DNS over TLS: Three ways of not using port 53

Paul Hoffman

Assumptions

- Goal is privacy for DNS requests and responses from stubs to recursives
- TLS works well for client-server interactions, and is well understood
- Middleboxes exist between stubs and recursives, and some of them do a poor job of transmitting valid data
- TCP is fine

Assumed middlebox brokenness

- Acting as DNS forwarders, but poorly
- Acting as DNS resolvers, but poorly
- Blocking ports they don’t know about
- Blocking TLS traffic that they don’t like
- Other things that we don’t like and can’t predict

If you think that TLS over port 53 is fine

- See draft-hzhwm-dprive-start-tls-for-dns
- Uses a STARTTLS-style mechanism
- However, it is probably susceptible to stupid and/or malicious middleboxes

Three ways to use TLS but not port 53

- Plain DNS-over-TCP, over port 443: draft-hoffman-dprive-dns-tls-alpn
- Barely wrap DNS queries and responses in HTTP: draft-hoffman-dprive-dns-tls-https
- Use a port that is not 443: draft-hoffman-dprive-dns-tls-newport

Plain DNS-over-TCP

- ALPN lets the TLS negotiation say what the protocol that will run after TLS is set up will be
- Downsides
  - Not all TLS stacks support ALPN
  - An aggressive middlebox can see the APNL and stop the TLS negotiation

Barely wrap DNS queries and responses in HTTP

- Take the octets from the DNS request and make them into a URI
- Example: https://8.8.8.8/.well-known/dns-in-https/TN4AAAABAAAAAAAAB2V4YW1wbGUDY29tAAABAAE=
- Response is an unmodified binary blob
- Downsides
  - Many will consider this a misuse of HTTP, but it is allowed by RFC 3205

Use a port that is not 443

- Port is TBD
- Downsides
  - – Middleboxes that block ports they don’t know

What I like, at least for today

- Barely-wrap will get through anything that lets 443 through currently
- A new port may be OK because the client can tell immediately if the port is unavailable and fall back to unprotected DNS
- ALPN is nice, but it is not likely to be widely-enough supported