Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 2, 2019

S. Huque
P. Aras
Salesforce
J. Dickinson
Sinodun
J. Vcelak
NS1

July 1, 2018

Multi Provider DNSSEC models
draft-huque-dnsop-multi-provider-dnssec-03

Abstract

Many enterprises today employ the service of multiple DNS providers to distribute their authoritative DNS service. Deploying DNSSEC in such an environment can have some challenges depending on the configuration and feature set in use. This document will present several deployment models that may be suitable.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2019.

Copyright Notice

Table of Contents

1. Introduction and Motivation

RFC EDITOR: PLEASE REMOVE THE FOLLOWING PARAGRAPH BEFORE PUBLISHING: The source for this draft is maintained in GitHub at: https://github.com/shuque/multi-provider-dnssec

Many enterprises today employ the service of multiple DNS providers to distribute their authoritative DNS service. Two providers are fairly typical and this allows the DNS service to survive a complete failure of any single provider. This document outlines some possible models of DNSSEC [RFC4033] [RFC4034] [RFC4035] deployment in such an

environment.

2. Deployment Models

The two main models discussed are (1) where the zone owner runs a master signing server and essentially treats the managed DNS providers as secondary servers, the "Serve Only" model, and (2) where the managed DNS providers each act like primary servers, signing data received from the zone owner and serving it out to DNS queriers, the "Sign and Serve" model. Inline signing and hybrid models are also briefly mentioned. A large part of this document discusses the Sign and Serve models, which present novel challenges and requirements.

2.1. Serve Only model

The most straightforward deployment model is one in which the zone owner runs a primary master DNS server, and manages the signing of zone data. The master server uses DNS zone transfer mechanisms (AXFR/IXFR) [RFC5936] [RFC1995] to distribute the signed zone to multiple DNS providers.

This is also arguably the most secure model because the zone owner holds the private signing keys. The managed DNS providers cannot serve bogus data (either maliciously or because of compromise of their systems) without detection by validating resolvers.

One notable limitation of this model is that it may not work with DNS authoritative server configurations that use certain non-standardized DNS features. Some of these features like DNS based Global Server Load Balancing (GSLB), dynamic failover pools, etc. rely on querier specific responses, or responses based on real-time state examination, and so, the answer and corresponding signature has to be determined at the authoritative server being queried, at the time of the query, or both. (If all possible answer sets for these features are known in advance, it would be possible to pre-compute these answer sets and signatures, but the DNS zone transfer protocol cannot be used to distinguish or transfer such data sets, or the rules used to select among the possible answers.)

2.2. Sign and Serve model

In this category of models, multiple providers each independently sign and serve the same zone. The zone owner typically uses provider-specific APIs to update zone content at each of the providers, and relies on the provider to perform signing of the data. A key requirement here is to manage the contents of the DNSKEY and DS RRset in such a way that validating resolvers always have a viable path to authenticate the DNSSEC signature chain no matter which provider they query and obtain responses from. This requirement is

achieved by having each provider import the Zone Signing Keys of all other providers into their DNSKEY RRsets.

These models can support DNSSEC even for the non-standard features mentioned previously, if the DNS providers have the capability of signing the response data generated by those features. Since these responses are often generated dynamically at query time, one method is for the provider to perform online signing (also known as on-the-fly signing). However, another possible approach is to pre-compute all the possible response sets and associated signatures and then algorithmically determine at query time which response set needs to be returned.

In the first two of these models, the function of coordinating the DNSKEY or DS RRset does not involve the providers communicating directly with each other, which they are unlikely to do since they typically have a contractual relationship only with the zone owner.

The following descriptions consider the case of two DNS providers, but the model is generalizable to any number.

2.2.1. Model 1: Common KSK, Unique ZSK per provider

- Zone owner holds the KSK, manages the DS record, and is responsible for signing the DNSKEY RRset and distributing the signed DNSKEY RRset to the providers.

- Each provider has their own ZSK which is used to sign data.

- Providers have an API that owner uses to query the ZSK public key, and insert a combined DNSKEY RRset that includes both ZSKs and the KSK, signed by the KSK.

- Key rollovers need coordinated participation of the zone owner to update the DNSKEY RRset (for KSK or ZSK), and the DS RRset (for KSK).

