Domain Name System Operations   P. Ebersman
Internet-Draft   Comcast
Intended status: Informational   C. Griffiths
Expires: November 13, 2015

                                                       W. Kumari
                                                          Google
                                                    J. Livingood
                                                         Comcast
                                                        R. Weber
                                                         Nominum
                                                    May 12, 2015

            Definition and Use of DNSSEC Negative Trust Anchors
               draft-ietf-dnsop-negative-trust-anchors-09

Abstract

   DNS Security Extensions (DNSSEC) is now entering widespread
   deployment. However, domain signing tools and processes are not yet
   as mature and reliable as those for non-DNSSEC-related domain
   administration tools and processes. Negative Trust Anchors
   (described in this document) can be used to mitigate DNSSEC
   validation failures.

   [RFC Editor: Please remove this before publication. This document is
   being stored in github at https://github.com/wkumari/draft-livingood-
   dnsop-negative-trust-anchors . Authors accept pull requests, and keep
   the latest (edit buffer) versions there, so commenters can follow
   along at home.]

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction and motivation

This document defines a Negative Trust Anchor, which can be used
during the transition to ubiquitous DNSSEC deployment. Negative
Trust Anchors (NTAs) are configured locally on a validating DNS
recursive resolver to shield end users from DNSSEC-related
authoritative name server operational errors. Negative Trust Anchors
are intended to be temporary, and should not be distributed by IANA
or any other organization outside of the administrative boundary of
the organization locally implementing a Negative Trust Anchor.
Finally, Negative Trust Anchors pertain only to DNSSEC and not to
Public Key Infrastructures (PKI) such as X.509.

DNSSEC has now entered widespread deployment. However, the DNSSEC
signing tools and processes are less mature and reliable than those
for non-DNSSEC-related administration. As a result, operators of DNS
recursive resolvers, such as Internet Service Providers (ISPs),
occasionally observe domains incorrectly managing DNSSEC-related
resource records. This mismanagement triggers DNSSEC validation
failures, and then causes large numbers of end users to be unable to
reach a domain. Many end users tend to interpret this as a failure
of their ISP or resolver operator, and may switch to a non-validating
resolver or contact their ISP to complain, rather than seeing this as
a failure on the part of the domain they wanted to reach. Without
the techniques in this document, this pressure may cause the resolver
operator to disable (or simply not deploy) DNSSEC validation. Use of
a Negative Trust Anchor to temporarily disable DNSSEC validation for
a specific misconfigured domain name immediately restores access for
end users. This allows the domain's administrators to fix their
misconfiguration, while also allowing the organization using the
Negative Trust Anchor to keep DNSSEC validation enabled and still
reach the misconfigured domain.

It is worth noting the following text from [RFC4033] - "In the final
analysis, however, authenticating both DNS keys and data is a matter
of local policy, which may extend or even override the protocol
extensions defined in this document set." A responsibility (one of
many) of a caching server operator is to "protect the integrity of
the cache."

1.1. Definition of a Negative Trust Anchor

Trust Anchors are defined in [RFC5914]. A trust anchor should be
used by a validating caching resolver as a starting point for
building the authentication chain for a signed DNS response. By way
of analogy, negative trust anchors stop validation of the
authentication chain. Instead, the validator treats any upstream
responses as if the zone is unsigned and does not set the AD bit in

responses it sends to clients. Note that this is a behavior, and not
a separate resource record. This Negative Trust Anchor can
potentially be implemented at any level within the chain of trust and
would stop validation from that point in the chain down. Validation
starts again if there is a positive trust anchor further down in the
chain. For example, if there is a NTA at example.com, and a positive
trust anchor at foo.bar.example.com, then validation resumes for
foo.bar.example.com and anything below it.

1.2. Domain Validation Failures

A domain name can fail validation for two general reasons: a
legitimate security failure such as due to an attack or compromise of
some sort, or as a result of misconfiguration on the part of an
domain administrator. As domains transition to DNSSEC, the most
common reason for a validation failure has been misconfiguration.
Thus, domain administrators should be sure to read [RFC6781] in full.
They should also pay special attention to Section 4.2, pertaining to
key rollovers, which appear to be the cause of many recent validation
failures.

It is also possible that some DNSSEC validation failures could arise
due to differences in how different software developers interpret
DNSSEC standards and/or how those developers choose to implement
support for DNSSEC. For example, it is conceivable that a domain may
be DNSSEC signed properly, and one vendor's DNS recursive resolvers
will validate the domain but other vendors' software may fail to
validate the domain.

