Workgroup:
Domain Name System Operations (dnsop)
Published: 24 May 2022
Intended Status: Standards Track
Expires: 25 November 2022
Authors: U. Wisser
The Swedish Internet Foundation
S. Huque
Salesforce

DNSSEC automation

Abstract

This document describes an algorithm and a protocol to automate DNSSEC Multi-Signer [RFC8901] "Multi-Signer DNSSEC Models" setup, operations and decomissioning. Using Model 2 of the Multi-Signer specification, where each operator has their own distinct KSK and ZSK sets (or CSK sets), [RFC8078] "Managing DS Records from the Parent via CDS/CDNSKEY" and [RFC7477] "Child-to-Parent Synchronization in DNS" to accomplish this.

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 https://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 25 November 2022.

Copyright Notice

Table of Contents

1. Introduction

[RFC8901] describes the necessary steps and API for a Multi-Signer DNSSEC configuration. In this document we will combine [RFC8901] with [RFC8078] and [RFC7477] to define an automatable algorithm for setting up, operating and decomissioning of a Multi-Signer DNSSEC configuration.

One of the special cases of Multi-Signer DNSSEC is the secure change of DNS operator. Using Multi-Signer Model 2 the secure change of DNS operator can be accomplished.

1.1. Out-Of-Scope

In order for any Multi-Signer group to give consistent answers across all nameservers, the data contents of the zone also have to be synchronized (in addition to infrastructure records like NS, DNSKEY, CDS etc). This content synchronization is out-of-scope for this document (although there are a number of methods that can be used, such as making the the same updates to each operator using their respective APIs, using zone transfer in conjunction with "inline signing" at each operator, etc.)

1.2. Notation

Short definitions of expressions used in this document

Signer

An entity signing a zone

Multi-Signer Group

A group of signers that sign the same zone

Controller

An entity controlling the multi-signer group. Used in the decentralized model.

1.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Use Cases

2.1. Maintaining a Multi-Signer group

As described in [RFC8901] a Multi-Signer DNSSEC configuration has some challenges that can be overcome with the right infrastructure and following a number of steps for setup and operation.

In this document we describe, except for the initial trust, how the steps in the Multi-Signer DNSSEC setup can be automated.

2.2. Secure Nameserver Operator Transition

Changing the nameserver operator of a DNSSEC signed zone can be challenging. Currently the most common method is temporarily "going insecure". This is poor for security, and for users relying on the security of the zone. Furthermore, when DNSSEC is being used for application security functions like DANE [RFC6698], it is critical that the DNSSEC chain of trust remain unbroken during the transfer.

Multi-Signer DNSSEC Model 2 provides a mechanism for transitioning from one nameserver operator to another without "going insecure". A new operator joins the current operator in a temporary Multi-Signer group. Once that is accomplished and stable the old operator leaves the Multi-Signer group completing the transition.

3. Automation Models

Automation of the necessary steps can be categorized into two main models, centralized and decentralized. Both have pros and cons, and a zone operator should carefully choose the model that works best.

3.1. Centralized

In a centralized model the zone operator will run controller that executes all steps necessary and controls all signers.

A centralized controller needs to have authorized access to all signers. This can be achieved in a variety of different ways. For example will many service providers offer access through a REST API. Another possibility is access through Dynamic Update [RFC2136] with TSIG authentication.

3.2. Decentralized

In the decentralized models all signers will communicate with each other and execute the necessary steps on their instance only. For this signers need a specialized protocol to communicate configuration details that are not part of the zone data.

3.3. Capabilities

In order for any of the models to work the signer must support the following capabilities.

1. Add DNSKEY records (without the private key)

2. Remove (previously added) DNSKEY record(s)

3. Add CDS and CDNSKEY records for keys not in the DNSKEY set

4. Remove (previously added) CDS and CDNSKEY records

5. Add CSYNC record

6. Remove CSYNC record

4. Algorithms

4.1. Prerequisites

Each Signer to be added, including the initial Signer, must meet the following prerequisites before joining the Multi-Signer Group

1. A working setup of the zone, including DNSSEC signing.

2. Uses the same algorithm for DNSSEC signing as the Multi-Signer group uses or will use.

3. Signer or controller must be able to differentiate between its own keys and keys from others signers

4. Signer controller must be able to differentiate between NS records that are updated by itself and NS records that receive updates from other signers.

