Workgroup: DNSOP Working Group
Internet-Draft:
draft-johani-tld-zone-pipeline-00
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023

Authors: J. Stenstam J. Schlyter
The Swedish Internet Foundation Kirei AB

TLD Zone Pipeline: Requirements And Design Principles

Abstract

Today most TLD registries publish DNSSEC signed zones. The sequence of steps from generating the unsigned zone, via DNSSEC signing and various types of verification is referred to as the "zone pipeline".

The robustness and correctness of the zone pipeline is of crucial importance and the zone pipeline is one of the most critical parts of the operations of a TLD registry.

After a serious incident in 2022, the .SE Registry decided to re-evaluate the requirements on the zone pipeline. This has led to several new design choices and a decision to create a more robust implementation from scratch.

The goal of this document is to describe the requirements that the .SE Registry choose in preparation for the implementation of the new zone pipeline. The document also describes some of the design consequences that follow from the requirements. Hence this document is intended to work as a guide for understanding the actual implementation, which is planned to be released as open source.

TO BE REMOVED: This document is being collaborated on in Github at: https://github.com/johanix/draft-johani-tld-zone-pipeline. The most recent working version of the document, open issues, etc. should all be available there. The authors (gratefully) accept pull requests.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/johanix/draft-johani-tld-zone-pipeline.

Status of This Memo

This Internet-Draft will expire on 14 September 2023.

Copyright Notice

Table of Contents

1. Introduction

Today most TLD registries publish DNSSEC signed zones. This typically leads to a zone production "pipeline" that consists of several steps, including generation of the unsigned zone, signing of the zone and various types of verifications that the zone is correct before publication on the Internet.

In some cases, including the .SE Registry, the zone pipeline was not the result of a clear requirements process, nor was it the result of a concious design and implementation. Rather, it was the result of combining various tools in a mostly ad-hoc way that achieved the goal of moving the zone via signing and verifications towards publication.

When a critical part of the operation of a TLD registry is the result of an ad-hoc process there are likely to be hidden risks. That was the case for the .SE registry, which had a serious incident in February 2022. Because of that incident, .SE decided to re-evaluate the requirements on the zone pipeline and then create a more robust implementation from scratch.

This document aims to describe the requirements for zone production (also known as "zone pipeline") that the .SE Registry choose in preparation for the implementation of the new zone pipeline. It is developed for the needs of the .SE and .NU TLD Registries, but the conclusions are intended to be generally applicable.

1.1. Requirements Notation

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Purpose

A TLD Registry has a total responsibility towards society and the Internet community to ensure, at any given time, public access to correct versions of the DNS zones under their management. In order to meet this commitment, three components are essentially required:

*Generation of unsigned zones from the Registry System.

*Quality control and signing of the zones.

*Publication of resulting signed zones.

The first step is handled by the Registry System. The third step is handled by a combination of external providers of authoritative DNS handledservice and in-house DNS service. Both of these steps are out-of-scope for this document

The sole purpose of this document is to provide a correct set of requirements for the second step, between zone generation in the Registry System and zone publication on the public Internet.

3. Terminology

Well lit Software or other system that is successfully used in production, similar to our needs, in a large number of other places around the world.

Upstream Further up in the zone pipeline, i.e. in the direction of the Registry System.

Downstream Further down the zone pipeline, i.e. towards the public Internet.

4. Basic Design Principles

A number of fundamental principles are defined for the design of the system. The intention of these principles is to minimize the risk that the zone data (as generated by the Registry System) can somehow change at any stage through the zone pipeline.

*Critical path for zone data must be via proven and well-reviewed standard software. This critical path is called the "zone pipeline".

Rationale: Using well-established software used by others in the industry reduces development needs for the Registry. By not being critically dependent on self-developed software, the dependence on individuals is reduced.

*Standardized protocols shall be used as far as possible.

Rationale: Individual components must be replaceable as easily as possible.

*Consequences of inaccuracies in custom software must be limited as far as possible and must never affect published zone data.

Rationale: Obvious opportunities for risk minimization of a critical system within the business.

*Verification, signing and publication of the zone must be able to take place independently and without dependence on infrastructure outside the operating facility.

Rationale: The ability to always maintain and publish an updated zone is the most important responsibility of the Registry. To ensure the ability to always maintain this ability the zone production must be self-contained.

5. Interface to the Registry

Interface to the Registry System must be done according to standardized protocols. This requirement has the following consequences:

*Zone data must be retrieved from the Registry System using AXFR and IXFR [RFC5936].

*The request for publication of new zone data must be notified with DNS NOTIFY [RFC1995].

*Zone data published by the Registry System must contain a checksum in the form of ZONEMD [RFC8976].

6. Local Updates

During normal operation, no changes to the zone data retrieved from the Registry may take place. However, there may be situations where the Registry is not reachable (nor is it expected to be reachable within a reasonable time) and where local updates of zone data must be able to be carried out. This can for example,. be redirection of socially important infrastructure.

In a crisis situation (emergency operation), zone updates must be able to take place locally. Updates that take place in this way are introduced into regular systems as soon as possible. Return to normal operation may only take place after all changes made during emergency operation have been introduced into the regular system.

Local updates must be applied using a strict and traceable method. It must be clear at all times whether local updates have been applied, what these are and who requested them.

In cases where local updates have taken place, ZONEMD must be updated.

N.B. Local updates are an extraordinary measure and must not be used except in emergency situations. Procedures for who may request these are decided by the Internet foundation's crisis management.

7. Ingress Verification

Before signing, a number of checks must be performed on the zone contents. The reason why checking must take place before signing is to ensure that the zone being signed is always correct and can thus continue to be re-signed in the event of problems upstream. The exact checks to be carried out are set out in a separate specification and are subject to local policy.

