Root & TLD DNSSEC

Early Deployment Observations

Edward Lewis
ed.lewis@neustar.biz
NANOG 56
October 23, 2012

neustar.

Introduction

- DNSSEC is an extension to DNS to add some security to the system, "in the works" for almost 20 years
- To deploy DNSSEC, an operator has to set some parameters

- This effort began to determine the commonly chosen values so we could set our parameters
- The effort yield some insight into the transition of a protocol extension from engineering to operations

Ground Rule

- I stick to a rule of "name no names"
- Data is based on averages and smoothed to eliminate random outages that hide design choices
- Stories are related anonymously, to highlight underlying influences that are interesting

- There is no attempt to grade the actions of an operator because we all have different requirements to meet

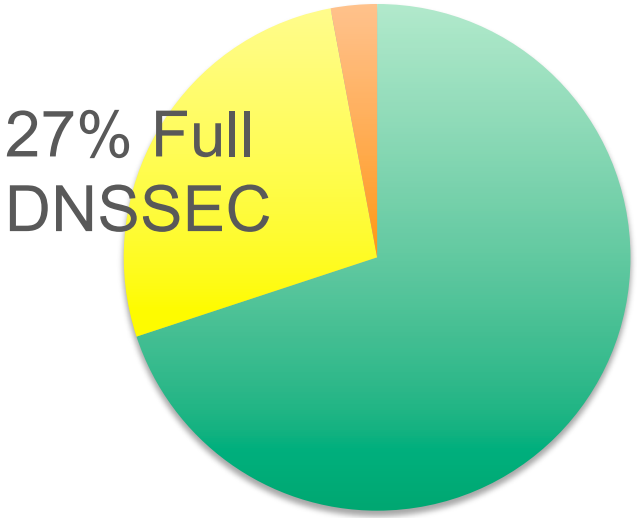
Expectation vs. Observation

- Protocol engineers began from a "how to make this secure" and used IETF workshops to bang DNSSEC into shape
- During this period (1999-2004) a number of expectations were set and appeared in RFC documents

- Besides the "trivia" of what DNSSEC parameters are, it's interesting to see how observed behaviors match up against the RFC documents and thus the expectations of the protocol engineers

Adoption in upper zones

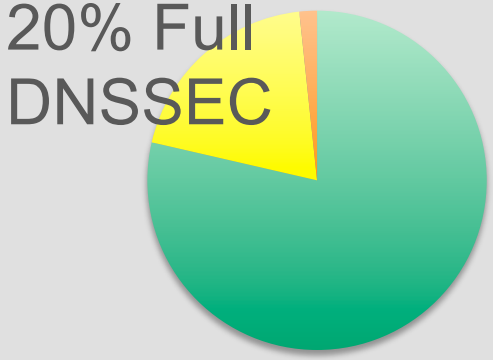
October 16, 2012
(306* total zones)



Remainder (small slice) have deployed but not "linked up"

* - ICANN operates 11 signed test TLDs which are not included

July 1, 2011



Deployment...slowly

- Adoption at the top of the DNS tree is greater than in the lower portions of the tree
  - Not too surprising, scaling in a hierarchical environment
- What is kind of interesting is that the adoption seems to be seasonal
  - Few zones take on DNSSEC during summer and winter
  - Most work is done in fall and spring
  - Perhaps coincidently, most DNS conferences are then too

  - ...while generating these slides a week before NANOG two more zones were signed...one as I was doing a final revision...

Parameters of DNSSEC

- Key Management (RFC 4641 and a-soon-to-be update)
  - Roles - what are particular keys used for?
  - Cryptography - algorithms, lengths
  - Lifetime - durations of key use, signatures, methods of change
- Building Trust Chain (RFC 4509)
  - The DS resource record
- Negative Answer Style (RFC 5155)
  - NSEC (plaintext) and NSEC3 (salted hash)
- And non-parameters
  - Message size

