

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 7, 2016

W. Kumari  
Google  
G. Huston  
APNIC  
E. Hunt  
Internet Systems Consortium  
R. Arends  
ICANN  
October 5, 2015

Signalling of DNS Security (DNSSEC) Trust Anchors  
draft-wkumari-dnsop-trust-management-01

Abstract

[ Editor note: This originally included a mechanism to actually roll the keys (like [RFC5011](#) does), but feedback from the Prague meeting indicated a strong preference for signalling only. ]

This document describes a simple method for validating recursive resolvers to signal their configured list of DNSSEC trust anchors. This mechanism allows the trust anchor maintainer to monitor the progress of the migration to a new trust anchor, and so predict the effect before decommissioning the existing trust anchor.

It is primarily aimed at the root DNSSEC trust anchor, but should be applicable to trust anchors elsewhere in the DNS as well.

[ Ed note - informal summary: One of the big issues with rolling the root key is that it is unclear who all is using [RFC5011](#), who all has successfully fetched and installed the new key, and, most importantly, who all will die when the old key is revoked. By having resolvers query for a special QNAME, comprised of the list of TAs that it knows about, we effectively signal "up stream" to the authoritative server. By querying for this name, the recursive exposes its list of TAs to this authoritative server. This allows the TA maintainer to gather information relating to the state of TAs on resolvers.]

[ Ed note: Text inside square brackets ([ ]) is additional background information, answers to frequently asked questions, general musings, etc. They will be removed before publication.]

[ This document is being collaborated on in Github at: <https://github.com/wkumari/draft-wkumari-dnsop-trust-management>. The most recent version of the document, open issues, etc should all be available here. The authors (gratefully) accept pull requests ]

Internet-Draft [draft-wkumari-dnsop-trust-management](#)

October 2015

## 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 April 7, 2016.

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">1.1.</a>	Requirements notation . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Trust Anchor Telemetry . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	TAT Name Format . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Sending the Trust Anchor Telemetry Query . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Known issues and limitations . . . . .	<a href="#">5</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">6</a>
<a href="#">7.</a>	Contributors . . . . .	<a href="#">7</a>

<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">7</a>
<a href="#">9.</a>	References . . . . .	<a href="#">7</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">7</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">7</a>
<a href="#">Appendix A.</a>	Changes / Author Notes. . . . .	<a href="#">7</a>

---

Internet-Draft      [draft-wkumari-dnsop-trust-management](#)      October 2015

Authors' Addresses . . . . .	<a href="#">8</a>
------------------------------	-------------------

## [1.](#) Introduction

When a DNSSEC-aware resolver performs validation, it requires a trust anchor to validate the DNSSEC chain. An example of a trust anchor is the so called DNSSEC "root key". For a variety of reasons, this trust anchor may need to be replaced or "rolled", to a new key (potentially with a different algorithm, different key length, etc.).

[RFC5011] provides a secure mechanism to do this, but operational experience has demonstrated a need for some additional functionality that was not foreseen.

During the current efforts to roll the IANA DNSSEC "root key", it has become clear that, in order to predict (and minimize) outages caused by rolling the key, real-time information about the uptake of the new key will be needed.

This document defines a mechanism ("trust anchor telemetry") by which validating resolvers can provide information about their configured trust anchors. Readers of this document are expected to be familiar with the contents of [\[RFC7344\]](#) and [\[RFC5011\]](#).

### [1.1.](#) 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.](#) Trust Anchor Telemetry

The purpose of the mechanism described in this document is to allow the trust anchor maintainer to determine how widely deployed a given trust anchor is. This information is signaled from the validating resolver to the authoritative servers serving the zone in which the

trust anchor lives by sending a periodic query to that zone. The query type of the TAT Query is NULL. The query name is a TAT Name, a format which encodes the list of the trust anchors for that zone that are currently in use by the validating resolver, along with status information about each key. Telemetry information can be retrieved by the trust anchor maintainer by examining logged queries that match the TAT Name format.

