Signaling Authoritative DNS Encryption

draft-rescorla-dprive-adox

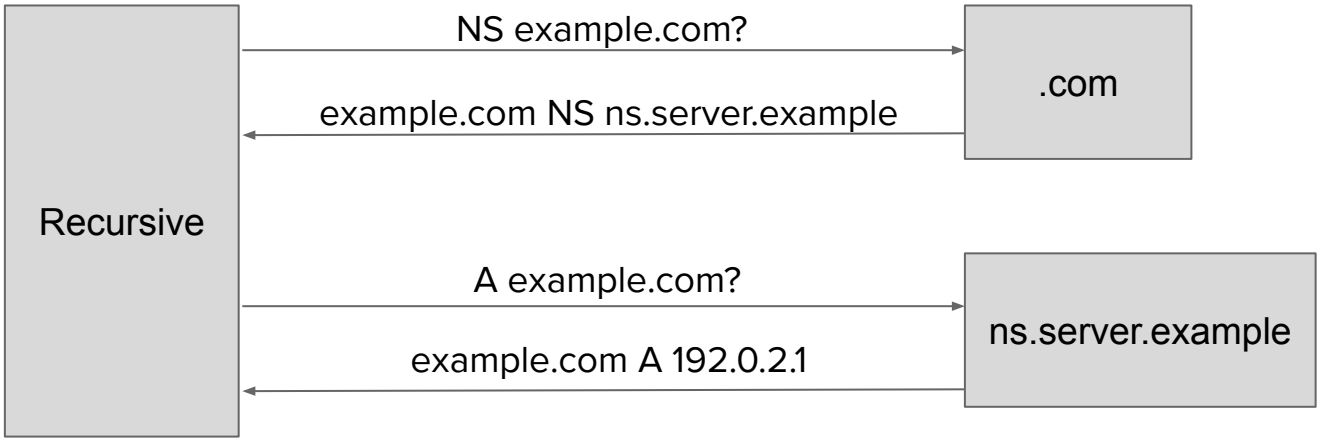
Tommy Pauly, Eric Rescorla,
David Schinazi, Chris Wood

Threat Model

- Active attacker on-path between recursive and **all** authoritatives
  - What RFC 3552 calls the "Internet Threat Model"
- Passive attacker on-path between recursive and **all** authoritatives
- Passive attacker on-path between recursive and **some** authoritatives
  - Worth considering but not here

What needs to be encrypted?

- Generally need to encrypt to both the parent and the child



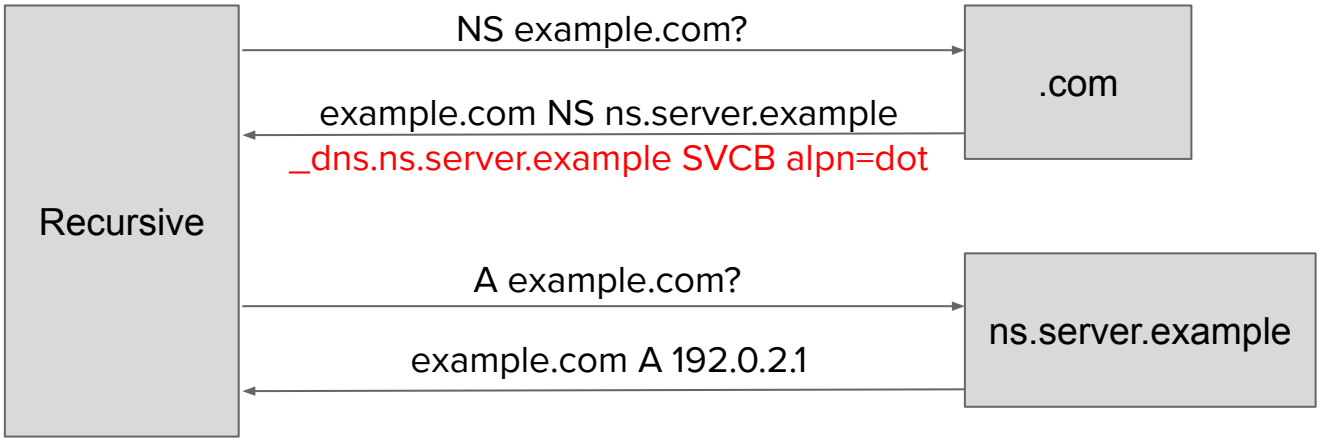
- Both of these queries reveal the target domain
  - Exception: queries for X.example.com where X is part of a large anonymity set (i.e., not www, web, etc.)

Basic Idea: Use SVCB

- The authoritative has a SVCB record
  - This indicates (1) that it supports encryption (2) what protocol (DoT, DoH, DoQ, it supports)
  - Served by the parent in additional data
- The recursive connects to the indicated server with TLS/QUIC
  - Authenticates in the usual fashion (WebPKI, DANE, etc.)
- Hard fail when you can't negotiate TLS or can't authenticate
  - i.e., don't fall back to Do53

Example

- Generally need to encrypt to both the parent and the child



What if you can't use additional data glue?

- The recursive can connect to the authoritative and ask for SVCB
- This will be over Do53
  - Or at best opportunistically
  - Because you don't yet have SVCB
- Only secure if the authoritative zone is DNSSEC-signed
  - Because TLS is providing integrity from the parent
- Special case: SVCB not currently permitted at the root zone
  - Fix: pre-configure recursives with the TLS status of the TLD authoritatives
- Side note: you're also going to want SVCB for ECH

Aren't you trusting the parent?

- Yes. You need that to get the NS record for the child
- What about DNSSEC?
  - NS records in the parent zone are unsigned
  - By the time you have connected to the (bogus) NS server and found out the NS records are bad, it's too late
- You should still validate NS and SVCB when zone is signed
  - This allows for detecting attacks retrospectively

How do you authenticate the resolver?

- The usual way
- You have the NS record from the parent and hence the name
- Choices
  - WebPKI
  - DANE (you'll want the TLS extension)
- Warning: potential disagreement between recursive and authoritative on supported methods
  - How do we distinguish between mismatch and attack?
  - Need some way for the authoritative to indicate what kind of credentials it has
    - Add a new SvcKey to SVCB

What if there are no common auth methods?

- OK to proceed with unauthenticated TLS
  - This may provide some defense against passive attack
  - This allows for incremental adoption of new auth methods
- Also useful for retrieving SVCB and NS

How does security work?

- Connection is secure if…
  - TLS certificate checks out **AND**
  - … NS name checks out (referred over TLS or NS signed by DNSSEC) **AND**
  - … SVCB record is OK (sent over TLS or signed by DNSSEC)
- Referrals sent over TLS allow recursive security
  - If referral/SVCB are secured by TLS…
  - … then child records are delivered securely if child TLS certificate valid
- Security propagates recursively… TLS all the way down
  - TLS trust anchors can be configured for TLDs, roots, etc…
  - … or bootstrapped from DNSSEC signatures when NS/SVCB are signed
- SVCB checks also protect against downgrade attacks!

Next steps?

- Pull in ideas from NS2?
- WG adoption?

Known Contentious Issues

- DANE vs. WebPKI
- DoT vs. DoH
- Draft position: why not both? SVCB is plenty flexible