DNSSEC: what every sysadmin should know to keep things working

Roland van Rijswijk - Deij

roland.vanrijswijk@surfnet.nl

SURF NET

About SURFnet

- National Research and Education Network (NREN)
- Founded in 1986
- > 11000km dark-fibre network
- Shared ICT innovation centre
- > 160 connected institutions
  ± 1 million end users

DNSSEC: recap in 1 slide

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

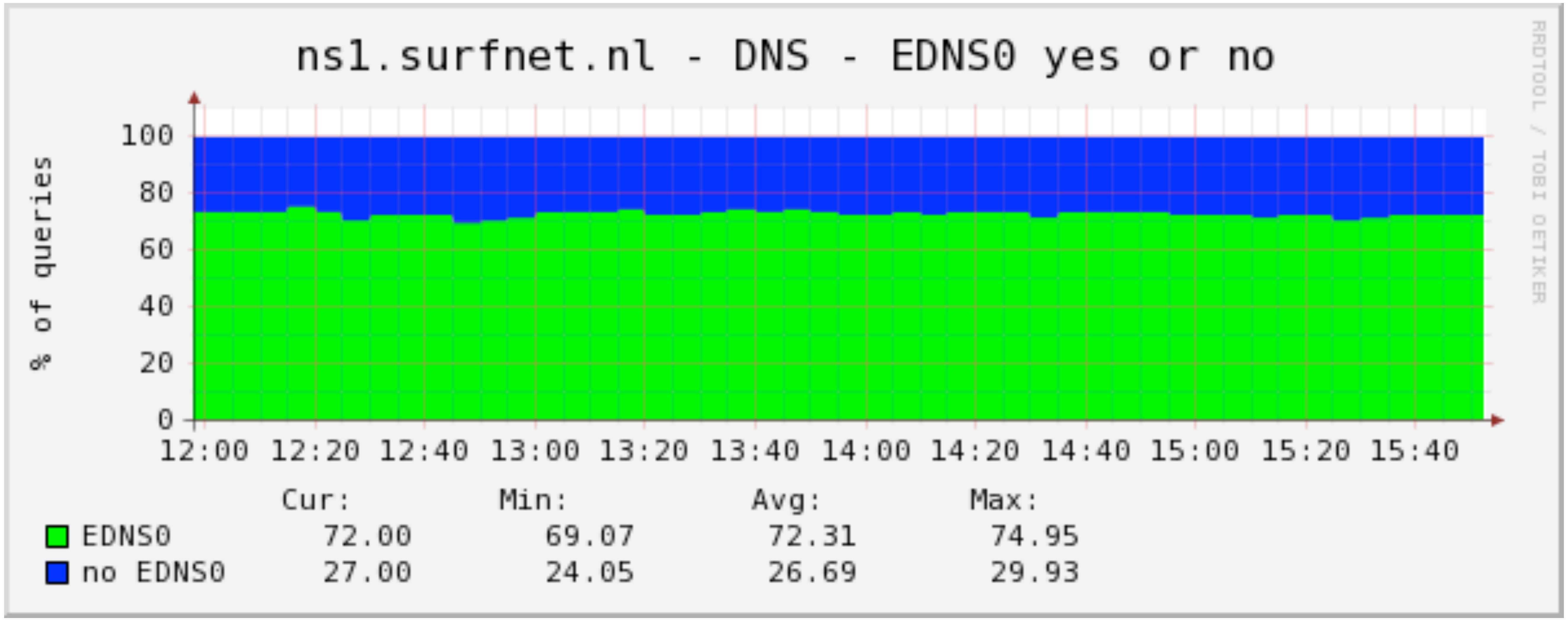
- DNSSEC adds this using digital signatures

