DNSSEC meets real world
Lessons learned from 7 years of DNSSEC in practice

Roland van Rijswijk-Deij
r.m.vanrijswijk@utwente.nl
About SURFnet

- National Research and Education Network (NREN)
- Founded in 1986, incorporated 1988
  - 2013: 25 year anniversary!
- >11000km dark-fibre network
- Shared ICT innovation centre
- >160 connected institutions
  ±1 million end users
A little bit about me

• M.Sc. in Computer Science from the UT

• Worked in security past 14 years

• Ph.D. student (part-time) in the DACS group

• Love scuba diving ;-)
DNSSEC recap in 1 slide

• Plain DNS does not allow you to check the authenticity or integrity of a message

• DNSSEC adds this using digital signatures

• DNSSEC has two perspectives:
  – Domain owners sign their zone and publish the signed zone on their authoritative name servers
  – Querying hosts validate the digital signatures they receive in answers, along a chain of trust
DNSSEC relies on EDNS0

• EDNS0
- is an extension to DNS that allows for additional flags and large(r) DNS answers over UDP
- is enabled by default in most modern DNS servers
DNS servers using EDNS0 almost always ask for DNSSEC data

• EDNS0 introduces the “DNSSEC OK” flag (DO)
  - if set in a query, indicates that the querying host wants to receive DNSSEC information if available
  - again, enabled by default on most modern DNS servers
So lots of people use DNSSEC!

• Even if you never specifically asked for DNSSEC, it is likely your recursive name servers (resolvers) are in the >60% of hosts that have it enabled

• EDNS0 & DNSSEC OK are enabled by default in:
  – BIND 9.x (DNSSEC OK on by default from 9.5 and up)
  – Unbound
  – Microsoft Windows Server 2008R2
  – Microsoft Windows Server 2012
  – that covers the vast majority of DNS servers on the planet
One of the options set in an EDNS0 query is the maximum UDP payload size.

- RFC 6891 defines this as: *the number of octets of the largest UDP payload that can be reassembled and delivered in the sender's network stack*.
- The default value for most servers still is 4096 bytes.
- ±90% of hosts we see use this value.
So what?

- Recapping: >60% of querying hosts use EDNS0 and ask for DNSSEC data, 90% of those hosts ask for answers as large as 4096 bytes by default.

- As an indication:
  
  \$ dig +dnssec +bufsize=4096 MX comcast.net

  ... 

  ;; MSG SIZE  rcvd: 3229

- That will get fragmented into 3 packets!
Why fragmentation is a problem...

1. Recursive caching name server → firewall → query, buffer size 4096 → Authoritative name server

2. Recursive caching name server → firewall → answer, fragment 1 → Authoritative name server
   answer, fragment 2

3. Recursive caching name server → firewall → ICMP fragment reassembly time-out → Authoritative name server
So why are fragments blocked?

- In the 1990s there was a host of fragment-related attacks (e.g. “ping-of-death”)

- Many vendors still have outdated KB-articles and HOWTO’s floating around

- Some security auditors force people to block fragments, or worse, to block TCP on port 53
  - Not based on proven security issues, but based on “gut feeling” (it used to be bad in the past so it must still be bad)
How big a problem is this?

• 9% of all internet hosts may have problems receiving fragmented UDP messages [1];

• 2% – 10% of all resolving name servers experience problems receiving fragmented DNS responses [2]


Solutions

• Operators of DNS resolvers must be aware of this issue —> training, KB articles, HOWTO’s

• But operators of signed zones also suffer!

• If resolvers cannot reach you, that is a problem

• Luckily, zone operators can also take action
Why did we research this?

• It was a cold night just before Christmas 2010... and I had a queue of colleagues complaining by my desk that I had broken “The Internet”

turned out it wasn’t me, but their ISP...
Resolver experiments (1) normal operation

Response time (ms.)

Time (ms.)

900
800
700
600
500
400
300
200
100
0

Windows Server 2012
Unbound
BIND

Response (>me (ms.))

785
687
388
381
281
150
109
83
105

0
100
200
300
400
500
600
700
800
900
Resolver experiments (2) blocking fragments

Response time (ms.) [0/5 altered Authoritative Name Servers]

- Windows Server 2012: Time x100+ (!!!)
- Unbound: Time x2
- BIND: Time x10 (!)

Mean time: \( \bar{x} = 17,787 \)

[24,195;12,167]
Resolver experiments (3)

limit response size on 1 auth. NS

Response time (ms.) [1/5 altered Authoritative Name Servers]

- **Max. = 16,162**
- Unbound: 638 ms., 117 ms.
- BIND: 2,126 ms., 1,118 ms.

UNIVERSITY OF TWENTE.
Resolver experiments (4)

*limit response size on 2 auth. NS*

**Response time (ms.) [2/5 altered Authoritative Name Servers]**

- **Windows Server 2012**
  - Time x10: 3.295 ms.
  - Time x1.5: 1.756 ms.
  - Time x2: 1.036 ms.

- **Unbound**
  - Time x10: 2.90 ms.
  - Time x1.5: 513 ms.
  - Time x2: 651 ms.