2.2.2. Model 2: Unique KSK and ZSK per provider

- Each provider has their own KSK and ZSK.

- Each provider offers an API that the Zone Owner uses to import the ZSK of the other provider into their DNSKEY RRset.

- DNSKEY RRset is signed independently by each provider using their own KSK.

- Zone Owner manages the DS RRset that includes both KSKs.

- Key rollovers need coordinated participation of the zone owner to update the DS RRset (for KSK), and the DNSKEY RRset (for ZSK).

2.2.3. Model 3: Shared KSK/ZSK Signing Keys

Other possible models could involve the KSK and/or ZSK signing keys shared across providers. Preliminary discussion with several providers has revealed that this is not a model they are comfortable with, again because they want to be independently responsible for securing the signing keys without involvement of other parties they don't have contractual relationships with. A possible way to mitigate this concern might be for the zone owner to operate a networked Hardware Security Module (HSM) which houses the shared signing keys and performs the signing operations. The signing instructions and results are communicated over a secure network channel between the provider and HSM. This could work, but may also pose performance bottlenecks, particularly for providers that perform on-the-fly signing. Due to open questions about the operational viability of this model, it is not discussed further.

2.3. Inline Signing model

In this model, the zone owner runs a master server but does not perform zone signing, instead pushing out the zone (typically via zone transfer mechanisms) to multiple providers, and relying on those providers to sign the zone data before serving them out. This model has to address the same set of requirements as the Sign-and-Serve model regarding managing the DNSKEY and DS RRsets. However, assuming standardized zone transfers mechanisms are being used to push out the zone to the providers, it likely also has the limitation that non-standardized DNS features cannot be supported or signed. This model is not discussed further.

2.4. Hybrid model

In the hybrid model, the zone owner uses one provider as the primary, operating in Sign and Serve mode. The other providers operate in Serve Only mode, i.e., they are configured as secondary servers, obtaining the signed zone from the primary provider using the DNS zone transfer protocol. This model suffers from the same limitations as the Serve-Only model. It additionally requires the signing keys to be held by the primary provider.

3. Signing Algorithm Considerations

In the Serve Only and Hybrid models, one entity (the Zone Owner in the former, and the primary provider in the latter) performs the signing and hence chooses the signing algorithm to be deployed. The

more interesting case is the Sign and Serve model (Section 2.2), where multiple providers independently sign zone data.

Ideally, the providers should be using a common signing algorithm (and common keysizes for algorithms that support variable key sizes). This ensures that the multiple providers have identical security postures and no provider is more vulnerable to cryptanalytic attack than the others.

It may however be possible to deploy a configuration where different providers use different signing algorithms. The main impediment is that current DNSSEC specifications require that if there are multiple algorithms in the DNSKEY RRset, then RRsets in the zone need to be signed with at least one DNSKEY of each algorithm, as described in RFC 4035 [RFC4035], Section 2.2. However RFC 6781 [RFC6781],

Section 4.1.4, also describes both a conservative and liberal interpretation of this requirement. When validating DNS resolvers follow the liberal approach, they do not expect that zone RRsets are signed by every signing algorithm in the DNSKEY RRset, and responses with single algorithm signatures can be validated corectly assuming a valid chain of trust exists. In fact, testing by the .BR Top Level domain for their planned algorithm rollover [BR-ROLLOVER], demonstrates that the liberal approach works.

4. Authenticated Denial Considerations

Authentiated denial of existence enables a resolver to validate that a record does not exist. For this purpose, an authoritative server presents in a response to the resolver special NSEC (Section 3.1.3 of [RFC4035]) or NSEC3 (Section 7.2 of [RFC5155]) records. The NSEC3 method enhances NSEC by providing opt-out for signing insecure delegations and also adds limited protection against zone enumeration attacks.

An authoritative server response carrying records for authenticated denial is always self-contained and the receiving resolver doesn't need to send additional queries to complete the denial proof data. For this reason, no rollover is needed when switching between NSEC and NSEC3 for a signed zone.

Since authenticated denial responses are self-contained, NSEC and NSEC3 can be used by different providers to serve the same zone. Doing so however defeats the protection against zone enumeration provided by NSEC3. A better configuration involves multiple providers using different authenticated denial of existence mechanisms that all provide zone enumeration defense, such as pre-computed NSEC3, NSEC3 White Lies [RFC7129], NSEC Black Lies [BLACKLIES], etc. Note however that having multiple providers

offering different authenticated denial mechanisms may impact how effectively resolvers are able to make use of the caching of negative responses.

