DNSOP
Internet-Draft
Intended status: Standards Track
Expires: August 16, 2018

G. Huston
J. Damas
APNIC
W. Kumari
Google

February 12, 2018

A Sentinel for Detecting Trusted Keys in DNSSEC
draft-ietf-dnsop-kskroll-sentinel-01

Abstract

The DNS Security Extensions (DNSSEC) were developed to provide origin authentication and integrity protection for DNS data by using digital signatures. These digital signatures can be verified by building a chain of trust starting from a trust anchor and proceeding down to a particular node in the DNS. This document specifies a mechanism that will allow an end user to determine the trusted key state of the resolvers that handle that user's DNS queries.

There is an example / toy implementation of this at http://www.ksk-test.net .

[This document is being collaborated on in Github at: https://github.com/APNIC-Labs/draft-kskroll-sentinel.. The most recent version of the document, open issues, etc should all be available here. The authors (gratefully) accept pull requests. Text in square brackets will be removed before publication.]

[NOTE: This version uses the labels "kskroll-sentinel-is-ta-<tag-index>", "kskroll-sentinel-not-ta-<tag-index>"; older versions of this document used "_is-ta-<tag-index>", "_not-ta-<tag-index>".]

This Internet-Draft will expire on August 16, 2018.

Table of Contents

1. Introduction

The DNS Security Extensions (DNSSEC) [RFC4033], [RFC4034] and [RFC4035] were developed to provide origin authentication and integrity protection for DNS data by using digital signatures. DNSSEC uses Key Tags to efficiently match signatures to the keys from which they are generated. The Key Tag is a 16-bit value computed from the RDATA portion of a DNSKEY RR using a formula not unlike a ones-complement checksum. RRSIG RRs contain a Key Tag field whose value is equal to the Key Tag of the DNSKEY RR that validates the signature.

This document specifies how validating resolvers can respond to certain queries in a manner that allows a querier to deduce whether a

particular key has been loaded into that resolver's trusted key store. In particular, this response mechanism can be used to determine whether a certain Root Zone KSK is ready to be used as a trusted key within the context of a key roll by this resolver.

This new mechanism is OPTIONAL to implement and use, although for reasons of supporting broad-based measurement techniques, it is strongly preferred if configurations of DNSSEC-validating resolvers enabled this mechanism by default, allowing for local configuration directives to disable this mechanism if desired.

1.1. Terminology

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 RFC 2119.

Address Record: Throughout this document we use the term Address Record (AR) to mean an A or AAAA record. We are using example.com, AAAA records and the IPv6 documentation prefix (2001:DB8::/32) as examples; these are only examples - A records (or CNAMES), other IPs, other domains work just as well. [Ed note: There was some earlier confusion on this, being explicit!]

2. Use Case

[Ed note: This is currently towards the front of the document; we will make it an appendix at publication time, but until then it is worth having up front, as it makes the rest of the document much easier to understand]

This section provides a non-normative example of how the sentinel mechanism could be used, and what each participant does. It is provided in a conversational tone to be easier to follow.

Alice is in charge of the DNS root KSK (Key Signing Key), and would like to roll / replace the key with a new one. She publishes the new KSK, but would like to be able to predict / measure what the impact will be before removing/revoking the old key. The current KSK has a key ID of 1111, the new KSK has a key ID of 2222

Bob, Charlie, Dave, Ed are all users. They use the DNS recursive resolvers supplied by their ISPs. They would like to confirm that their ISPs have picked up the new KSK and will not break. Bob's ISP does not perform validation. Charlie's ISP does validate, but the resolvers have not yet been upgraded to support sentinel. Dave and Ed's resolvers have been upgraded to support sentinel; Dave's

resolver has the new KSK, Ed's resolver hasn't managed to install the 2222 KSK in its trust store yet.

Geoff is a researcher, and would like to both provide a means for Bob, Charlie, Dave and Ed to be able to perform tests, and also would like to be able to perform Internet wide measurements of what the impact will be (and report this back to Alice).

