Flexible Testbed for Recursive Resolver Software

Paul Hoffman

DNS-OARC 27, San Jose, CA
30 September 2017

ICANN

v1

Overview

- The resolver server testbed
- The “test root” creator
- Use case: getting ready for the KSK rollover
- Next steps
- Some observations on creating tools for researchers

Resolver testbed overview (1)

- Goal: be able to launch resolvers with test-specific configurations, send queries, and see the results

- Filled with open-source resolvers (BIND, Knot Resolver, Unbound, PowerDNS Recursor, ...) but should also be able to handle proprietary servers (Windows Server, Nominum, InfoBlox, ...)

Resolver testbed overview (2)

- Testing should be automated
  - Target resolvers are tagged (“bind9”, “validator”, ..., and “all”)
  - Capture all output of queries sent to the resolvers in a test
  - Use tcpdump to capture all the traffic between the resolvers and the roots
- Can use alternate roots (coverd later in this presentation)

Not much of a man page

```
# ./resolver_test.py
Available commands are:
help:           Show this text
list:           Show the resolver names and their tags
update_sources: Update all resolver tarballs
fill_configs:   Update all resolver configuration files
make_bases:     Make the base images; this is probably
                only run once
build:          Build one or more resolvers; needs
                exactly one argument, either "all",
                the name of a resolver, or a tag
test:           Run a test; needs exactly two
                arguments, the test name and either
                "all", the name of a resolver, or a tag
```

Configuration file

Single file, JSON-formatted

```
"bind-9.10.0":
      { "tags": [ "bind9", "recent", "oct" ],
      "url": "ftp://ftp.isc.org//isc/bind9/9.10.0/bind-
9.10.0.tar.gz",
      "base": "common",
      "conf_type": "bind9",
      "make_str": "!bind9make",
      "start": "!bind9start"  },
```

Test root creator (1)

- People have done this before, I did it again, maybe you won’t need to the next time
- Needed something different than what I had seen, namely signing delegation that looks like the current root
  - Two KSKs and two ZSKs on October 10, but just one on October 11
  - Delegation to something that looks like root-servers.net

Test root creator (2)

- Takes in a configuration file, puts out a directory full of files
- Makes the needed keys and a BIND named.conf for running the test root on this box
- Also puts out configuration files for BIND, Unbound, and Knot Resolver to be able to use this root's addresses and trust anchors

Configuration file

Single file, ini-formatted

```
[confs]
directory = /root/oct10
ipv4 = 1.2.3.4 5.6.8.8
ipv6 = 2001::2001 2600:dead::beef
ksk-number = 2
zsk-number = 2

# zsk-type = rsa2048

# Suffix for the root name servers
#   Defaults to "some-servers.p53”
# suffix = something.notreal

# wrong-trust-anchor = false
```

Use case: the KSK rollover (1)

- <insert KSK rollover intro here>
- There will be some resolvers on October 11 (or 12 or 13, depending on TTLs) which will have only KSK2010
  - We tried; we still trying
- If there are a lot of them, can we identify them by the software they are running?
- Goal: get fingerprints of how resolvers look on October 10, then on October 11

Use case: the KSK rollover (2)

- Create test root that looks like October 10 (two KSKs, two ZSKs) and use the correct trust anchor
- Create test root that looks like October 11 (two KSKs, one ZSK) and use the wrong trust anchor
- See what we can learn by watching the tcpdump pcaps

Use case: the KSK rollover (3)

- We got fingerprints, but they weren’t that informative
  - Different versions of BIND look pretty much alike
  - BIND does look different than Unbound
  - PowerDNS Recursor doesn’t show much
- Still, if we have widespread problems, this might be useful

What we tested

- bind
  - 9.10.0, 9.10.3, 9.10.5, 9.10.6, 9.11.0, 9.11.1, 9.11.2, 9.2.9, 9.3.6, 9.4.3, 9.5.2, 9.6-ESV-R11, 9.6.3, 9.7.4, 9.7.7, 9.8.0, 9.8.4, 9.8.8, 9.9.0, 9.9.10, 9.9.11, 9.9.5

- pdns-recursor
  - 3.2, 3.3, 3.5, 3.5.1, 3.5.2, 3.5.3, 3.6.0, 3.6.1, 3.6.2, 3.6.3, 3.6.4, 3.7.1, 3.7.2, 3.7.3, 3.7.4, 4.0.0, 4.0.1, 4.0.2, 4.0.3, 4.0.4, 4.0.5, 4.0.6

- unbound
  - 1.3.4, 1.4.20, 1.4.21, 1.4.22, 1.5.10, 1.5.8, 1.5.9, 1.6.0, 1.6.1, 1.6.2, 1.6.3, 1.6.4

- Could have easily done more versions of these packages

- Could not test Knot Resolver because it cannot import a new set of root hints and new trust anchor

What’s next

- How resolvers pick and change authoritative servers based on timing and availability
  - Compare results with “Recursives in the Wild” and come up with additional tests
- Catalog current behavior for proposed changes in the DNSOP WG
- Stuff that you want

Offering these tools to others

- Creating tools for other researchers forced me to give these structure, but it might be fruitless
  - This type of tool quickly becomes personalized for the use case at hand
- Am hoping for collaboration, but forking is fine too

Please use these tools!

- https://github.com/icann/resolver-testbed
- https://github.com/icann/test-root
- If you think of interesting tests but find it too hard to get started, let’s talk