4.1. Single Method

Usually, the NSEC and NSEC3 methods are used exclusively (i.e. the methods are not used at the same time by different servers). This

configuration is prefered because the behavior is well-defined and it's closest to the current operational practice.

4.2. Mixing Methods

Compliant resolvers should be able to serve zones when different authoritative servers for the same zone respond with different authentiated denial methods because this is normally observed when NSEC and NSEC3 are being switched or when NSEC3PARAM is updated.

Resolver software may be however designed to handle a single transition between two authenticated denial configurations more optimally than permanent setup with mixed authenticated denial methods. This could make caching on the resolver side less efficient and the authoritative servers may observe higher number of queries. This aspect should be considered especially in context of Aggresive Use of DNSSEC-Validated Cache [RFC8198].

In case all providers cannot be configured for a matching authentiated denial, it is advised to find lowest number of possible configurations possible across all used providers.

Note that NSEC3 configuration on all providers with different NSEC3PARAM values is considered a mixed setup.

5. Validating Resolver Behavior

From the point of view of the Validating Resolver, the Sign and Serve models (Section 2.2), that employ multiple providers signing the same zone data with distinct keys, are the most interesting. In these models, for each provider, the Zone Signing Keys of the other providers are imported into the DNSKEY RRset and the DNSKEY RRset is re-signed. If this is not done, the following situation can arise (assuming two providers A and B):

- The validating resolver follows a referral (delegation) to the zone in question.
- It retrieves the zone's DNSKEY RRset from one of provider A's nameservers.

- At some point in time, the resolver attempts to resolve a name in

the zone, while the DNSKEY RRset received from provider A is still viable in its cache.

- It queries one of provider B's nameservers to resolve the name, and obtains a response that is signed by provider B's ZSK, which it cannot authenticate because this ZSK is not present in its cached DNSKEY RRset for the zone that it received from provider A.

- The resolver will not accept this response. It may still be able to ultimately authenticate the name by querying other nameservers for the zone until it elicits a response from one of provider A's nameservers. But it has incurred the penalty of additional roundtrips with other nameservers, with the corresponding latency and processing costs. The exact number of additional roundtrips depends on details of the resolver's nameserver selection algorithm and the number of nameservers configured at provider B.

- It may also be the case that a resolver is unable to provide an authenticated response because it gave up after a certain number of retries or a certain amount of delay. Or that downstream clients of the resolver that originated the query timed out waiting for a response.

Zone owners will want to deploy a DNS service that responds as efficiently as possible with validatable answers only, and hence it is important that the DNSKEY RRset at each provider is maintained with the active ZSKs of all participating providers. This ensures that resolvers can validate a response no matter which provider's nameservers it came from.

Details of how the DNSKEY RRset itself is validated differs. In Sign and Serve model 1 (Section 2.2.1), one unique KSK managed by the Zone Owner signs an identical DNSKEY RRset deployed at each provider, and the signed DS record in the parent zone refers to this KSK. In Sign and Serve model 2 (Section 2.2.2), each provider has a distinct KSK and signs the DNSKEY RRset with it. The Zone Owner deploys a DS RRset at the parent zone that contains multiple DS records, each referring to a distinct provider's KSK. Hence it does not matter which provider's nameservers the resolver obtains the DNSKEY RRset from, the signed DS record in each model can authenticate the associated KSK.

6. Key Rollover Considerations

The Sign-and-Serve (Section 2.2) models introduce some new requirements for DNSSEC key rollovers. Since this process necessarily involves co-ordinated actions on the part of providers

and the Zone Owner, one reasonable strategy is for the Zone Owner to initiate key rollover operations. But other operationally plausible models may also suit, such as a DNS provider initiating a key rollover and signaling their intent to the Zone Owner in some manner.

The descriptions in this section assume that KSK rollovers employ the commonly used Double Signature KSK Rollover Method, and that ZSK rollovers employ the Pre-Publish ZSK Rollover Method, as described in detail in [RFC6781]. With minor modifications, they can also be easily adapted to other models, such as Double DS KSK Rollover or Double Signature ZSK rollover, if desired.