- 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**

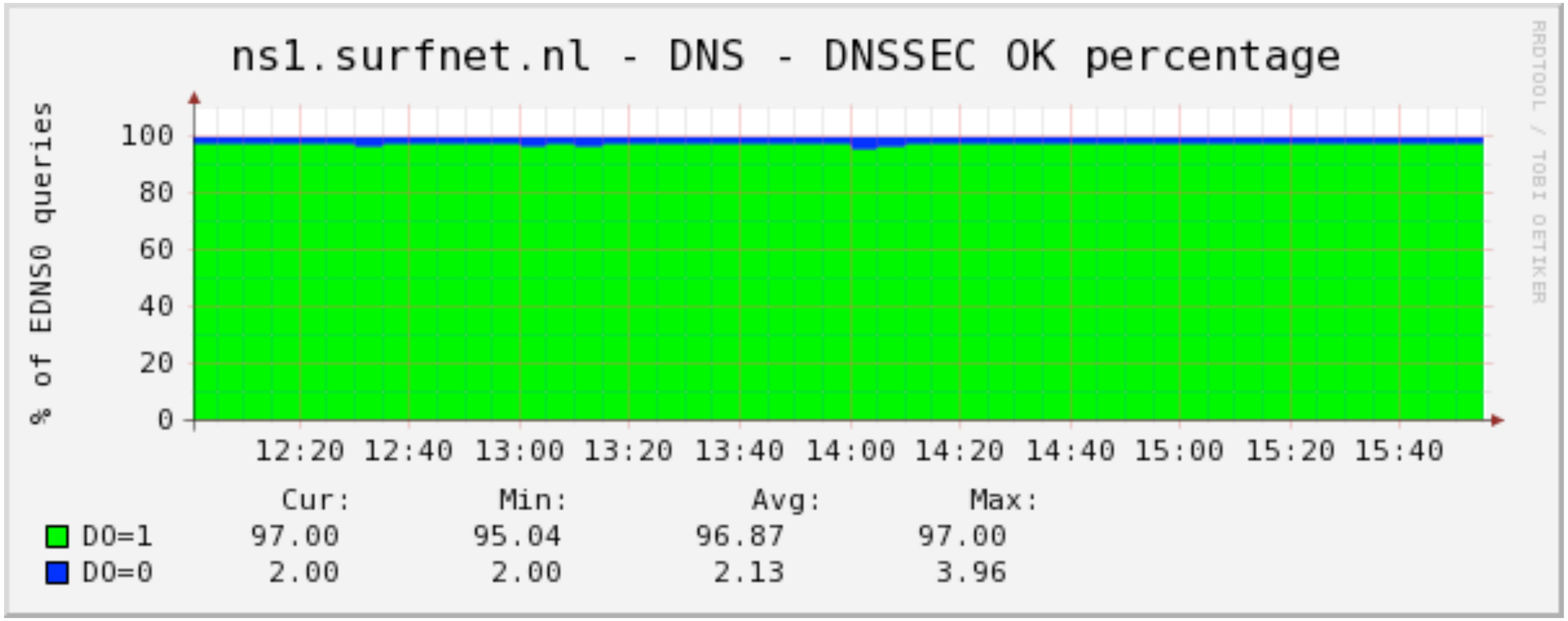
You are most likely using EDNS0

- EDNS0 (RFC 2671)
  - 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



And if you use EDNS0, you are probably asking for DNSSEC

- 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



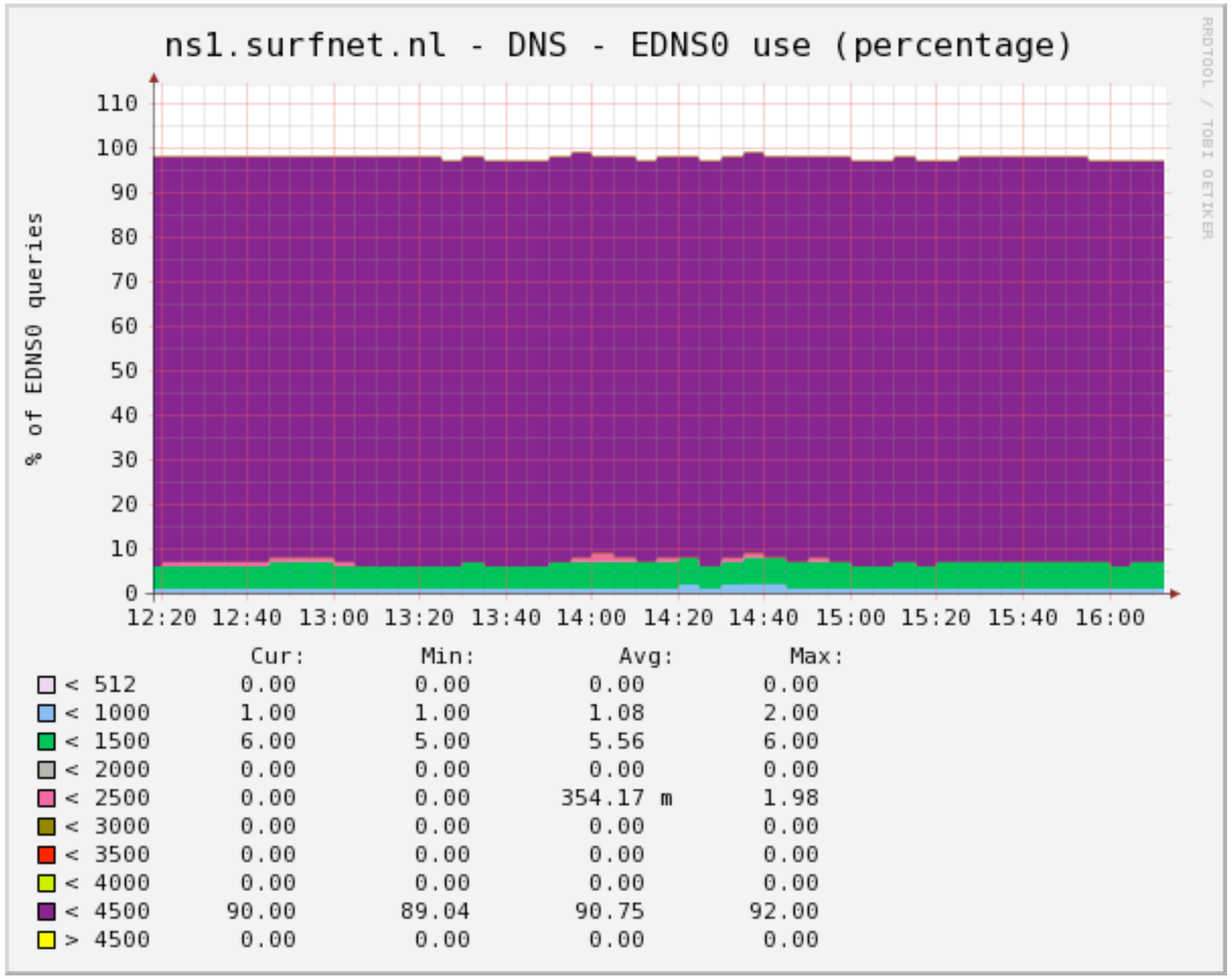
So it's likely you're using DNSSEC

- Even if you never specifically asked for DNSSEC, it is likely your recursive name servers (resolvers) are in the ±70% 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**

EDNS0 max. UDP payload size

- One of the options set in an EDNS0 query is the maximum UDP payload size
  - RFC 2671 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 is 4096 bytes
  - ±90% of hosts we see use the default value



So what?

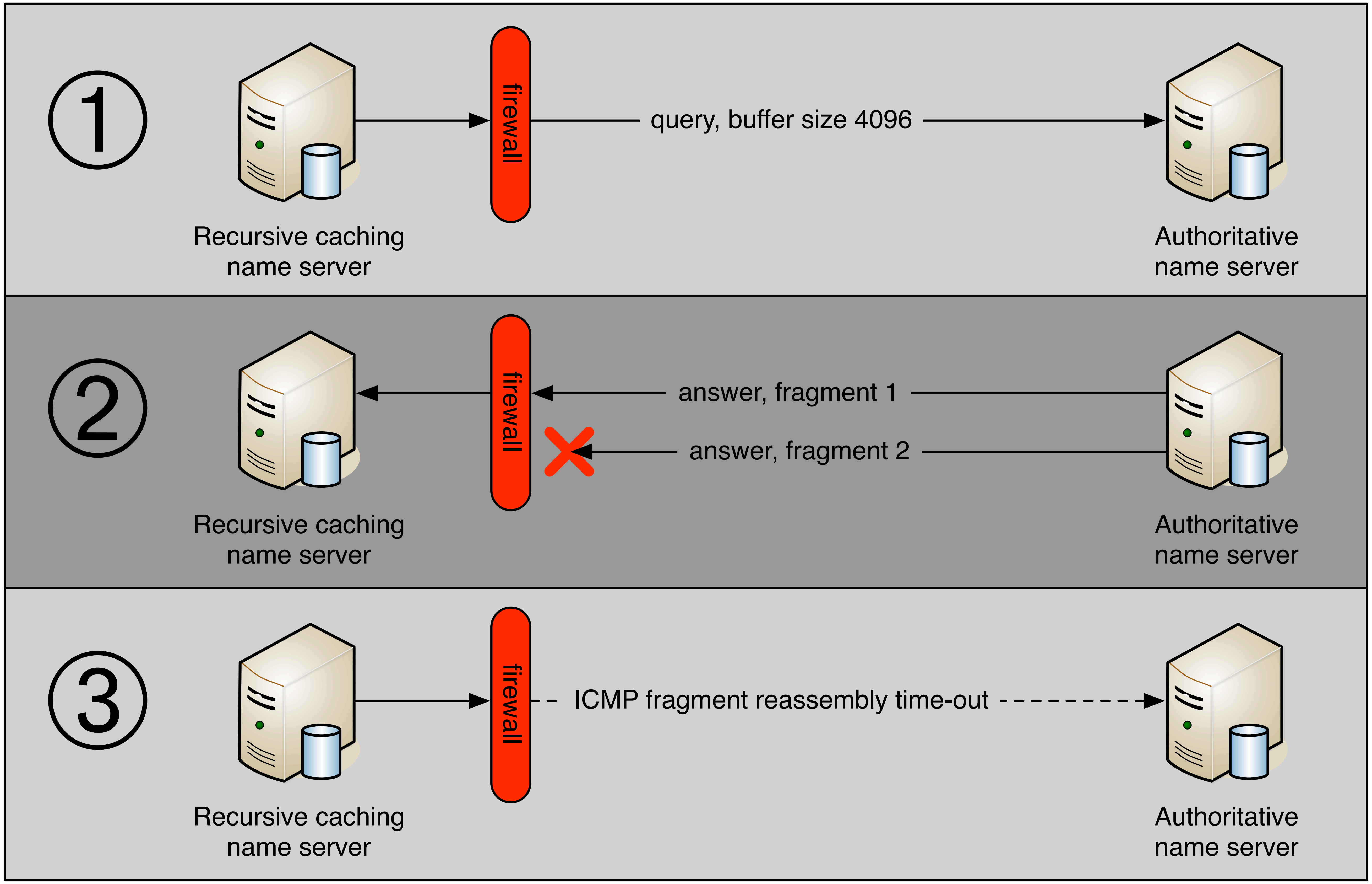
- Recapping: ±70% of querying hosts use EDNS0 and ask for DNSSEC data, 90% of those hosts ask for answers as large as 4096 bytes by default
- As an indication:

```
$ dig +dnssec +bufsize=4096 MX comcast.net
...
;; MSG SIZE  rcvd: 3229
```

- That will get fragmented into 3 packets!