7.1. Requirements on ingress verification

*The ingress verification must prevent an updated incorrect zone from being signed. An already approved previous version of the zone must continue to be signed until a new zone is approved.

*New zone controls must be able to be added without the component for ingress verification of zone data needing to be redesigned

*The interface from the zone pipeline to the ingress verification function must be DNS AXFR. This means that all controls must logically sit to the side and not be part of the critical path. I.e. the verification code is not part of the zone pipeline.

*All zone controls, self-developed or imported, must have local ownership within the organization.

7.2. Examples of ingress verification checks:

*Check that the zone data is complete.

*Check that delegation information for the zone itself is correct.

*Check that the delta (i.e. the difference) between the current zone and the previous version is within approved limits.

*Check that certain crucial records are present and correct (the zone's SOA record is one example).

8. Signing

The task of the signing step is to keep an approved and received zone signed for an arbitrary length of time until a new zone is received from upstream (i.e. from Registry via Ingress Verification).

8.1. Key management

The following requirements apply to the management of cryptographic keys for signing zone data:

- The key material must be stored and used in an HSM that meets the security requirements set by the CISO.

- The interface for accessing the key material shall be PKCS#11.

- All keys must, to facilitate replication between different signing entities, be generated in advance.In order to facilitate replication between different signing entities, all keys must be generated in advance.

- Exchange of KSK can be initiated automatically or manually, and is automatically terminated when the DS record in the parent zone is updated (after according to an appropriate safety margin).

- Changing the KSK may only be completed if the DS record in the parent zone is updated.

- Replacement of ZSK must be done automatically.

8.2. Zone signing

The following requirements apply to signing zone data:

- Signing must be done using key material via PKCS#11.

- The signing function must support algorithm rollover, e.g,. from RSA/SHA-256 to ECC/P-256/SHA-256.

- Signing must be done with either NSEC or NSEC3 semantics.

- If NSEC3 salt is used, the salt must be periodically changed automatically.

- Change of DNSSEC signing semantics from NSEC to NSEC3 and vice versa must be possible automatically.

- Previous signatures that are valid within a configurable time window shall be reused when re-signing, in order to reduce the rate of change on the zone.

- The signing function shall recreate the ZONEMD RR per [RFC8976] after the zone has been signed.

9. Egress Verification

After signing, several checks must be performed on the zone contents. Apart from the obvious validation of generated DNSSEC signatures it is also important to ensure that the signing step only added DNSSEC- information without in any way modifying the unsigned data.

9.1. Requirements on egress verification

- All generated DNSSEC signatures (RRSIG records) must be validated.

- The NSEC (or NSEC3) chain must be verified to be complete.

- The non-DNSSEC content of the signed zone must be provably identical to the corresponding unsigned zone that entered the signing step.

10. Distribution

The following requirements apply to distribution of the signed zone:

- The signed zone must be retrieved from signing and egress verification to the distribution points with AXFR (not IXFR).

- The signed zone shall be distributed to the designated authoritative name server services using AXFR/IXFR.

- To reduce convergence time towards the public Internet, the signed zone must be distributed with IXFR as far as possible.

- At least two complete zone publishing chains must be operational and always active.

- Choice of zone publishing chain is an active configuration choice in each distribution point and must always be the same for all distribution points.

- The signed zone from the selected zone publishing chain must be retrieved by all distribution points in all operating facilities.

11. Resulting Design Consequences

- The requirement on being able to prove that no unsigned data has been modified during signing is most efficiently fullfilled by computing the ZONEMD checksum on the unsigned data after signing (i.e. the signed zone modulo the DNSSEC related records DNSKEY, RRSIG. NSEC, NSEC3, NSEC3PARAM, apex CDS and CDNSKEY) and comparing that to the ZONEMD checksum for the corresponding unsigned zone.

- The ZONEMD checksums need to be stored outside the zone pipeline, indexed by the unsigned zone that each checksum corresponds to.

- The signed zone may (and will) change the SOA Serial independently of the unsigned zone. For this reason the ZONEMD checksums can not be stored using the SOA Serial as the index. Therefore a separate, unique, identifier is attached to each new version of the zone as a TXT record. A UUID is used as the identifier.

- Each unsigned zone MUST have a ZONEMD and a UUID index to store it under. Therefore changes to the unsigned zone via the local update facility must update the UUID in addition to any other change that is executed.

- The Registry runs multiple, parallel zone pipelines for the same zone with the requirement to be able to switch which pipeline is "active" at any time. As the local update facility is responsible for updating the UUID if a local change is needed, the same UUID will identify the exact same zone in all pipelines.

- The requirement that the zone pipeline only consists of proven and widely used software forces all local and custom software (including various tests and verification modules) to be located outside the zone pipeline. This cause a need for a component inside the zone pipeline with the ability to call an external "verifier" for verifications. At present only one such component is known (the authoritative nameserver NSD with its "verify:" attribute), but we hope that there will be more alternatives in the future.

12. Acknowledgements

13. Normative References

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

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[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/rfc/rfc5936>.

[RFC8174]
Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8976] Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", RFC 8976, DOI 10.17487/RFC8976, February 2021, <https://www.rfc-editor.org/rfc/rfc8976>.

Appendix A. Change History (to be removed before publication)

*draft-johani-tld-zone-pipeline-00

Initial public draft.

Authors' Addresses

Johan Stenstam
The Swedish Internet Foundation

Email: johan.stenstam@internetstiftelsen.se

Jakob Schlyter
Kirei AB

Email: jakob@kirei.se