February 2026
Geoff Huston

From the Stupid DNS Tricks Department: ipasn.net

There have been a number of services that allow a lookup of an IP address or Autonomous System Number (ASN) and return information about that IP number resource. The Regional Internet Registries (RIRs) each operate a database that records (among other data items) the number resource and the details of the entity that is described in the relevant number registration record. The data can be queried using individual queries using the *whois* query tool, by giving an IP Address or an Autonomous System Number in the query, and the response is based on the registry's database entries relating to that IP address or AS Number in response.

> There are many other similar services. An Internet search using the term "tools to retrieve information about an IP address" will reveal quite a collection of such services.

One such service that caught my attention was the ip-to-asn mapping service provided by Team Cymru (https://www.team-cymru.com/ip-asn-mapping), and in particular the DNS variant of this service.

This service requires some pre-processing of the query string, in that the octets of the IP address need to be reversed and in the case of an IPv6 address the ':' delimiter needs to be replaced by the same reverse order of the string of 8-bit octets. Here's a couple of examples:

```
$ dig +short TXT  31.108.90.216.origin.asn.cymru.com
"3561 | 216.88.0.0/14 | US | arin | 1998-09-25"

$ dig +short TXT 8.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.0.b.0.6.8.4.1.0.0.2.origin6.asn.cymru.com.
"15169 | 2001:4860::/32 | US | arin | 2005-03-14"
```

That IPv6 address query is just ugly, and the IPv4 address used in the query is really not any better! This need for reverse ordering and translating IPv6 addresses seems to me to be an unnecessary step. What would be easier is to query directly for an IPv4 address or an IPv6 address without any pre-processing. Can we do this?

"Classic" DNS uses the concept of a "zone file", which lists all the defined labels in the zone. When a server receives a query name (and query type), the to resolve the query the server looks for an exact match of the query name with the labels defined in the zone file, and if a match is found it assembles the matching records as its response. However, the concept of a "zone file" is convenient, but not necessary. Some servers use a database backend in place of a zone file and perform a database lookup to find the response record. More generally, in place of an exact match in a zone file, or a database lookup, a DNS authoritative server could execute any procedure it wanted that generated a DNS response that is correlated in some manner to the query it received.

Secondly, we can revisit the question of what characters are permitted in a DNS query. The answer is that, perhaps in spite of the strictures described in various RFCs, the DNS in practice is surprisingly flexible here as to what characters can be used to form DNS query names. Even the '.' character, normally used to define the "cut points" from one zone to a delegated subordinate zone can also be used within a DNS zone label. The DNS appears not to care!

To assemble this particular stupid DNS trick, we can put these two observations together. We need to use a DNS authoritative server platform that supports the concept of using a "plug-in" backend server in place of a zone file lookup. Luckily, the PowerDNS authoritative server supports this function. And, unlike the ip-asn-mapping DNS tool provided by Team Cymru, it's possible to create an authoritative server that does not require reversing octets.

The result is similar to the ip-asn map:

```
$ dig +short TXT 216.88.0.0.origin.asn.ipasn.net
"216.88.0.0/14 | 3561 | US | arin | 1998-09-25"
```

and

```
$ dig +short TXT 2401:2000:6660::.origin6.asn.ipasn.net
"2401:2000::/32 | 4608 | AU | apnic | 2007-06-19"
```

This data is assembled by looking up the address prefix in a current BGP routing table snapshot in order to retrieve the origin AS, a lookup into a geolocation database to get the country code where the address prefix is located, and then a lookup into RIRs' statistics reports to retrieve the RIR that holds the registration record and the date of registration.

Maybe we can push this concept a bit further. For example:

```
$ dig +short TXT 216.88.0.0.ipasn.net
"216.88.0.0|IPv4|ADVERTISED|216.88.0.0/14|3561|CenturyLink_Communications,_LLC|US|United_States_of_A
merica|arin|216.88.0.0/14|assigned|1998-09-25|VLD|216.88.0.0/14|24|3561|ARIN"
```

This output shows that the advertised address prefix is a /14, the originating AS is AS3561, the name of the organisation that holds this prefix is "CenturyLink_Communications, LLC" in the United States, the registry status of this address prefix is "assigned" by ARIN, and the date of the registration is 25 September 1998. There is a valid ROA published for this address, using a maximum length of 24 and an originating AS of AS3561, using the RPKI trust anchor for this ROA is operated by ARIN.

It's possible to generate the same information formatted as a json data object:

```
$ dig +short TXT 216.88.0.0.json.ipasn.net
"{\"Address\":\"216.88.0.0\",\"Class\":\"IPv4\",\"BGP\":\"ADVERTISED\",\"Advertised_Prefix\":\"216.8
8.0.0/14\",\"Origin_AS\":\"3561\",\"Org_Name\":\"CenturyLink_Communications,_LLC\",\"CC\":\"US\",\"C
C_Name\":\"United_States_of_America\",\"RIR\":\"arin\",\"RIR_Prefix\":\"216.88.0.0/14\",\"Reg_Sta"
"tus\":\"assigned\",\"Reg_Date\":\"1998-0-25\",\"ROV\":\"VLD\",\"ROA_Prefix\":\"216.88.0.0/14\",
\"ROA_Maxlen\":\"24\",\"ROA_AS\":\"3561\",\"ROA_TAL\":\"ARIN\"}"
```

DNS recursion is also supported, by having the back-end process perform its own DNS resolution operation.

```
$ dig +short TXT www.potaroo.net.dns.ipasn.net
"2401:2000:6660:0:0:0:0:108|IPv6|ADVERTISED|2401:2000::/32|4608|APNIC_Research_and_Development|AU|
Australia|apnic|2401:2000::/32|assigned|2007-06-19|VLD|2401:2000::/32|35|4608|APNIC_RPKI_Root"
```

Or, if you wanted the IPv4 address record rather than the default IPv6 record:

```
$ dig +short TXT www.potaroo.net.a.dns.ipasn.net
"203.133.248.108|IPv4|ADVERTISED|203.133.248.0/24|4608|APNIC_Research_and_Development|JP|Japan|apnic
|203.133.248.0/22|assigned|2007-05-22|VLD|203.133.248.0/23|24|4608|APNIC_RPKI_Root"
```

Individual attributes can also be queried:

```
$ dig +short TXT 216.88.0.0.cc.ipasn.net
"US"
```

```
$ dig +short TXT 216.88.0.0.rpki.ipasn.net
"VLD_216.88.0.0/14-24_3561_ARIN"
```

Is this useful?

Frankly I'm not sure if it's useful to anyone, but this exercise illustrates the inherent flexibility that lurks in the DNS, where it's possible to dynamically generate DNS responses based on the query string. The DNS system is not just a distributed directory but can be seen as a form of a distributed processing system, where queries can be seen as coded instructions, which are executed on DNS servers.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston AM, B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net