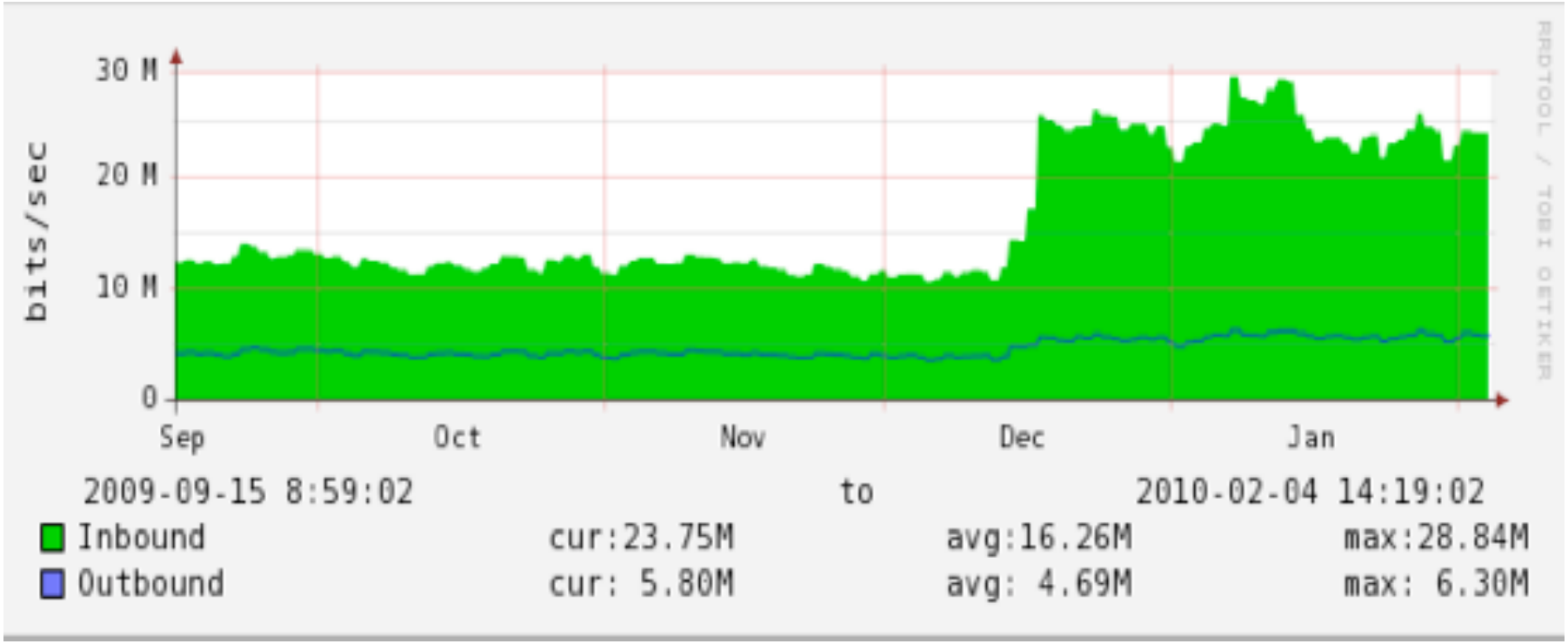
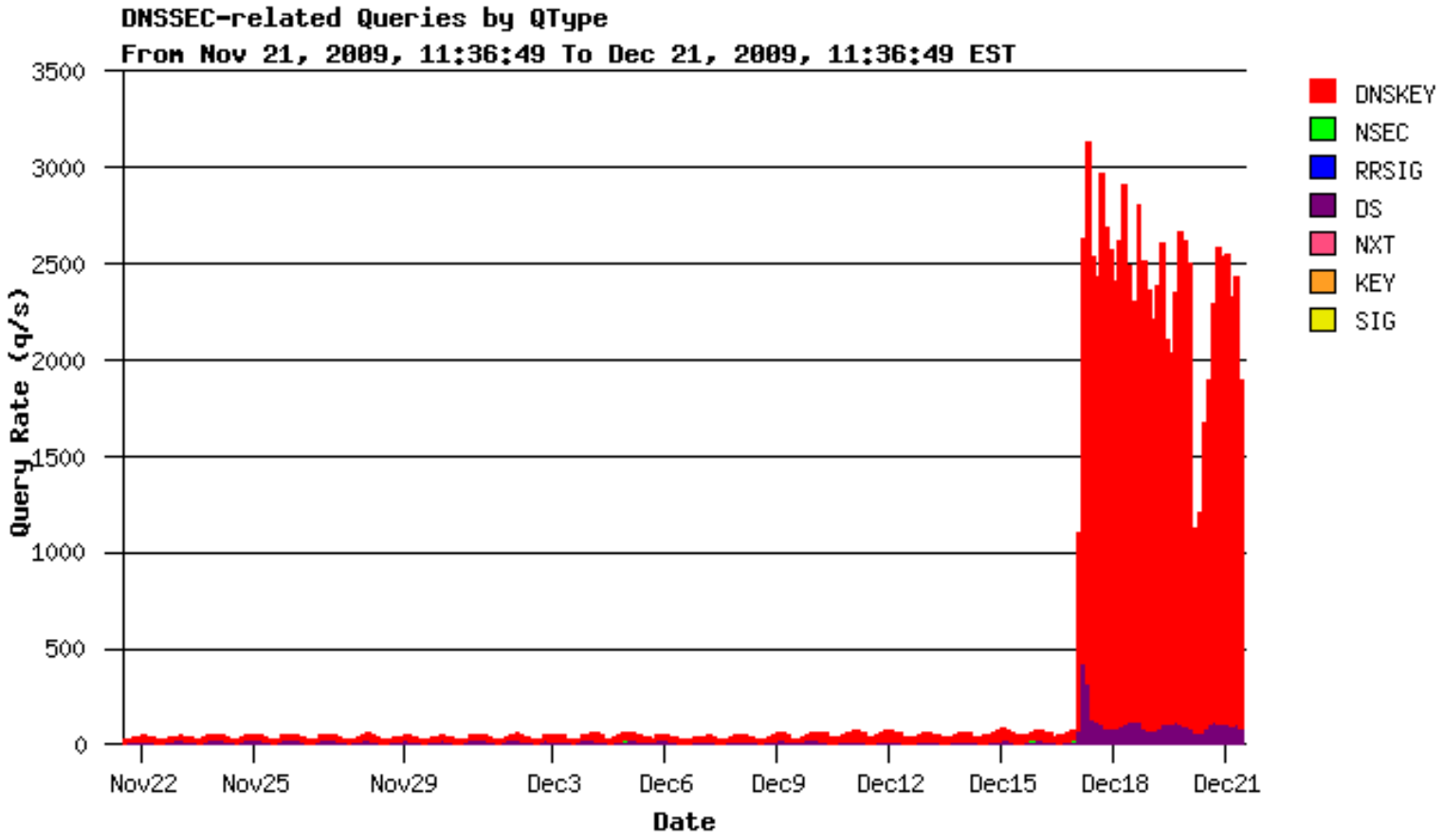
Roll over and Die

