Workgroup: Network Working Group
Internet-Draft:
draft-hardaker-dnsop-intentionally-temporary-insec-01
Published: 20 October 2021
Intended Status: Best Current Practice
Expires: 23 April 2022
Authors: W. Hardaker
USC/ISI

Intentionally Temporarily Degraded or Insecure

Abstract

Performing DNSKEY algorithm transitions with DNSSEC signing is unfortunately challenging to get right in practice without decent tooling support. This document weighs the correct, completely secure way of rolling keys against an alternate, significantly simplified, method that takes a zone through an insecure state.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Performing DNSKEY [RFC4035] algorithm transitions with DNSSEC [RFC4033] signing is unfortunately challenging to get right in practice without decent tooling support. This document weighs the correct, completely secure way of rolling keys against an alternate, significantly simplified, method that takes a zone through an insecure state.

Section 4.1.4 of [RFC6781] describes the necessary steps required when a new signing key is published for a zone that uses a different signing algorithm than the currently published keys. These are the steps that MUST be followed when zone owners wish to have uninterrupted DNSSEC protection for their zones. The steps in this document are designed to ensure that all DNSKEY records and all DS [RFC4509] records (and the rest of a zone records) are properly validatable by validating resolvers throughout the entire process.

Unfortunately, there are a number of these steps that are challenging to accomplish either because the timing is tricky to get right or because current software doesn't support automating the process easily. Some examples:

1. The second step in Section 4.1.4 of [RFC6781] requires that a new key with the new algorithm (which we refer to as K_new) be created, but not yet published. This step also requires that both the old key (K_old) and K_new sign and generate signatures for the zone, but with only the K_old key is published even though signatures from K_new are included. After this odd mix has been published for a sufficient time length, based on the

TTL, can K_new be safely introduced and published into the zone as well.

2. The new algorithm to be deployed isn't supported in the existing DNSSEC signing software and it is not possible (or not desired) to move the private key into the DNSSEC signer that supports the new algorithm choice.

Although many DNSSEC signing solutions may automate the algorithm rollover steps (making operator involvement unnecessary), many other tools do not support automated algorithm updates. In these environments, the most challenging step is requiring that certain RRSIGs be published without the corresponding DNSKEYs that created them. This will likely require operators to use a text editor on the contents of a signed zone to carefully select zone records to extract before publication. This introduces potentially significant operator error(s).

This document proposes an alternate, potentially more operationally robust but less secure, approach to performing algorithm DNSKEY rollovers for use in these situations.

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. Temporary transition mechanisms

2.1. Transitioning temporarily through insecurity

An alternate approach to rolling DNSKEYs, especially when the toolsets being used do not provide easy algorithm rollover approaches, is to intentionally make the zone become insecure while the DNSKEYs and algorithms are swapped. At a high level, this means removing all DS records from the parent zone during the removal of the old key and the introduction of a new key using a new algorithm. Zone TTLs may be significantly shortened during this period to minimize the period of insecurity.

Below are the enumerated steps required by this alternate transition mechanism. Note that there are still two critical waiting time requirements (steps 2 and 6) that must be followed carefully.

1. Optional: lower the TTLs of the zone's DS record (if possible), and the TTL of the DNSKEY RRset.

2. Remove all DS records from the parent zone.

3. Ensure the zone is considered unsigned by all validating resolvers by waiting 2 times the maximum TTL length for the DS record, and/or 2 times the largest TTL found in the zone (whichever is larger) to expire from caches. This is the most critical timing. The author of this document failed to wait the required time once. It was not pretty.

4. Replace the old DNSKEY(s) with the old algorithm with new DNSKEY(s) with the new algorithm(s) in the zone and publish the zone.

5. Wait 2 times the largest TTL found in the zone to ensure the new DNSKEYs will be found by validating resolvers.

6. Add the DS record(s) for the new DNSKEYs to the parent zone.

7. If the TTLs were modified in the optional step 1, change them back to their preferred values.

2.2. Transitioning using two DNS servers

Another option for performing an algorithm roll is to make use of two (or more) NS records, where one of them continues to serve a zone signed by the old algorithm and the other authoritative server switches to serving the zone using the new DNSKEY and its new algorithm. This allows for clients that end up at the wrong NS to eventually give up and switch to the other, containing the expected algorithm. The downside of this approach is the deliberate delay in resolutions for resolvers that query the wrong authoritative server for the DS record they are trying to match.

