Technical and Business Considerations for DNSSEC Deployment

Jim Reid
jim@rfc1035.com
Objectives

• Understanding the main technical & business impacts
• Are these helping or hindering DNSSEC deployment?
• What could or should be done to improve things?
TECHNICAL CONSIDERATIONS AND MYTHS

Zone size
CPU load
Traffic levels, need for rate-limiting & traffic shaping
Key rotation (rollover)
Tooling & debugging
Use cases
Zone & Cache Size

- Signed zones are typically 5-10 times larger than unsigned ones
  - Approximately 4 times as many resource records
  - RRSIGs are big
- So what?
  - Commodity RAM is cheap: ~$20 for 4GB of DDR3
  - Disk space is cheap too: 1TB costs ~$50
- Name server’s RAM footprint might be an issue
CPU Load Considerations

• 1024-bit RSASHA1 keys, 3GHz Xeon, 1 CPU
• Signing:
  • ~1800 signatures/second
• Validation:
  • ~24,000 validations/second
• Off-the-shelf hardware & software should be good enough most (or all?) of the time
Network Considerations

• DNSSEC requires EDNS0
  • DO bit, larger payloads, etc.
  • Want to avoid truncation in both DNS response and the underlying MTU for packets/frames

• Lots of broken stuff assumes DNS only goes over UDP and always has packets < 512 bytes
  • DNSSEC kills these false assumptions
  • Firewalls sometimes get misconfigured like this
  • CPE is notorious for getting this wrong too
DDoS Exposure

- DNSSEC is a vector for DDoS amplification attacks
  - 50-60 byte query typically generates 1-2KB of response
- Use Response Rate Limiting (RRL)
  - Provided in BIND9.10+ source distribution
    - Patches available for BIND9.9
  - Also in recent versions of other DNS implementations
- Traffic shaping might also help
  - Edge routers and/or kernels
Secure Hardware

- Hardware Security Modules (HSMs)
  - Tamper-proof hardware for storing keys
    - Expensive and concerns about managing lots of keys
    - Largely a niche solution for very important security-critical zones
  - Security protocols for managing access tokens
- HSM hardware isn’t really needed most of the time
  - HSM in software (OpenDNSSEC) might be good enough for a large registrar or small TLD registry
DNSSEC SIGNING CHOICES

The main choices and trade-offs to consider when deploying Secure DNS include:

- Which versions of DNSSEC to use
- Crypto algorithms
- Managing keys & signatures
- Monitoring & reporting
- Tools & tooling
- What functionality & UIs should customers see?
Key Lengths - 1

• How large should the key signing keys (KSK) and zone signing keys (ZSK) be?

• Obvious Goldilocks trade-offs:
  • Not too big or too small; just nice

• Don’t assume large keys are “stronger” than small ones or vice versa

• Could key size (or algorithms) be something to discuss with stakeholders?
  • Some might care (a lot), most probably won’t
Key Lengths - 2

• What’s the window for cryptanalysis of some key?

• Some rules of thumb:
  • Short-lived keys don’t need to be big
  • Long-lived keys should be reasonably big

• Suggestion:
  • Change a small ZSK once a week/month (maybe)
  • Change a large KSK once a year (maybe)

• Whatever’s done for the .tld zone or at the root should be more than good enough for your zones
Signature Duration

- Signature expiry intervals need careful thought too

- Obvious Goldilocks trade-offs again:
  - Not too short or too long; just nice
  - Short expiry => more CPU cycles for validators
  - Long expiry => “stale” signatures may cause validation failures when something has to be changed in a hurry

- Potential for cryptanalysis of long lived signatures?

- Might be best to start off by following what the parent zone does and adjust in light of experience
Key Rollover - I
Key Rollover - 2

• This should happen at regular, planned intervals
  • Might have to happen sooner in an emergency: ie when current key(s) are considered tainted/compromised

• Hard to get the choreography right

• Too many moving parts for KSK rollovers

• Signing tools are getting a lot better at managing key rollovers and signing policies in general
  • Metadata in BIND’s $K*.private$ files
  • OpenDNSSEC
Tooling

- Early signing and validating tools were clunky
- Much improved now but could be better still
  - Stronger focus on supporting local policies
  - Use obvious primitives that hide icky detail:
    - `pdnssec secure-zone/add-zone-key` in PowerDNS
    - Windows GUI-based DNS Manager
  - Essentially just click “Sign My Zone”
UI Considerations

• Most end users and administrators won’t want or need to see DNSSEC

  • How best to “hide” DNSSEC operations on the control panel or web site?