George Michaelson ggm@apnic.net
Patrik Wallstrom pawal@blipp.com
Roy Arends roy@dnss.ec
Geoff Huston gih@apnic.net

Operations Group notice an interface has gone feral…



DNS query load in DNSSEC has gone feral



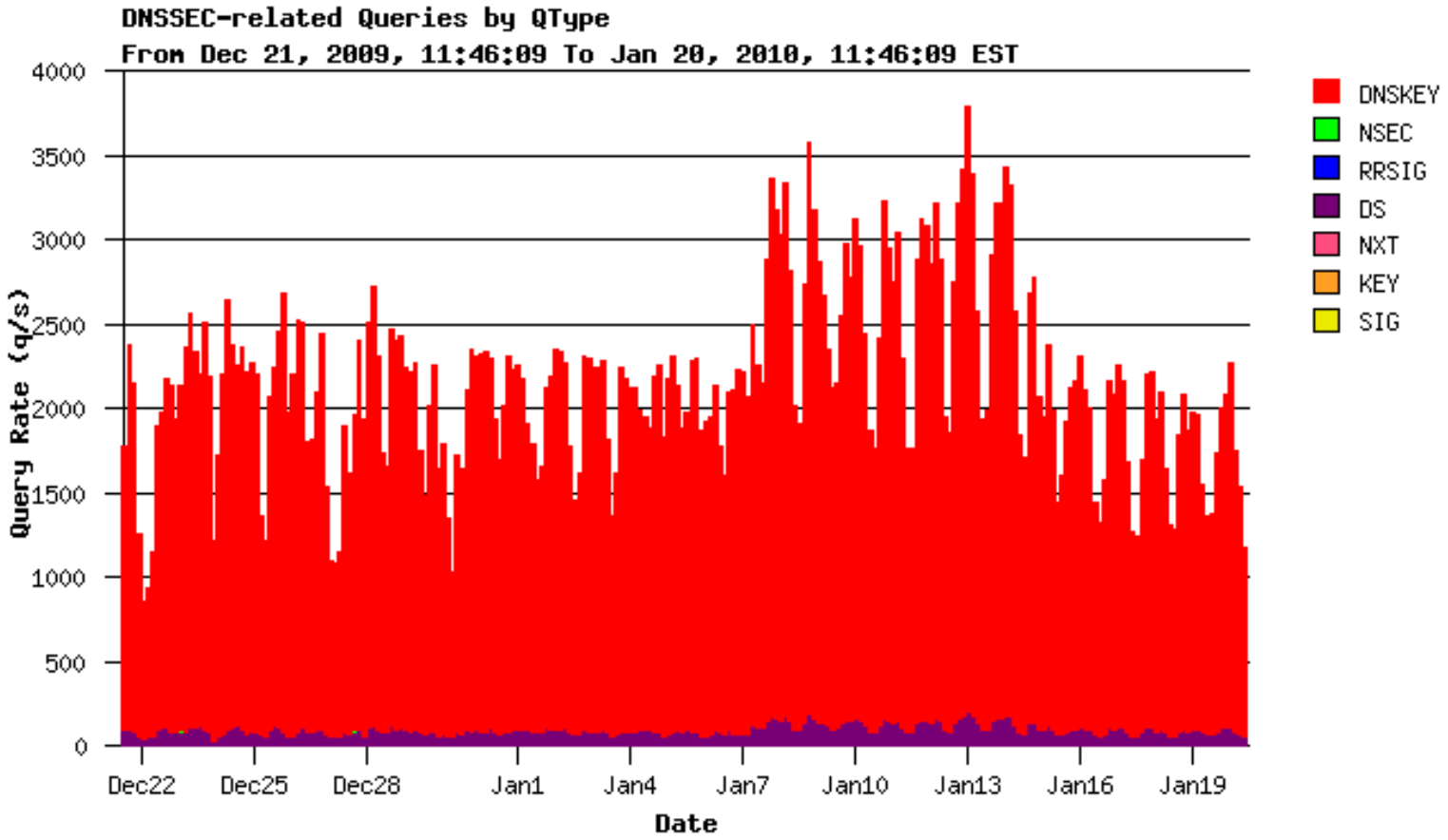
Whats going on?

- DoS attack on DNS servers?
- DoS attack using DNS reflection?
- Application(s) gone feral?
- Some other reason …

No biggie?

- 3000 q/sec per node
  - 1000byte response (150b more normal)
  - That's 30mbit/sec of excess
  - 3000 extra sessions
  - Lots of source IPS
- No biggie, but unexpected
  - Re-scaling of servers probably needs to be brought forward
- But what about the future?

It continues..



What does it look like?

- Time to break out the dnscap/tcpdump…

What does it look like?

```
[76] 2010-01-20 03:06:46.755545 [#56 eth2 0] \
     [212.126.213.158].38868 [202.12.29.59].53  \
     dns QUERY,NOERROR,11038,cd \
     1 211.89.in-addr.arpa,IN,TYPE43 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

Requests DS records for a Specific subdomain in a Signed zone.

```
[577] 2010-01-20 03:06:46.755634 [#57 eth2 0] \
      [202.12.29.59].53 [212.126.213.158].38868  \
      dns QUERY,NOERROR,11038,qr|aa \
      1 211.89.in-addr.arpa,IN,TYPE43 0 \
      4 89.in-addr.arpa,IN,SOA,7200,ns-pri.ripe.net,dns-help.ripe.net,
    2010012011,3600,7200,1209600,7200 \
      89.in-addr.arpa,IN,TYPE46,172800,[185] \
      210.89.in-addr.arpa,IN,TYPE47,7200,[31] \
      210.89.in-addr.arpa,IN,TYPE46,7200,[185] \
      1 .,CLASS4096,OPT,0,[0]
```

Reply says ‘no such delegation’
And sends RRSIG and NSEC from Parent zone, and surrounding records

But wait. There is more

```
[72] 2010-01-20 03:06:47.817759 [#60 eth2 0] \
     [212.126.213.158].24482 [202.12.29.59].53  \
     dns QUERY,NOERROR,226,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

Requests DNSKEY for parent zone

```
[1188] 2010-01-20 03:06:47.817835 [#61 eth2 0] \
     [202.12.29.59].53 [212.126.213.158].24482  \
     dns QUERY,NOERROR,226,qr|aa \
     1 89.in-addr.arpa,IN,TYPE48 \
     5 89.in-addr.arpa,IN,TYPE48,3600,[264] \
     89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE46,3600,[291] \
     89.in-addr.arpa,IN,TYPE46,3600,[185] 0 \
     1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and
RRSIG set for parent zone

```
[72] 2010-01-20 03:06:48.892744 [#66 eth2 0] \
     [212.126.213.158].59720 [202.12.29.59].53  \
     dns QUERY,NOERROR,25901,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

Re- Requests DNSKEY for parent zone

No, really. There is more

```
[1188] 2010-01-20 03:06:48.892825 [#67 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].59720  \
    dns QUERY,NOERROR,25901,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and
RRSIG set for parent zone

```
[72] 2010-01-20 03:06:49.963730 [#70 eth2 0] \
    [212.126.213.158].41028 [202.12.29.59].53  \
    dns QUERY,NOERROR,41199,cd \
    1 89.in-addr.arpa,IN,TYPE48 0 0 \
    1 .,CLASS4096,OPT,32768,[0]
```

Re- Requests DNSKEY for parent zone

```
[1188] 2010-01-20 03:06:49.963821 [#71 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].41028  \
    dns QUERY,NOERROR,41199,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and
RRSIG set for parent zone

Any more?

```
[72] 2010-01-20 03:06:51.024725 [#74 eth2 0] \
     [212.126.213.158].32988 [202.12.29.59].53  \
     dns QUERY,NOERROR,15282,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

And again..