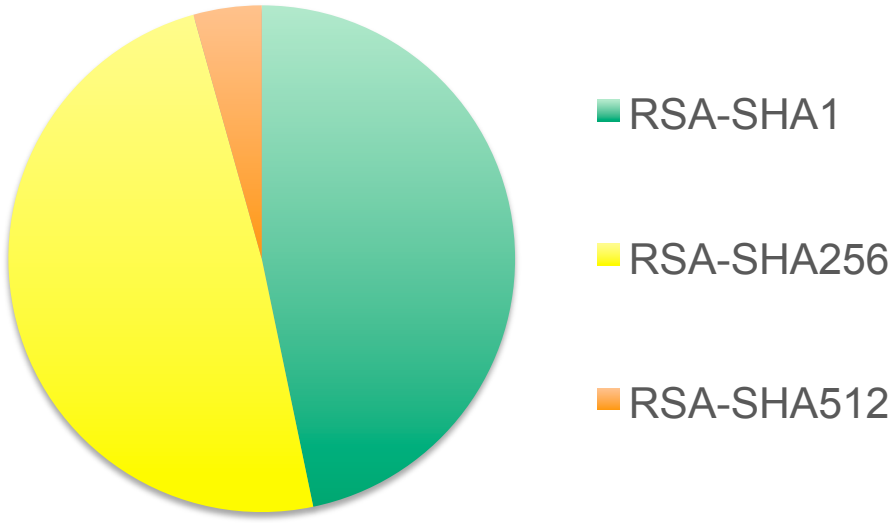
Key Roles

- DNSSEC usually employs two roles for keys
  - KSK - an "externally facing" key that is thought to be a pain to update and therefore has to be stronger
    - Longer key length, less frequent changes
  - ZSK - an "internal" key that is thought to be easy to change and therefore does not need to be as strong (as far as being 'cracked')
    - Shorter key length, more frequent changes
- These roles are optional, but all of the studied zones use these roles

Cryptography - Algorithms

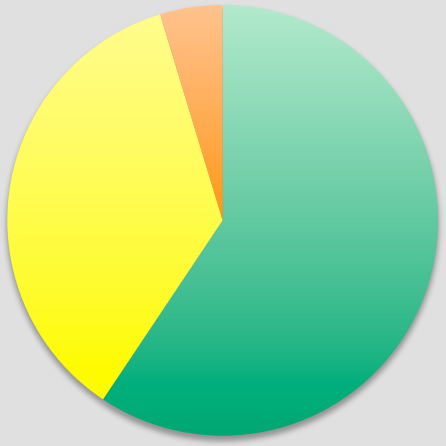
- Algorithms
  - During design, it was assumed various algorithms would be used to help interoperability
  - Operators use just one (at a time)
- Changes over time
  - After a number of operators began DNSSEC, a newer ("better?") algorithm was defined (RSA-SHA256)
  - Generally, operators beginning after the definition use RSA-SHA256
  - Of the operators predating the definition, only 2 have switched

Which Algorithm?



Older deployments use RSA/SHA1, newer tend to use RSA/SHA256

July 1, 2011

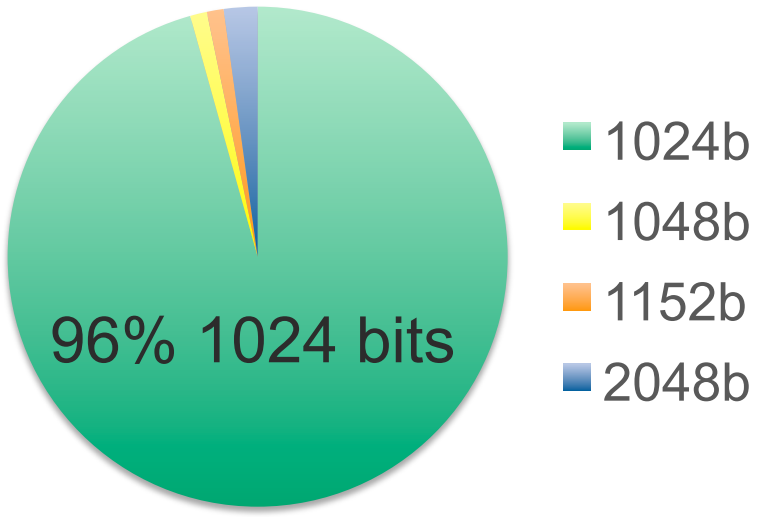


As of October 16, 2012: RSA-SHA256 (yellow) is in the majority.