The steps for deploying this technique to switch algorithms is as follows:

1. Optional: lower the TTLs of the zone's DS record (if possible) and the SOA's negative TTL (MINIMUM) [RFC1035].

2. Ensure your zone has matching NS records in both the child data and in the parent data.

3. Leaving the old algorithm DS record in the parent zone. Resign the child zone using a new DNSKEY with the new algorithm and publish it on roughly 50% of the zone's authoritative nameservers.

4. Wait a period of time equal to max(TTL in the zone, DS record).

5. Simultaneously remove the old DS record from the parent, and publish a new DS record that refers to the new DNSKEY (and its new algorithm).

6. Wait a period of time equal to max(TTL in the zone, DS record).

7. Update the authoritative nameservers that remained publishing the older copy of the zone. All authoritative servers can now publish the updated zone with the new DNSKEYs.

Credit for this idea goes to Tuomo Soini and Paul Wouters.

3. Operational considerations

The process of replacing a DNSKEY with an older algorithm, such as RSAMD5 or RSASHA1 with a more modern one such as RSASHA512 or ECDSAP256SHA256 can be a daunting task if the zone's current tooling doesn't provide an easy-to-use solution. This is the case for zone owners that potentially use command line tools that are integrated into their zone production environment.

This document describes an alternative approach to rolling DNSKEY algorithms that may be significantly less prone to operational mistakes. However, understanding of the security considerations of using this approach is paramount.

The document recommends waiting 2 times TTL values in certain cases for added assurance that the waiting period is long enough for caches to expire. In reality, waiting only 1 TTL may be sufficient assuming all clocks around the world are operating with perfection.

4. Security considerations

DNSSEC provides an data integrity protection for DNS data. This document specifically calls out a reason why a zone owner may desire to deliberately turn off DNSSEC while changing the zone's DNSKEY's cryptographic algorithms. Thus, this is deliberately turning off security which is potentially harmful if an attacker knows when this will occur and can use that time window to launch DNS modification attacks (for example, cache poisoning attacks) against validating resolvers or other validating DNS infrastructure.

Most importantly, this will deliberately break certain types of DNS records that must be validatable for them to be effective. This includes for example, but not limited to, all DS records for child zones, DANE [RFC6698][RFC7671][RFC7672], PGP keys [RFC7929], and SSHFP[RFC4255]. Zone owners must carefully consider which records within their zone depend on DNSSEC being available before using the procedure outlined in this document.

Given all of this, it leaves the question of: "why would a zone owner want to deliberately turn off security temporarily then?", to which there is one principal answer. Simply put, if the the complexity of doing it the correct way is difficult with existing tooling then the chances of performing the more complex procedure and introducing an error, likely making the entire zone unavailable during that time period, may be significantly higher than the chances of the zone being attacked during the transition period of the simpler approach where zone availability is less likely to be impacted. Simply put, an invalid zone created by a botched algorithm roll is potentially worse than an unsigned but still available zone.

5. References

5.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[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>.

[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>.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, DOI 10.17487/RFC4509, May 2006, <https://www.rfc-editor.org/info/rfc4509>.

[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>.

[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/info/rfc8174>.

5.2. Informative References

[RFC4255]
Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, DOI 10.17487/RFC4255, January 2006, <https://www.rfc-editor.org/info/rfc4255>.

[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>.

[RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <https://www.rfc-editor.org/info/rfc7671>.

[RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <https://www.rfc-editor.org/info/rfc7672>.

[RFC7929] Wouters, P., "DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP", RFC 7929, DOI 10.17487/RFC7929, August 2016, <https://www.rfc-editor.org/info/rfc7929>.

Appendix A. Acknowledgments

The author has discussed the pros and cons of this approach with multiple people, including:

* Viktor Dukhovni

* Warren Kumari.

* Tuomo Soini

* Paul Wouters

Appendix B. Github Version of this document

While this document is under development, it can be viewed, tracked, issued, pushed with PRs, ... here:

https://github.com/hardaker/draft-hardaker-dnsop-intentionally-temporarily-insecure

Author's Address

Wes Hardaker
USC/ISI

Email: ietf@hardakers.net