```
[1188] 2010-01-20 03:06:51.024820 [#75 eth2 0] \
     [202.12.29.59].53 [212.126.213.158].32988  \
     dns QUERY,NOERROR,15282,qr|aa \
     1 89.in-addr.arpa,IN,TYPE48 \
     5 89.in-addr.arpa,IN,TYPE48,3600,[264] \
     89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE46,3600,[185] \
     89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
     1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

```
[72] 2010-01-20 03:06:52.091457 [#78 eth2 0] \
     [212.126.213.158].35975 [202.12.29.59].53  \
     dns QUERY,NOERROR,13437,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

And again

This is bad

```
[1188] 2010-01-20 03:06:52.091544 [#79 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].35975  \
    dns QUERY,NOERROR,13437,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

```
[72] 2010-01-20 03:06:53.451761 [#84 eth2 0] \
    [212.126.213.158].6458 [202.12.29.59].53  \
    dns QUERY,NOERROR,53340,cd \
    1 89.in-addr.arpa,IN,TYPE48 0 0 \
    1 .,CLASS4096,OPT,32768,[0]
```

And again

```
[1188] 2010-01-20 03:06:53.451852 [#85 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].6458  \
    dns QUERY,NOERROR,53340,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

Nearly there

```
[72] 2010-01-20 03:06:54.556688 [#88 eth2 0] \
     [212.126.213.158].45030 [202.12.29.59].53  \
     dns QUERY,NOERROR,11754,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

And again

```
[1188] 2010-01-20 03:06:54.556779 [#89 eth2 0] \
     [202.12.29.59].53 [212.126.213.158].45030  \
     dns QUERY,NOERROR,11754,qr|aa \
     1 89.in-addr.arpa,IN,TYPE48 \
     5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE48,3600,[158] \
     89.in-addr.arpa,IN,TYPE48,3600,[264] \
     89.in-addr.arpa,IN,TYPE46,3600,[185] \
     89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
     1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

```
[72] 2010-01-20 03:06:55.335413 [#92 eth2 0] \
     [212.126.213.158].57584 [202.12.29.59].53  \
     dns QUERY,NOERROR,8453,cd \
     1 89.in-addr.arpa,IN,TYPE48 0 0 \
     1 .,CLASS4096,OPT,32768,[0]
```

And again

Neeeeeerly there

```
[1188] 2010-01-20 03:06:55.335556 [#93 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].57584  \
    dns QUERY,NOERROR,8453,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

```
[72] 2010-01-20 03:06:56.463737 [#96 eth2 0] \
    [212.126.213.158].37422 [202.12.29.59].53  \
    dns QUERY,NOERROR,47232,cd \
    1 89.in-addr.arpa,IN,TYPE48 0 0 \
    1 .,CLASS4096,OPT,32768,[0]
```

And again

```
[1188] 2010-01-20 03:06:56.463818 [#97 eth2 0] \
    [202.12.29.59].53 [212.126.213.158].37422  \
    dns QUERY,NOERROR,47232,qr|aa \
    1 89.in-addr.arpa,IN,TYPE48 \
    5 89.in-addr.arpa,IN,TYPE48,3600,[264] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE48,3600,[158] \
    89.in-addr.arpa,IN,TYPE46,3600,[185] \
    89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
    1 .,CLASS4096,OPT,0,[0]
```

Receives DNSKEY and RRSIG set for parent zone

done!

```
[72] 2010-01-20 03:06:58.046152 [#102 eth2 0] \
      [212.126.213.158].14831 [202.12.29.59].53  \
      dns QUERY,NOERROR,14439,cd \
      1 89.in-addr.arpa,IN,TYPE48 0 0 \
      1 .,CLASS4096,OPT,32768,[0]

[1188] 2010-01-20 03:06:58.046272 [#103 eth2 0] \
      [202.12.29.59].53 [212.126.213.158].14831  \
      dns QUERY,NOERROR,14439,qr|aa \
      1 89.in-addr.arpa,IN,TYPE48 \
      5 89.in-addr.arpa,IN,TYPE48,3600,[158] \
      89.in-addr.arpa,IN,TYPE48,3600,[264] \
      89.in-addr.arpa,IN,TYPE48,3600,[158] \
      89.in-addr.arpa,IN,TYPE46,3600,[185] \
      89.in-addr.arpa,IN,TYPE46,3600,[291] 0 \
      1 .,CLASS4096,OPT,0,[0]
$
```

And again

Receives DNSKEY and RRSIG set for parent zone

What did I just see?