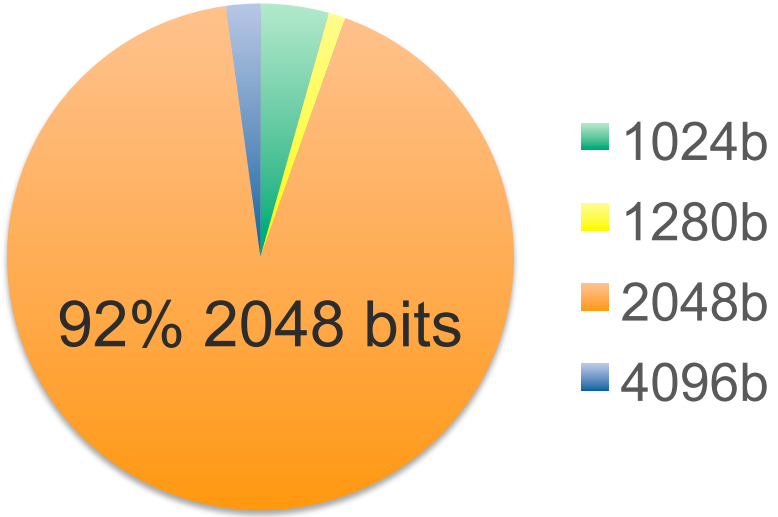
Key Lengths

» RFC 4641 "suggested" 1024 bits for ZSK, 2048 bits for KSK

ZSK

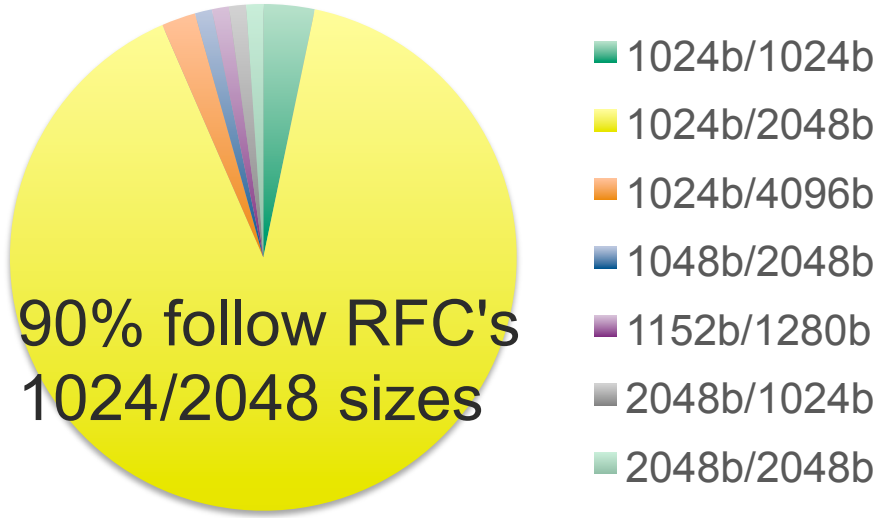


KSK



Key Combinations

- And to put a cap on this...looking and ZSK/KSK size pairings



- Only two zones do not use 1024-"or"-2048
  - I believe one is as a mistake, one did this intentionally
- Yes, a herd mentality when it comes to key lengths

A Story...

- It's pretty obvious most operators took the numbers from the RFC so one case stands out
  - One operator chooses key lengths to minimize the size of DNS responses, a rather important consideration - but only one operator!
- A story told to me from ***another*** operator: they commissioned a study to select key sizes and the study agreed with that one operator. But when the study was presented to a review committee the committee decided to "follow the herd."
- One sign that "better" guidance on crypto is needed!

Key Lifetimes, Durations

- RFC 4641 suggested monthly changes of the ZSK and annual changes of KSK
- Much later on crypto "experts" said DNSSEC might never need to change the keys unless there was an emergency
- Operators have said that they will change anyway to make sure they can change in an emergency
- Parameters to look at:
  - Frequency of ZSK changes
  - Frequency of KSK changes