## [2.1.](#) TAT Name Format

The TAT Name is generated as follows:

1. For each trust anchor that the resolver knows and/or is using, generate a string consisting of the key's Algorithm in decimal format, followed by an underscore ('\_'), followed by the derived Key Tag in decimal format. [NOTE: If we used hex, this could just be AAKKKK, no need for a punctuation mark, but it would be less human-readable.]
2. Follow each string with a character indicating the status of the key from the resolver's point of view:  
  
S Static trust anchor, not subject to [\[RFC5011\]](#)  
  
A Accepted trust anchor  
  
P Pending trust anchor, not yet accepted  
  
R Revoked trust anchor
3. Sort the list in numerically ascending order of Algorithm and Key Tag.
4. Concatenate the list, with each string used as a label in a domain name.
5. Append \_tat.<domain>

Assuming no more than two digits for the Algorithm and five for the Key Tag, a TAT Name for the root zone can encode up to 24 trust anchors. [ Someone should probably check my math. QUESTION: Do we need to specify what will happen in the crazy case of a resolver having configured more than 24 trust anchors? -each ]

Examples:

- o If the resolver has a single trust anchor statically configured for the root zone, with an algorithm of RSASHA256 and a Key Tag of 19036, it would emit a query for 8\_19036S.\_tat.
- o If the resolver were configured to use [[RFC5011](#)] trust anchor management, it would send 8\_19036A.\_tat.
- o If a new key with Key Tag 1999 was added to the root zone and had been seen by the resolver, but was too recent to have been accepted as a trust anchor, then the resolver would send a query

for 8\_1999P.8\_19036A.\_tat. After the hold-down timer ([RFC5011](#) [Section 2.2](#)) had expired, the resolver would send a query for 8\_1999A.8\_19036A.\_tat.

- o If there is a separate static trust anchor configured for example.com with an algorithm of RSASHA1 and a Key Tag of 1234, the resolver would send a query for 5\_1234S.\_tat.example.com.

NOTE: The format of the TAT Name requires that Key Tags MUST be unique, at least within "recent" history. If (e.g. during a Key Ceremony) a new DNSKEY is generated whose derived Key Tag collides with an existing one (statistically unlikely, but not impossible) this DNSKEY MUST NOT be used, and a new DNSKEY MUST be generated. [ Ed note: This is to prevent two successive keys having the same keytag (e.g: 123), and then seeing "8\_123A." - which 123 key was that?! [RFC4034 Appendix B](#) admonition: "Implementations MUST NOT assume that the key tag uniquely identifies a DNSKEY RR", but this appears to be targeted at validating resolver implementations.]

### [3.](#) Sending the Trust Anchor Telemetry Query

When a compliant validating resolver performs the "Active Refresh"

query as part of its [RFC5011](#) ([\[RFC5011\] Section 2.3](#)) processing it will also send a query for the TAT Name. This SHOULD be the default for compliant resolvers.

It will receive back either a negative response (e.g. NXDOMAIN), or a (nonsensical) answer. As the entire purpose of this query is to send information from a recursive resolver to the nameservers that serve the zone containing a trust anchor, the response to the query contains no useful information and MUST be ignored.

#### [4.](#) Known issues and limitations

This solution is designed to provide a rough idea of the rate of uptake of a new key during a key rollover; perfect visibility is not achievable. In particular:

1. Only compliant resolvers will send telemetry queries; no information is provided from legacy resolvers, or from those who choose to disable this functionality.
2. The trust anchor maintainer has no way to differentiate a query that is emitted by the resolver itself from a query that is forwarded through the resolver. (Note, however, that forwarded queries are likely to be infrequent; responses to TAT queries will in most cases be negatively cached with an NXDOMAIN covering

the \_TAT subdomain; subsequent client queries would be answered from the cache rather than forwarded to the trust anchor zone.)

