dnsop
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2017

W. Hardaker
USC/ISI
W. Kumari
Google

June 27, 2017

Security Considerations for RFC5011 Publishers
draft-ietf-dnsop-rfc5011-security-considerations-02

Abstract

This document extends the RFC5011 rollover strategy with timing advice that must be followed in order to maintain security. Specifically, this document describes the math behind the minimum time-length that a DNS zone publisher must wait before signing with only recently added DNSKEYs. This document also describes the minimum time-length that a DNS zone publisher must wait after publishing a revoked DNSKEY before assuming that all active RFC5011 resolvers should have seen the revocation-marked key and removed it from their list of trust anchors.

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 December 29, 2017.

Copyright Notice

Table of Contents

1. Introduction

[RFC5011] defines a mechanism by which DNSSEC validators can extend their list of trust anchors when they've seen a new key published in a zone. However, RFC5011 [intentionally] provides no guidance to the publishers of DNSKEYs about how long they must wait before switching to using only recently published keys for signing records, or how long they must wait before removing a revoked key from a zone. Because of this lack of guidance, zone publishers may derive incorrect assumptions about safe usage of the RFC5011 DNSKEY advertising, rolling and revocation process. This document describes the minimum security requirements from a publisher's point of view and is intended to compliment the guidance offered in RFC5011 (which is written to provide timing guidance solely to a Validating Resolver's point of view).

1.1. Document History and Motivation

To verify this lack of understanding is wide-spread, the authors reached out to 5 DNSSEC experts to ask them how long they thought they must wait before signing a zone using a new KSK [RFC4033] that was being rolled according to the 5011 process. All 5 experts answered with an insecure value, and we determined that this lack of operational guidance is causing security concerns today and wrote this companion document to RFC5011. We hope that this document will rectify this understanding and provide better guidance to zone publishers that wish to make use of the RFC5011 rollover process.

1.2. Safely Rolling the Root Zone's KSK in 2017/2018

One important note about ICANN's [currently upcoming] 2017/2018 KSK rollover plan for the root zone: the timing values chosen for rolling the KSK in the root zone appear completely safe, and are not affected by the timing concerns introduced by this draft

1.3. Requirements notation

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. Background

The RFC5011 process describes a process by which a RFC5011 Validating Resolver may accept a newly published KSK as a trust anchor for validating future DNSSEC signed records. It also describes the process for publicly revoking a published KSK. This document augments that information with additional constraints, as required from the DNSKEY publication and revocation's points of view. Note that it does not define any other operational guidance or recommendations about the RFC5011 process and restricts itself to solely the security and operational ramifications of switching to using only recently added keys or removing a revoked keys too soon. Failure of a DNSKEY publisher to follow the minimum recommendations associated with this draft will result in potential denial-of-service attack opportunities against validating resolvers. Failure of a DNSKEY publisher to publish a revoked key for a long enough period of time may result in RFC5011 Validating Resolvers leaving a key in their trust anchor storage beyond their expected lifetime.

3. Terminology

Trust Anchor Publisher The entity responsible for publishing a DNSKEY that can be used as a trust anchor.

Zone Signer The owner of a zone intending to publish a new Key-Signing-Key (KSK) that will become a trust anchor by validators following the RFC5011 process.

RFC5011 Validating Resolver A DNSSEC Validating Resolver that is using the RFC5011 processes to track and update trust anchors. Sometimes referred to as a "RFC5011 Resolver"

Attacker An attacker intent on foiling the RFC5011 Validator's ability to successfully adopt the Zone Signer's new DNSKEY as a new trust anchor or to prevent the RFC5011 Validator from removing an old DNSKEY from its list of trust anchors.

Also see Section 2 of [RFC4033] and [RFC7719] for additional terminology.

4. Timing Associated with RFC5011 Processing

4.1. Timing Associated with Publication

RFC5011's process of safely publishing a new key and then making use of that key falls into a number of high-level steps to be performed by the Trust Anchor Publisher:

1. Publish a new DNSKEY in the zone, but continue to sign the zone with the old one.

2. Wait a period of time.

3. Begin using only recently published DNSKEYs to sign the appropriate resource records.

This document discusses step 2 of the above process. Some interpretations of RFC5011 have erroneously determined that the wait time is equal to RFC5011's "hold down time".