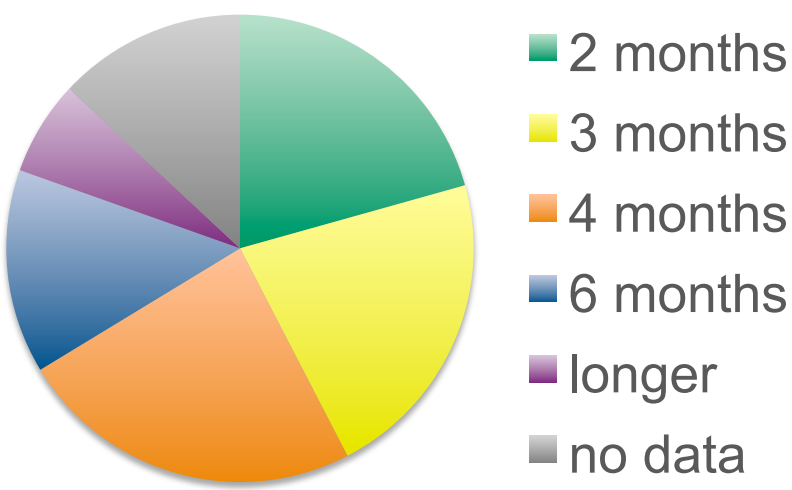
Frequency of changes

- Frequency of KSK changes are expected to be on the order of a year, but even with more than a year of data, there aren't enough KSKs to study. Only 4 have come and gone.

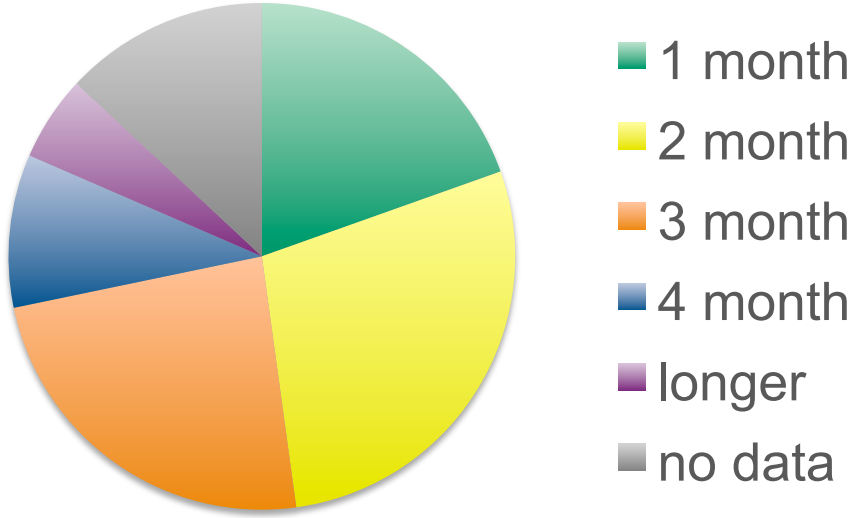
- For ZSK there are many examples of keys completing a lifecycle
  - How long is a ZSK seen? (Seen > In Use ... just because)
  - How long is a ZSK in use?
- Expectations from RFCs would be one month "in-use" and a about two months (or less) of total time "seen"

ZSK cycles

» Appears in Zone



» Key is "In Use"

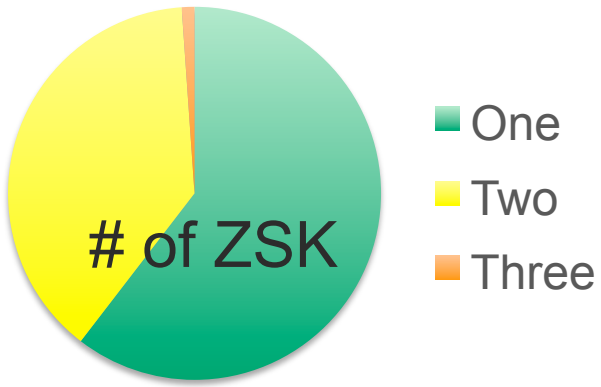


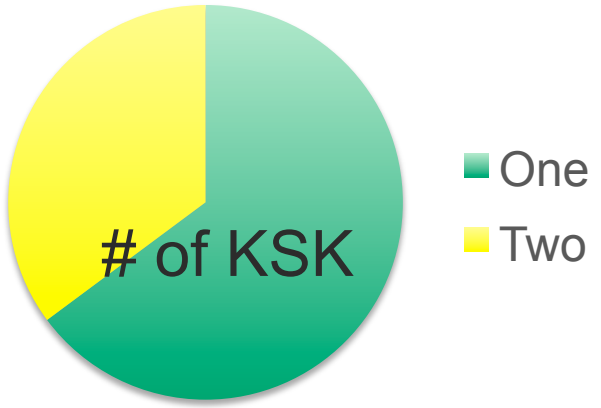
» Only 20% follow the assumptions in the RFC