Geoff sets an authoritative DNS server for example.com, and also a webserver (www.example.com). He adds 3 AAAA records to example.com:

```
invalid IN AAAA 2001:DB8::1

kskroll-sentinel-is-ta-2222 IN AAAA 2001:DB8::1

kskroll-sentinel-not-ta-2222 IN AAAA 2001:DB8::1
```

Geoff then DNSSEC signs the example.com zone, and intentionally makes the invalid.example.com record invalid/bogus (for example, by editing the signed zone and entering garbage for the signature). Geoff also configures his webserver to listen on 2001:DB8::1 and serve a resource (for example, a 1x1 GIF, 1x1.gif) for all of these names. The webserver also serves a webpage (www.example.com) which contains links to these 3 resources (http://invalid.example.com/1x1.gif, http://kskroll-sentinel-is-ta-2222.example.com/1x1.gif, http://kskroll-sentinel-not-ta-2222.example.com/1x1.gif).

Geoff then asks Bob, Charlie, Dave and Ed to browse to www.example.com. Using the methods described in this document, the users can figure out what their fate will be when the 1111 KSK is removed.

Bob is not using a validating resolver. This means that he will be able to resolve invalid.example.com (and fetch the 1x1 GIF) - this tells him that the KSK roll does not affect him, and so he will be OK.

Charlie's resolvers are validating, but they have not been upgraded to support the KSK sentinel mechanism. Charlie will not be able to fetch the http://invalid.example.com/1x1.gif resource (the invalid.example.com record is bogus, and none of his resolvers will resolve it). He is able to fetch both of the other resources - from this he knows (see the logic below) that he is using legacy, non-validating resolvers. The KSK sentinel method cannot provided him with a definitive answer.

Dave's resolvers implement the sentinel method, and have picked up the new KSK. For the same reason as Charlie, he cannot fetch the

"invalid" resource. His resolver resolves the kskroll-sentinel-is-ta-2222.example.com name normally (it contacts the example.com authoritative servers, etc); as it supports the sentinel mechanism, just before Dave's recursive server send the reply to Dave's stub, it performs the KSK Sentinel check (see below). The QNAME starts with "kskroll-sentinel-is-ta-", and the recursive resolver does indeed have a key with the Key ID of 2222 in its root trust store. This means that that this part of the KSK Sentinel check passes (it is true that 2222 is in the Trust Anchor store), and the recursive resolver replies normally (with the answer provided by the authoritative server). Dave's recursive resolver then resolves the kskroll-sentinel-not-ta-2222.example.com name. Once again, it performs the normal resolution process, but because it implements KSK Sentinel (and the QNAME starts with "kskroll-sentinel-not-ta-"), just before sending the reply, it performs the KSK Sentinel check. As it has 2222 in it's trust anchor store, the "Is this *not* a trust anchor" is false, and so the recursive resolver does not reply with the answer from the authoritative server - instead, it replies with a SERVFAIL (note that replying with SERVFAIL instead of the original answer is the only mechanism that KSK Sentinel uses). This means that Dave cannot fetch "invalid", he can fetch "kskroll-sentinel-is-ta-2222", but he cannot fetch "kskroll-sentinel-not-ta-2222". From this, Dave knows that he is behind an upgraded, validating resolver, which has successfully installed the new, 2222 KSK. Dave has nothing to worry about - he will be fine with the old (1111) KSK is removed.

Now for Ed. Just like Charlie and Dave, Ed cannot fetch the "invalid" record. This tells him that his resolvers are validating. When his (upgraded) resolver performs the KSK Sentinel check for "kskroll-sentinel-is-ta-2222", it does *not* have the (new, 2222) KSK in it's trust anchor store. This means check fails, and Ed's recursive resolver converts the (valid) 2001:DB8::1 answer into a SERVFAIL error response. It performs the same check for kskroll-sentinel-not-ta-2222.example.com; as it does not have the 2222 KSK, it is true that this is not a trust anchor for it, and so it replies normally. This means that Ed cannot fetch the "invalid" resource, he also cannot fetch the "kskroll-sentinel-is-ta-2222" resource, but he can fetch the "kskroll-sentinel-not-ta-2222" resource. This tells Ed that his resolvers have not installed the new KSK, and, when the old KSK is removed, his DNS will break.