SURF NET

Why fragmentation is a problem



So why are fragments blocked?

- In the 1990s there was a host of fragment-related attacks (remember the ping-of-death, anyone?)
- 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)

Extent of the problem

- 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]

[1] Weaver, N., Kreibich, C., Nechaev, B., and Paxson, V.: Implications of Netalyzr's DNS Measurements. In: Proceedings of the First Workshop on Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom, (2011).

[2] Van den Broek, J., Van Rijswijk, R., Pras, A., Sperotto, A., "DNSSEC and firewalls - Deployment problems and solutions", Private Communication, Pending Publication, (2012).

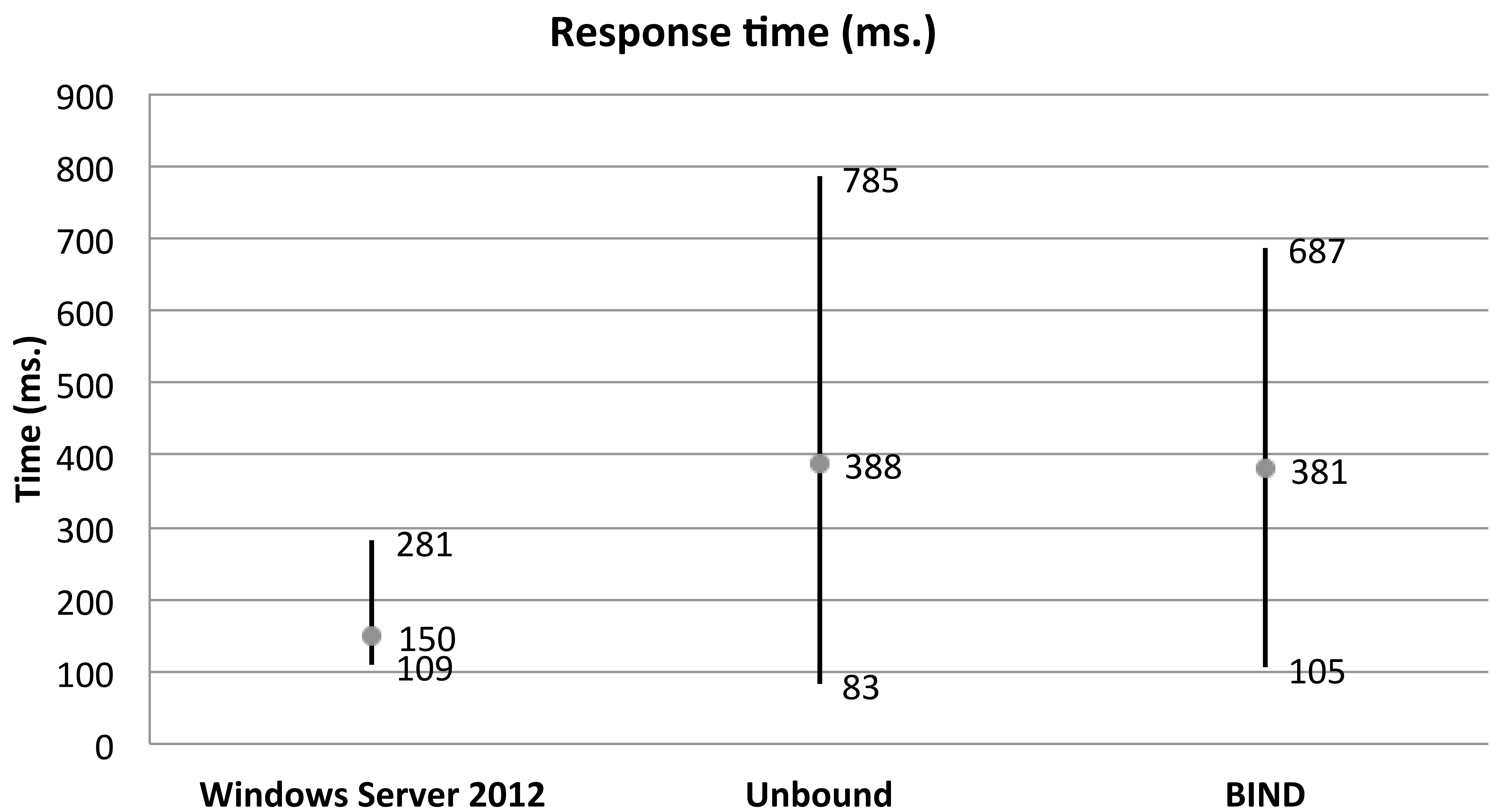
What you should do on your resolver

- Make sure you know the maximum packet size you can receive
- Use tools like the DNS-OARC reply-size tester
  - https://www.dns-oarc.net/oarc/services/replysizetest
- Reconfigure your firewall not to block fragments
  - e.g. older Cisco firewalls block DNS UDP >512 bytes + frags by default (!)
- Make sure you don't block TCP port 53!