» And there's a move to longer cycles in evidence

Response Sizes

- This is a space vs. time tradeoff
- Operators can choose to publish backup keys to speed time to recovery in an emergency, at the cost of larger messages
- If an operator uses no backup, the message size is less than 1000 bytes. With a backup, roughly up to 1700 bytes.
- Showing (average) counts of keys in a response:





Thoughts on This

- The size of a DNS response is a significant concern
  - Historically 512 byte limits were in firewalls/filtering devices
  - Large sized DNS responses are a tool of amplification attacks
  - And there's anxiety over UDP fragmentation

- In looking at the data over time there's one more important lesson
  - Operators are tinkering with the key management
  - There isn't a lot of experience with cryptography
  - There is a need to learn more here to help tune the system

Antithesis

- Some operators are not changing unless needed
- After a full year of (my) collecting data there were 6 TLDs that had not changed their keys at all
  - I chatted with one operator that had set up DNSSEC as "fire and forget" - with no problem and full knowledge of the situation
- Another operator "recycles" keys, meaning that once a key has run it's course, it isn't disposed it gets brought back
- Two examples of operators not following the expectations but no one has seen any trouble

Key Management Summary

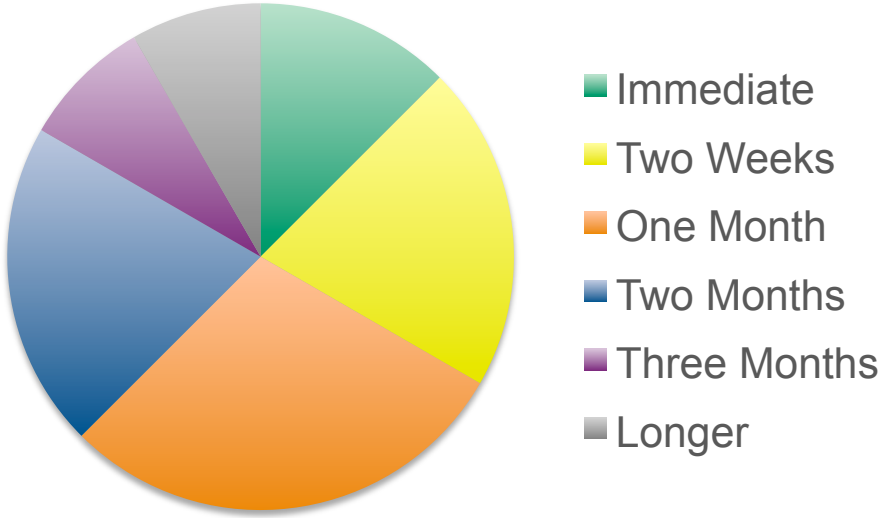
- The IETF issued RFC 4641 as it's guidance, NIST has documents too, and the IETF is preparing to issue an update to RFC 4641
- These documents are discussions and recommendations, not requirements nor qualify as compliance documents
- There are no universally accepted "best practices" or other studies in existence that give really good guidance yet
  - That would be nice for many folks, i.e., those writing RFPs, as well as operators

Building the trust chain

- For DNSSEC to scale, there has to be a chain of secure data from the root downwards to the TLDs and then to lower delegations
  - The vehicle for this is the Delegation Signer Resource Record
- How long it takes for the DS record to appear is interesting
- And RFC 4509 has a operational recommendation in it
  - The recommendation relates to backwards compatibility