1.3. End User Reaction

End users generally do not know of, understand, or care about the
resolution process that causes connections to happen. This is by
design: the point of the DNS is to insulate users from having to
remember IP addresses through a friendlier way of naming systems. It
follows from this that end users do not, and should not, be expected
to know about DNSSEC, validation, or anything of the sort. As a
result, end users may misinterpret the failure to reach a domain due
to DNSSEC-related misconfiguration . They may (incorrectly) assume
that their ISP is purposely blocking access to the domain or that it
is a performance failure on the part of their ISP (especially of the
ISP's DNS servers). They may contact their ISP to complain, which
will incur cost for their ISP. In addition, they may use online
tools and sites to complain of this problem, such as via a blog, web
forum, or social media site, which may lead to dissatisfaction on the
part of other end users or general criticism of an ISP or operator of
a DNS recursive resolver.

As end users publicize these failures, others may recommend they
switch from security-aware DNS resolvers to resolvers not performing
DNSSEC validation. This is a shame since the ISP or other DNS
recursive resolver operator is actually doing exactly what they are
supposed to do in failing to resolve a domain name; this is the
expected result when a domain can no longer be validated and it
protects end users from a potential security threat. Use of a
Negative Trust Anchor would allow the ISP to specifically remedy the
failure to reach that domain, without compromising security for other
sites. This would result in a satisfied end user, with minimal
impact to the ISP, while maintaining the security of DNSSEC for
correctly maintained domains.

1.4. Switching to a Non-Validating Resolver is Not Recommended

As noted in Section 1.3, some people may consider switching to an
alternative, non-validating resolver themselves, or may recommend
that others do so. But if a domain fails DNSSEC validation and is
inaccessible, this could very well be due to a security-related
issue. In order to be as safe and secure as possible, end users
should not change to DNS servers that do not perform DNSSEC
validation as a workaround, and people should not recommend that
others do so either. Domains that fail DNSSEC for legitimate reasons
(versus misconfiguration) may be in control of hackers or there could
be other significant security issues with the domain.

Thus, switching to a non-validating resolver to restore access to a
domain that fails DNSSEC validation is not a recommended practice, is
bad advice to others, is potentially harmful to end user security.

2. Use of a Negative Trust Anchor

Technical personnel trained in the operation of DNS servers MUST
confirm that a failure is due to misconfiguration, as a similar
breakage could have occurred if an attacker gained access to a
domain's authoritative servers and modified those records or had the
domain pointed to their own rogue authoritative servers. They should
also confirm that the domain is not intentionally broken, such as for
testing purposes as noted in Section 6. Finally, they should make a
reasonable attempt to contact the domain owner of the misconfigured
zone, preferably prior to implementing the Negative Trust Anchor.
Involving trained technical personnel is costly, but operational
experience suggests that this is a very rare event, usually on the
order of once per quarter (or even less).

It is important for the resolver operator to confirm that the domain
is still under the ownership / control of the legitimate owner of the
domain in order to ensure that disabling validation for a specific

domain does not direct users to an address under an attacker's
control. Contacting the domain owner and telling them the DNSSEC
records that the resolver operator is seeing allows the resolver
operator to determine if the issue is a DNSSEC misconfiguration or an
attack.

In the case of a validation failure due to misconfiguration of a TLD
or popular domain name (such as a top 100 website), content or
services in the affected TLD or domain could be inaccessible for a
large number of users. In such cases, it may be appropriate to use a
Negative Trust Anchor as soon as the misconfiguration is confirmed.
An example of a list of "top N" websites is the "Alexa Top 500 Sites
on the Web" [Alexa], , or a list of the of the most-accessed names in
the resolver's cache.

Once a domain has been confirmed to fail DNSSEC validation due to a
DNSSEC-related misconfiguration, an ISP or other DNS recursive
resolver operator may elect to use a Negative Trust Anchor for that
domain or sub-domain. This instructs their DNS recursive resolver to
temporarily NOT perform DNSSEC validation at or in the misconfigured
domain. This immediately restores access to the domain for end users
while the domain's administrator corrects the misconfiguration(s).
It does not and should not involve turning off validation more
broadly.

A Negative Trust Anchor MUST only be used for a limited duration.
Implementors SHOULD allow the operator using the Negative Trust
Anchor to set an end time and date associated with any Negative Trust
Anchor. Optimally, this time and date is set in a DNS recursive
resolver's configuration, though in the short-term this may also be
achieved via other systems or supporting processes. Use of a
Negative Trust Anchor MUST NOT be automatic.

Finally, a Negative Trust Anchor SHOULD be used only in a specific
domain or sub-domain and MUST NOT affect validation of other names up
the authentication chain. For example, a Negative Trust Anchor for
zone1.example.com would affect only names at or below
zone1.example.com, and validation would still be performed on
example.com, .com, and the root ("."). This Negative Trust Anchor
also SHOULD NOT affect names in another branch of the tree (such as
example.net). In another example, a Negative Trust Anchor for
example.com would affect only names within example.com, and
validation would still be performed on .com, and the root ("."). In
this scenario, if there is a (probably manually configured) trust
anchor for zone1.example.com, validation would be performed for
zone1.example.com and subdomains of zone1.example.com.

