IETF Lightning Talks

Aggressive-NSEC
EDNS-KeyTags
DNSEXTLANG

Warren Kumari

Aggressive use of NSEC/NSEC3

draft-ietf-dnsop-nsec-aggressiveuse

What’s the problem?

Not so much a problem as an opportunity…

- Name-server / DNS optimization
  - Increase performance / decreases latency
  - Decrease resource utilization on name-servers
  - Increase resilience to certain DoS attacks
- Improve privacy

What's the solution?

Deduce answers…

- DNSSEC provides authentication of both *positive* **and** *negative* answers
- Positive answers get a signature proving that they are valid; negative answers include a signature proving that the name doesn't exist
  - NSEC (Next SECure) records list the alphabetical records on each side of the non-existing name, and then signs that

Err, what?

Example…

```
wkumari$ dig +dnssec  belkin
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 41230
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6,
ADDITIONAL: 1
;; QUESTION SECTION:
;belkin.              IN  A
;; AUTHORITY SECTION:
.          1795   IN  SOA a.root-servers.net. nstld.verisign-grs.com.
2016070901 1800 900 604800 86400
beer.         21512  IN  NSEC    bentley. NS DS RRSIG NSEC
beer.         21512  IN  RRSIG  NSEC 8 1 86400 20160719170000
20160709160000 46551 . AoT2Oe3eVZ3pC1DousLXDYABGuTTvkyP4rbBXvquGp3T/
Lg7Rer3Vx2g oC9p5u6T+lj/3u879htWNRO62wSdODkvOdtVFA5iJxN9DJ5EtuJdbuL/
xJuPhoin+0Fc6Vtf0X0l7e5TBtxYAyPZqUq6dxm6qE/NW6Ft1nAv3GYX jlg=
;; Query time: 222 msec
```

Ok, so?…

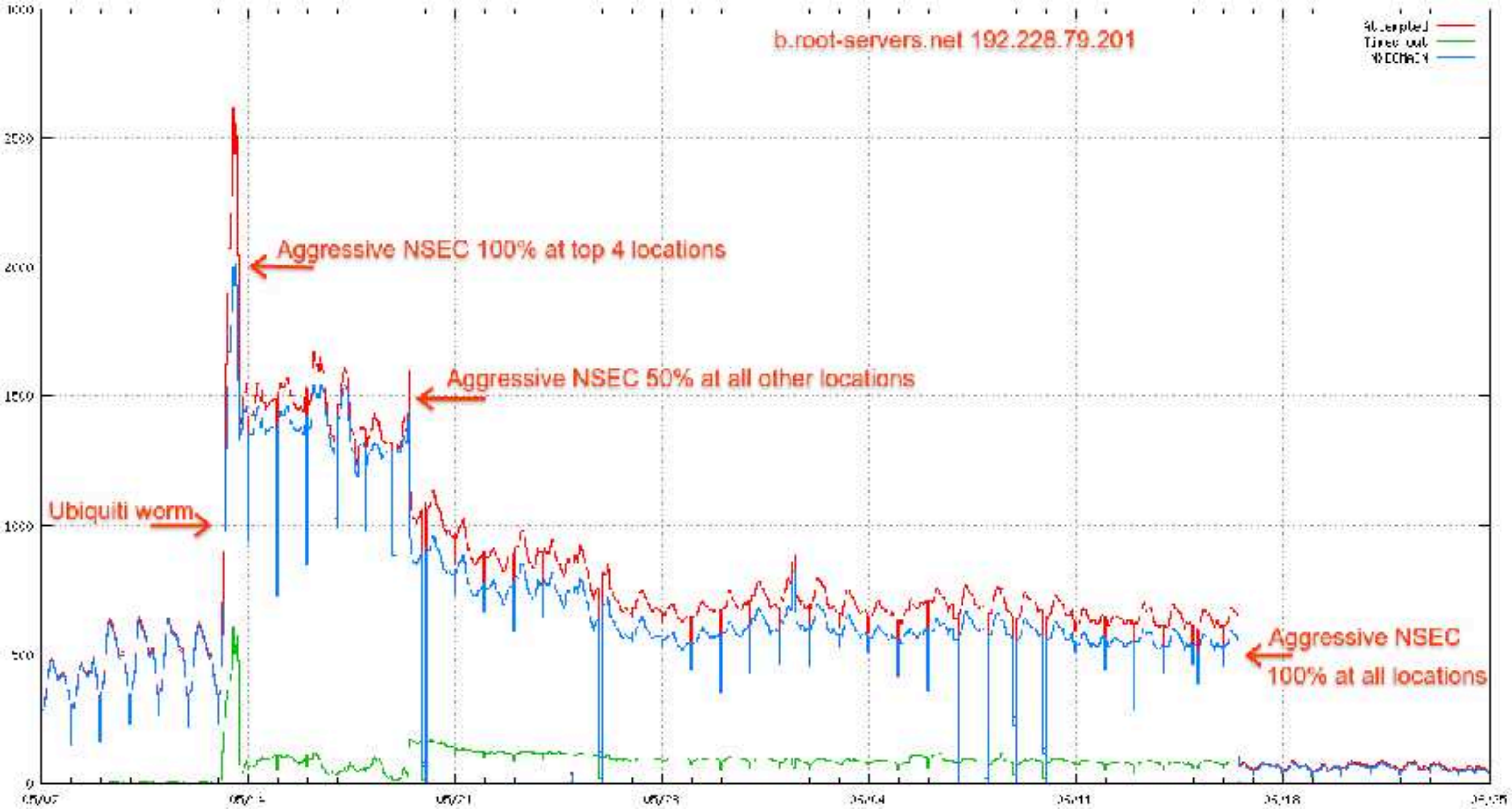
Deduce answers…

- Currently this NSEC is only used for the specific question
  - Like being told a shop only has Brie and Stilton in stock
    - but then asking about Edam, Camembert and Gouda
- This document allows name-servers to use the (signed) information in the NSEC record to synthesize answers … and to do the same thing for wildcards

Does this really help?