    • Perhaps just add a “click here to sign” button?

• Might be the end of the road for anyone managing DNS zone files with *emacs* or *vi* or *perl*

  • No user serviceable parts inside...
Monitoring & Reporting

• Add DNSSEC elements to name server monitoring and reporting systems
  • Check that zones get signed when expected (or not)
  • Look out for zones with close-to-expiring signatures
  • Monitor key rollovers closely
• Check that zone signing behaves as expected
• Are KSK changes getting notified to the parent?
  • Does the parent’s DS match the child’s KSK?
  • Does the parent publish new DS records in good time?
Key Management for Customer & End User Zones

- Who will generate/manage the keys for example.com, you or the customer?
- Risks and benefits in both approaches
  - Who gets blamed if something breaks?
  - Is this an extra (chargeable?) service?
  - Who has responsibility for choosing the keys, rotating them, revoking them, etc?
- What about customer support and helpdesk staff?
Validation Challenges

• Switching on validation breaks response rewriting
  • RPZ, content filtering, anti-abuse protection measures, NXDOMAIN rewriting, parental controls, etc.
• Government and law enforcement blacklists
• IPR takedown/block requests

• Clumsy workarounds
  • Forward “vanilla” queries to validating resolvers, send the dodgy ones elsewhere
  • Might be seen as needless extra complexity
What deployment barriers?

• The software and tools work reasonably well

• However it’s not always clear:
  • How to put them together and use/debug things
  • Define policies or understand the trade-offs

• Where are the white papers, case studies and business justifications?
  • “We switched on DNSSEC and….”
  • Are experiences at the root or a TLD registry or a big ISP meaningful for example.com?
Advice Needed!

- DNS administrators need guidance on DNSSEC deployment considerations and trade-offs:
  - Local policy choices on: key selection, signature duration, rollovers, negative trust anchors, etc.
- NIST report 800-81-2 is wonderful
  - Tough reading for non-experts — 130+ pages!
- Need something similar (and shorter) on business justifications
  - Convince the CEO that using DNSSEC is a Good Thing
Where’s the killer app?

• Not much software uses or needs DNSSEC today
  • Proof of concept web plug-ins mostly

• DANE could/should be the driver
  • Lookup certificates and other crypto material from the DNS
  • IETF’s ACME WG standards coming to browsers Real Soon Now
    • Let’s Encrypt certificates as an alternative to ones issued by conventional CAs
  • DANE support emerging in MTAs
Deployment Today - I

- Most of the important TLDs are signed
- Contractual requirement for new gTLDs
- Some European ccTLDs have 70%+ signed delegations
  - Registrar discounts rather than DNSSEC enthusiasm
- Very few of the busiest domain names are signed
  - Just paypal.com and nih.gov of the Alexa top 100
- The number of signed zones looks OK-ish quantitatively but not so good qualitatively
• Validation uptake is a lot healthier

• Around 15% of users world-wide depend on a validating resolver

• See https://stats.labs.apnic.net/dnssec

• Much of this is attributed to just three providers:
  • Comcast, google’s 8.8.8.8 and TeliaSonera

• This is great, but are they validating anything that actually matters?

• What are the other big ISPs doing?
Externalities

- For signers:
  - Why sign if almost nobody is thought to be validating?
  - What's the impact when/if someone else's validator fails?