Geoff would like to do a large scale test and provide the information back to Alice. He uses some mechanism (such as an advertising network) to cause a large number of users to attempt to resolve the 3 resources, and then analyzes the results of the tests to determine what percentage of users will be affected by the KSK rollover event.

The above description is a simplified example - it is not anticipated that Bob, Charlie, Dave and Ed will actually look for the absence or presence of web resources; instead, the webpage that they load would likely contain JavaScript (or similar) which displays the result of the tests. An example of this is at http://www.ksk-test.net. This KSK mechanism does not rely on the web - this method can equally be used by trying to resolve the names (for example, using 'dig') and checking which result in a SERVFAIL.

[Note that the KSK Sentinel mechanism measures a very different (and, in our opinion, much more useful!) metric than RFC8145 -- RFC8145 relied on resolvers reporting the list of keys that they have -- this doesn't reflect what the *user* impact of the KSK roll will be. As we cannot get perfect visibility into all resolvers, we will have to aim for "do the least harm", not "do no harm"]

3. Sentinel Mechanism

DNSSEC-Validating resolvers that implement this mechanism MUST be performing validation of responses in accordance with the DNSSEC response validation specification [RFC4035].

This sentinel mechanism makes use of 2 special labels, "kskroll-sentinel-is-ta-<tag-index>." (intended to be used in a query where the response can answer the question: Is this the key tag a trust anchor which the validating DNS resolver is currently trusting?) and "kskroll-sentinel-not-ta-<tag-index>." (intended to be used in a query where the response can answer the question: Is this the key tag of a key that is NOT in the resolver's current trust store?). The use of the positive question and its inverse allows for queries to detect whether resolvers support this sentinel mechanism. Note that the test is "Is there a key with this KeyID in the resolver's current trust store for the DNS root", not "Is there any key with this KeyID in the trust store", nor "Was a key with this KeyID used to validate this query?". [This is still an active discussion on the DNSOP list]

If the outcome of the DNSSEC validation process on the response RRset indicates that the response RRset is authentic, and if the left-most label of the original query name matches the template "kskroll-sentinel-is-ta-<tag-index>.", then the following rule should be applied to the response: If the resolver has placed a Root Zone Key Signing Key with tag index value matching the value specified in the query into the local resolver's store of trusted keys, then the resolver should return a response indicating that the response contains authenticated data according to section 5.8 of [RFC6840]. Otherwise, the resolver MUST return RCODE 2 (server failure). Note

that the <tag-index> is specified in the DNS label using hexadecimal notation.

If the outcome of the DNSSEC validation process applied to the response RRset indicates that the response RRset is authentic, and if the left-most label of the original query name matches the template "kskroll-sentinel-not-ta-<tag-index>.", then the following rule should be applied to the response: If the resolver has not placed a Root Zone Key Signing Key with tag index value matching the value specified in the query into the local resolver's store of trusted keys, then the resolver should return a response indicating that the response contains authenticated data according to section 5.8 of [RFC6840]. Otherwise, the resolver MUST return RCODE 2 (server failure). Note that the <tag-index> is specified in the DNS label using hexadecimal notation.

In all other cases the resolver MUST NOT alter the outcome of the DNS response validation process.

This mechanism is to be applied only by resolvers that are performing DNSSEC validation, and applies only to RRset responses to an A or AAAA query (Query Type value 1 or 28) where the resolver has authenticated the response RRset according to the DNSSEC validation process and where the query name contains either of the labels described in this section as its left-most label. In this case, the resolver is to perform an additional test following the conventional validation function, as described in this section. The result of this additional test determines whether the resolver will alter its response that would have indicated that the RRset is authentic to a response that indicates DNSSEC validation failure via the use of RCODE 2.