Section 5 describes an attack based on this (common) erroneous belief, which results in a denial of service attack against the zone if that value is used.

4.2. Timing Associated with Revocation

RFC5011's process of advertising that an old key is to be revoked from RFC5011 validating resolvers falls into a number of high-level steps:

1. Set the revoke bit on the DNSKEY to be revoked.

2. Sign the revoked DNSKEY with itself.

3. Wait a period of time.

4. Remove the revoked key from the zone.

This document discusses step 3 of the above process. Some interpretations of RFC5011 have erroneously determined that the wait time is equal to RFC5011's "hold down time".

This document describes an attack based on this (common) erroneous belief, which results in a revoked DNSKEY potentially staying in a RFC5011 validating resolver long past its expected usage.

5. Denial of Service Attack Considerations

If an attacker is able to provide a RFC5011 Validating Resolver with past responses, such as when it is in-path or able to otherwise perform any number of cache poisoning attacks, the attacker may be able to leave compliant RFC5011-Validating Resolvers without an appropriate DNSKEY trust anchor. This scenario will remain until an administrator manually fixes the situation.

The following timeline illustrates this situation.

5.1. Enumerated Attack Example

The following example settings are used in the example scenario within this section:

TTL (all records) 1 day

DNSKEY RRSIG Signature Validity 10 days

Zone resigned every 1 day

Given these settings, the sequence of events in Section 5.1.1 depicts how a Trust Anchor Publisher that waits for only the RFC5011 hold time timer length of 30 days subjects its users to a potential Denial of Service attack. The timing schedule listed below is based on a

Trust Anchor Publisher publishing a new Key Signing Key (KSK), with the intent that it will later become a trust anchor. We label this publication time as "T+0". All numbers in this sequence refer to days before and after this initial publication event. Thus, T-1 is the day before the introduction of the new key, and T+15 is the 15th day after the key was introduced into the fictitious zone being discussed.

In this dialog, we consider two keys being published:

K_old An older KSK and Trust Anchor being replaced.

K_new A new KSK being transitioned into active use and becoming a Trust Anchor via the RFC5011 process.

5.1.1. Attack Timing Breakdown

The following series of steps depicts the timeline in which an attack occurs that foils the adoption of a new DNSKEY by a Trust Anchor Publisher that starts signing with the new DNSKEY too quickly.

T-1 The last RRSIGs are published by the Zone Signer that signs only K_old key using the K_old key itself. [It may also be signing ZSKs as well, but they are not relevant to this event so we will not talk further about them; we are only talking about RRSIGs that cover the DNSKEYs.] The Attacker queries for, retrieves and caches this DNSKEY set and corresponding RRSIG signatures.

T-0 The Zone Signer adds K_new to their zone and signs the zone's key set with K_old. The RFC5011 Validator (later to be under attack) retrieves this new key set and corresponding RRSIGs and notices the publication of K_new. The RFC5011 Validator starts the (30-day) hold-down timer for K_new.

T+5 The RFC5011 Validator queries for the zone's keyset per the RFC5011 Active Refresh schedule, discussed in Section 2.3 of RFC5011. Instead of receiving the intended published keyset, the Attacker successfully replays the keyset and associated signatures that they recorded at T-1. Because the signature lifetime is 10 days (in this example), the replayed signature and keyset is accepted as valid (being only 6 days old) and the RFC5011 Validator cancels the hold-down timer for K_new, per the RFC5011 algorithm.

T+10 The RFC5011 Validator queries for the zone's keyset and discovers the new kset which includes K_new (again), signed by K_old. Note: the attacker is unable to replay the records cached

at T-1, because they have now expired. The RFC5011 Validator starts (anew) the hold-timer for K_new.

T+15,T+20, and T+25 The RFC5011 Validator continues checking the zone's key set at the prescribed regular intervals. The RFC5011 Validator's hold-down timer keep running without being reset assuming all of the validations succeed (again: the attacker can no longer replay traffic to their benefit).

