DNS Zone Transfer-over-TLS (XoT)

draft-ietf-dprive-xfr-over-tls

Sara Dickinson
Willem Toorop
Shivan Sahib
Pallavi Aras
Allison Mankin
XoT - Background

Why XoT?

- Zone data can be collected via passive monitoring on-the-wire
- Zone owner may desire privacy for personal, organizational, or regulatory/policy reasons
- The main motivation for XoT is to prevent zone data collection during transfer

What is XoT?

- Encryption of DNS zone transfer (AXFR & IXFR) using TLS as a transport
- Draft adopted by DPRIVE in Nov 2019
Use cases

- **Confidentiality**: Encrypting zone transfers will defeat zone content leakage that can occur via passive surveillance

- **Authentication**: Use of single or mutual TLS authentication can complement TSIG/ACLs

- **Performance**:
  - Existing XFR implementation must be backwards compatible [RFC1034]/[RFC1035]
  - Current usage of TCP for IXFR is sub-optimal in some cases
  e.g. TCP connections are frequently closed after a single IXFR
IXFR : Existing mechanisms vs IXoT

**Existing**

- NOTIFY Request
- NOTIFY Response
- SOA Request
- SOA Response
- IXFR Request
- IXFR Response
- XOT-Based IXFR

**IXFR** Request

**IXFR** Response (Zone Data)

Retry over TCP if required.

**XOT-Based IXFR**

- NOTIFY Request
- NOTIFY Response
- SOA Request
- SOA Response
- IXFR Request
- IXFR Response
- IXFR Request 1
- IXFR Response 1
- IXFR Request 2
- IXFR Response 2

UDP or TCP

UDP or TCP

UDP (or part of TLS session)

TLS session

UDP
IXFR : Existing mechanisms vs IXoT

Existing

XOT-Based IXFR

<table>
<thead>
<tr>
<th>Secondary</th>
<th>Primary</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTIFY</td>
<td>UDP</td>
</tr>
<tr>
<td>NOTIFY Response</td>
<td></td>
</tr>
<tr>
<td>SOA Request</td>
<td>UDP or TCP</td>
</tr>
<tr>
<td>SOA Response</td>
<td></td>
</tr>
<tr>
<td>IXFR Request</td>
<td>UDP or TCP</td>
</tr>
<tr>
<td>IXFR Response (Zone Data)</td>
<td></td>
</tr>
<tr>
<td>IXFR Request</td>
<td></td>
</tr>
</tbody>
</table>
IXFR: Existing mechanisms vs IXoT

Existing IXFR:
- NOTIFY
- NOTIFY Response
- SOA Request
- SOA Response
- IXFR Request
- IXFR Response (Zone Data)

IXFR Request
- IXFR Request 1
- IXFR Response 1 (Zone Data)
- IXFR Request 2
- IXFR Response 2 (Zone Data)

Retry over TCP if required.

XOT-Based IXFR:
- NOTIFY
- NOTIFY Response
- SOA Request
- SOA Response
- IXFR Request
- IXFR Response (Zone Data)

UDP or TCP
- UDP
- UDP or TCP
- UDP

TLS session
- UDP (or part of TLS session)

Encrypted
-02 updates (July 2020)

- ALPN: Introduced use of ‘xot’ ALPN and term ‘XoT connection’- for *XFR + SOA only
  ○ RFC5936 states ‘Non-AXFR session traffic can also use an open connection.’
  ○ Currently no RFC for recursive to auth encryption (ADoT)....
  ○ Want to remove any assumption/dependency on ADoT solution or deployment
  ○ ALPN removes any requirement on the authoritative to (indirectly) support DoT
  ○ Server SHOULD REFUSE other queries (with extended error code ‘Not supported’)

- RFC7766 (TCP) - Tried to address issues around num of client/server connections
  ○ “…SHOULD be...one for regular queries, one for zone transfers for TCP…”
  ○ “… and one for each protocol that is being used on top of TCP…”
  ○ XoT draft updates this so all transports behave the same
-02 updates (July 2020)

- -02 minimally updates RFC1995 (IXFR) to clarify SHOULD do connection reuse (RFC7766)
- -02 discusses RFC5936 (AXFR) but does not currently update
- Both mechanisms are optimised specifically for XoT use case