4. Sentinel Processing

This proposed test that uses the sentinel detection mechanism described in this document is based on the use of three DNS names that have three distinct DNS resolution behaviours. The test is intended to allow a user to determine the state of their DNS resolution system, and, in particular, whether or not they are using validating DNS resolvers that have picked up an incoming trust anchor as a trusted key in a root zone KSK roll scenario.

The name format can be defined in a number of ways, and no name form is intrinsically better than any other in terms of the test itself. The critical aspect of the DNS names used in any such test is that they contain the specified label for either the positive and negative test as the left-most label in the query name.

The sentinel detection process is envisaged to use a test with three query names:

a. a query name containing the left-most label "kskroll-sentinel-is-ta-<tag-index>.". This corresponds to a a validly-signed RRset in the zone, so that responses associated with queried names in this zone can be authenticated by a DNSSEC-validating resolver. Any validly-signed DNS zone can be used for this test.

b. a query name containing the left-most label "kskroll-sentinel-not-ta-<tag-index>.". This is also a validly-signed name. Any validly-signed DNS zone can be used for this test.

c. a third query name that is signed with a DNSSEC signature that cannot be validated (i.e. the corresponding RRset is not signed with a valid RRSIG record).

The responses received from queries to resolve each of these names would allow us to infer a trust key state of the resolution environment. To describe this process of classification, we can classify resolvers into four distinct behavior types, for which we will use the labels: "Vnew", "Vold", "Vleg", and "nonV". These labels correspond to resolver behaviour types as follows:

- Vnew: A DNSSEC-Validating resolver that is configured to implement this mechanism has loaded the nominated key into its local trusted key store will respond with an A or AAAA RRset response for "kskroll-sentinel-is-ta" queries, SERVFAIL for "kskroll-sentinel-not-ta" queries and SERVFAIL for the invalidly signed name queries.

- Vold: A DNSSEC-Validating resolver that is configured to implement this mechanism that has not loaded the nominated key into its local trusted key store will respond with an SERVFAIL for "kskroll-sentinel-is-ta" queries, an A or AAAA RRset response for "kskroll-sentinel-not-ta" queries and SERVFAIL for the invalidly signed name queries.

- Vleg: A DNSSEC-Validating resolver that does not implement this mechanism will respond with an A or AAAA RRSET response for "kskroll-sentinel-is-ta", an A record response for "kskroll-sentinel-not-ta" and SERVFAIL for the invalid name.

- nonV: A non-DNSSEC-Validating resolver will respond with an A record response for "kskroll-sentinel-is-ta", an A record response for "kskroll-sentinel-not-ta" and an A record response for the invalid name.

Given the clear delineation amongst these three cases, if a client directs these three queries to a simple resolver, the variation in response to the three queries should allow the client to determine the category of the resolver, and if it supports this mechanism, whether or not it has loaded a particular key into its local trusted key stash.

Type\Query	is-ta	not-ta	invalid
Vnew	A	SERVFAIL	SERVFAIL
Vold	SERVFAIL	A	SERVFAIL
Vleg	A	A	SERVFAIL
nonV	A	A	A

A "Vnew" response pattern says that the nominated key is trusted by the resolver and has been loaded into its local trusted key stash. A "Vold" response pattern says that the nominated key is not yet trusted by the resolver in its own right. A "Vleg" response pattern is indeterminate, and a "nonV" response pattern indicates that the resolver does not perform DNSSEC validation.

5. Sentinel Test Result Considerations

The description in the previous section describes a simple situation where the test queries were being passed to a single recursive resolver that directly queried authoritative name servers, including the root servers.

There is also the common case where the end client is configured to use multiple resolvers. In these cases the SERVFAIL responses from one resolver will prompt the end client to repeat the query against one of the other configured resolvers.

