Bridging DNS

Over HTTP(S) using JSON

brian.peter.dickson at gmail.com

What Are The Problems?

- There are several similar and/or related problems
- Each interferes with secure/available DNS
- They include:
  - Obsolete client OSen (no DNSSEC) (Old resolver libs)
  - Obsolete, broken, cheap consumer boxen (UDP, 512B, 103[345] only)
  - Hotels, ISPs, hotspots monetizing DNS (rewriting answers)
  - Hostile networks in the middle (blocking DNSSEC and rewriting)
  - Pervasive surveillance, active attacks, passive attacks
  - Lack of wide deployment of DNSSEC validation

One fix? How?

- Turn the “last mile” into the “last meter” (at both ends!!!)
- Use mostly-workable service (http)
  - bypass broken boxes & hostile networks
  - enable host- or site-specific validating clients
- Privacy via https (optional but advisable)
- Sit below the application (DNS) and bridge it across
  - No DNS skillz required (mostly)
  - Opaque (forwarder model) or Transparent (for the recursives!!!)

Where Should This ID Live?

- It kind of looks like an application…
- But it helps DNS resolver operators…
- And may help DNS authority operators…
- And definitely helps DNSSEC get deployed…
- And Snowden-esque (last mile protection)
- DNSOP? Maybe, or maybe just partly…

Elements of the Solution

1. DNS descriptor “language” for helping build compatible parsers
2. JSON encoding scheme mostly derived from DNS RFCs
    - Uses DNS descriptor language in reference implementation
3. Client/Server “rules of the road” for compatibility
    - Interoperable implementations is the goal

No Bike-Shedding!

- Based on development of actual running code - no need to argue design
  - “Bike shed factory” - build your own bike shed, any size, any colo(u)r
- However, original code not available (yet/now/ever?)
  - To be reimplemented using only these drafts
  - Hard work already done (dns descriptor language)
- One-pass transliteration on server, 1-pass + swap on client
  - Negligible latency on modest server (Mac Mini +~1ms RTT)

Finer Points

- JSON scheme:
  - bidirectional (DNS->JSON->DNS)
  - 100% fidelity (bit for bit)
- HTTP using POST - no URI parsing
- DNS-implementation neutral (by design)
  - DNS hosts may not even be aware of bridges
- Topologically flexible - deploy between DNS client/server and/or as a forwarder
- No DNS fiddling, no user-serviceable parts
  - No DNS knowledge needed to deploy
  - Minimal DNS knowledge needed to implement

FIN

- Did anyone get a chance to read the drafts?
- Questions? Comments? Opinions?
- Is this work you want to see implemented & standardized
- Anyone interested in: Co-authoring? Developing? Reviewing?
- Where does it belong?
- Is any of it ready to adopt and/or merge with other drafts?

Thank you for your time and patience.

brian.peter.dickson@gmail.com