- **BIND**
  - Time x10: 1.756 ms.
  - Time x1.5: 651 ms.
  - Time x2: 1.408 ms.
## Experiments on live auth. NS

<table>
<thead>
<tr>
<th>Traffic (IPv4 + IPv6)</th>
<th>Normal Operations</th>
<th>Max. response size 1232 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragmented responses</td>
<td>28.9%</td>
<td>0.0%*</td>
</tr>
<tr>
<td>Fragment receiving resolvers</td>
<td>57.3%</td>
<td>0.0%*</td>
</tr>
<tr>
<td>Truncated UDP responses</td>
<td>0.8%</td>
<td>0.9%</td>
</tr>
<tr>
<td>ICMP FRTE messages</td>
<td>5649/h</td>
<td>&lt; 1/h*</td>
</tr>
<tr>
<td>ICMP FRTE sending resolvers</td>
<td>1.3%</td>
<td>0.0%*</td>
</tr>
<tr>
<td>Total retries</td>
<td>25.8%</td>
<td>25.5%</td>
</tr>
</tbody>
</table>

*Statistically significant difference between experiments
Rise in truncated answers

• Experiment:
  - Querying 995 zones in .com, .edu, .mil, .net and .nl
  - All zones are signed and have a www-node
  - Results:

<table>
<thead>
<tr>
<th>Max. response</th>
<th>A for www</th>
<th>AAAA for www</th>
<th>DNSKEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1472</td>
<td>1.8%</td>
<td>1.8%</td>
<td>8.1%</td>
</tr>
<tr>
<td>1232</td>
<td>2.9%</td>
<td>3.5%</td>
<td>40.0%</td>
</tr>
</tbody>
</table>

- 30% truncations were expected for a maximum response size of 1232 bytes [3]
RFC 6891 updates EDNS0 obsoleting RFC 2671

6.2.3. Requestor's Payload Size

... 

The requestor **SHOULD** place a value in this field *that it can actually receive*. For example, *if a requestor sits behind a firewall that will block fragmented IP packets, a requestor SHOULD NOT choose a value that will cause fragmentation*

... 

Choosing a very large value will guarantee fragmentation at the IP layer, and may prevent answers from being received due to loss of a single fragment or to misconfigured firewalls.
So why fragmentation?

- Fragmentation is part of the core Internet design
- But: not well understood, lots of folklore, implementation errors and other misery

Dilbert Comic © 1995 United Feature Syndicate / Scott Adams
And now for something completely different...

Let’s talk about attacks :-)

photo © Dutch Photo Press, Patrick van Katwijk
DNS(SEC) amplification

Amplification = \( \frac{\text{Response size}}{\text{Query size}} \)

e.g.: amplification of 100x means: send 10Mbit of attack traffic, victim receives 1Gbit!
DNSSEC and amplification

• Problem: DNSSEC responses are generally much bigger than classic DNS

• This is an important criticism of DNSSEC, e.g. notable critic Dan Bernstein:

  “DNSSEC is a remote-controlled double-barreled shotgun, the worst DDoS amplifier on the Internet”
Remember our comcast.net example?

$ tcpdump -n -v -i en0 host xxxx

... 11:00:19.411981 IP (... proto UDP (17), length 68)

... 11:00:19.430637 IP (... proto UDP (17), length 1500)
    xxxx.53 > yyyy.55023: 36075$ 3/6/29 comcast.net. MX ...

11:00:19.430640 IP (... length 1500)
    xxxx > yyyy: udp

11:00:19.430641 IP (... length 297)
    xxxx > yyyy: udp

Send: 68 bytes, recv: 3297 bytes, amp. ≈ 48.5x!
Ground truth!

• Intuitively, DNSSEC is an amplifier but how bad is it really?

• We measured ground truth for all signed domains in .com, .net, .org, .nl, .se and .uk [4]
  – 2.5 million DNSSEC-signed domains, compared to 2.5 million “regular” domains

Worst amplifier: ANY queries

The chart shows the percentage of domains against the amplification factor for different domain suffixes with and without DNSSEC combined. The theoretical maximum amplification of regular DNS is indicated by the peak. The amplification factor is binned at 0.1 intervals.
Problematic amplifier: DNSKEY

Amplification factor [bin=0.1]

Percentage of domains

theor. maximum amplification of regular DNS

com, net, org, uk, se, nl
Lots of solutions...

• Amplification is hard to tackle. Only real solution: no IP address spoofing (“BCP 38”)
  – Far away in practice; some 30% of networks still allow spoofing

• Lots of other partial solutions, none of which provides a 100% solution

• Amplification & fragmentation are related!

  less amplification  ↔  less fragmentation
Root cause: conservative crypto

• The DNSSEC standard has only one mandatory signature algorithm: RSA

• 99.99%* of .com, .net and .org domains with DNSSEC use RSA keys

• Main cause of fragmentation and amplification
  – RSA 1024-bit: 128 bytes per signature, ±132 bytes in DNSKEY
  – RSA 2048-bit: 256 bytes per signature, ±260 bytes in DNSKEY

Why not use Elliptic Curve crypto?

- ECDSA standardised for DNSSEC (RFC 6605)
  - with two curves, P-256 and P-384
  - signatures and keys up to 4x smaller than w. current RSA keys

- But: ECC crypto has issues; most importantly: verification of signatures is (much) slower

- We are currently researching this
  - Preliminary results: fragmentation virtually disappears, amplification no longer an issue
Research matters!

• EDNS0 RFC and resolver software was changed because of fragmentation research (and roadshow)

• Paper quantifying amplification influenced major CDN in designing its DNSSEC deployment

• Deploying ECDSA without studying its impact would repeat old mistakes
  – Fool me once, shame on you, fool me twice...
Thank you for your attention!

Questions or remarks?