DNSSEC Operations

Choice of Signing Parameters

What choices do we have to make?

- Signing algorithm and key sizes
- NSEC or NSEC3
- Signature validity period
- Separate ZSK and KSK, or a single-purpose key
- Key rollover periods
- Delegation Signer hash algorithms
- Timers of many kinds

Software often offers defaults that are frequently reasonable, but it’s worth being able to have an opinion on that!

Signing Algorithm and Key Sizes

- Many algorithms have been standardized, and more will be standardized in the future
  - https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml
- Considerations of your choice include software support in signers, software support in validators, support in the surrounding machinery (e.g. HSMs) and cryptographic strength
- You can crowd-source your research into all of this by looking to see what your peers have done
  - E.g. survey algorithms in use by ICANN in the root zone, and by TLD operators
  - There might be little cryptographic benefit in using a new, strong algorithm that has questionable support in the real world if your parents are using an older algorithm

Signing Algorithm and Key Sizes

- Some algorithms require a corresponding choice of key size (e.g. RSA) but some do not (e.g. ECDSA)
  - CA/Browser Forum recommends/requires 2048-bit minimum for RSA
- Governments sometimes publish advice about the use of cryptosystems in general, and even specifically about DNSSEC
  - https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines
- SHA-1 as a hash algorithm has recently fallen out of favour
  - The SHAppening,
  - SHAttered
  - Birthday-Near Collision Attack

Signing Algorithm and Key Size

- Common choices for DNSSEC in 2020
  - Algorithm 8, RSA/SHA-256 with 2048-bit keys
  - Algorithm 13, ECDSA Curve P-256 with SHA-256
- As an aside, ORG is currently signed with algorithm 5
  - RSASHA1-NSEC3-SHA1
  - Choice was made back in 2009
  - Decreasing confidence in SHA-1 is prompting a refresh
  - PIR is planning an algorithm roll to algorithm 8 in 2020

NSEC or NSEC3

- Simple advice for this workshop
  - Use NSEC3 if you have requirements not to disclose the contents of your zone, but be aware that NSEC3 doesn’t provide great protection and other ways of enumerating your zone exist
  - Use NSEC3 if you need opt-out, e.g. because you are running a TLD registry and most of your customers don’t publish DS RRSets
  - If you do use NSEC3, research best
- Otherwise use NSEC
  - Easier to understand
  - Simple things are good and easier to troubleshoot
  - Avoiding opt-out makes zone size more predictable

Separate ZSK and KSK

- It is entirely plausible to use a single key to sign everything in your zone
  - No distinction necessary between ZSK and KSK
  - Looks like a KSK that is used as a ZSK
- However, many tools exist in a world where a ZSK/KSK split is conventional
  - By departing from the convention you might cause yourself headaches, trigger false alarms, exercise unusual code-paths and expose bugs
- There are operational differences too
  - Provisioning and management of the KSK involves a third-party

Key Rollover Periods

- Why change your keys?
  - Compromise (crypto, availability, disclosure, breach in chain of custody)
    - Roll as necessary, or as often as needed to manage confidence that there has been no compromise
  - Familiarity (execute key rollovers so that you know how to do them)
    - Roll regularly, as often as needed to make sure you have staff that know how to do it
    - … but be careful about running out of keys!
- ZSK rollovers can be well-automated
  - Some automation possible with BIND9, as we have seen
  - Better automation possible with other software
- KSK rollovers might be well-automatable in the future
  - E.g. CDS/CDNSKEY provides some automatable mechanisms for managing DS RRSets in parent zones

Delegation Signer Hash Algorithms

- Not all hash algorithms are supported by every registry
  - For domains whose parents are managed by such people
- Pragmatic choice today is to use SHA-256
  - Widely supported
  - Widely accepted
  - Not SHA-1
- There’s no dramatic harm in using SHA-1 today
  - But there is advice against it, and no strong reason to include it
- Remember you can publish multiple DS records corresponding to the same KSK, each with a different algorithm
  - So e.g. you could use SHA-256 and others (but perhaps not SHA-1)

Timers!

- DNS contains timers that trip people up
  - TTLs
  - SOA.RDATA timers for zone transfers (refresh, retry, expire)
- DNSSEC includes all of those timers plus excitingly many more
  - Signature validity periods (inception dates, expiry dates, jitter)
  - Pre-publish intervals for incoming keys and DS records
  - Hold intervals for outgoing keys and DS records
- All of these timers interact, and the timing can be complicated to manage by hand. Fortunately software is good at this.

References

- RFC 6781, “DNSSEC Operational Practices, Version 2”, O. Kolkman, W. Mekking, R. Gieban, Decmber 2012