3. Managing Negative Trust Anchors

   While Negative Trust Anchors have proven useful during the early
   stages of DNSSEC adoption, domain owners are ultimately responsible
   for managing and ensuring their DNS records are configured correctly.

   Most current implementations of DNS validating resolvers currently
   follow [RFC4033] on configuring a Trust Anchor using either a public
   key as in a DNSKEY RR or a hash of a public key as in a DS RR.

   Different DNS validators may have different configuration names for a
   Negative Trust Anchor. For examples see Appendix A.

   An NTA placed at a node where there is a configured positive trust
   anchor MUST take precedence over that trust anchor, effectively
   disabling it. Implementations SHOULD issue a warning or
   informational message when this occurs, so that operators are not
   surprised when this happens.

3.1. Alerting Users to NTA Use

   End users of a DNS recursive resolver or other people may wonder why
   a domain that fails DNSSEC validation resolves with a supposedly
   validating resolver. As a result, implementors should consider
   transparently disclosing those Negative Trust Anchors which are
   currently in place or were in place in the past, such as on a website
   [Disclosure-Example].

   This is particularly important since there is currently no special
   DNS query response code that could indicate to end users or
   applications that a Negative Trust Anchor is in place. Such
   disclosures should optimally include both the data and time that the
   Negative Trust Anchor was put in place and when it was removed.

4. Removal of a Negative Trust Anchor

   As explored in Section 8.1, using an NTA once the zone correctly
   validates can have security considerations. It is therefore
   RECOMMENDED that NTA implementors SHOULD periodically attempt to
   validate the domain in question, for the period of time that the
   Negative Trust Anchor is in place, until such validation is again
   successful. NTAs MUST expire automatically when their configured
   lifetime ends. The lifetime MUST NOT exceed a week. Before removing
   the Negative Trust Anchor, all authoritative resolvers listed in the
   zone should be checked (due to anycast and load balancers it may not
   be possible to check all instances).

Once all testing succeeds, a Negative Trust Anchor should be removed
as soon as is reasonably possible. One possible method to
automatically determine when the NTA can be removed is to send a
periodic query for type SOA at the NTA node; if it gets a response
that it can validate (whether the response was an actual SOA answer
or a NOERROR/NODATA with appropriate NSEC/NSEC3 records), the NTA is
presumed no longer to be necessary and is removed. Implementations
SHOULD, by default, perform this operation. Note that under some
circumstances this is undesirable behavior (for example, if
www.example.com has a bad signature, but example.com/SOA is fine) and
so implementations may wish to allow the operator to override this
spot-check / behavior.

When removing the NTA, the implementation SHOULD remove all cached
entries at and below the NTA node.

5. Comparison to Other DNS Misconfigurations

Domain administrators are ultimately responsible for managing and
ensuring their DNS records are configured correctly. ISPs or other
DNS recursive resolver operators cannot and should not correct
misconfigured A, CNAME, MX, or other resource records of domains for
which they are not authoritative. Expecting non-authoritative
entities to protect domain administrators from any misconfiguration
of resource records is therefore unrealistic and unreasonable, and in
the long-term is harmful to the delegated design of the DNS and could
lead to extensive operational instability and/or variation.

With DNSSEC breakage, it is often possible to tell that there is a
misconfiguration by looking at the data and not needing to guess what
it should have been.

6. Intentionally Broken Domains

Some domains, such as dnssec-failed.org, have been intentionally
broken for testing purposes
[Measuring-DNSSEC-Validation-of-Website-Visitors] [Netalyzr]. For
example, dnssec-failed.org is a DNSSEC-signed domain that is broken.
If an end user is querying a validating DNS recursive resolver, then
this or other similarly intentionally broken domains should fail to
resolve and should result in a "Server Failure" error (RCODE 2, also
known as 'SERVFAIL'). If such a domain resolved successfully, then
it is a sign that the DNS recursive resolver is not fully validating.

Organizations that utilize Negative Trust Anchors should not add a
Negative Trust Anchor for any intentionally broken domain. Such
additions are prevented by the requirement that the operator attempt
to contact the administrators for the zone that has broken DNSSEC.

Organizations operating an intentionally broken domain may wish to consider adding a TXT record for the domain to the effect of "This domain is purposely DNSSEC broken for testing purposes".

[7](). Discovering broken domains

Discovering that a domain is DNSSEC broken as result of an operator error instead of an attack is not trivial, and the examples here should be vetted by an experienced professional before taking the decision on implementing an negative trust anchor.