- New (limited) discussion of padding
  - In -02 only the goals of padding and minimum requirements are discussed
    - Currently identified a need to receive ‘empty’ AXFRs to future proof padding
  - Traffic analysis and padding policies will be addresses in a separate draft
More recent questions/comments

- Review pointed to the need to revise the proposed updates to both RFC1995 (IXFR) & RFC5936 (AXFR)
  - Clarification of behaviour on a single connection when intermingling both IXFR and AXFR

- Review requested more discussion of limits on transfer rates or concurrent AXFRs
  - BIND has some controls for this already
  - Is more signalling from primary on transfer rate and concurrency limits useful?
    - Allows primary to throttle transfer rates when under heavy load
    - This could influence which primary is used and therefore allow load balancing
More recent questions/comments

- Better analysis of ‘non-Strict XoT’ use cases
  - Any need to allow fallback to TCP?
  - Handy on primary during testing/rollout (but allows downgrade, so block on secondary?)

- Clarify server cert config options:
  - e.g. one XoT cert (multiple SANs?) vs one per zone
  - Beyond server certs, mutual TLS is discussed as an additional option...

- Name compression limits packet size to ~16k because of the size of the compression pointer
  - For XoT is an option to disable this and have 64k packets beneficial?
Moving forward

- Spec is maturing - more reviews please!!
- Implementations - work starting on NSD patch, discussions with ISC on BIND support
- Future interop on this would be really beneficial
- Aware of a demand to deploy this
- Hopefully looking for WGLC in IETF 109 timeframe
Moving forward

- Spec is maturing - more reviews please!!
- Implementations - work starting on NSD patch, discussions with ISC on BIND support
- Future interop on this would be really beneficial
- Aware of a demand to deploy this
- Hopefully looking for WGLC in IETF 109 timeframe

Questions Please!
Additional Slides
XoT - Authentication mechanisms

<table>
<thead>
<tr>
<th>Method</th>
<th>Secondary</th>
<th>Primary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Auth</td>
<td>Channel Conf</td>
</tr>
<tr>
<td>TSIG</td>
<td></td>
<td><img src="#" alt="Green" /></td>
</tr>
<tr>
<td>Oppo</td>
<td></td>
<td><img src="#" alt="Green" /></td>
</tr>
<tr>
<td>Strict</td>
<td></td>
<td><img src="#" alt="Gray" /></td>
</tr>
<tr>
<td>Mutual</td>
<td></td>
<td><img src="#" alt="Gray" /></td>
</tr>
<tr>
<td>ACL on master</td>
<td></td>
<td><img src="#" alt="Green" /></td>
</tr>
</tbody>
</table>

**Conclusion:** Using TSIG, Strict TLS and an ACL on the primary provides all 3 properties for both parties with reasonable overhead.
Policy Management for XoT

- ‘Transfer Group’ - entire group of servers involved in transfers of a given zone (all primaries, all secondaries)

- The entire transfer group SHOULD have the same policy wrt (no weak point):
  - TSIG, TLS (O, S or m), IP ACL

- CHALLENGE: How to configure, enforce and test policy implementation?
  - Often involves different operators, different software, hidden servers
  - Feedback please 😊
Padding experiments

IXFR transfer sizes and rates are VERY context specific. Re-using connections for multiple zones hides patterns.

<table>
<thead>
<tr>
<th>Update rate</th>
<th>Zone size</th>
<th>DNSSEC</th>
<th>Update frequency</th>
<th>Order of Update size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Yes</td>
<td>Low</td>
<td>100s</td>
</tr>
<tr>
<td>Low</td>
<td>Very Large</td>
<td>No</td>
<td>High</td>
<td>1,000s</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Yes</td>
<td>High</td>
<td>10,000+</td>
</tr>
</tbody>
</table>

Jittered resigning
RRSIGs still significant
Simplest IXFR pattern (unsigned zone with regular updates)

- Unsigned zone with records added every 10 seconds
- Smallest XFR response packet possible would be 5 records:
  - 1 new record
  - 4 SOAs
- Order of few hundred bytes (~250 in this case)
- Packet size can indicate record changes but adding and changing are hard to distinguish (and name compression happens)
Multiple IXFRs for large DNSSEC NSEC3 signed zone (one update shown)

- Periodic resigning dominates
- Transfers every 5s, on a separate TCP connection
- Responses clustered around multiples of 3k bytes (1 SOA change) - note no condensation of changes
- Anomaly at 77s is caused by a single record update to the zone