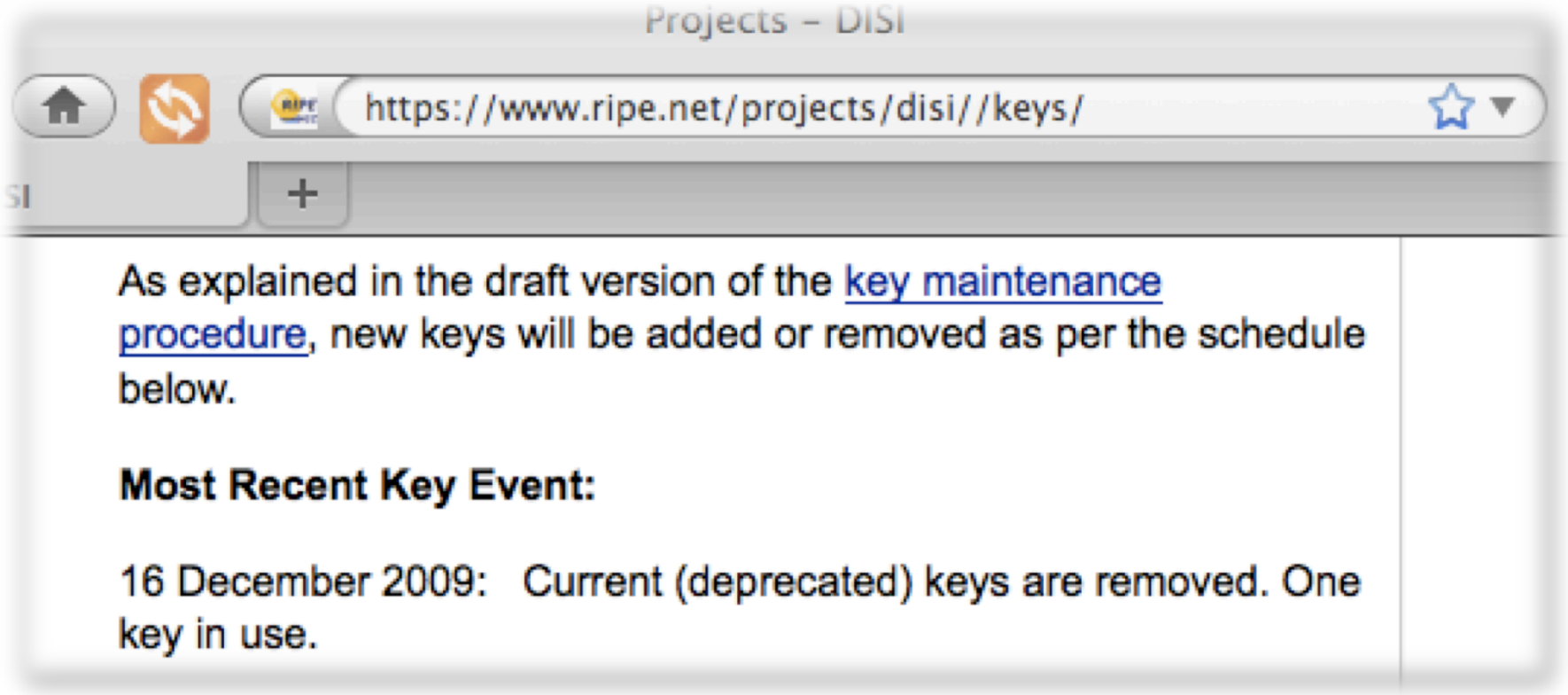
- 11 queries as a long tail, for one trigger event
  - Attempt to fetch 211.89.in-addr.arpa with sigcheck
  - Get 89.in-addr.arpa with RRSIG/NSEC set
  - Then fetch DNSKEY of 89.in-addr.arpa
  - Get 89.in-addr.arpa with DNSKEY/RRSIG set

    - Again
    - And again
    - And again

There’s something about DNSSEC

- Have the keys changed?
- Time to look back…

RIPE-NCC announced key roll



Its just in-addr.arpa, right?

- No. remember, in-addr.arpa is just another domain
- No special magic here. Its all DNS
- So, in principle this can happen in any domain
- Is it happening in any other DNSSEC enabled domain?
- …

Surely not seen before?

- Patrik Wallstrom (.se) was seeing large packetflows for .se, after a keychange..

- After investigation, finds it is his own resolver!
  - Opportunity to use tcpdump, examine resolver side view of this situation.
- So Patrik goes digging. What does he see?

.se keychange validation failure

06:50:18.045007 IP 94.254.84.99.6197 > 130.239.5.114.53: 34306% [1au] DNSKEY? se. (31)
06:50:18.059296 IP 94.254.84.99.7496 > 199.254.63.1.53: 8090% [1au] DNSKEY? se. (31)
06:50:18.088664 IP 94.254.84.99.10736 > 81.228.10.57.53: 62708% [1au] DNSKEY? se. (31)
06:50:19.075008 IP 94.254.84.99.18234 > 81.228.8.16.53: 8774% [1au] DNSKEY? se. (31)
06:50:19.078576 IP 94.254.84.99.57591 > 81.228.8.16.53: P 0:33(33) ack 1 win 78
<nop,nop,timestamp 428416112 3213455965>58695% [1au] DNSKEY? se. (31)06:50:19.081633 IP
94.254.84.99.50469 > 81.228.10.57.53: 36119% [1au] DNSKEY? se. (31)06:50:20.685057 IP
94.254.84.99.24744 > 81.228.10.57.53: 60621% [1au] DNSKEY? se. (31)06:50:20.711385 IP
94.254.84.99.42081 > 81.228.10.57.53: P 0:33(33) ack 1 win 78 <nop,nop,timestamp 428416275
3215995958>15873% [1au] DNSKEY? se. (31)06:50:20.727943 IP 94.254.84.99.18180 >
194.146.106.22.53: 639% [1au] DNSKEY? se. (31)06:50:20.729360 IP 94.254.84.99.14715 >
130.239.5.114.53: 42673% [1au] DNSKEY? se. (31)06:50:20.740681 IP 94.254.84.99.8412 >
192.36.135.107.53: 43484% [1au] DNSKEY? se. (31)06:50:20.748103 IP 94.254.84.99.15091 >
199.7.49.30.53: 48966% [1au] DNSKEY? se. (31)06:50:20.771257 IP 94.254.84.99.26304 >
199.254.63.1.53: 5749% [1au] DNSKEY? se. (31)06:50:20.800054 IP 94.254.84.99.43607 >
192.36.144.107.53: 42701% [1au] DNSKEY? se. (31)06:50:20.801602 IP 94.254.84.99.38911 >
192.71.53.53.53: 48332% [1au] DNSKEY? se. (31)06:50:20.803462 IP 94.254.84.99.60254 >
192.36.133.107.53: 644% [1au] DNSKEY? se. (31)06:50:20.815856 IP 94.254.84.99.31948 >
81.228.8.16.53: 17008% [1au] DNSKEY? se. (31)06:50:21.625015 IP 94.254.84.99.19712 >
81.228.10.57.53: 19280% [1au] DNSKEY? se. (31)06:50:22.542092 IP 94.254.84.99.41822 >
81.228.8.16.53: 22224% [1au] DNSKEY? se. (31)06:50:23.515012 IP 94.254.84.99.55978 >
81.228.10.57.53: 29847% [1au] DNSKEY? se. (31)

.se keychange validation failure

06:50:18.045007 IP 94.254.84.99.6197 > 130.239.5.114.53: 34306% [1au] DNSKEY? se. (31)
06:50:18.059296 IP 94.254.84.99.7496 > 199.254.63.1.53: 8090% [1au] DNSKEY? se. (31)
06:50:18.088664 IP 94.254.84.99.10736 > 81.228.10.57.53: 62708% [1au] DNSKEY? se. (31)
06:50:19.075008 IP 94.254.84.99.18234 > 81.228.8.16.53: 8774% [1au] DNSKEY? se. (31)
06:50:19.078576 IP 94.254.84.99.57591 > 81.228.8.16.53: P 0:33(33) ack 1 win 78
<nop,nop,timestamp 428416112 3213455965>58695% [1au] DNSKEY? se. (31)06:50:19.081633 IP
94.254.84.99.50469 > 81.228.10.57.53: 36119% [1au] DNSKEY? se. (31)06:50:20.685057 IP
94.254.84.99.24744 > 81.228.10.57.53: 60621% [1au] DNSKEY? se. (31)06:50:20.711385 IP
94.254.84.99.42081 > 81.228.10.57.53: P 0:33(33) ack 1 win 78 <nop,nop,timestamp 428416275
3215995958>15873% [1au] DNSKEY? se. (31)06:50:20.727943 IP 94.254.84.99.18180 >
194.146.106.22.53: 639% [1au] DNSKEY? se. (31)06:50:20.729360 IP 94.254.84.99.14715 >
130.239.5.114.53: 42673% [1au] DNSKEY? se. (31)06:50:20.740681 IP 94.254.84.99.8412 >
192.36.135.107.53: 43484% [1au] DNSKEY? se. (31)06:50:20.748103 IP 94.254.84.99.15091 >
199.7.49.30.53: 48966% [1au] DNSKEY? se. (31)06:50:20.771257 IP 94.254.84.99.26304 >
199.254.63.1.53: 5749% [1au] DNSKEY? se. (31)06:50:20.800054 IP 94.254.84.99.43607 >
192.36.144.107.53: 42701% [1au] DNSKEY? se. (31)06:50:20.801602 IP 94.254.84.99.38911 >
192.71.53.53.53: 48332% [1au] DNSKEY? se. (31)06:50:20.803462 IP 94.254.84.99.60254 >
192.36.133.107.53: 644% [1au] DNSKEY? se. (31)06:50:20.815856 IP 94.254.84.99.31948 >
81.228.8.16.53: 17008% [1au] DNSKEY? se. (31)06:50:21.625015 IP 94.254.84.99.19712 >
81.228.10.57.53: 19280% [1au] DNSKEY? se. (31)06:50:22.542092 IP 94.254.84.99.41822 >
81.228.8.16.53: 22224% [1au] DNSKEY? se. (31)06:50:23.515012 IP 94.254.84.99.55978 >
81.228.10.57.53: 29847% [1au] DNSKEY? se. (31)