5. The domain must be covered by a CDS/CDNSKEY scanner and a CSYNC scanner. Otherwise updates to the parent zone have to be made manually.

4.2. Definitions

4.2.1. DS Waiting Time

Once the parent has picked up and published the new DS record set, the any further changes MUST to be delayed until the new DS set has propagated.

The minimum DS Waiting Time is the TTL of the DS RRset.

4.2.2. DNSKEY Waiting Time

Once the DNSKEY sets of all signers are updated, any further changes MUST to be delayed until the new DNSKEY set has propagated.

The minimum DNSKEY Waiting Time is the maximum of all DNSKEYS TTL values from all signers plus the time it takes to publish the zone on all secondaries.

4.2.3. NS Waiting Time

Once the parent has picked up and published the new NS record set, any further changes MUST be delayed until the new NS set has propagated.

The minimum NS Waiting Time is the maximum of the TTL value of the NS set in the parent zone and all NS sets from all signers.

4.3. Setting up a new Multi-Signer group

The zone is already authoritatively served by one DNS operator and is DNSSEC signed. For full automation both the KSK and ZSK or CSK must be online.

This would be a special case, a Multi-Signer group with only one signer.

4.4. A Signer joins the Multi-Signer group

1. Confirm that the incoming Signer meets the prerequisites.
2. Establish a trust mechanism between the Multi-Signer group and the Signer.
3. Add ZSK for each signer to all other Signers.
4. Calculate CDS/CDNSKEY Records for all KSKs/CSKs represented in the Multi-Signer group.
5. Configure all Signers with the compiled CDS/CDNSKEY RRSET.
6. Wait for Parent to publish the combined DS RRset.
7. Remove CDS/CDNSKEY Records from all Signers. (optional)
8. Wait maximum of DS-Wait-Time and DNSKEY-Wait-Time
9. Compile NS RRSET including all NS records from all Signers.
10. Configure all Signers with the compiled NS RRSET.
11. Compare NS RRSET of the Signers to the Parent, if there is a difference publish CSYNC record with NS and A and AAAA bit set on all signers.
12. Wait for Parent to publish NS.
13. Remove CSYNC record from all signers. (optional)

4.5. A signer leaves the Multi-Signer group

1. Remove exiting Signer's NS records from remaining Signers
2. Compare NS RRSET of the Signers to the Parent, if there is a difference publish CSYNC record with NS and A and AAAA bit set on remaining signers.
3. Wait for Parent to publish NS RRSET.
4. Remove CSYNC record from all signers. (optional)
5. Wait NS-Wait-Time
6. Stop the exiting Signer from answering queries.
7. Calculate CDS/CDNSKEY Records for KSKs/CSKs published by the remaining Signers.
8. Configure remaining Signers with the compiled CDS/CDNSKEY RRSET.
9. Remove ZSK of the exiting Signer from remaining Signers.
10. Wait for Parent to publish the updated DS RRset.
11. Remove CDS/CDNSKEY set from all signers. (Optional)

4.6. A Signer performs a ZSK rollover

1. The signer introduces the new ZSK in its own DNSKEY RRset.
2. Update all signers with the new ZSK.
3. Wait DNSKEY-Wait-Time
4. Signer can start using the new ZSK.
5. When the old ZSK is not used in any signatures by the signer, the signer can remove the old ZSK from its DNSKEY RRset.
6. Remove ZSK from DNSKEY RRset of all signers.

4.7. A Signer performs a CSK or KSK rollover

1. Signer publishes new CSK / KSK in its own DNSKEY RRset.
2. In case of CSK, add CSK to DNSKEY set of all other Signers.
3. Signer signs DNSKEY RRset with old and new CSK / KSK.