- For validators:
  - Why validate if hardly anyone important signs their zones?
  - When someone else's signer screws up, you end up taking the hit(s) for their mistake(s). What will the impact(s) be?
    - i.e. A rival ISP's customers have no problem getting to badlysigned.com because that ISP doesn't validate while your customers can't because we are validating
Customer Support

• Training & education for helpdesk/support staff
  • Ditto for customers

• DNSSEC troubleshooting

• N-th level support staff will need training
  • How to tell the difference between a Secure DNS validation error SERVFAIL and a regular SERVFAIL
  • How to troubleshoot validation problems and fix or escalate them
  • Expired signatures, key mismatches, etc.
Documentation

• Make sure everything gets properly documented:
  • Design/architecture, deployment plans & roadmap
  • Policies (e.g. key sizes, rotation, signing interval, audit, etc.)
  • Write a DNSSEC Policy/Practice Statement - DPS
    • RFC6841 is a good starting point
  • DPS for the root zone is an excellent template:
    • https://www.iana.org/dnssec/icann-dps.txt
• Document DNSSEC tools and new processes
• Produce white papers and use cases for stakeholders?
Training

• How will your staff get trained?
  • Knowledge transfer from in-house and external experts
  • Commercial training courses, webinars, etc.

• Not just for your DNS team
  • Network operations & system administrators
  • Developers & helpdesk
  • Customer relations & support staff
  • Upper management
The “Last Mile” Issue

- AD header bit is set by the validating resolver
- Vulnerable to spoofing by an attacker
- Can the path between some stub resolver and its resolving server be trusted?
  - Windows uses IPsec to protect this
- If the local net isn’t considered secure, run a validating resolver on the edge device itself
- IETF’s DNS-over-TLS could be the answer
Validator Crunch Time!

- Major test is just around the corner
- First ever rollover of root zone’s KSK due Real Soon Now
- Validators which don’t support or use RFC5011 key rollover will almost certainly break
  - BIND configurations with trusted-keys{} statements
  - Old/clunky/buggy DNSSEC implementations
- IANA’s giving plenty of advance warning
- Nothing important should fail
  - Famous last words….
Negative Trust Anchors

• How to tell the validator that DNSSEC for some domain(s) are broken
  • Don't validate these domains for now, but carry on validating elsewhere
  • Provably insecure (and possibly bad) data might be better than nothing

• Features in BIND and unbound to support this
  • A necessary evil - unfortunately
  • Workarounds for other people's mistakes
DNSSEC Troubleshooting

• Can be very painful

• Debugging vanilla DNS problems is hard enough

• DNSSEC makes things even harder
  • Checking DNSSEC-related RRtypes, keeping track of key IDs & flags, algorithm numbers, etc.

• Tools are getting better, but still far too difficult for many DNS administrators
  • Error messages should be clearer: “you forgot to re-sign”, “that signature has the wrong hash value”, “there's no DS record for this KSK”, etc.
drill

• The best DNSSEC debugging tool by far
  • Also replicates dig’s commonly used functionality
  • Open source from NLnetLabs

• Can illustrate validation in action
  • Shows which keys (algorithms & key lengths) are used
  • Work on a single signature or top-down from the root

• Pinpoint stale keys & signatures

• Identify DS record and KSK mismatches

• drill -TD some-domain is just awesome!
delv

- ISC's answer to `drill`
- Distributed in BIND9.10+
- Command-line options almost identical to `dig`
- Not as chatty as `drill`
- Use the `+vtrace` option to see the validations
DNSVIZ

• A web-based DNS visualisation tool
  • Draws nice pictures
  • Can display details of revoked keys

• Visit dnsviz.net and type in your favourite signed domain name
  • Uses the web site's validating resolver, not yours
  • Doesn't check your validator's configuration
  • Can't check internal-only signed domains

• Download DNSVIZ source code and run it locally?
DNSVIZ Example

- Hover over elements to see details of the key, algorithm, TTL, key tags, etc
- Shaded elements are the KSKs
- Double circle around the trust anchor: the root's KSK
A Never Ending Task?

- Keeping DNSSEC software and tools up-to-date
  - Can you rely on your vendor/supplier?
- Crypto arms race
- Changing DNSSEC keys regularly
- Tweaking DNSSEC policies
- Interactions with parent zone(s)
- New operational problems and failure modes
- Customer service/support and helpdesk issues
QUESTIONS?