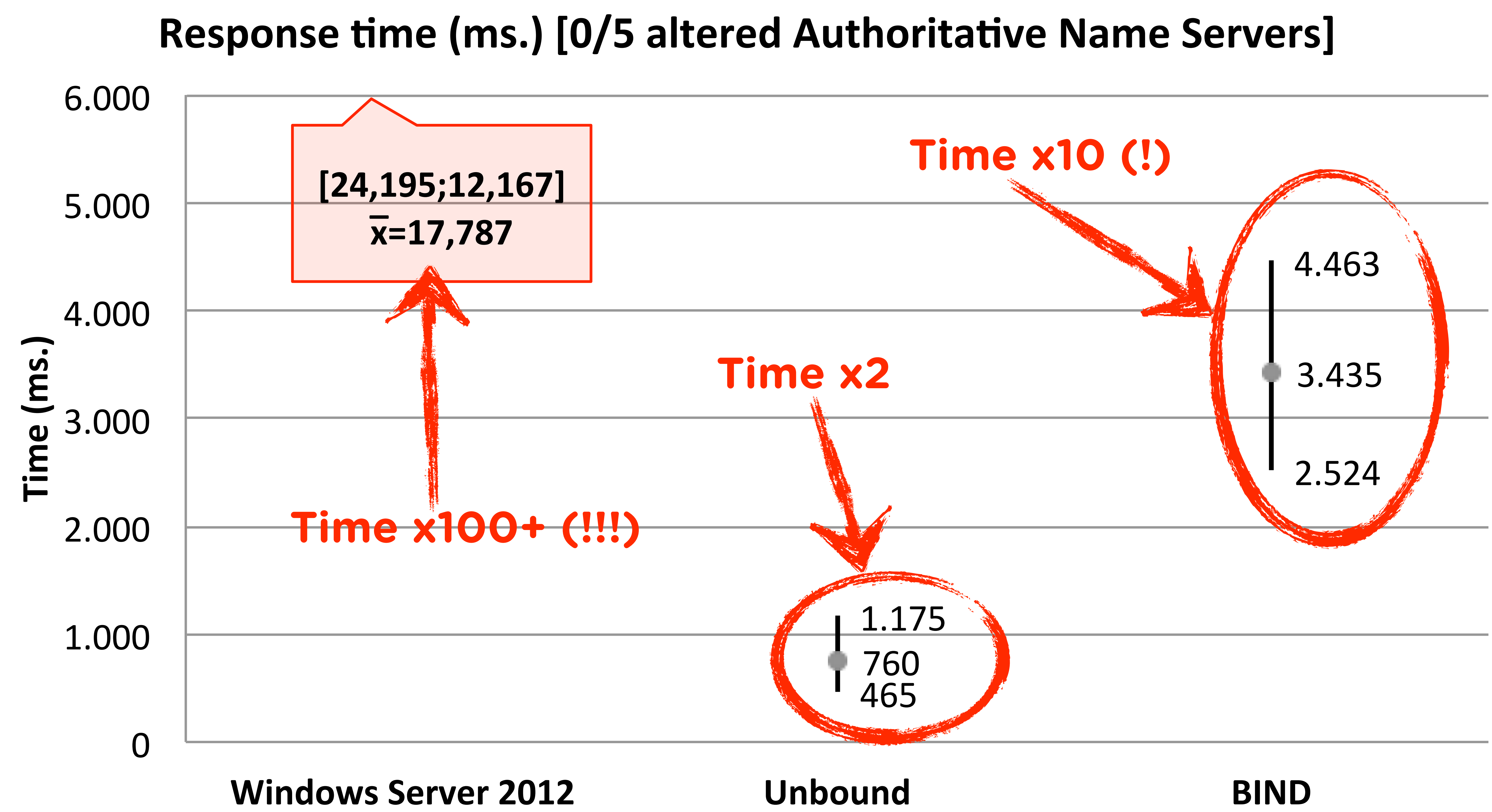
But I operate a signed zone...

- If you operate a DNSSEC signed zone, servers sending you queries may suffer from this problem...
- You want to be/stay resolvable, right?
- Luckily, there are some things you can do
- Let's dive into some resolver behaviour

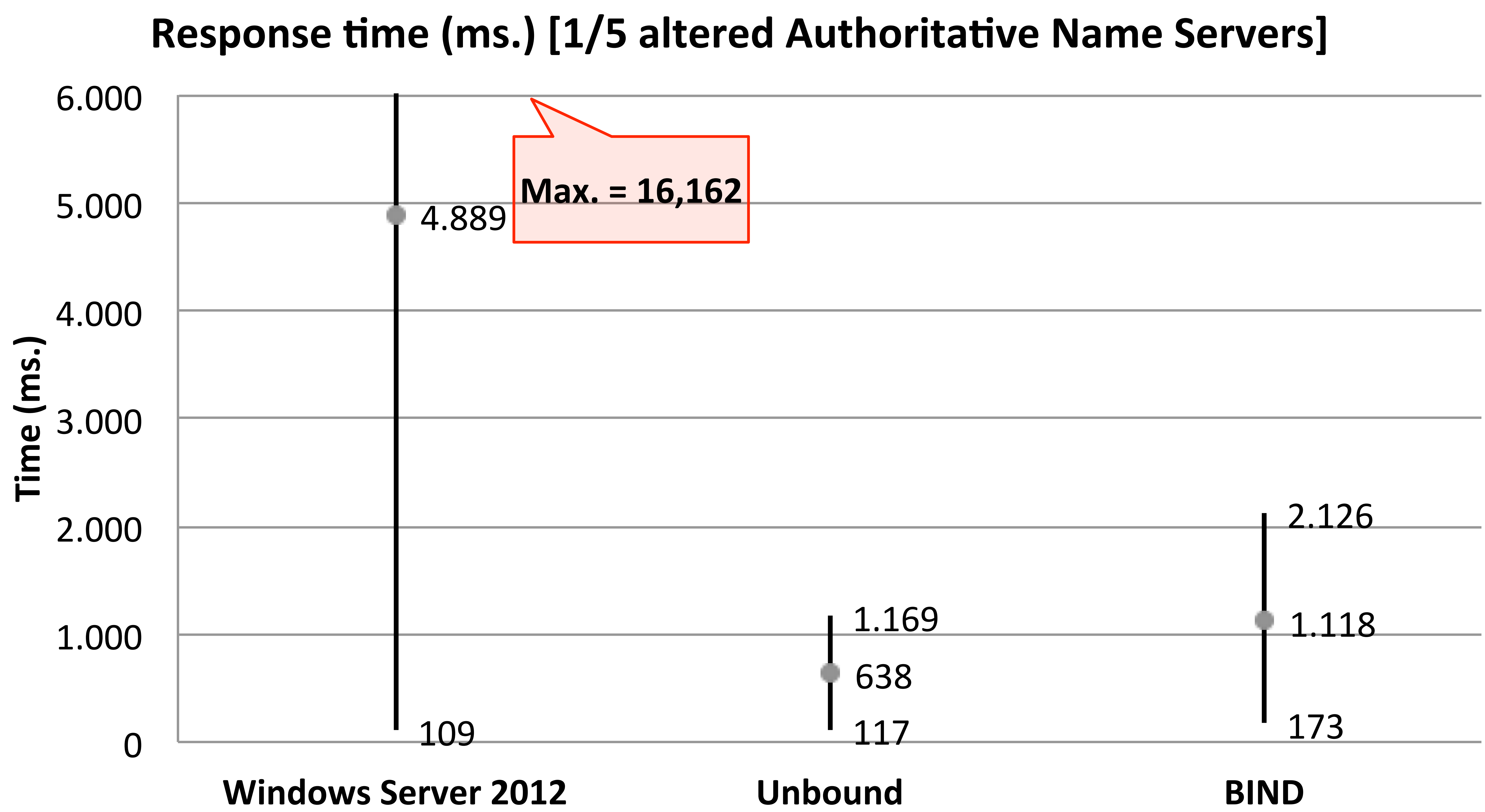
Resolver experiments (1) Normal operations