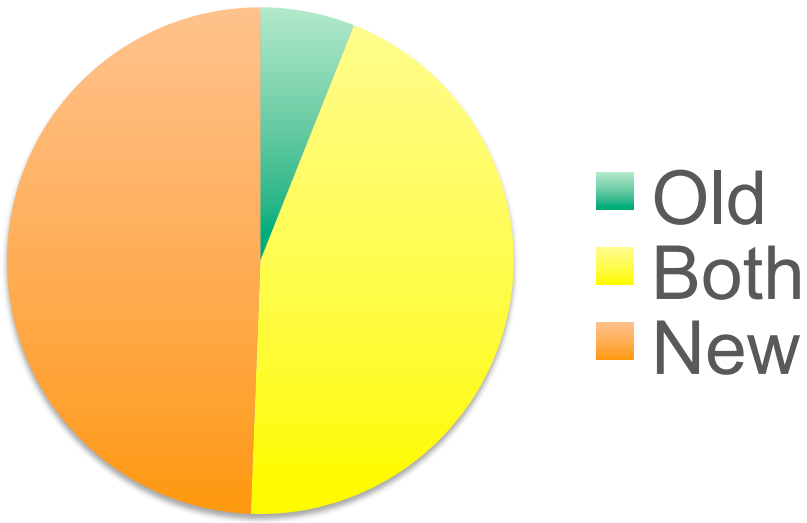
DS "Delay"

- Once a TLD zone initially signs, then next step it to "link up" to the root
  - Once the link up is made, "security matters"
  - The number of days that pass between these steps gives an measure of how aggressive the TLD has been



RFC 4509's recommendation

- The RFC says to publish both an "old" and "new" style DS until the time comes when the old is no longer needed
  - As if we could tell...;)
- Here's what operators do today:



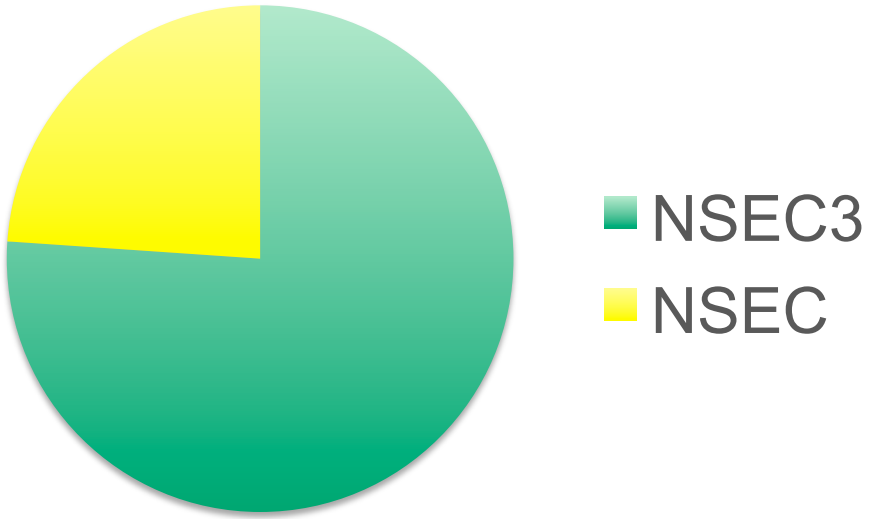
"Old (only)" and "New (only)" are incompatible - the burden is on clients to continue to support both old and new.

Impact

- DNS has been fairly good at rolling out new features with backwards compatibility in mind
- The weakness has been in know when to retire a phased-out value or feature
  - There are many "dead" RR types that are kept on the books because we can't tell if they are used somewhere and sometimes new RR types don't gain traction because of the backwards compatibility features
- DNS needs to come up with a way to have clients and other "mid-net" elements alert servers about their capabilities

"No": NSEC or NSEC3

- TLDs favor NSEC3 for two reasons (confirmed outside of the observations)
  - Opt-out of unsigned delegations (a size consideration)
  - Concerns about exposing the zone's list of names
- The choices made for a TLD might not be applicable elsewhere