.se keychange validation failure

06:50:18.045007 IP 94.254.84.99.6197 > 130.239.5.114.53: 34306% [1au] DNSKEY? se. (31)
06:50:18.059296 IP 94.254.84.99.7496 > 199.254.63.1.53: 8090% [1au] DNSKEY? se. (31)
06:50:18.088664 IP 94.254.84.99.10736 > 81.228.10.57.53: 62708% [1au] DNSKEY? se. (31)
06:50:19.075008 IP 94.254.84.99.18234 > 81.228.8.16.53: 8774% [1au] DNSKEY? se. (31)
06:50:19.078576 IP 94.254.84.99.57591 > 81.228.8.16.53: P 0:33(33) ack 1 win 78
<nop,nop,timestamp 428416112 3213455965>58695% [1au] DNSKEY? se. (31)06:50:19.081633 IP
94.254.84.99.50469 > 81.228.10.57.53: 36119% [1au] DNSKEY? se. (31)06:50:20.685057 IP
94.254.84.99.24744 > 81.228.10.57.53: 60621% [1au] DNSKEY? se. (31)06:50:20.711385 IP
94.254.84.99.42081 > 81.228.10.57.53: P 0:33(33) ack 1 win 78 <nop,nop,timestamp 428416275
3215995958>15873% [1au] DNSKEY? se. (31)06:50:20.727943 IP 94.254.84.99.18180 >
194.146.106.22.53: 639% [1au] DNSKEY? se. (31)06:50:20.729360 IP 94.254.84.99.14715 >
130.239.5.114.53: 42673% [1au] DN .8412 >
192.36.135.107.53: 43484% [1au] D 9.15091 >
199.7.49.30.53: 48966% [1au] DNS 6304 >
199.254.63.1.53: 5749% [1au] DNS 3607 >
192.36.144.107.53: 42701% [1au] D 9.38911 >
192.71.53.53.53: 48332% [1au] DN 60254 >
192.36.133.107.53: 644% [1au] DN 31948 >
81.228.8.16.53: 17008% [1au] DNS 9712 >
81.228.10.57.53: 19280% [1au] DN 41822 >
81.228.8.16.53: 22224% [1au] DNSKEY? se. (31)06:50:25.515012 IP 94.254.84.99.55978 >
81.228.10.57.53: 29847% [1au] DNSKEY? se. (31)

Notice how its querying **all** of the NS set of the domain being DNSSEC tested.

So this is not restricted to just one NS in the set…

.se keychange validation failure

06:50:18.045007 IP 94.254.84.99.6197 > 130.239.5.114.53: 34306% [1au] DNSKEY? se. (31)
06:50:18.059296 IP 94.254.84.99.7496 > 199.254.63.1.53: 8090% [1au] DNSKEY? se. (31)
06:50:18.088664 IP 94.254.84.99.10736 > 81.228.10.57.53: 62708% [1au] DNSKEY? se. (31)
06:50:19.075008 IP 94.254.84.99.18234 > 81.228.8.16.53: 8774% [1au] DNSKEY? se. (31)
06:50:19.078576 IP 94.254.84.99.57591 > 81.228.8.16.53: P 0:33(33) ack 1 win 78
<nop,nop,timestamp 428416112 3213455965>58695% [1au] DNSKEY? se. (31)06:50:19.081633 IP

Lots of DNSSEC related queries for his zone.

Endless amounts of DNSSEC in fact.

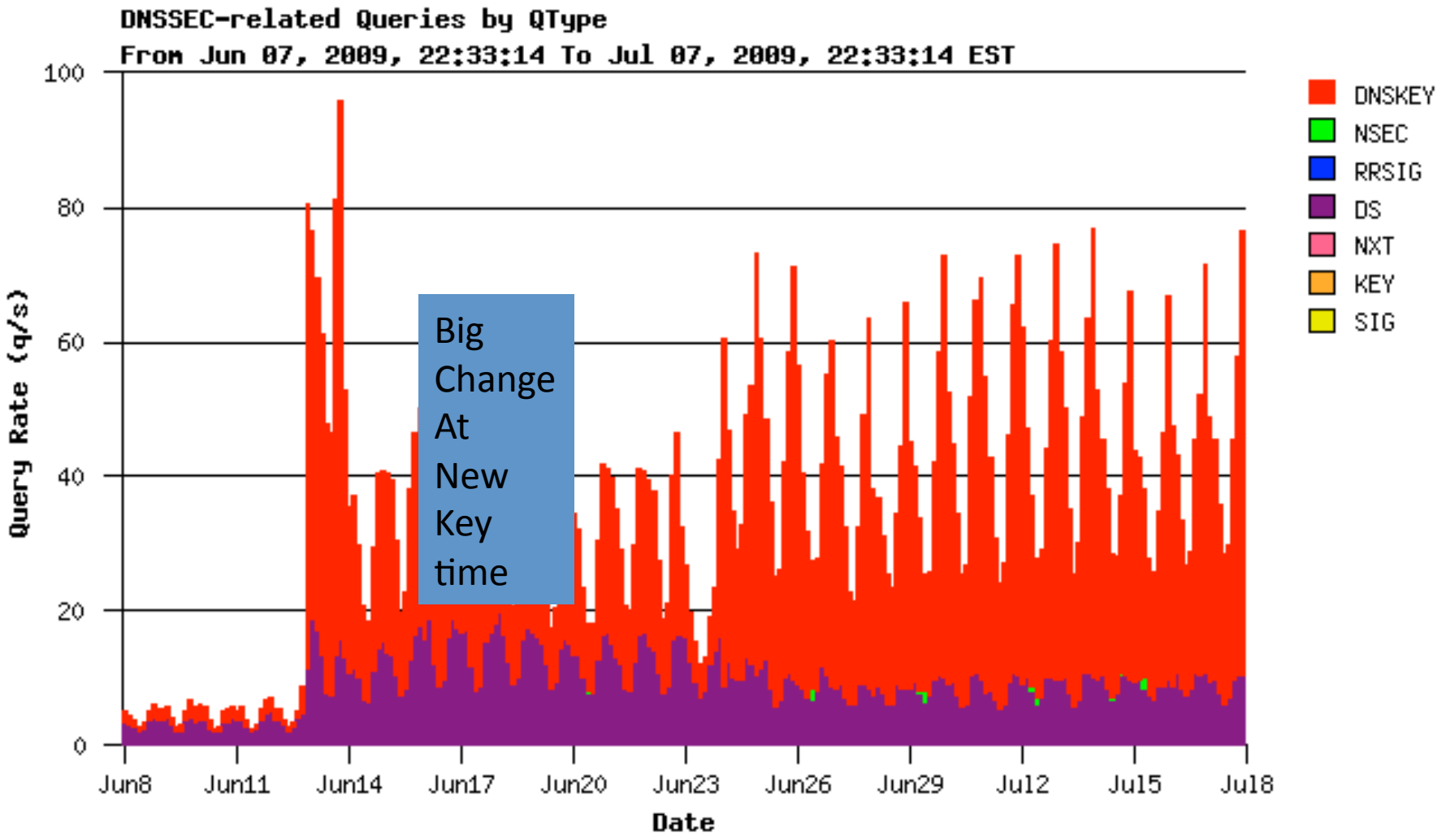
Re-requesting the DNSKEY state for his zone.