3. An attacker could forge TAT queries to trick the trust anchor maintainer into a false impression of the adoption rate of a new trust anchor, if there were a perceived advantage to doing so.

#### [ Open Questions:

1: In order to disambiguate queries from resolvers versus those forwarded through resolvers (or being recursed because of users behind the resolver) we *could* add craziness like having resolvers include ephemeral UUIDs or something...). Is this worth doing? (Personally I think not...)

2: We *\*could\** also specify that compliant resolvers **MUST NOT** forward queries of type TDS to try limit this. Worth doing? This is some of the reason for having a defined type.

3: The authoritative server *\*could\** return a record with a long TTL to stop queries (if it knows that it is not doing a rollover in the near future). This seems like a simple option, worth doing? (I think so). (each thinks not.)

## 5. IANA Considerations

[ Ed note: This is largely a place holder. The real IANA considerations section will require updating things like the DPS, etc. ]

The format of the TAT query requires that Key Tags **MUST** be unique, at least within an interval. If, during a Key Ceremony, a new DNSKEY is generated whose derived Key Tag collides with an exiting one (statistically unlikely, but not impossible) this DNSKEY **MUST NOT** be used, and a new DNSKEY **MUST** be generated.

There will need to be some text added to the DNSSEC Ceremony to handle this.

## 6. Security Considerations

[ Ed note: a placeholder as well ]

This mechanism causes a recursive resolver to disclose the list of trust anchors that it knows about to the authoritative servers serving the zone containing the TA (or attackers able to monitor the path between these devices). It is conceivable that an attacker may be able to use this to determine that a resolver trusts an outdated /

revoked trust anchor and perform a MitM attack. This would also require the attacker to have factored the private key. This seems farfetched....

## 7. Contributors

A number of people contributed significantly to this document, including Joe Abley, Paul Wouters, Paul Hoffman. Wes Hardaker and

David Conrad.

## 8. Acknowledgements

## 9. References

### 9.1. 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [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, <<http://www.rfc-editor.org/info/rfc4034>>.
- [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>>.
- [RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", [RFC 7344](#), DOI 10.17487/RFC7344, September 2014, <<http://www.rfc-editor.org/info/rfc7344>>.

### 9.2. Informative References

- [I-D.ietf-sidr-iana-objects]  
Manderson, T., Vegoda, L., and S. Kent, "RPKI Objects issued by IANA", [draft-ietf-sidr-iana-objects-03](#) (work in progress), May 2011.

## Appendix A. Changes / Author Notes.

[RFC Editor: Please remove this section before publication ]

From -00 to -01.1:

Kumari, et al.

Expires April 7, 2016

[Page 7]

---

Internet-Draft

[draft-wkumari-dnsop-trust-management](#)

October 2015

- o Ripped all the actual keyroll logic out.



- o Added Geoff, Evan and Roy as authors.
- o Added some limitations and known issues.
- o Renamed to TAT, added tag describing the state of the TA.

From -00.1 to -00 (published):

- o Integrated comments and feedback from DRC and Paul Hoffman.
- o Use \_ as a prefix to make clear it is meta-type (drc)

From -00.0 to -00.1

- o Initial draft, written in an airport lounge.

#### Authors' Addresses

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

Email: warren@kumari.net

Geoff Huston  
 APNIC  
 6 Cordelia St  
 South Brisbane QLD 4001  
 AUS

Email: gih@apnic.net

Evan Hunt  
 Internet Systems Consortium  
 950 Charter St  
 Redwood City, CA 94063  
 US

Email: each@isc.org

---

Internet-Draft

[draft-wkumari-dnsop-trust-management](#)

October 2015

Roy Arends  
ICANN  
12025 Waterfront Drive, Suite 300  
Los Angeles, CA 90094-2536  
US

Email: [roy.arends@icann.org](mailto:roy.arends@icann.org)

