0|0|1273:12250 2200:1000 2200:2200 | NAG | |
```

Now you can use your favorite scripting language to extract IP to AS mappings and perform a best prefix match.

Fortunately, existing software tools may help you in the process of implementing you own BGP-based IP to AS mapping service. Consider taking a look at:

1. [Net-NfDump](#): a perl library that makes use of the [Net::IP::LPM](#) implementation of Longest Prefix Match algorithm.
2. [IP-ASN-history](#): a client/server (with a client web interface) software to store efficiently the history of BGP announces and quickly lookup IP addresses origins.

From:

<http://wiki.lahoud.fr/> - **wikiroute**

Permanent link:

http://wiki.lahoud.fr/doku.php?id=who_is_using_this_ip_address&rev=1391948604

Last update: **2014/02/09 13:23**

