Recommendations for DNS Privacy Client Applications

IETF 104, Prague

Vittorio Bertola <vittorio.bertola@open-xchange.com>

What is this?

- Similar to draft-ietf-dprive-bcp-op, but for Do* client authors rather than for Do* server operators
- Different from (and complementary to) the two DoH operator drafts
  - Aims to document and address all concerns by all parties systematically
  - Not focusing only on centralization
  - Not dealing only with the issues for network operators
- Since most of the DoH concerns come not from the protocol but from the deployment plans, it makes sense to address them as recommended best practices for Do* deployment in applications

What is this not?

- Not meant to stop Do* deployment or advocate against it
  - In fact, it is meant to make deployment easier by promoting and documenting consensus on how to do it without controversy
  - No explicit recommendation to deploy Do*, but would not mind adding it
- Not meant to promote a single side of the story
  - What is in the -00 is of course just mine
  - But I tried to reflect the concerns accurately, and also reflect the advantages that Do* would indeed bring
  - The objective is to test whether it is possible to get to consensus on how to address at least some of the concern areas
  - Happy to incorporate more views and take co-authors

How is it structured?

- Covers both DoH and DoT (and indeed even issues that already existed with Do53 public resolvers)
- An attempt to turn the DoH concerns into an organized and constructive discussion:
  1. A bit of terminology
  2. An organized breakdown of the discussion into issues, though not completely possible due to interdependencies
  3. A write-up of each issue (as balanced as possible)
  4. Recommendations on policies and deployment choices that could alleviate each issue while being broadly acceptable

Terminology

- «Local» vs «remote» resolver
- «Network-level» vs «application-level» name resolution architecture

Network-level	Application-level
All applications use the same resolver (the operating system one)	Each application uses its own resolver
The default is usually the resolver automatically suggested by the network	The default is usually supplied by the application (no local resolver discovery)
The user is in charge, either accepting the default or changing it in a single place	The application is at least partly in charge, choosing the default and/or constraining the choice to its own «trusted resolvers»

Issues in the current version

1. Trust model and user choice
2. Consolidation
3. Namespace fragmentation
4. Privacy
5. Content access control
6. Security and network management
7. Jurisdiction
8. Disaster recovery
9. User support

Example issue: Namespace fragmentation

- If each application uses a different resolver, different replies to the same queries can appear (user confusion, security, support issues)
- This can be minimized if applications only use a different resolver (than the operating system) when the user really wants it
- However, popular applications could direct users to resolvers that they control, and use this to assert de-facto control of the namespace
- So applications SHOULD NOT adopt alternate roots (unless for specific non-mainstream use cases)
- And they SHOULD follow community standards/governance policies

Questions to start with

- Is this useful? Would people with different viewpoints want to try to reach consensus?
- Is this in the mandate of the IETF (fully, partially, totally not)?
- Is this in DPRIVE's charter? If so, would the group want to work on it?
- Otherwise, where can we go?
- How do we involve non-technical stakeholders?