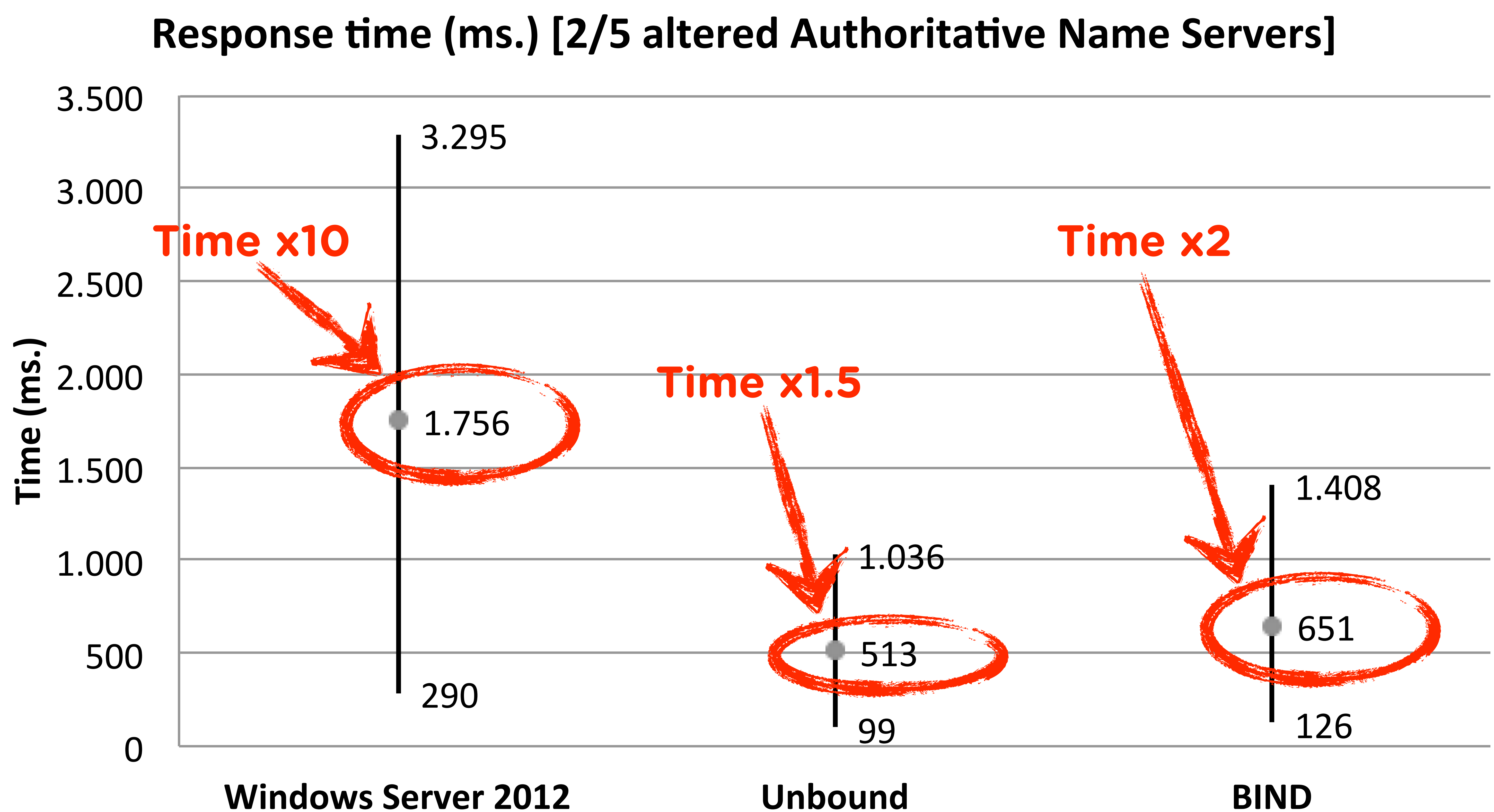
Resolver experiments (2)
Blocking fragments



Resolver experiments (3)
Max. resp. size on 1 authNS



Resolver experiments (4)
Max. resp. size on 2 authNS



Experiment on live authNS

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

*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:

Max. response	A for www	AAAA for www	DNSKEY
4096	0.0%	0.0%	0.0%
1472	1.8%	1.8%	8.1%
1232	2.9%	3.5%	**40.0%**

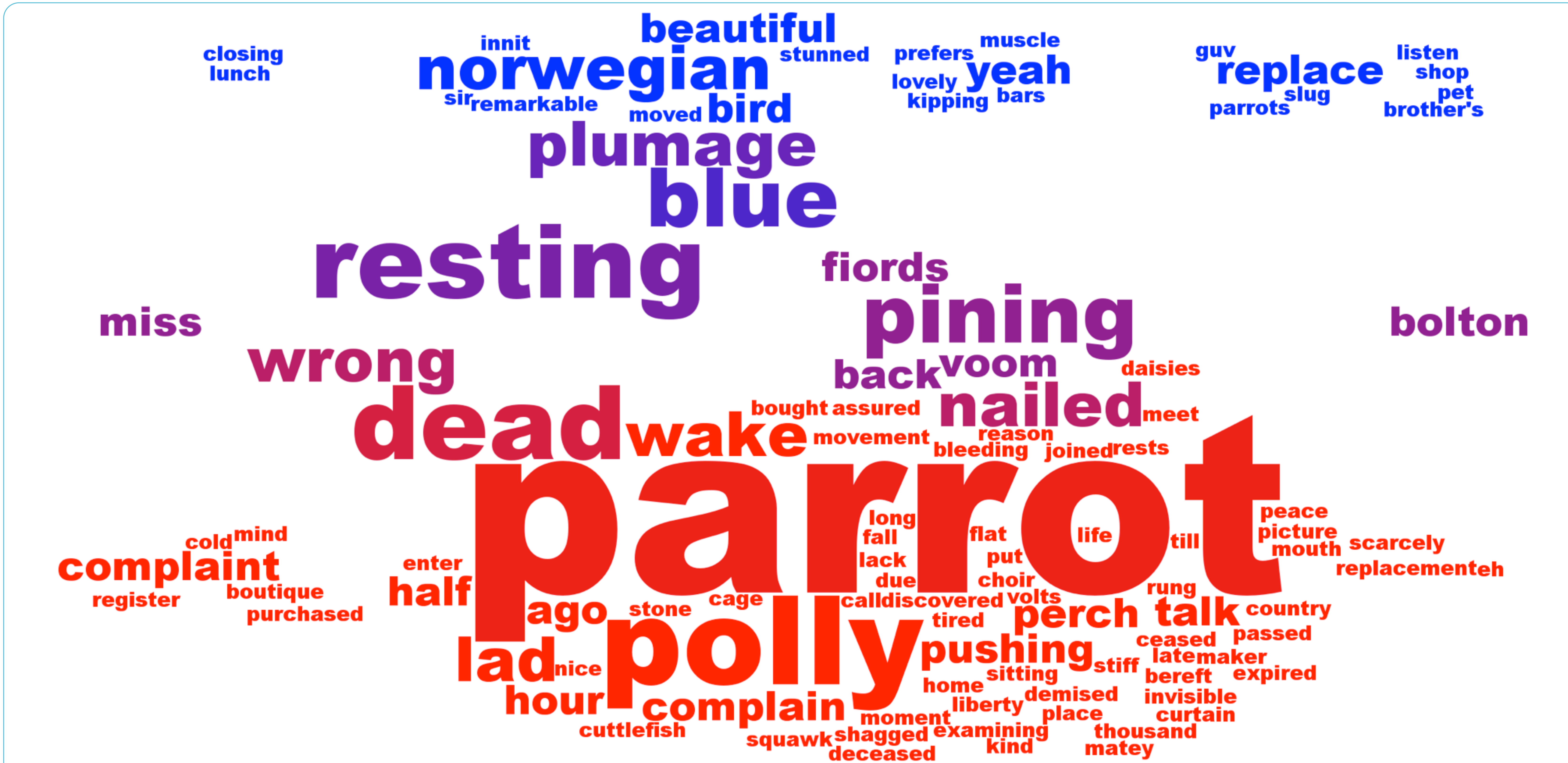
- 30% truncations were expected for a maximum response size of 1232 bytes by Rikitake, K., Nogawa, H., Tanaka, T., Nakao, K. and Shimojo, S. “An Analysis of DNSSEC Transport Overhead Increase”, IPSJ SIG Technical Reports 2005-CSEC-28, Vol. 2005, No. 33, pp. 345-350,

So what can you do?