Yes… depends on what and where…

- Currently >60% of root answers are NXDOMAIN
  - Drops to <1%
- Random-subdomain DoS attacks



Ubiquiti worm

Aggressive NSEC 100% at top 4 locations

Aggressive NSEC 50% at all other locations

Aggressive NSEC 100% at all locations

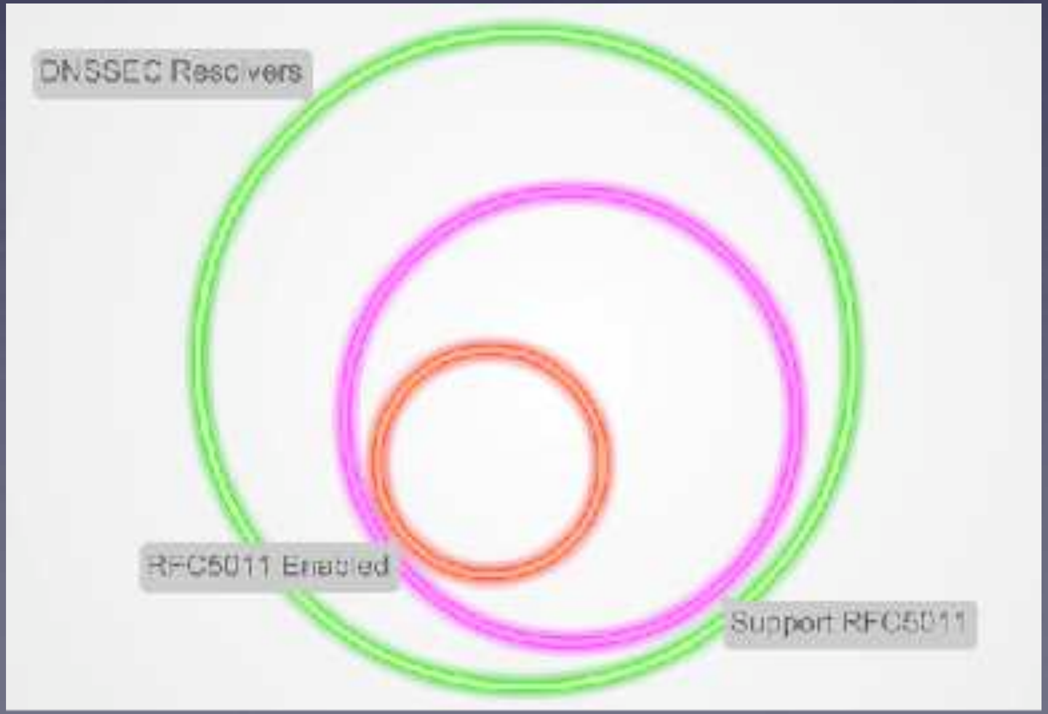
Questions?

Signaling Trust Anchor Knowledge in DNS Security Extensions

draft-ietf-dnsop-edns-key-tag

What's the problem?

- DNSSEC KSK (trust anchor) is rolling
  - https://www.icann.org/resources/pages/ksk-rollover
  - October 2017: New KSK used for signing
  - January 2018: Revocation of old KSK
- RFC 5011 — process for introducing the new key
  - Some (?) nameservers don't support RFC5011
  - Many do, but some (?) have RFC5011 disabled
- EDNS KeyTags provides a means to measure this



What’s the problem?

> Measuring before the actual key roll has proved to be challenging. The potential to signal whether a validating resolver that relies on a configured trust anchor for the root zone follows the implicit key roll signals defined in RFC 5011 has been the subject of further investigation the Design Team. The conclusion is that it is not possible to devise such a signal or test in the current environment. In other words, when a new KSK is published in the root zone, it is not possible to use a third-party measurement technique to determine which resolvers have picked up the new KSK, nor is it possible at this juncture to determine which resolvers have not picked up the new KSK. Two IETF Internet-draft documents[14] [15] propose to add explicit trust anchor signaling into the DNS specification. Either approach, if adopted, would add some further visibility to the situation, but would also complicate the analysis.
>
> [14] https://tools.ietf.org/html/draft-wkumari-dnsop-trust-management-01
>
> [15] https://tools.ietf.org/html/draft-wessels-edns-key-tag-00
>
> — ROOT ZONE KSK ROLLOVER PLAN, P 14

What’s the solution?

Measure…um..once, cut, errm, something..?

- Resolvers signal which KSK they know about
    - Signal “upstream” to root servers
        - Using special queries (`_ta-1984-4242`)
            - Example:
              `_ta-1984 → _ta-1984-4242 → _ta-4242`
        - and special options (EDNS option-code N)
- Allows the root servers to see who has which key, and predict what percentage, **and who** will break.

Yay, solved!….?

Yay….?!

Image by Pete Unseth (https://commons.wikimedia.org/wiki/User:Pete_unseth)
File: Glass half full kind of day.JPG CC BY-SA 3.0

What’s the solution?

More unknown unknowns…

- Deployments written before RFC5011 (Sept 2007) were written before EDNS KeyTags introduced
- Cannot measure who doesn’t do EDNS KeyTag
- Not useful for this KSK roll — but may be for the next one

Questions?