RFC8806 – is it enough?

Wes Hardaker

November 14, 2021

1 Background

[RFC8806](`https://datatracker.ietf.org/doc/html/rfc8806`) defines a mechanism where local resolvers can pre-cache the root zone with a defined goal of:

> The primary goals of this design are to provide more reliable answers for queries to the root zone during network attacks that affect the root servers and to prevent queries and responses from being visible on the network.

A project by the author, [LocalRoot](`https://localroot.isi.edu/`), extends this concept allowing users to pre-cache other zones as well, such as *root-servers.net* and the *arpa* TLD.

We briefly study the effects of turning off access to the root when a RFC8806 or *LocalRoot* resolver is in use.

2 Results

In the following scenarios we implement a local resolver with and without *RFC8806* and *LocalRoot* support and then perform a number of simple network tasks to test the resiliency of the offered service.

To firewall off the existing root servers to ensure they can't be reached, we run the simple 'bash' script in 4.1:

2.1 querying for normal domains

When we query the local *Unbound* resolver, with a disconnected root, that supports either *RFC8806* or *LocalRoot* we find the resolver able to properly return responses. Thus, even without the ability to contact the *root-servers.net* domain the resolver is sufficiently willing to trust the authoritative answers from the loaded zone 4.2.

2.2 traceroute

On the other hand, without the *arpa* zone being available, we see difference in traceroute output (captured in 4.3).

With the results from a *LocalRoot* enabled resolver returning some reverse lookup domains. One conclusion to draw from this is that debugging network issues is harder when there is a global or local outage to the DNS root's core DNS servers, which also serve *.arpa*.

3 Questions to be answered

Though we have quickly highlighted some difference between support offered by generic *RFC8806* resolvers and *LocalRoot* enabled resolvers, we wonder what other research questions should be studied. What other zones would be beneficial to pre-cache besides the root and the additional ones that *LocalRoot* supports?

4 Appendixes

4.1 Fire-walling off existing roots

«sec:roots»

```
iptables -N ROOTS
roots=$(dig @localhost . ns | grep '\tA\t' | awk '{print $NF}')
for addr in $roots ; do
  echo $addr
  iptables -A ROOTS -s $addr -j DROP
  iptables -A ROOTS -d $addr -j DROP
done
iptables -A ROOTS -j RETURN
```

4.2 Resolve

«sec:resolve»

The following shell code was used to study answers from with and without LocalRoot:

```
name_checks="www.google.com www.ietf.org www.icann.org www.example"
for item in $name_checks ; do
   echo " --------- $item"
   /bin/time dig +short @localhost $item 2>&1
done
```

4.3 Traceroute outputs

We capture two different traceroute outputs to *google.com* after the roots have been blocked (4.1), one run with LocalRoot enabled and another when the roots have been blocked.

```
traceroute www.google.com
```

With the results from a *LocalRoot* enabled resolver returning some reverse lookup domains:

```
traceroute to www.google.com (216.58.194.164), 30 hops max, 60 byte packets
 1  router (10.0.0.1)  0.262 ms  0.228 ms  0.206 ms
 2  68.78.72.22 (68.78.72.22)  1.687 ms  2.203 ms  2.701 ms
 3  * * *
 4  71.147.199.98 (71.147.199.98)  25.423 ms  25.904 ms  26.493 ms
 5  12.122.160.166 (12.122.160.166)  35.190 ms  35.704 ms  36.140 ms
 6  12.122.2.78 (12.122.2.78)  34.752 ms  38.533 ms  36.608 ms
 7  ggr3.sffca.ip.att.net (12.122.136.17)  36.130 ms  22.929 ms  21.965 ms
 8  12.255.10.244 (12.255.10.244)  24.710 ms  25.511 ms 12.255.10.240 (12.255.10.240)  26.204 ms
 9  * * *
10  108.170.243.1 (108.170.243.1)  31.414 ms 142.251.65.138 (142.251.65.138)  32.730 ms 142.251.224
11  sfo07s13-in-f4.1e100.net (216.58.194.164)  32.214 ms 108.170.243.14 (108.170.243.14)  33.294 ms
```

But a *RFC8806*-only resolver failing to provide the additional useful context:

```
traceroute to www.google.com (216.58.194.164), 30 hops max, 60 byte packets
 1  router (10.0.0.1)  0.174 ms  0.142 ms  0.175 ms
 2  68.78.72.22 (68.78.72.22)  62.951 ms  63.319 ms  63.700 ms
 3  * * *
 4  71.147.199.98 (71.147.199.98)  87.307 ms  89.229 ms  91.099 ms
 5  12.122.160.166 (12.122.160.166)  105.116 ms  103.417 ms  128.799 ms
```

```
 6  12.122.2.78 (12.122.2.78)  127.487 ms  126.465 ms  127.789 ms
 7  12.122.136.17 (12.122.136.17)  122.757 ms  22.747 ms  24.157 ms
 8  12.255.10.242 (12.255.10.242)  28.092 ms  29.589 ms  30.590 ms
 9  * * *
10  216.58.194.164 (216.58.194.164)  43.589 ms 142.251.70.104 (142.251.70.104)  44.532 ms  44.932 m
```