Why would .SE be doing this?

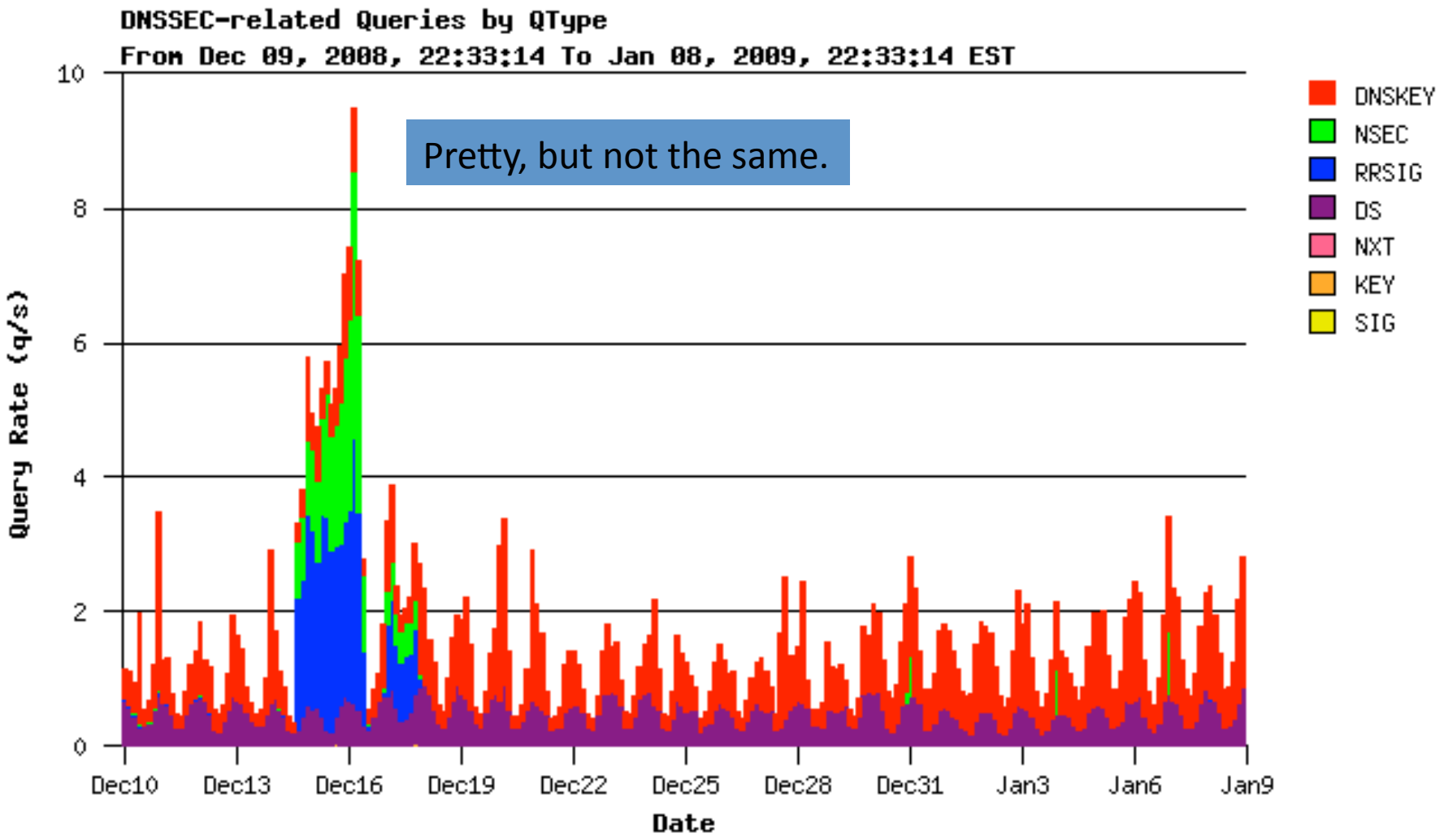
Any more?

- Maybe people need to be looking back into the query logs, DSC graphs?
- Re-examination of key rollover for .SE
  - January rollover, one resolver did 700 q/sec

