Multi Provider DNSSEC

draft-huque-dnsop-multi-provider-dnssec-02

Shumon Huque

March 22nd 2018

DNSOP Working Group, IETF101, London, U.K.

Note to the DNS Camel*

- This document does not propose any new extensions to the DNS protocol.
- It merely outlines operational deployment models for DNSSEC with multiple providers.

* https://datatracker.ietf.org/meeting/101/materials/slides-101-dnsop-sessa-the-dns-camel-01

Note to the DNS Camel*

- This document does not propose any new extensions to the DNS protocol.
- It merely outlines operational deployment models for DNSSEC with multiple providers.

*

https://datatracker.ietf.org/meeting/101/materials/slides-101-dnsop-sessa-the-dns-camel-01

Problem statement

- Many organizations employ the services of multiple DNS providers to distribute their authoritative DNS service.
- We want to successfully deploy DNSSEC in such an environment.
- Certain types of DNS configuration/features pose challenges.

Deployment models

- **Serve Only**
- **Sign and Serve**
- Inline Signing
- Hybrid

- Last two models are really variations/combinations of the first two, and are not ideal because they combine the weaknesses of both.

Serve Only

- Zone owner runs master server that signs zone data.
- Pushes out zone to multiple providers via DNS zone transfer.
- Providers serve the zone to the world.
- Zone owner holds signing keys: so managed DNS providers cannot serve false data, without detection by validating resolvers.

- Well understood model. Has been deployed in the field. Works.

Serve Only

- Zone owner runs master server that signs zone data.
- Pushes out zone to multiple providers via DNS zone transfer.
- Providers serve the zone to the world.
- Zone owner holds signing keys: so managed DNS providers cannot serve false data, without detection by validating resolvers.

- Well understood model. Has been deployed in the field. Works.

- **Notable limitation: doesn't work with non-standardized DNS features that are fairly widely used in the DNS industry today.**

Non-standard response mechanisms

- Sometimes called “Traffic management”:
  - Global Server Load Balancing, Probe and Failover records, custom scripted responses, etc.
- These types of responses are often querier-specific or dependent on inspecting dynamic state in the network
  - So answer and signature typically have to be determined at the authoritative server itself, at the time of the query, or both.

- **Also known by other colorful names**:
  - P. Vixie, “What the DNS is not”, acmqueue, November 2009
- DNS protocol purity vs. the reality of how extensively these mechanisms are already deployed.

Sign and Serve

- Each provider independently signs and serves zone data.
- Zone owner typically uses provider specific zone management API's to update zone content.

- This model presents some novel challenges, and is essentially the primary focus of this document.

Sign and Serve

- Can support the non-standard DNS features *if* the provider is capable of signing the response data generated by these features.
- Common strategies for doing so:
  - On-the-fly signing
  - Pre-compute & sign all possible response sets, and then algorithmically determine at query time which response + signature needs to be returned.

Sign and Serve

- Key requirement: manage the contents of the DNSKEY and DS RRsets such that validation is always possible, not matter which provider you query and obtain the response from.

- Strategy: each provider has to import the zone signing (public) keys of the other providers into their DNSKEY RRset.

Sign and Serve models

- Probably a range of possible models
- We focus on a small set (currently 2) that we've deemed to be operationally viable and palatable, based on discussion with actual managed DNS providers.
- Constraint: providers only want to directly interact with the zone owner and not with other providers (contractual reasons).
- Model descriptions assume 2 providers (but generalizable to more).

Model 1: Common KSK, Unique ZSK per prov

- Common KSK; Unique ZSK per provider.
- Zone Owner holds the KSK and manages the DS record.
- Each provider uses their own ZSK to sign zone data.
- Zone owner uses provider APIs to extract ZSKs, assemble them into a common DNSKEY RRset, signs it, and distributes it to the providers.
- Key rollovers need coordinated participation of the Zone Owner to update the DNSKEY RRset (KSK and ZSK) and DS RRset (KSK).

Model 2: Unique KSK & ZSK per provider

- Unique KSK and ZSK per provider.
- Each provider has their own KSK and ZSK.
- Zone Owner uses provider API to import the ZSKs of other providers into the DNSKEY RRset.
- DNSKEY RRset is independently signed by each provider's KSK.
- Zone Owner manages the DS RRset that includes both provider's KSKs.
- Key Rollovers need coordinated participation of the Zone Owner to update the DS (KSK) and DNSKEY (for ZSK).

Validating Resolver Behavior

- Read this section to understand some of the subtleties of this configuration and why ZSK cross sharing is needed to ensure that all answers are validatable.

Questions/Discussion/Feedback

- Is this a useful document for the DNSOP working group?
- If we ask for adoption, what category should be aimed for?
  - Informational? BCP? Something else?
- Are there other models that should be documented?
- For Sign and Serve, should we recommend one specific model?