6.1. Model 1: Common KSK, Unique ZSK per provider

- Key Signing Key Rollover: In this model, the two managed DNS providers share a common KSK which is held by the Zone Owner. To initiate the rollover, the Zone Owner generates a new KSK and obtains the DNSKEY RRset of each DNS provider using their respective APIs. The new KSK is added to each provider's DNSKEY RRset and the RRset is re-signed with both the new and the old KSK. This new DNSKEY RRset is then transferred to each provider. The Zone Owner then updates the DS RRset in the parent zone to point to the new KSK, and after the necessary DS record TTL period has expired, proceeds with updating the DNSKEY RRSet to remove the old KSK.

- Zone Signing Key Rollover: In this model, each DNS provider has separate Zone Signing Keys. Each provider can choose to roll their ZSK independently by co-ordinating with the Zone Owner. Provider A would generate a new ZSK and communicate their intent to perform a rollover (note that Provider A cannot immediately insert this new ZSK into their DNSKEY RRset because the RRset has to be signed by the Zone Owner). The Zone Owner obtains the new ZSK from Provider A. It then obtains the current DNSKEY RRset from each provider (including Provider A), inserts the new ZSK into each DNSKEY RRset, re-signs the DNSKEY RRset, and sends it back to each provider for deployment via their respective key management APIs. Once the necessary time period is elapsed (i.e. all zone data has been re-signed by the new ZSK and propagated to all authoritative servers for the zone, plus the maximum zone TTL value of any of the data in the zone signed by the old ZSK), Provider A and the zone owner can initiate the next phase of removing the old ZSK.

6.2. Model 2: Unique KSK and ZSK per provider

- Key Signing Key Rollover: In Model 2, each managed DNS provider has their own KSK. A KSK roll for provider A does not require any change in the DNSKEY RRset of provider B, but does require co-ordination with the Zone Owner in order to get the DS record set in the parent zone updated. The KSK roll starts with Provider A generating a new KSK and including it in their DNSKEY RRSet. The DNSKey RRset would then be signed by both the new and old KSK. The new KSK is communicated to the Zone Owner, after which the Zone Owner updates the DS RRset to replace the DS record for the old KSK with a DS record for the new ZSK. After the necessary DS RRset TTL period has elapsed, the old KSK can be removed from provider A's DNSKEY RRset.

- Zone Signing Key Rollover: In Model 2, each managed DNS provider has their own ZSK. The ZSK roll for provider A would start with them generating new ZSK and including it in their DNSKEY RRset and re-signing the new DNSKEY RRset with their KSK. The new ZSK of provider A would then be communicated to the Zone Owner, who will initiate the process of importing this ZSK into the DNSKEY RRsets of the other providers, using their respective APIs. Once the necessary Pre-Publish key rollover time periods have elapsed, provider A and the Zone Owner can initiate the process of removing the old ZSK from the DNSKEY RRset of all providers.

7. IANA Considerations

This document includes no request to IANA.

8. Security Considerations

[TBD]

9. Acknowledgments

This document benefited from discussions with and review from Duane Wessels and David Blacka.

10. References

10.1. Normative References

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, DOI 10.17487/RFC6781, December 2012, <https://www.rfc-editor.org/info/rfc6781>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/info/rfc8198>.

10.2. Informative References

[BLACKLIES]
Valsorda, F. and O. Gudmundsson, "Compact DNSSEC Denial of Existence or Black Lies", <https://tools.ietf.org/html/draft-valsorda-dnsop-black-lies>.

[BR-ROLLOVER]
Neves, F., ".br DNSSEC Algorithm Rollover Update", in ICANN 62 DNSSEC Workshop, June 2018, <https://static.ptbl.co/static/attachments/179548/1529933472.pdf>.

[RFC7129] Gieben, R. and W. Mekking, "Authenticated Denial of Existence in the DNS", RFC 7129, DOI 10.17487/RFC7129, February 2014, <https://www.rfc-editor.org/info/rfc7129>.

Authors' Addresses

Shumon Huque
Salesforce

Email: shuque@gmail.com

Pallavi Aras
Salesforce

Email: paras@salesforce.com

John Dickinson
Sinodun

Email: jad@sinodun.com

Jan Vcelak

NS1

Email: jvcelak@ns1.com