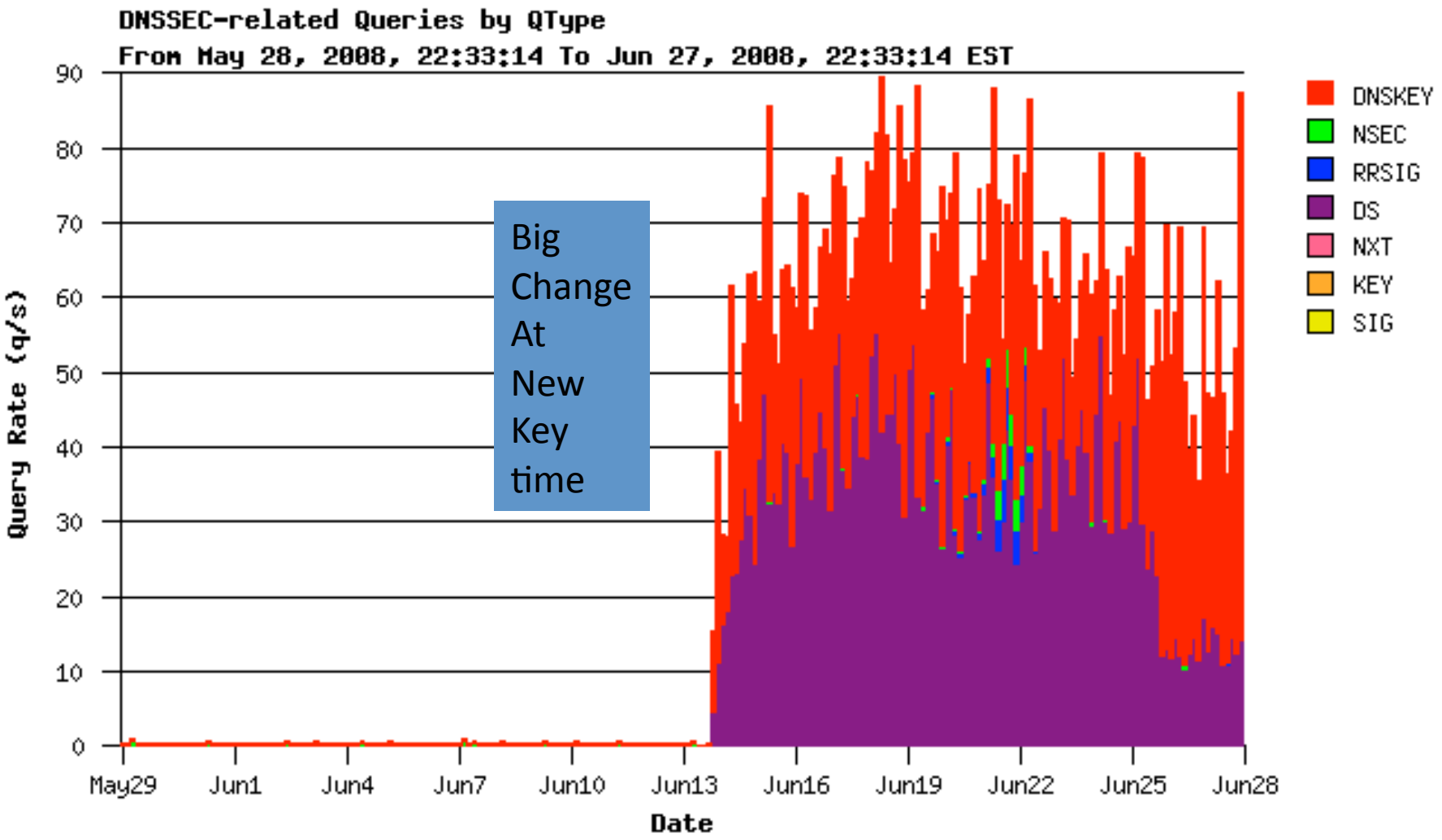
How about back in June 09?



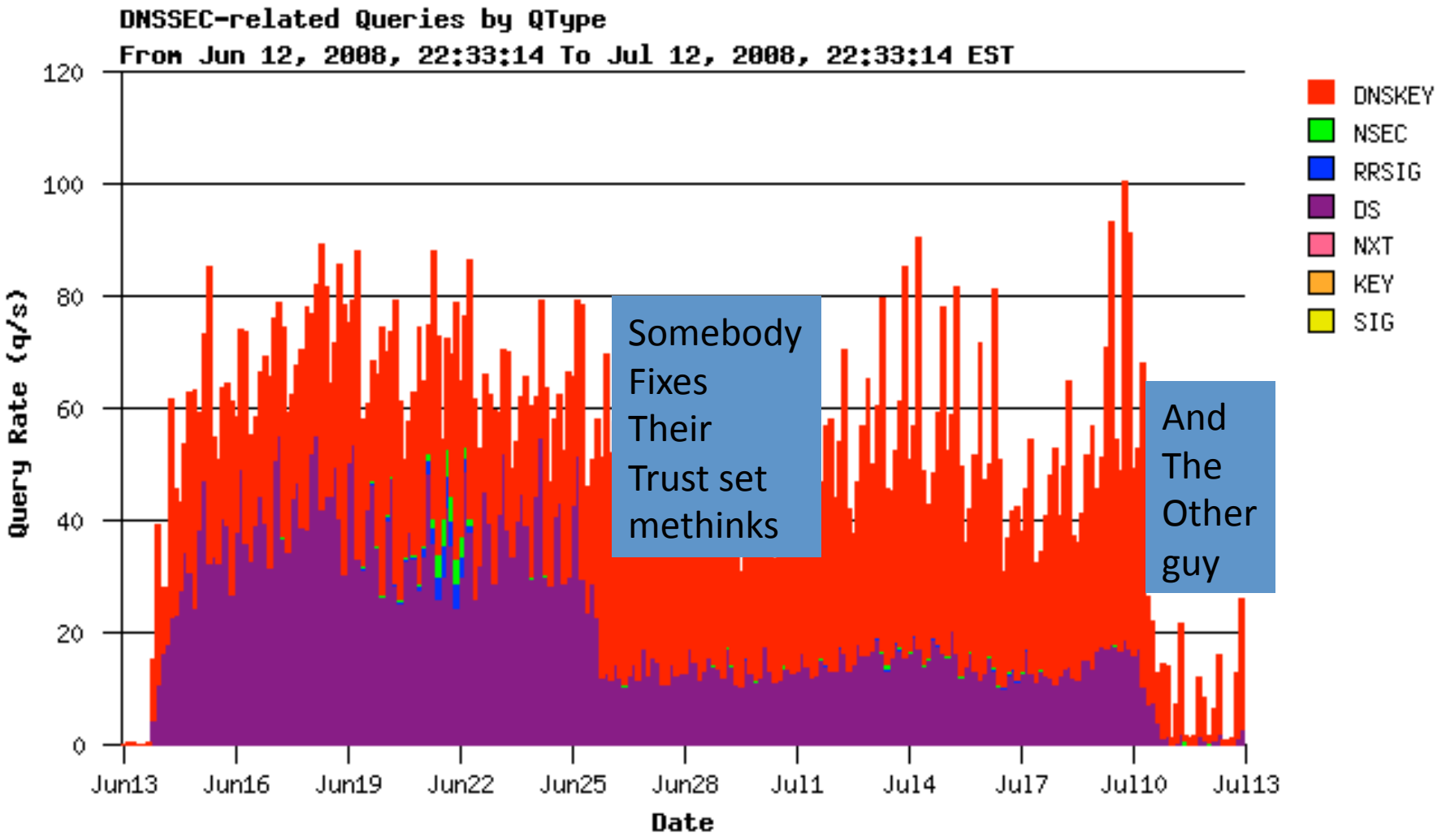
How about back in December 08?



How about back in June 08?



And the fix..


DNSSEC-related Queries by QType
From Jun 12, 2008, 22:33:14 To Jul 12, 2008, 22:33:14 EST
Query Rate (q/s)
120
100
80
60
40
20
0
Jun13
Jun16
Jun19
Jun22
Jun25
Jun28
Jul1
Jul4
Jul7
Jul10
Jul13
Date
DNSKEY
NSEC
RRSIG
DS
NXT
KEY
SIG
Somebody Fixes Their Trust set methinks
And The Other guy

Did we miss something?