4. Calculate new CDS/CDNSKEY RRset and publish on all signers.

5. Wait for parent to pickup and publish new DS RR set.

6. Wait DS-Wait-Time + DNSKEY-Wait-Time

7. Signer removes old CSK/KSK from its DNSKEY RR set. And removes all signatures done with this key.

8. In case of CSK, remove old CSK from DNSKEY set of all other signers.

9. Calculate new CDS/CDNSKEY RRset and publish on all signers.

10. Wait for parent to pickup and publish new DS RR set.

11. Remove CDS/CDNSKEY RR sets from all signers.

4.8. Algorithm rollover for the whole Multi-Signer group.

1. All signers publish KSK and ZSK or CSK using the new algorithm.

2. All signers sign all zone data with the new keys.

3. Wait until all signers have signed all data with the new key(s).

4. Add new ZSK of each signer to all other Signers.

5. Calculate new CDS/CDNSKEY RRset and publish on all signers.

6. Wait for parent to pickup and publish new DS RR set.

7. Wait DS-Wait-Time + DNSKEY-Wait-Time

8. Removes all keys and signatures which are using the old algorithm.

9. Calculate new CDS/CDNSKEY RRset and publish on all signers.

10. Wait for parent to pickup and publish new DS RR set.

11. Remove CDS/CDNSKEY RR sets from all signers.

5. Signers with different algorithms in one Multi-Signer group

Section 2.2 of [RFC4035] states that a signed zone MUST include a DNSKEY for each algorithm present in the zone's DS RRset and expected trust anchors for the zone.

A setup where different signers use different key algorithms therefore violates [RFC4035].

According to Section 5.11 of [RFC6840] validators SHOULD NOT insist that all algorithms signaled in the DS RRset work, and they MUST NOT insist that all algorithms signaled in the DNSKEY RRset work.

So a Multi-Signer setup where different signers use different key algorithms should still validate.

This could be an acceptable risk in a situation where going insecure is not desirable or impossible and name servers have to be changed between operators which only support distinct set of key algorithms.

We have to consider the following scenarios

Validator supports both algorithms

Validation should be stable through all stages of the multi-signer algorithms.

Validator supports none of the algorithms

The validator will treat the zone as unsigned. Resolution should work through all stages of the multi-signer algorithms.

Validator supports only one of the algorithms

The validator will not be able to validate the DNSKEY RR set or any data from one of the signers. So in some cases the validator will consider the zone bogus and reply with a SERVFAIL response code.

The later scenario can be mitigated, but not fully eliminated, by selecting two well supported algorithms.

6. Acknowledgements

The authors would like to thank the following for their review of this work and their valuable comments: Steve Crocker, Eric Osterweil, Roger Murray, Jonas Andersson, Peter Thomassen.

7. IANA Considerations

8. Implementation Status

One implementation of a centralized controller which supports updates through Dynamic DNS or REST API's of several vendors has been implemented by the Swedish Internet Foundation.

The code can be found as part of the Multi-Signer project on Github https://github.com/DNSSEC-Provisioning/multi-signer-controller

9. Security Considerations

Every step of the multi-signer algorithms has to be carefully executed at the right time and date. Any failure could resolve in the loss of resolution for the domain.

Independently of the chosen model, it is crucial that only authorized entities will be able to change the zone data. Some providers or software installation allow to make more specific configuration on the allowed changes. All extra steps to allows as little access to change zone data as possible should be taken.

If used correctly the multi-signer algorithm will strengthen the DNS security by avoiding "going insecure" at any stage of the domain life cycle.

10. Normative References

[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/info/rfc2119>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

[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>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <https://www.rfc-editor.org/info/rfc6840>.

[RFC7477] Hardaker, W., "Child-to-Parent Synchronization in DNS", RFC 7477, DOI 10.17487/RFC7477, March 2015, <https://www.rfc-editor.org/info/rfc7477>.

[RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <https://www.rfc-editor.org/info/rfc8078>.

11. Informative References

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC8901] Huque, S., Aras, P., Dickinson, J., Vcelak, J., and D. Blacka, "Multi-Signer DNSSEC Models", RFC 8901, DOI 10.17487/RFC8901, September 2020, <https://www.rfc-editor.org/info/rfc8901>.

Appendix A. Change History

A.1. Change from 01 to 02

1. Trying to fix wording to be more precise
2. Added algorithm for ZSK rollover
3. Added algorithm for KSK rollover
4. Added algorithm for algorithm rollover

A.2. Change from 02 to 03

1. Fix sequnce of steps in the joining procedure
2. Excplicit handling of CSK cases in CSK/ KSK rollover

Authors' Addresses

Ulrich Wisser
The Swedish Internet Foundation
Box 92073
SE-12007 Stockholm
Sweden

Email: ulrich@wisser.se
URI: https://www.internetstiftelsen.se

Shumon Huque
Salesforce
415 Mission Street, 3rd Floor
San Francisco, CA 94105
United States of America

Email: shuque@gmail.com