Specifying That a Server Supports TLS: a HASTLS Resource Record for the DNS

Paul Hoffman

IETF 80, Prague

v.01

Overview of this presentation

- Perceived need
- Current draft
- Architectural issues raised
- Note: the assumption for this short presentation is that you have at least skimmed the draft or a recent version of it
  - I only have 15 minutes here

Problem statement

- Some apps (not web browsers) allow the client to say, in essence, “try to connect with TLS, but if that doesn’t work for some reason, try to connect without it”
- That app currently cannot tell whether or not the intended server actually offers the non-SSL version of the protocol
- This lets a man-in-the-middle (MITM) who can force a TLS session not to be set up to get the client to still communicate; there are many reasons why this is bad

Current draft

- draft-hoffman-server-has-tls-04
  - Will probably become an appsawg draft unless the following slides scare people away
- Gives the problem statement, describes the different types of clients and servers based on what their TLS-using policy is, proposes a concise solution in DNS, explains how to implement it based on the desired TLS-using policy

Types of clients and servers

- Clients:
  - CIO: insecure only
  - CSO: secure only
  - CFB: starts secure but willing to fall back
- Servers:
  - SIO: insecure only (doesn't even offer TLS)
  - SSO: secure only (doesn't offer non-TLS)
  - SSB: serves both

Where knowing definitively what the server offers will help

- A CIO that starts an insecure communication with a server, or a CFB that falls back to insecure communication with a server, has no idea whether the site they wish to communicate with even hosts an insecure server
- If either of them knew for sure that the host didn’t offer an insecure service, they would not try on the non-secure port
- This is probably more realistic than “you should only use secure communication”

Proposal: HASTLS in the DNS

- Query: _appname._protoname.hostname IN HASTLS
- Response: ins-port sec-port pol-pref
- Policy preference is “0” for “the server admin doesn’t care” and “1” for “the server admin prefers you to be CSO”
- The response should be gotten with DNSSEC; otherwise, a MITM can fool you into thinking there is an insecure port available

What it looks like, what it means

- Example:
  - _http._tcp.www.example.com IN HASTLS
  - 80 443 0
- Lots of explanation of what different types of clients do when they see different responses

Architectural issue 1: Is this a service discovery protocol?

- The draft says “no”, but there is nothing stopping clients from (mis)using it as such
- Many people *want* a service discovery protocol for secure ports, and they want this proposal to be changed to be one
- Architectural question: is this for increasing security, or also for announcing that security is offered?

Architectural issue 2: Should this data be carried with other DNS information?

- The proposed HASTLS record only talks about TLS availability
- Maybe this should be coupled with other security information (such as DANE certs) or other information (A and AAAA) in a single DNS lookup
- Architectural question: do we want focused records that require more DNS queries or kitchen sink records that have complex semantics?

Architectural issue 3: Applications don't know if DNSSEC was used

- Currently, both of the DNS "last hops" (application to its host's resolver, stub resolvers to recursive resolvers) are not cryptographically protected
- Architectural question: should we propose extensions like this (and DANE) before an application can know that the information is authentic, or wait until that has become real?

Questions

- Note that not all questions can be answered by appsawg; some will involve input from the DNS community