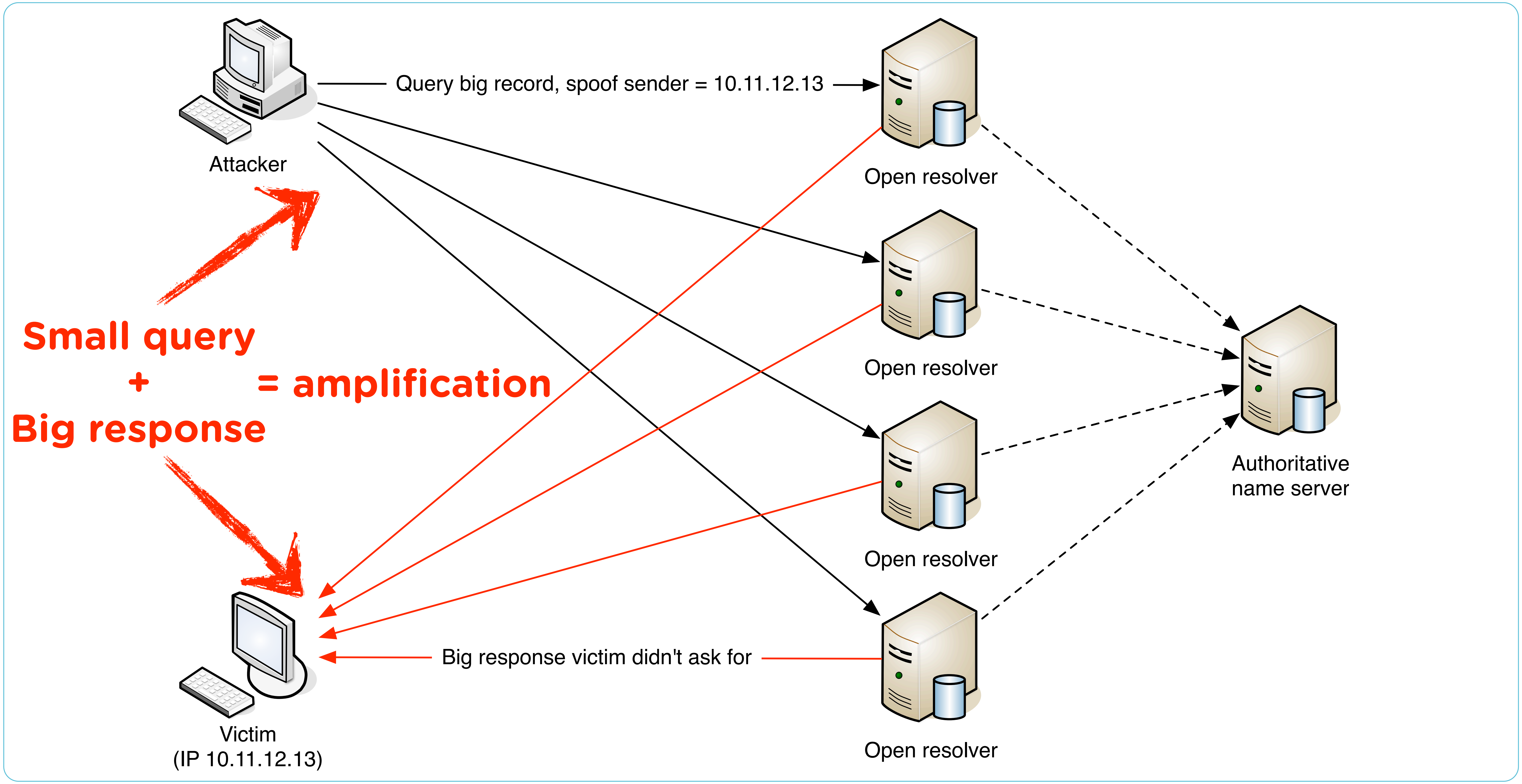
- If you use BIND: set “**minimal-responses: yes**”
- If you use NSD, make sure you use NSD ≥ 3.2.9

- Or: limit the maximum response size
  - Works well, as demonstrated in previous slides
  - BIND: set “edns-udp-size”
  - Windows Server: change “MaximumUdpPacketSize” in registry
  - Do this only on *some* of your authoritative servers
  - Choose a value below the PMTU (e.g. 1472 or 1232 bytes)
  - **And make sure your server can be reached over TCP!**

And now for something completely different



DNS(SEC) amplification



Remember that comcast.net MX query?

```
$ tcpdump -n -v -i en0 host xxxx
...
11:00:19.411981 IP (... proto UDP (17), length 68)
    yyyy.55023 > xxxx.53: 36075+ [1au] MX? comcast.net.
...
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
```

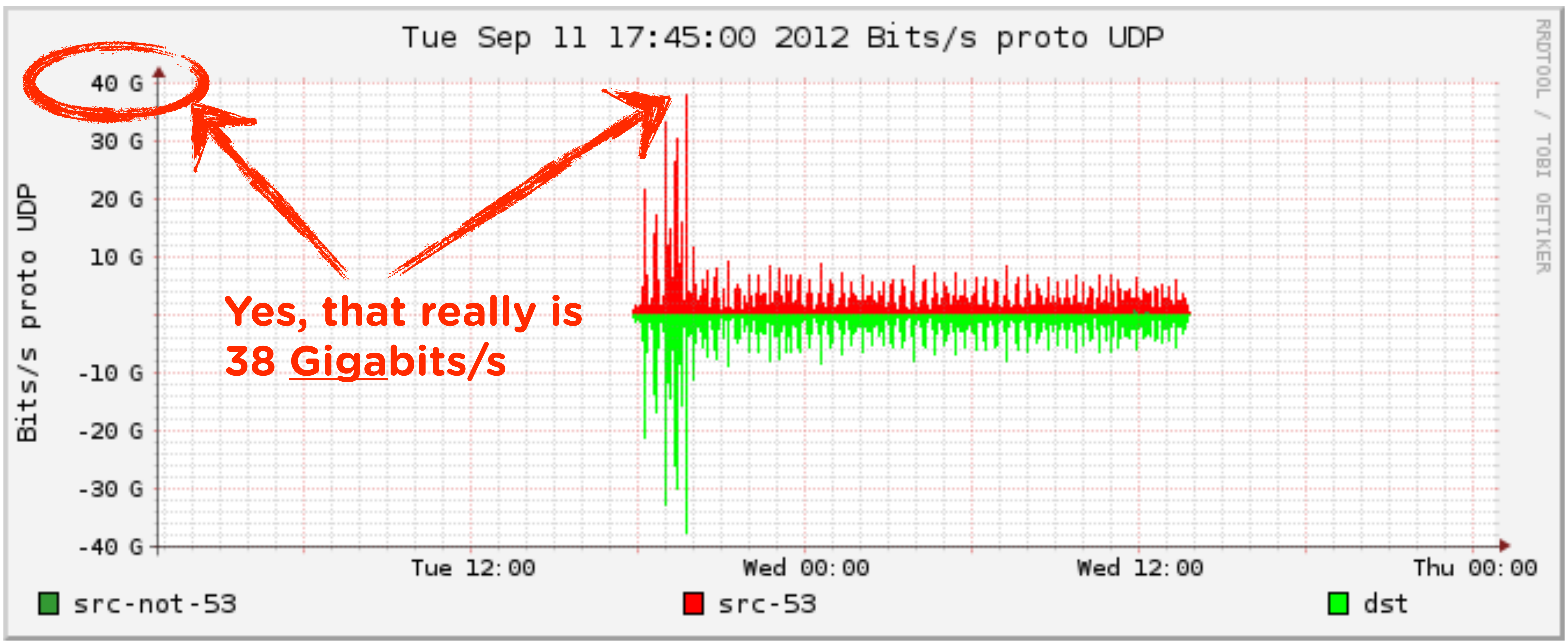
Send: 68 bytes, recv: 3297 bytes, amp. ≈ 48.5x !

