Authenticated DNS over TLS to Authoritative

Presentations to DPRIVE and DNSOP

Brian Dickson, GoDaddy
brian.peter.dickson@gmail.com

Problems To Solve

- Provide privacy protection for (some) queries resolver->authoritative
- Authoritative server’s identity must be authenticated
  - Need to secure the delegation itself to obtain the name server’s name
- MUST be deployable
  - Using current Registry mechanisms (EPP) as-is
  - Using only the DNS
  - Minimize new DNS protocol elements
- Explicit signaling of support
- Downgrade resistant

Other Drafts (and why)

- NSV: new DNSKEY algorithm (dnsop-ds-hack draft)
  - Protect delegation NS records via DNSSEC (in DS record)
  - Necessary
- DNST new RRTYPE (dprive-dnst draft)
  - Explicit transport signaling and discovery
  - Necessary
- Glueless guidance (dnsop-glueless draft)
  - How to avoid almost all A/AAAA glue
  - Optional but highly recommended
  - Avoids requirement for extra records to protect glue integrity
    - E.g. compare to DSGLUE which includes A/AAAA glue
    - Glue owner names differ from NS owner names
    - Gets very messy

TLS for ADoT (TLSADOT)

TLS needs the **IP address** of the server plus a **domain name** to use:

- The resolver connects to the **IP address** (glue data or glueless)
- The server name (validated by **NSV**) is sent via SNI
- The server presents a matching TLS **certificate**
- The TLS certificate must be **validated** *somehow:* WebPKI, or TLSA, or both
  - **TLSA** is a DNS standard (requires DNSSEC), can use any Cert type
    - For public DNS, the ICANN Root Trust Anchor is used
  - As long as the **TLSADOT** validates, the WebPKI validation can be skipped
- Question:
  - New RRTYPE (TLSADOT)?
  - Or just use un-prefixed TLSA at name server name?

NSV for Protecting Delegation NS

Solution: use a new DNSKEY algorithm “NSV” to create a special DS record.
The DNSKEY ”signature” for NSV is the RDATA from a single NS record.
However, no DNSKEY records are published in the child zone, which might not be signed.
NSV exists ONLY so DS records can be created; since DS records are signed, NS records are now able to be validated.

DS = hash(DNSKEY) = hash (NS RDATA) = hash (name server name).

Validate using the parent NS RDATA (name server name).
Validation of an NS record succeeds IFF a matching DS is found.

Unknown DNSKEY algorithms are treated as “insecure” rather than ”bogus”, so NSV-unaware resolvers will not have problems with this proposed addition.

DNST for Signaling ADoT Support

Supported transport protocols for each target (server) name are signaled.
DNST consists of flags for transports supported: **UDP TCP DOT**
DNST has the same owner name as ~~the~~ ***a*** name server's name.

For ADoT to be supported, all of the following are required:

- The name server's name must be in a DNSSEC signed zone
- A DNST record must be present
- The DNST record must have the DOT flag bit set

Other transports are also explicitly signaled by corresponding flags.
This could support servers that ONLY support DOT as a transport method (i.e. no UDP or TCP), such as for "private" zones (e.g. "enterprise" use cases.)

Downgrade Resistance

- NS records are protected with NSV-algorithm DS hashes
  - Only validated NS target server names will be used
- DNST flags ADoT support and is signed.
- TLSADOT records provide TLS certificate parameters and are signed.
  - Only the actual DNS server will be able to establish a TLS session.
  - No third-party certificates (issued by other CA providers, for example) will validate.
- An on-path adversary can interfere with the connection establishment but cannot impersonate the server.
  - The client may choose to fall back to UDP but cannot be tricked into doing so.

Bonus Section: Optional Elements

Adding these reduces round trips when they are both actually used, and the cache is “warm”.

Optional element 1: a new wildcard label.

- Matches only if both QTYPE and QNAME match.
- This could be polled for if NODATA answer is returned
- No change to the authority server’s logic

Eventually augment/replace ‘*’ wildcard label?

- Insert immediately before the ‘*’ section of the existing standard(s).

Optional element 2: OPT RR for soliciting NSEC records for *positive* answers

- Map of RRTYPEs present at the owner name

DNSSEC/DANE requirements

- Servers MUST do DNSSEC and TLSA for name server domains.
- Clients MUST be able to validate using DANE.
- This is the minimum requirement for interoperability, and necessary for the downgrade resistance aspects.
- Clients (resolvers) are free to choose not to do DNSSEC validation, and to do WebPKI validation, if they wish.

ONLY name server zones are required to be signed.
ADoT provides privacy for unsigned domains using ADOT name servers.
(DNSSEC is optional but strongly encouraged.)

Work in Progress

These drafts are pretty new and have not had a ton of review.

Feedback and suggestions are definitely welcome, as is work on initial development and testing. One goal was to make this as easy to implement and deploy as possible.

Questions?

Author:
Brian Dickson
GoDaddy
brian.peter.dickson@gmail.com