DSGLUE

DPRIVE, IETF 112, November 2021
Ben Schwartz
Slides v01

Background: Authenticated ADoT

- “Authenticated” -> An active network adversary cannot ever gain access to the DNS query or response.
- Authenticated ADoT (A2DoT) is possible without any modification to parents … if resolvers are very patient:
  - Always do NS revalidation before using a nameserver.
  - Also send a SVCB query for _dns.$NSNAME (in parallel).
  - Use DNSSEC to authenticate the answers (requires signed child).
- This slows down resolution of all domains, not just those that use ADoT. Not likely to be deployed at scale (?)

Background: ADoT Parent Signals

- A signal in the parent domain is purely a performance optimization
- Most resolvers are impatient, so enabling optimized performance may be a prerequisite for wide deployment.

Background: Design Space

- Can we slow down resolution of existing domains?
- Do we care about A2DoT under non-A2DoT parents?
  - i.e. protecting label N+1 after label N has leaked
  - Can we require that non-A2DoT parents are signed?
- Can we add new RR types to the glue?
- Can we add new digest types to the DS record?
- Do we care about the latency of A2DoT-enabled domains?
- Can the child atomically update NS/DS/glue RRSets together in the parent?

Design assumptions for DSGLUE

- Can we slow down resolution of existing domains? **NO**
- Do we care about A2DoT under non-A2DoT parents? **YES**
  - i.e. protecting label N+1 after label N has leaked
  - Can we require that non-A2DoT parents are signed? **YES**
- Can we add new RR types to the glue? **NO**
- Can we add new digest types to the DS record? **YES**
- Do we care about the latency of A2DoT-enabled domains? **YES**
- Can the child atomically update NS/DS/glue RRSets together in the parent? **NO**

DSGLUE Structure

- A DSGLUE record is a DS record with
    - Algorithm: DSGLUE (TBD1)
    - Digest type: VERBATIM (TBD2)
        - See draft-vandijk-dnsop-ds-digest-verbatim
    - Contents: One arbitrary RRSet in a compact TLV encoding
        - Name must be below the zone cut, so only the prefix is encoded.
- Nonexistence is indicated by encoding an empty RRSet

Interpretation

- DSGLUE records are DS records.
    - Covered by the usual parent RRSIG, Bogus if tampered or removed
- RRSets in DSGLUE are glue.
    - Only for delegation-following, not authoritative for the child zone.
    - DSGLUE can repeat ordinary glue to secure it.
        - If they disagree, DSGLUE overrides the unsigned glue RRSet.
    - Like glue, each DSGLUE record’s RRSet SHOULD actually exist.
- Any RR Type is expressible, but not all are allowed (yet).
    - We can add RR Types as we figure out what they mean.

Simple Example

```
;; Child zone
$ORIGIN child.example.
secret  AAAA 2001:db8::1
@       NS ns
ns      AAAA 2001:db8::2
@       DNSKEY ...
@       CDS <real DS>
```

```
;; Parent zone
$ORIGIN example.
child    NS ns.child
ns.child AAAA 2001:db8::2
child    DS <real DS>
```

Both zones are fully signed. RRSIGs and TTLs are omitted for brevity.

Simple Example with slow A2DoT

```
;; Child zone
$ORIGIN child.example.
secret  AAAA 2001:db8::1
@       NS ns
ns      AAAA 2001:db8::2
_dns.ns SVCB 1 ns alpn=dot
@       DNSKEY ...
@       CDS <real DS>
```

```
;; Parent zone
$ORIGIN example.
child    NS ns.child
ns.child AAAA 2001:db8::2
child    DS <real DS>
```

Simple Example with DSGLUE

```
;; Child zone
$ORIGIN child.example.
secret  AAAA 2001:db8::1
@       NS ns
ns      AAAA 2001:db8::2
_dns.ns SVCB 1 ns alpn=dot
@       DNSKEY ...
@       CDS <real DS>
        CDS $DSGLUE(., NS,
[ns.child.example.])
        CDS $DSGLUE(_dns.ns.,
SVCB, [1 ns.child.example.
alpn=dot])
```

```
;; Parent zone
$ORIGIN example.
child    NS ns.child
ns.child AAAA 2001:db8::2
child    DS <real DS>
         DS $DSGLUE(., NS,
[ns.child.example.])
         DS $DSGLUE(_dns.ns.,
SVCB, [1 ns.child.example.
alpn=dot])
```

Closing thoughts

- DSGLUE shows that A2DoT is achievable even under very challenging assumptions.
    - No slowdown for non-participating zones
    - Minimum additional latency for participating zones
    - The child zone doesn’t have to be signed.
- This is a very large design space with a lot of options to consider.
- We are not blocked. We can start testing slow A2DoT while we figure out what we want from parent signals.