DNS(SEC) amplification is on the rise

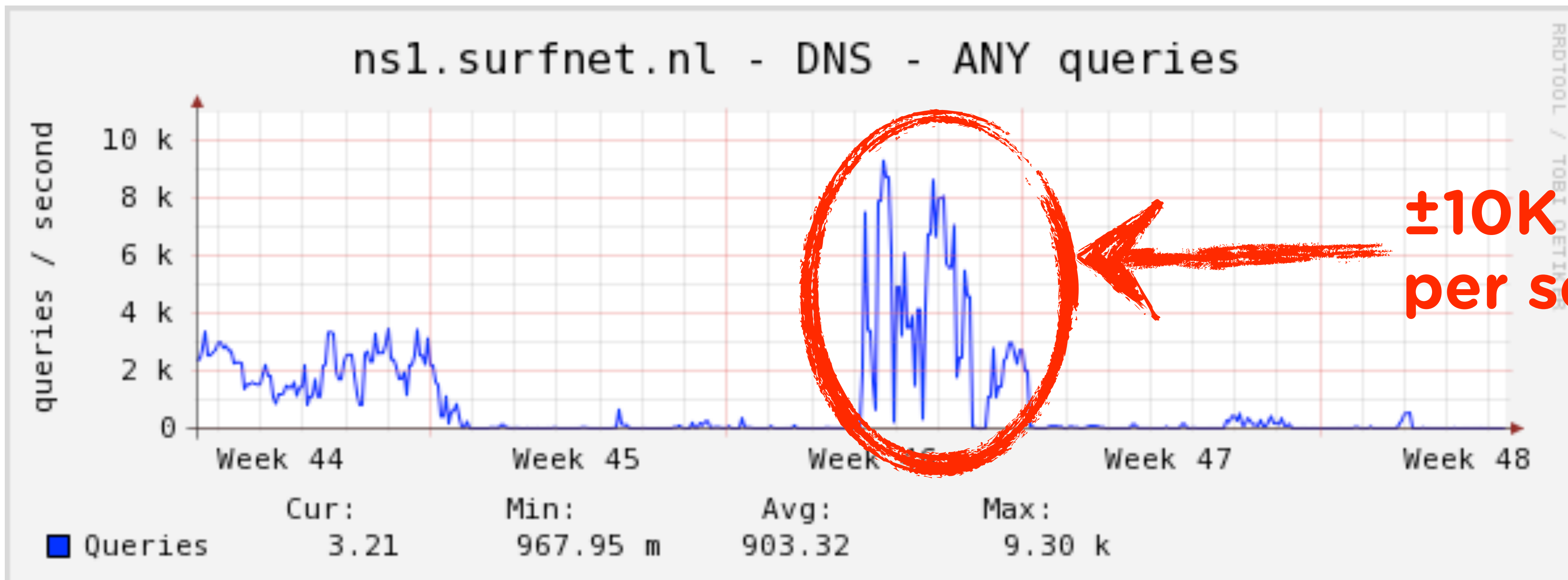
- Our CERT team sees both abuse of our name servers as well as the attack being used against us and our constituency
- Seems to be popular among “evildoers”
- Hasn’t gotten any better with the introduction of DNSSEC (larger answers!) but was already a problem with plain old DNS

A small (?) example

- Attack against some infrastructure we host:

