DS and DNSKEY
low TTL experiments

Petr Špaček
2023-09-06
pspacek@isc.org
Outline

- TTL is wrong
- TTL does not behave
- Experimental setup
- Test data set
- Evaluation
- Real-world experience – by Viktor Dukhovni
Time To Live – always wrong

- Too short
  - Outage if servers are down
    - see *Cache Me If You Can: Effects of DNS Time-to-Live*

- Too long
  - Long outages with bad data
  - *Slack.com, .NZ, …*
Time To Live – full of surprises

- Supposedly an upper bound
- But …
  RFC 8767: Serving Stale Data to Improve Resiliency
- See also
  Measuring TTL Violation of DNS Resolvers at scale
- See also
  The role of DNS in residential Internet use
RIPE NCC lowered TTL
- NS 2 days ⇒ 1 day
- DS 1 day ⇒ 1 hour

- expecting at least 2x times query load increase …?
- "There was NO increase in query rates at RIPE NCC’s servers"
  - see RIPE NCC's presentation
Time To Live – full of surprises #3

- CZ TLD lowered TTL: 5 h ⇒ 1 h
- expecting 5x times query load increase …?
- see article

before - 250 k QPS

after - 250 k QPS!
Lab Experiment
Lab Experiment

- **DS & DNSKEY** TTL cap = 5 minutes
  - other RR types unaffected
- Enforce cap inside resolver side
  - but unlimited cache size
- Evaluate
  - answer latency
  - queries to auths
BIND v9.18.15 hack – TTL cap

diff --git a/lib/dns/resolver.c b/lib/dns/resolver.c
index b09813d444..e45d062369 100644
--- a/lib/dns/resolver.c
+++ b/lib/dns/resolver.c
@@ -6255,6 +6255,12 @@ cache_name(fetchctx_t *fctx, dns_name_t *name, dns_message_t
               if ((rdataset->type == dns_rdatatype_ds ||
                    rdataset->type == dns_rdatatype_dnskey) &&
                   rdataset->ttl > 5 * 60) {
+               rdataset->ttl = 5 * 60;
+           }
if (rdataset->ttl > res->view->maxcachettl) {
    rdataset->ttl = res->view->maxcachettl;
}
Experimental setup

- query stream replay
- measure latency
- repeat 10 times
  - with and without cap
Test data set #1

- Real traffic capture
  - Anonymized
  - 1 real telco resolver, 1 hour of traffic
  - Telco in northern Europe, March 2023
  - Mix of landline & mobile clients
- DNSSEC in nordics* – signed domains
  - .NO ~ 60%
  - .SE ~ 55%
  - .NU ~ 51%
Test data set: query rate

- Short, large spikes
- Period ~ 11 minutes?!
Test data set: query distribution

- 20% clients (2.5% queries)
- 34% clients (39% queries)
- 38% clients (40% queries)
- 6% clients (49% queries)
Latency: whole test (1 h) – telco

1 % queries slower than 100 ms

> 94 % queries answered within 1 ms
Latency: whole test (1 h) – telco

- **min across 10 test runs**
- **max across 10 test runs**
- **average**

Graph showing response time in milliseconds against the slowest percentile.
Latency: 05:00 – 10:00 minutes
# Queries upstream

<table>
<thead>
<tr>
<th>Resolver</th>
<th>DS queries</th>
<th>DNSKEY queries</th>
<th>All queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIND v9.18.15</td>
<td>70 411</td>
<td>11 372</td>
<td>3 241 806</td>
</tr>
<tr>
<td>+ TTL cap 5 minutes</td>
<td>82 311 (+ 17 %)</td>
<td>23 653 (+ 108 %)</td>
<td>3 259 009 (+ 0.5 %)</td>
</tr>
</tbody>
</table>

Number of queries sent to upstream servers:

- **DS**
  - BIND v9.18.15
  - TTL cap 5 minutes

- **DNSKEY**
  - BIND v9.18.15
  - TTL cap 5 minutes
Test data set #2

- Real traffic capture
  - Anonymized
  - 1 PoP, Washington D.C. area, 1 hour of traffic
  - Public resolver
  - Composition of clients unknown

- US – signed domains
  - .COM ~ 4 %
  - .NET ~ 5 %
  - .GOV ~ 8 %
Test data set: query distribution
Latency: whole test (1 h) – public

Spot the difference!
+20 % DS queries – significant?