T+30 The Zone Signer knows that this is the first time at which some validators might accept K_new as a new trust anchor, since the hold-down timer of a RFC5011 Validator not under attack that had queried and retrieved K_new at T+0 would now have reached 30 days. However, the hold-down timer of our attacked RFC5011 Validator is only at 20 days.

T+35 The Zone Signer (mistakenly) believes that all validators following the Active Refresh schedule (Section 2.3 of RFC5011) should have accepted K_new as a the new trust anchor (since the hold down time of 30 days + 1/2 the signature validity period would have passed). However, the hold-down timer of our attacked RFC5011 Validator is only at 25 days; The replay attack at T+5 means its new hold-time timer actually started at T+10, and thus at this time it's real hold-down timer is at T+35 - T+10 = 25 days, which is less than the RFC5011 required 30 days and the RFC5011 won't consider it a valid trust anchor addition yet.

T+36 The Zone Signer, believing K_new is safe to use, switches their active signing KSK to K_new and publishes a new RRSIG, signed with K_new, and covering the DNSKEY set. Non-attacked RFC5011 validators, with a hold-down timer of at least 30 days, would have accepted K_new into their set of trusted keys. But, because our attacked RFC5011 Validator has a hold-down timer for K_new at only 26 days, it will fail to accept K_new as a trust anchor. Since K_old is no longer being used, all the DNSKEY records from the zone signed by K_new will be treated as invalid. Subsequently, all keys in the key set are now unusable, invalidating all of the records in the zone of any type and name.

6. Minimum RFC5011 Timing Requirements

Given the attack description in Section 5, the correct minimum length of time required for the Zone Signer to wait before using K_new is:

```
  waitTime = addHoldDownTime
             + (DNSKEY RRSIG Signature Validity)
             + MAX(MIN((DNSKEY RRSIG Signature Validity) / 2,
                       MAX(original TTL of K_old DNSKEY RRSet) / 2,
                       15 days),
                   1 hour)
             + 2 * MAX(TTL of all records)
```

The RFC5011 "Active Refresh" requirements state that:

> A resolver that has been configured for an automatic update of keys from a particular trust point MUST query that trust point (e.g., do a lookup for the DNSKEY RRSet and related RRSIG records) no less often than the lesser of 15 days, half the original TTL for the DNSKEY RRSet, or half the RRSIG expiration interval and no more often than once per hour.

The important timing constraint introduced by this memo relates to the last point at which a validating resolver may have received a replayed the original DNSKEY set (K_old) without the new key. It's the next query of the RFC5011 validator that the assured K_new will be seen without a potential replay afterward. Thus, the latest time a RFC5011 validator may begin their hold down timer is an "Active Refresh" period after the last point that an attacker can replay the K_old DNSKEY set.

The "Active Refresh" interval used by a RFC5011 validator is determined by the larger of (DNSKEY RRSIG Signature Validity) and (original TTL for the DNSKEY RRSet). The Following text assumes that (DNSKEY RRSIG Signature Validity) is larger of the two, which is operationally more common today.

Thus, the worst case scenario of this attack is when the attacker can replay K_old just before (DNSKEY RRSIG Signature Validity). If a RFC5011 validator picks up K_old at this this point, it will not have a hold down timer started as it will have been reset by previous replays. It's not until the next "Active Refresh" time that they'll pick up K_new with assurance, and thus start their (final) hold down timer. Thus, this is not at (DNSKEY RRSIG Signature Validity) time past publication but may be significantly longer based on the zone's DNSSEC parameters.

The extra 2 * MAX(TTL of all records) is the standard added safety margin when dealing with DNSSEC due to caching that can take place. Because the 5011 steps require direct validation using the signature

validity, the authors aren't yet convinced it is needed in this particular case, but it is prudent to include it for added assurance.

For the parameters listed in Section 5.1, our example:

```
  waitTime = 30
             + 10
             + 10 / 2
             + 2 * (1)        (days)

  waitTime = 47               (days)
```

This hold-down time of 47 days is 12 days longer than the (frequently perceived) 35 days in the example at T+35 above.

It is important to note that this value affects not just the publication of new DNSKEYs intended to be used as trust anchors, but also the length of time required to publish a DNSKEY with the revoke bit set. Both of these publication timing requirements are affected by the attacks described in this document.