- YES!
- Well, when the DNSSEC query load is down in the 10s and 20s its hard to tell what is signal and what is noise.
- But clearly, DSC did ‘see’ DNSSEC traffic at past key rollover events.
- And yes, it looks like large changes in DS/DNSKEY query pattern take place, at key rollover time
  - Hand-installed trust looks to be a problem

Hand installed trust?

- Fedora shipped with an ‘easy DNSSEC’ .rpm
- It enabled DNSSEC testing, with a set of keys for the RIPE-NCC
  - Keys which went stale when RIPE rolled keys…

Patrik asks a question

- “what if I configure validation under DURZ”?
  - DURZ key is invalid.
  - Installing any invalid key over root zone has same effect
- And performs same operation across root zone priming, from his bind/resolver instance.
- What does he see?

Root validation failure

boa$~>sudo tcpdump -n -i eth0 port 53 and src host 94.254.84.99 |grep DNSKEYtcpdump: verbose output suppressed, use -v or -vv for full protocol decodelistening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

06:54:30.955121 IP 94.254.84.99.44783 > 81.228.8.16.53: 1589% [1au] DNSKEY? se. (31)
06:54:30.958933 IP 94.254.84.99.44957 > 81.228.8.16.53: P 0:33(33) ack 1 win 78 <nop,nop,timestamp 428441300 3214623164>14744% [1au] DNSKEY? se. (31)06:54:31.027191 IP 94.254.84.99.1685 > 199.7.49.30.53: 25918% [1au] DNSKEY? se. (31)06:54:31.050182 IP 94.254.84.99.9095 > 192.36.144.107.53: 33074% [1au] DNSKEY? se. (31)06:54:31.051917 IP 94.254.84.99.27830 > 192.71.53.53.53: 47079% [1au] DNSKEY? se. (31)06:54:31.053896 IP 94.254.84.99.50673 > 194.146.106.22.53: 56424% [1au] DNSKEY? se. (31)06:54:31.055281 IP 94.254.84.99.50160 > 192.36.133.107.53: 56756% [1au] DNSKEY? se. (31)06:54:31.067474 IP 94.254.84.99.58543 > 199.254.63.1.53: 44551% [1au] DNSKEY? se. (31)06:54:31.096343 IP 94.254.84.99.62048 > 192.36.135.107.53: 530% [1au] DNSKEY? se. (31)06:54:31.103815 IP 94.254.84.99.51312 > 81.228.10.57.53: 57996% [1au] DNSKEY? se. (31)06:54:31.905008 IP 94.254.84.99.54482 > 81.228.8.16.53: 27498% [1au] DNSKEY? se. (31)06:54:32.715001 IP 94.254.84.99.28921 > 130.239.5.114.53: 10609% [1au] DNSKEY? se. (31)06:54:32.779093 IP 94.254.84.99.48402 > 81.228.10.57.53: 27809% [1au] DNSKEY? se. (31)06:54:33.685010 IP 94.254.84.99.60356 > 81.228.8.16.53: 45228% [1au] DNSKEY? se. (31)06:54:33.688613 IP 94.254.84.99.34477 > 81.228.8.16.53: P 0:33(33) ack 1 win 78 <nop,nop,timestamp 428441573 3213710601>24152% [1au] DNSKEY? se. (31)06:54:33.691431 IP 94.254.84.99.59347 > 81.228.10.57.53: 34220% [1au] DNSKEY? se. (31)06:54:35.295104 IP 94.254.84.99.58846 > 81.228.10.57.53: 6498% [1au] DNSKEY? se. (31)

Root validation failure

```
boa$~>sudo tcpdump -n -i eth0 port 53 and src host 94.254.84.99 |grep DNSKEYtcpdump: verbose
    output suppressed, use -v or -vv for full protocol decodelistening on eth0, link-type EN10MB
    (Ethernet), capture size 96 bytes
    06:54:30.955121 IP 94.254.84.99.44783 > 81.228.8.16.53: 1589% [1au] DNSKEY? se. (31)
    06:54:30.958933 IP 94.254.84.99.44957 > 81.228.8.16.53: P 0:33(33) ack 1 win 78
    <nop,nop,timestamp 428441300 3214623164>14744% [1au] DNSKEY? se. (31)06:54:31.027191 IP
```

Lots of DNSSEC related queries for .SE aimed at the root.

Endless amounts of DNSSEC in fact.

Why would the querier be doing this?

Things to notice

- Unlike normal resolver query path, it is not limited to the first-to-answer NS of the space
  - It promiscuously queries all of the NS it sees
    - Hunting a valid signature/keyset?
- Any subsequent query over the DNSSEC state of the 'query-root' (the parent domain of the subzone being requested) will re-trigger
  - Tested with BIND 9.5.0
    - (shipped with Ubuntu 8.10)
  - and BIND 9.7.0rc2.

The story continues..

- That's interesting..
  - "mis configured trust can cause large increases in DNS query load for large responses, to all NS of a zone"
- And its only below the root right?
  - Nope. It applies at any zone you enabled validation, if you mis-configure your trust set.

Just bind, right?

- Alas no.
  - Unbound also has a problem (Roy Arends)
    - Remembers for a minute, so queries under 1min delay are ‘cached’ per name
      - Should limit to a longer cachetime, but doesn’t appear to
    - But, with multiple failing names, each will be tested
      - And asks another 5 servers from a compiled in value

```
#define VAL_MAX_RESTART_COUNT 5
in unbound-1.4.1/validator/validator.h
```

- Not as bad as bind

What tickled this bug?

- All of:
  - Enabling validation
  - Having a signed zone to query into
  - Hand installed stale trust,
    - failure to track/update
  - Resolver-side caching behaviour
    - (or lack thereof)

Validation & Key rollover

- Normally, in key rollover, signature changes
  - And you re-acquire trust over the new key and the problem goes away
  - But if the resolver doesn't have its trust updated
    - Then this bug is tickled, and the resolver goes into a spin of endles DNSKEY fetches
- What if it's the priming query?
  - If it's the priming query, then *all* queries through the resolver will 'tickle' this problem
  - It should 'fail hard' but it appears, it doesn't
- Ie failure to follow root/trust key rollover by resolvers might cause a 'storm' of DNSKEY query on all NS in the chain to the invalid trustpoint.
  - You'd have to think we'd see this
  - But if you have more than one resolver, you might miss it because DNS will continue to work via the alternate nserver entry.

RFC5011 fixes this, right?

- Well, yes, for Bind 9.7 and later, this is cleaner
  - But what about the long tail?
  - ISC have just released patches for 9.4, 9.6 and 9.7 which address the heavy hitting query load, but only 9.7 has the RFC5011 behaviour.

How bad can it be?

- The further down in the validation chain, the worse it can get. Eg:
- test.example.com
  - 283024 (2*2*14*14*19*19) root DNSKEY requests to 19 root name servers,
  - 14896 (2*2*14*14*19) .com DS requests to 19 root name servers,
  - 784 (2*2*14*14) .com DNSKEY requests to 14 .com name servers,
  - 56 (2*2*14) example.com DS requests to 14 .com name servers,
  - 4 (2*2) example.com DNSKEY requests to 2 example.com name servers.

It continues..