If any of the client's resolvers are non-validating resolvers, the tests will result in the client reporting that it has a non-validating DNS environment ("nonV"), which is effectively the case.

If all of the client resolvers are DNSSEC-validating resolvers, but some do not support this trusted key mechanism, then the result will be indeterminate with respect to trusted key status ("Vleg"). Simlarly, if all the client's resolvers support this mechanism, but some have loaded the key into the trusted key stash and some have not, then the result is indeterminate ("Vleg").

There is also the common case of a recursive resolver using a forwarder.

If the resolver is non-validating, and it has a single forwarder clause, then the resolver will presumably mirror the capabilities of the forwarder target resolver. If this non-validating resolver it has multiple forwarders, then the above considerations will apply.

If the validating resolver has a forwarding configuration, and uses the CD flag on all forwarded queries, then this resolver is acting in a manner that is identical to a standalone resolver. The same consideration applies if any one one of the forwarder targets is a non-validating resolver. Similarly, if all the forwarder targets do not apply this trusted key mechanism, the same considerations apply.

A more complex case is where the following conditions all hold:

- both the validating resolver and the forwarder target resolver support this trusted key sentinel mechanism, and
- the local resolver's queries do not have the CD bit set, and
- the trusted key state differs between the forwarding resolver and the forwarder target resolver

then either the outcome is indeterminate validating ("Vleg"), or a case of mixed signals (SERVFAIL in all three responses), which is similarly an indeterminate response with respect to the trusted key state.

6. Security Considerations

This document describes a mechanism to allow users to determine the trust state of root zone key signing keys in the DNS resolution system that they use.

The mechanism does not require resolvers to set otherwise unauthenticated responses to be marked as authenticated, and does not alter the security properties of DNSSEC with respect to the interpretation of the authenticity of responses that are so marked.

The mechanism does not require any further significant processing of DNS responses, and queries of the form described in this document do not impose any additional load that could be exploited in an attack over the the normal DNSSEC validation processing load.

7. IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this version of the document.]

8. Acknowledgements

This document has borrowed extensively from [RFC8145] for the introductory text, and the authors would like to acknowledge and thank the authors of that document both for some text excerpts and for the more general stimulation of thoughts about monitoring the progress of a roll of the Key Signing Key of the Root Zone of the DNS.

The authors would like the especially thank Joe Abley, Mehmet Akcin, Mark Andrews, Richard Barnes, Ray Bellis, Stephane Bortzmeyer, David Conrad, Ralph Dolmans, Steinar Haug, Bob Harold, Wes Hardaker, Paul Hoffman, Matt Larson, Edward Lewis, George Michaelson, Benno Overeinder, Matthew Pounsett, Andreas Schulze, Mukund Sivaraman, Petr Spacek. Andrew Sullivan, Paul Vixie, Duane Wessels and Paul Wouters for their helpful feedback.

[TODO: Add people who have contributed!]

9. Change Log

Note that this document is being worked on in GitHub - see Abstract. The below is mainly large changes, and is not authoritative.

From -00 to 01:

- Added a conversational description of how the system is intended to work.
- Clarification that this is for the root.
- Changed the label template from _is-ta-<tag> to kskroll-sentinel-is-ta-<tag-index>. This is because BIND (at least) will not allow records which start with an underscore to have address records (CNAMEs, yes, A/AAAA no). Some browsers / operating systems also will not fetch resources from names which start with an underscore.

10. References

10.1. Normative References

[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>.

[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>.

10.2. Informative References

[RFC8145] Wessels, D., Kumari, W., and P. Hoffman, "Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)", RFC 8145, DOI 10.17487/RFC8145, April 2017, <https://www.rfc-editor.org/info/rfc8145>.

Authors' Addresses

Geoff Huston

Email: gih@apnic.net
URI: http://www.apnic.net

Joao Silva Damas

Email: joao@apnic.net
URI: http://www.apnic.net

Warren Kumari

Email: warren@kumari.net