DNS RRL In Action

Paul Vixie, ISC

DNS-OARC, Toronto

October, 2012

RRL Motivations

- Internet mostly lacks admission control
  - Called “source address validation”, BCP38, SAC004
  - Is not the default due to economics and inertia
  - Means anybody can forge a packet from anybody
- DNS is a great DDoS reflector
  - Authority servers have to answer any client
  - Recursive servers are often open
  - DNSSEC makes it even better: amplification

What It Looks Like

```
[nsa:amd64] repeat 25 \
       dig +novc +ignore +retries=0 +time=1 vix.com aaaa \
              @ns.sql1.vix.com \
       | grep tc
;; flags: qr aa tc rd ad; QUERY: 1, ANS: 0, AUTH 0, ADD: 1
;; flags: qr aa tc rd ad; QUERY: 1, ANS: 0, AUTH: 0, ADD: 1
;; flags: qr aa tc rd ad; QUERY: 1, ANS: 0, AUTH: 0, ADD: 1
```

How It Works

```
options {
        directory "/var/local/named";
        pid-file "/var/run/named-nsa.pid";
        query-source address 149.20.48.227 port *;
        listen-on-v6 { ::1; 2001:4f8:3:30::3; };
        listen-on { 127.0.0.1; 149.20.48.227; };
        recursion yes;
        notify yes;
        dnssec-enable yes;
        dnssec-lookaside . trust-anchor dlv.isc.org.;
        dnssec-validation yes;
        rate-limit {
                responses-per-second 5;
                window 5;
        };
};
```

How You Can Use It

- In authority servers
  - RRL has no negative impact on real flows, because real clients have caches, will retry with UDP, will try TCP if given a truncated response
- In recursive servers
  - RRL would have a negative impact on real flows, because real clients do not have caches
  - But it should not be necessary, just use ACLs
  - Intentionally open recursives are outside of scope

Final Thoughts: DNS RRL

- RRL was first implemented in BIND but is not in any way BIND specific
- Other implementations would be welcome
- Please study the DNS RRL specification carefully, it’s intended to be implemented literally
- Please join the ratelimits@lists.redbarn.org mailing list if you want to discuss further