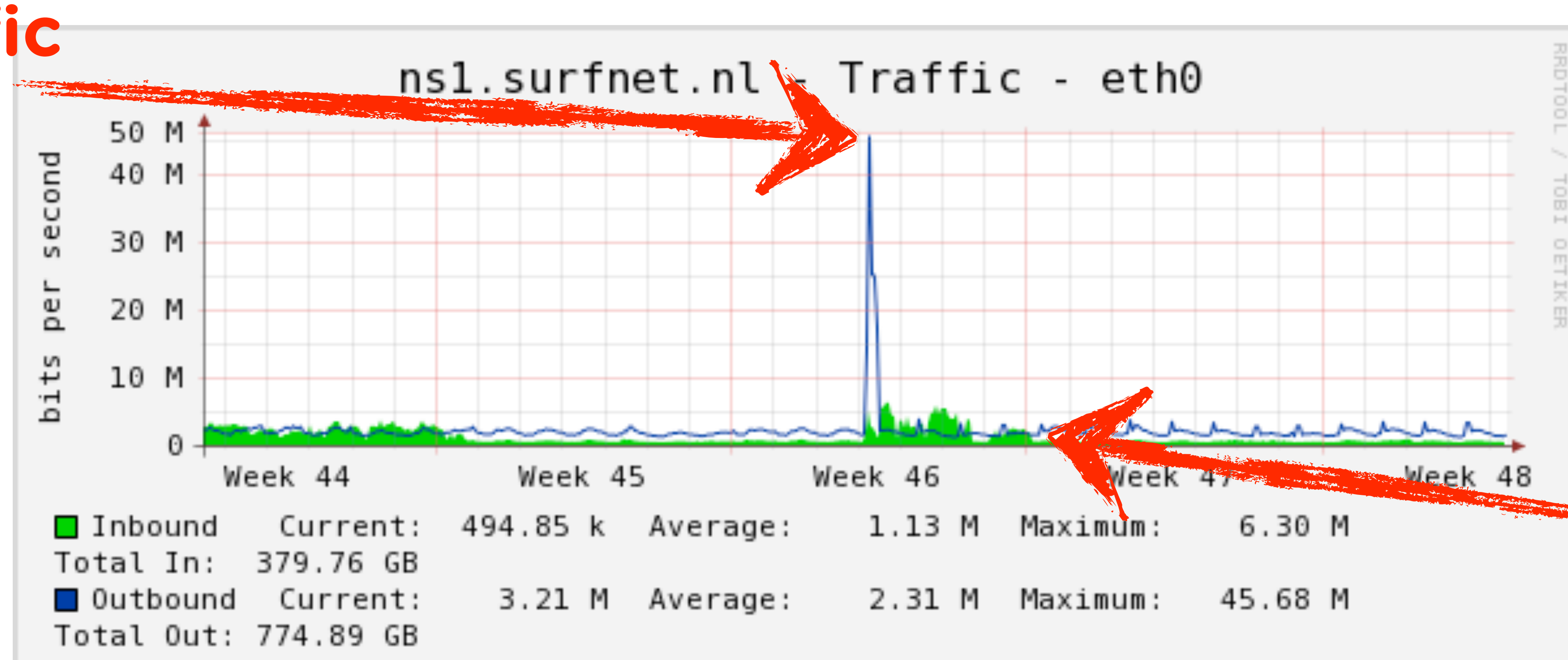


Another example: abuse of our authoritative name servers



±10K queries per second

Outbound traffic before filtering



Inbound traffic not very high

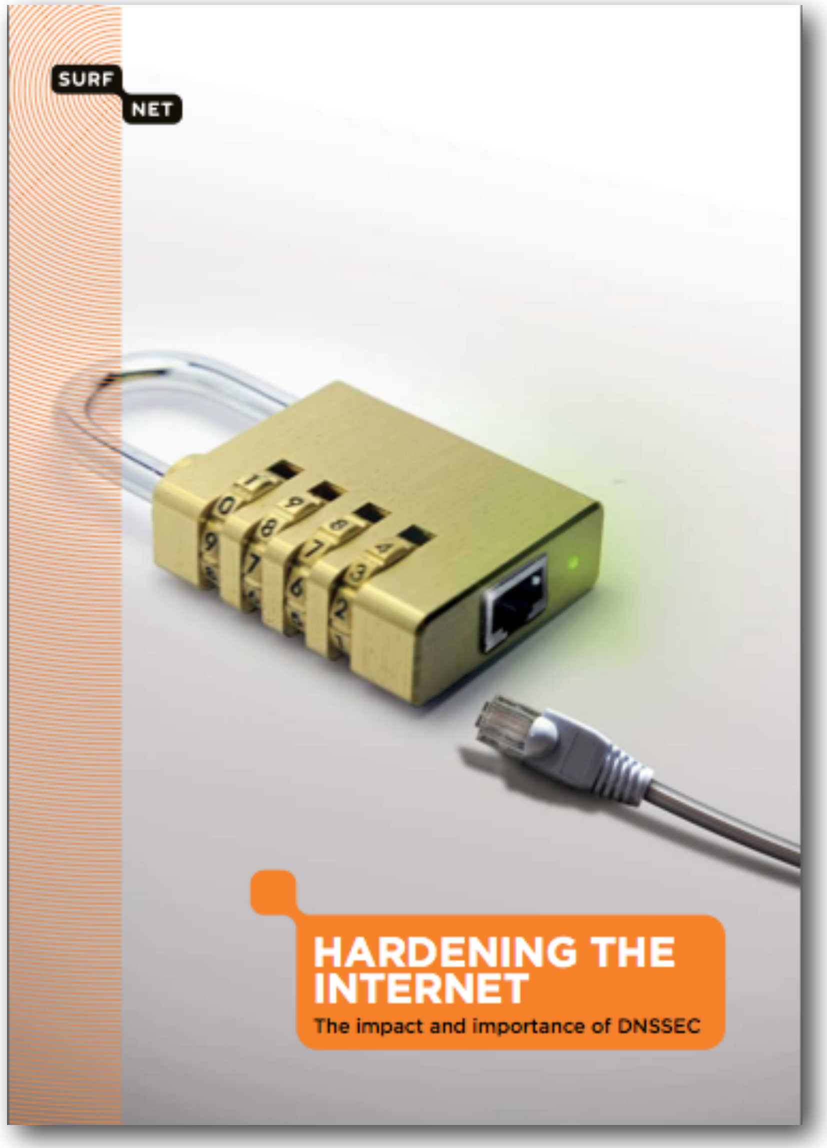
What can you do?

- Only real solution: **implement BCP38**
  - BCP38 = ingress filtering; only allow traffic into your network from end points with valid addresses --> http://tools.ietf.org/html/bcp38
- We actively monitor attacks and filter them
- Rate limiting DNS is being advocated a lot lately
  - Preliminary patch for BIND
  - Plans to implement in NSD
  - But can affect legitimate traffic, so be careful (!)

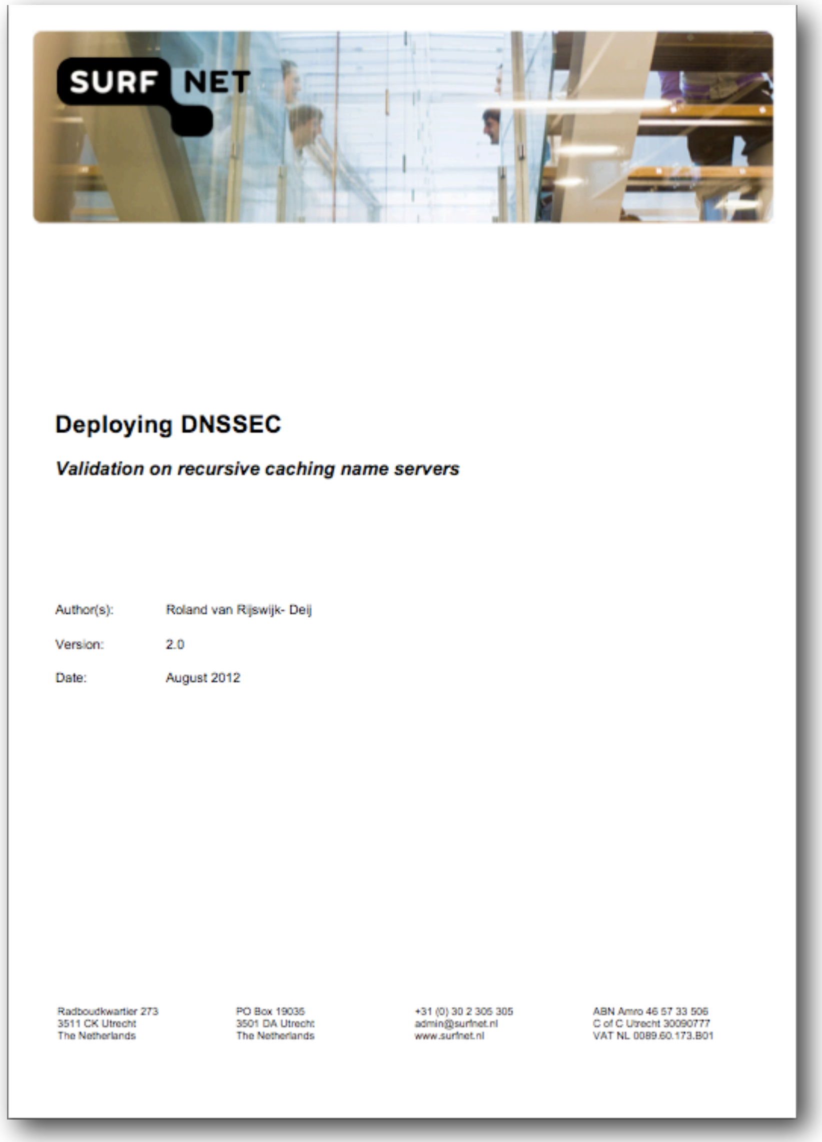
Conclusions

- It is very likely that you are using DNSSEC one way or another
- You may need to take action to make sure things keep working smoothly; DNSSEC is here to stay, the number of signed zones is on the rise
- We need to keep an eye out for "evil" behaviour that abuses DNS(SEC)

More information



http://bit.ly/sn-dnssec-2008


SURF NET

Deploying DNSSEC

Validation on recursive caching name servers

Author(s): Roland van Rijswijk- Deij
Version: 2.0
Date: August 2012

Radboudkwartier 273
3511 CK Utrecht
The Netherlands

PO Box 19035
3501 DA Utrecht
The Netherlands

+31 (0) 30 2 305 305
admin@surfnet.nl
www.surfnet.nl

ABN Amro 46 57 33 506
C of C Utrecht 30090777
VAT NL 0089.60.173.B01

http://bit.ly/sn-dnssec-vali

http://bit.ly/sn-cryptoweb

SURFnet DNSSEC blog: https://dnssec.surfnet.nl/

roland.vanrijswijk@surfnet.nl

nl.linkedin.com/in/rolandvanrijswijk

@reseauxsansfil

Questions? Comments?