7. IANA Considerations

This document contains no IANA considerations.

8. Operational Considerations

A companion document to RFC5011 was expected to be published that describes the best operational practice considerations from the perspective of a zone publisher and Trust Anchor Publisher. However, this companion document has yet to be published. The authors of this document hope that it will at some point in the future, as RFC5011 timing can be tricky as we have shown and we do not suggest "good operational practice" that might be associated with a BCP on the subject. This document is intended only to fill a single operational void that results in security ramifications (specifically a denial of service attack against an RFC5011 Validator). This document does not attempt to document any other missing operational guidance for zone publishers.

9. Security Considerations

This document, is solely about the security considerations with respect to the Trust Anchor Publisher of RFC5011 trust anchors / keys. Thus the entire document is a discussion of Security Considerations when rolling DNSKEYs using the RFC5011 process.

10. Acknowledgements

The authors would like to especially thank to Michael StJohns for his help and advice and the care and thought he put into RFC5011 itself. We would also like to thank Bob Harold, Shane Kerr, Matthijs Mekking, Duane Wessels, Petr Petr Spacek, and the dnsop working group who have assisted with this document.

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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, <http://www.rfc-editor.org/info/rfc4033>.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, DOI 10.17487/RFC5011, September 2007, <http://www.rfc-editor.org/info/rfc5011>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <http://www.rfc-editor.org/info/rfc7719>.

Appendix A. Real World Example: The 2017 Root KSK Key Roll

In 2017, ICANN expects to (or has, depending on when you're reading this) roll the key signing key (KSK) for the root zone. The relevant parameters associated with the root zone at the time of this writing is as follows:

```
addHoldDownTime:                    30 days
Old DNSKEY RRSIG Signature Validity:  21 days
Old DNSKEY TTL:                      2 days
```

Thus, sticking this information into the equation in Section Section 6 yields (in days):

```
waitTime = addHoldDownTime
           + (DNSKEY RRSIG Signature Validity)
           + MAX(MIN((DNSKEY RRSIG Signature Validity) / 2,
                     MAX(original TTL of K_old DNSKEY RRSet) / 2,
                     15 days),
                 1 hour)
           + 2 * MAX(TTL of all records)

waitTime = 30
           + (21)
           + MAX(MIN((21) / 2,
                     MAX(2 / 2,
                     15 days)),
                 1 hour)
           + 2 * MAX(2)

waitTime = 30 + 21 + MAX(MIN(11.5, MAX( 1, 15)), 1 hour) + 4

waitTime = 30 + 21 + 11.5 + 4

waitTime = 66.5 days
```

Thus, ICANN should wait 66.5 days before switching to the newly published KSK and before removing the old revoked key once it is published as revoked. ICANN's current plans are to wait 70 days before using the new KEY and 69 days before removing the old, revoked key. Thus, their current rollover plans are sufficiently secure from the attack discussed in this memo.

Appendix B. Changes / Author Notes.

From Individual-00 to DNSOP-00:

- Filename change.

From -00 to -01:

- Added Revocation processing (including "Timing Associated with Revocation")
- Added real world example.
- Fixed some typoes and missing references.

From Ind-00 to -02:

Additional background and clarifications in abstract.

Better separation in attack description between attacked and non-attacked resolvers.

Some language cleanup.

Clarified that this is maths (and math is hard, let's go shopping!)

Changed to " <?rfc include='reference....'?> " style references.

From -02 to -03:

Minor changes from Bob Harold

Clarified why 3/2 signature validity is needed

Changed min wait time math to include TTL value as well

From -03 to -04:

Fixed the waitTime equation to handle the difference between the usage of the expiration time and the Active Refresh time.

More clarification text and text changes proposed by Petr Spacek

From -04 to -05:

Clarifications about signing using only new keys, vs old ones too

Pre-DNSOP document: From hardaker-04 to ietf-00:

Just rebranding.

From ietf-00 to ietf-01:

Added discussion surrounding revocation everywhere

Fixed the text about the formula

Another complete re-read for word-smithing

Authors' Addresses

Wes Hardaker
USC/ISI
P.O. Box 382
Davis, CA 95617
US

Email: ietf@hardakers.net

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: warren@kumari.net