<table>
<thead>
<tr>
<th>TLD</th>
<th>Signed</th>
<th>DS TTL</th>
<th>current DS volume</th>
<th>projected DS volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>CZ</td>
<td>57 %</td>
<td>1 h</td>
<td>8 %</td>
<td>~ + 2 %</td>
</tr>
<tr>
<td>GOOG</td>
<td>7 %</td>
<td>1 h</td>
<td>30 %</td>
<td>~ + 6 %</td>
</tr>
<tr>
<td>NL</td>
<td>59 %</td>
<td>1 h</td>
<td>9 %</td>
<td>~ + 2 %</td>
</tr>
</tbody>
</table>
+108 % DNSKEY queries – significant?

<table>
<thead>
<tr>
<th>Operator</th>
<th>Signed</th>
<th>DNSKEY TTL</th>
<th>DNSKEY volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS1</td>
<td>0.1 %</td>
<td>1 h</td>
<td>0.3 %</td>
</tr>
<tr>
<td>GoDaddy</td>
<td>?</td>
<td>1 h ?</td>
<td>4 %</td>
</tr>
<tr>
<td>deSEC</td>
<td>59 %</td>
<td>1 h</td>
<td>8 %</td>
</tr>
</tbody>
</table>
Steady state: Suối tranh
What we provision for: Dray Nur
Lab Experiment Evaluation

- 5 minute DS TTL @ TLD
  - **No** impact on answer latency
  - **Minimal** impact on TLD steady-state
- 5 minute DNSKEY TTL @ apex
  - Surprisingly still no latency impact
- 0.5 % increase in auth query volume
Real-World Experience

data provided by

Viktor Dukhovni
dviktor@google.com
goog. TTL change

• DS TTL @ TLD
  • 1 h ⇒ 30 minutes ⇒ 15 minutes
  • Already deployed!

• DNSKEY TTL @ 2LD
  • 1h – unchanged
  • for domains on Google DNS servers
goog. TLD traffic before & after

- DS 1 h → 30 minutes
- accounting fix
- DS 30 → 15 minutes
**goog. TLD traffic before & after**

<table>
<thead>
<tr>
<th>TTL</th>
<th>DS volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 h</td>
<td>29 %</td>
</tr>
<tr>
<td>30 m</td>
<td>30 %</td>
</tr>
<tr>
<td>15 m</td>
<td>33 %</td>
</tr>
</tbody>
</table>
goog. 2LD traffic before & after

DS 1 h ⇒ 30 minutes

accounting fix

DS 30 ⇒ 15 minutes
goog. 2LD traffic before & after

<table>
<thead>
<tr>
<th>DS TTL</th>
<th>DNSKEY TTL</th>
<th>DNSKEY volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 h</td>
<td>1 h</td>
<td>3 %</td>
</tr>
<tr>
<td>30 m</td>
<td>1 h</td>
<td>3 %</td>
</tr>
<tr>
<td>15 m</td>
<td>1 h</td>
<td>4 %</td>
</tr>
</tbody>
</table>
Conclusions – for real

• 15 minute DS TTL @ TLD
  • **No** impact on answer latency
  • **Minimal** impact on TLD steady-state
  • **No** impact on 2LD
• Much faster recovery? Why not?
Thank you!

• Main website: https://www.isc.org
• Software downloads: https://www.isc.org/download or https://downloads.isc.org
• Presentations: https://www.isc.org/presentations
• Main GitLab: https://gitlab.isc.org