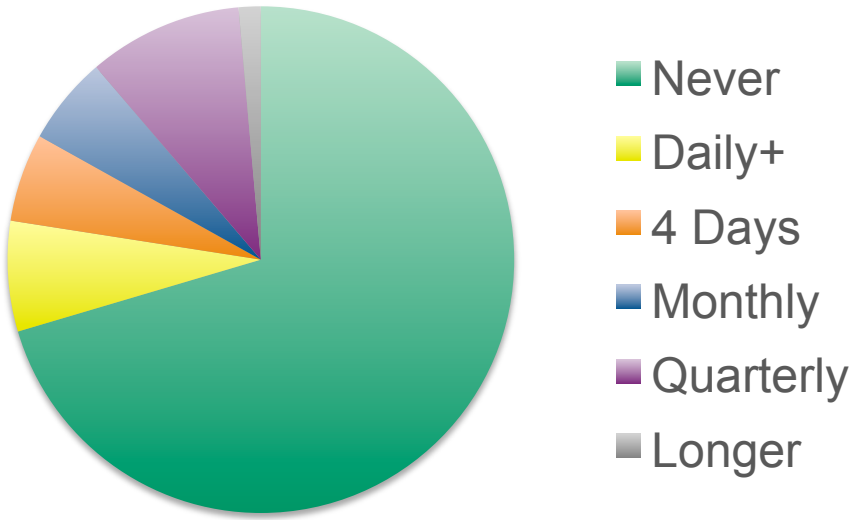
NSEC3 Recommendations

- Salt is a prefix added to the names to make reverse engineering them harder. The hashes are one-way but given the limited data space and other consideration a name could be discovered

- RFC 5155 recommends a new salt "every signing" but most zones don't batch "sign" anymore (it's all incremental)
- There is an iteration parameter, the number of times the hash algorithm is run
  - RFC 5155 say to make this low for better performance and does discuss upper effective limits

NSEC Salt Changes

- The changes of salt are RFC-supposed to be with every signing but here is the observed distribution



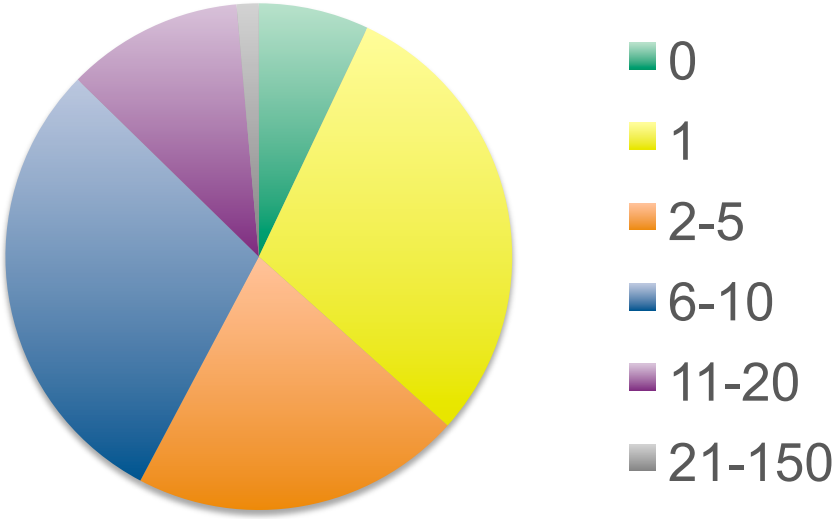
- Most zones don't bother to change
- At the other extreme, one changes every 10 minutes (the + in Daily)

NSEC3 Salt Values

- Rather insignificant but interesting is the value of the "salts"
- For genuine security, the salt would be a random value to eliminate predictability
- But here are some values seen, when I asked about them I got a few sheepish "oh, we forgot to change than from testing into production" or "we thought no one would notice"
  - BADFE11A
  - C0FFEE
  - BEEF
  - 5CA1AB1E
  - BA5EBA11

NSEC3 Iterations

- The RFC recommendation is to be a low number, with 150 being the absolute maximum that is workable
- Those choosing "0" are zones not making use of the hash (they "want" access a feature called opt-out) and one zone does use 150



- Number of iterations the hash function is run to convert plaintext names to NSEC3 owner names
- BTW, what's "low" ?

Summary/Final slide

- Not mentioned before - the role tools play
  - When asking some operators "why" the answer sometimes came down to "because the tool we used did it that way"
  - Algorithm change, as an example, isn't easy with many tools, so it isn't surprising it hasn't happened more
- Guidance on cryptography is needed
  - Seen in many measurements, the "herd mentality"
- DNS needs to be able to measure when old definitions are no longer needed
  - E.g., the DS hash example
- Operators avoid vacations and holidays when upgrading ;)