TCP for DNS security considerations

Fernando Gont
project carried out on behalf of
UK CPNI

76th IETF meeting, November 8-13, 2009
Hiroshima, Japan

What are the issues?

- TCP may be needed for DNS as a fall-back transport protocol when the answer to the query is large.
- Relying on TCP for DNS is going from stateless to stateful
- This opens the door to a number of Denial of Service vulnerabilities that are typical for a connection-oriented protocol:
  - Connection-flooding attacks: Naptha, FIN-WAIT-2 flooding, etc.
  - Send buffer issues: Netkill, closed windows, etc
  - Receive buffer issues: holes in the data stream
  - Blind attacks: that depend on predictable ISNs, ephemeral ports, etc.
  - Overhead issues arising from connection establishment/teardown

Background needed to understand them

- “Security Assessment of the Transmission Control Protocol (TCP)”, published by UK CPNI, available at: http://www.cpni.gov.uk/Docs/tn-03-09-security-assessment-TCP.pdf
- An IETF I-D version of the document is currently a TCPM WG item: draft-ietf-tcpm-tcp-security (but you probably want to look at draft-gont-tcp-security instead)

What alternatives exist?

- Go with larger UDP responses – this might lead to fragmentation… and fragmentation brings a lot of other issues (see draft-ietf-opsec-ip-security)
- Use some other connection-oriented protocol – probably with the same issues as TCP… (and additional issues for NAT traversal).
- A DNS-specific transport protocol? (e.g., draft-barwood-dnsext-dns-transport)
- Use a tuned/hardened TCP – i.e., configure TCP settings so that the aforementioned issues are less of an issue (better if much of the resiliency machinery is "built-in").

Choices that need to be made

- If connection establishment/teardown is a concern, some might think (actually, *have* thought) about "persistent connections" (as in HTTP)
  - □ Do they really make sense??
- To minimize state at the DNS server, the client could be required to perform the active close. Otherwise (e.g., idle connection) the server **resets** (RST) it.
- How "liberal" we are with respect to the client behavior?
  - □ Number of concurrent connections
  - □ Idle-connections
  - □ Connections in, e.g., FIN-WAIT-2 state, TIME-WAIT state, etc.
  - □ Timing issues: RTO, closed windows, etc.

Possible way forward

- TCP for DNS tuning document? (in DNSEXT or DNSOP?)
- Update Section 4.2.2 (“TCP usage”) in RFC 1035 in a DNSEXT document?
  - draft-ietf-dnsext-dns-tcp-requirements currently has *some* text on the subject
- Others?