Authenticated DNS over TLS to Authoritative

Presentations to DPRIVE and DNSOP

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

Goals for DNS over TLS to Authority

- Provide channel security between Resolver and Authoritative Server
  - Use TLS protocol for channel security
  - No restrictions/requirements on TLS certificate types
- Maximize interoperability
- MUST be deployable using current Registry mechanisms (EPP) as-is
  - Do not require non-DNS elements
  - Limit new DNS protocol elements to those strictly required
- Signal ADoT support and permit secure discovery
- Validate server identity (“authenticated” ADoT)
- Protect against downgrade attacks
- Resolver vs Authoritative Server roles:
  - Requirements apply strictly to authoritative servers
  - Recommendations apply to resolvers (strongly encouraged, however)

Non-Goals for ADoT

- Does NOT provide data integrity
  - Use DNSSEC for that
- Does NOT require use of WebPKI
  - WebPKI Certs MAY be used
  - Arbitrary CA Certs MAY be used
  - Self-signed Certs MAY be used
  - TLSA records MUST be used
  - Clients MAY ignore WebPKI elements of validation
- Does NOT require single/unique server identity
  - A server can go by multiple names (and use respective certs)
- Does NOT require Registry-side changes
- Does NOT require Registrar-side changes
  - (NSV might require Registrars to add new DNSKEY algorithm for DS)

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
    - Glue owner names differ from NS owner names
    - Gets very messy

Before TLS: Delegation NS Records and Glue

Delegation example:

```
example.com NS ns1.example.net
```

- Delegates a portion of namespace to another name server
- NS record is non-authoritative, and CANNOT be DNSSEC signed
- Target name (`ns1.example.net`) requires lookup of IP address (IPv4 and/or IPv6) or Glue
- Resolver contacts new name server *by IP address only*
- Target (server) name is NOT part of DNS on-the-wire protocol
- *Glue* is non-authoritative IP address information provided by the parent server
  - MAY be needed to prevent circular dependency
  - If NOT present, additional DNS lookup(s) are necessary
  - Non-Glue addresses CAN be DNSSEC signed
  - Glue addresses CANNOT be DNSSEC signed

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**
- The resolver must determine the server's name *somehow*
  - The NS record is non-authoritative, and CANNOT be DNSSEC signed
  - Some way of protecting the target (server) name from the NS is required
  - (**NSV** does this, we take that as read for now)
- The server name is sent via SNI, and the server presents a 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
- ADoT only sends DNS messages *after* TLS is established

NSV for Protecting Delegation NS

Some way of protecting the target (server) name from the NS is required.
NSV (draft dnsop-ds-hack) provides a mechanism for this.

Summary: DS records are hashes of DNSKEY records in the child zone.
NSV is a “fake” DNSKEY algorithm. Each NSV DS record encodes a single NS target name and is fed as input to the DS hash algorithm. The child zone does not have a corresponding NSV DNSKEY algorithm and MAY not even be a signed zone(!!).

An NSV-aware resolver performs a corresponding validation using the parent NS target names. Validation succeeds if a matching DS is found. Only validated NS records are used for ADoT. Resolvers MAY also enforce this for non-ADoT.

Unknown algorithms are treated as “insecure”, 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 need to be signaled.
DNST (draft dprive-dnst) provides such a signaling mechanism.
This RRTYPE has the same owner name as the 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. While the specific use cases have not been defined, this would support servers that ONLY support DOT as a transport method (i.e. no UDP or TCP).

Downgrade Resistance

The following elements provide resolvers the ability to detect on-path attempts to downgrade from TLS to plain DNS:

- NS records are protected with NSV-algorithm DS records
  - Only validated NS target server names will be used
- A and AAAA records are signed (courtesy of “glueless”)
- DNST flags signal ADoT support and are 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.

Glueless Name Server Names

The NSV mechanism to protect otherwise unsigned NS records is necessary, but involves the addition of a new RRTYPE and logic to handle it. Adding a similar protection for glue A and/or AAAA records is not technically necessary if there are no such glue records

Glueless (draft dnsop-glueless) provides information on ways to accomplish this.

There are no protocol changes or new RRTYPEs involved. Glueless involves one or two domains dedicated to name server names exclusively, which are DNSSEC signed. The A and AAAA records are located there, and being signed, are secure.

NB: This zone would be where DNST and TLSADOT records would be located as well, so the zone's existence is not actually optional for ADoT usage.

Bonus Section: Optional Elements

The ADoT draft includes some ideas for discussion and consideration, which have the potential to reduce the overhead of DNS queries needed for ADoT in a “warm” cache. Adding them reduces round trips when they are both actually used.

Optional element 1: a new wildcard label.
The concept is a wildcard which requires that the type matches the QTYPE.
This could be polled (no change to the authority server’s logic), or it could be returned in a manner similar to the ‘*’ wildcard label (which matches regardless of QTYPE). If the latter, this logic would be added immediately before the ‘*’ section of the existing standard(s).

Optional element 2: OPT RR for soliciting NSEC records for non-WC positive answers (thus providing the set of RRTYPEs present at the owner name).

DNSSEC/DANE interoperability 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.

NB: This all works for unsigned domains using ADOT name servers. There is NO requirement for any zones OTHER than the name server zones to be signed. (DNSSEC is strongly encouraged, but not strictly necessary for ADoT).

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