One of the key thing to look for when looking at a DNSSEC broken domain is consistency and history. It therefore is good if you have the ability to look at the server's DNS traffic over a long period of time or have a database that stores DNS names associated answers (this is often referred to as a "passive DNS database"). Another invaluable tool is dnsviz (http://www.dnsivz.net) which also stores DNSSEC related data historically. The drawback here is that you need to have it test the domain before the incident occurs.

The first and easiest thing to check is if the failure of the domain is consistent across different software implementations. If not, you want to inform the vendor where it fails so that the vendor can look more deeply into the issue.

The next thing is to figure out what the actual failure mode is. There are several tools to do this, an incomplete list includes:

o DNSViz (http://dnsviz.net)

o Verisign DNSSEC debugger (http://dnssec-debugger.verisignlabs.com)

o iis.se DNS check (http://dnscheck.iis.se)

most of these tools are open source and can be installed locally. However, using the tools over the Internet has the advantage of providing visibility from a different point.

Once you figure out what the error is, you need to check if it shows consistently around the world and from all authoritative servers. Use DNS Tools (dig) or DNS looking glasses to verify this. An error that is consistently the same is more likely to be operator caused than an attack. Also if the output from the authoritative server is consistently different from the resolvers output this hints to an attack rather then an error, unless there is EDNS0 client subnet (draft-ietf-dnsop-edns-client-subnet) applied to the domain.

A last check is to look at the actual DNS data. Is the result of the
query still the same or has it changed? While a lot of DNSSEC errors
occur on events that change DNSSEC data, the actual record someone
wants to go to often stays the same. If the data is the same, this
is an indication (not a guarantee) that the error is operator caused.
Keep in mind that with DNS being used to globally balance traffic the
data associated to a name might be different in different parts of
the Internet.

Here are some examples of common DNSSEC failures that have been seen
as operator signing errors on the Internet:

o RRSIG timing issue. Each signature has an inception time and
   expiry time, between which it is valid. Letting this time expire
   without creating a new signature is one of the most common DNSSEC
   errors. To a lesser extent, this also occurs if signatures were
   made active before the inception time. For all of these errors
   your primary check is to check on the data. Signature expiration
   is also about the only error we see on actual data (like
   www.example.com). All other errors are more or less related to
   dealing with the chain of trust established by DS records in the
   parent zone and DNSKEYs in the child zones. These mostly occur
   during key rollovers, but are not limited to that.

o DNSKEYs in child zone don't match the DS record in the parent
   zone. There is a big variation of this that can happen at any
   point in the key lifecycle. DNSViz is the best tools to show
   problems in the chain. If you debug yourself use dig +multiline
   so that you can see the key id of a DNSKEY. Common Variations of
   this can be:

   * DS pointing to a non existent key in the child zone. Questions
      for consideration here include: Has there ever been a key (and,
      if so, was it used)? Has there been a recent change in the
      DNSKEY RRSet (indicating a key rollover)? Has the actual data
      in the zone changed? Is the zone DNSSEC signed at all and has
      it been in the past?

   * DS pointing to an existent key, but no signatures are made with
      the key. The checks above should be done, with the addition of
      checking if another key in the DNSKEY RRSet is now used to sign
      the records.

   * Data in DS or DNSKEY doesn't match the other. This is more
      common in initial setup when there was a copy and paste error.
      Again checking history on data is the best you can do there.

All of the above is just a starting point for consideration when
deciding whether or not to deploy a trust anchor. It is not possible
to provide a simple checklist to run through to determine whether a
domain is broken because of an attack or an operator error.

8. Other Considerations

8.1. Security Considerations

End to end DNSSEC validation will be disabled during the time that a
Negative Trust Anchor is used. In addition, the Negative Trust
Anchor may be in place after the point in time when the DNS
misconfiguration that caused validation to break has been fixed.
Thus, there may be a gap between when a domain has been re-secured
and when a Negative Trust Anchor is removed. In addition, a Negative
Trust Anchor may be put in place by DNS recursive resolver operators
without the knowledge of the authoritative domain administrator for a
given domain name. However, attempts SHOULD be made to contact and
inform the domain administrator prior to putting the NTA in place.

One side effect of implementing an NTA is that it may break client
applications that assume that a domain is signed and expect an AD bit
in the response. It is expected that many application that require
DNSSEC for a domain will perform their own validation, and so this
should not be a major issue.

8.2. Privacy Considerations

There are no privacy considerations in this document.

8.3. IANA Considerations

There are no IANA considerations in this document.

9. Acknowledgements

Several people made contributions of text to this document and/or
played an important role in the development and evolution of this
document. This in some cases included performing a detailed review
of this document and then providing feedback and constructive
criticism for future revisions, or engaging in a healthy debate over
the subject of the document. All of this was helpful and therefore
the following individuals merit acknowledgement: Joe Abley,John
Barnitz, Tom Creighton, Marco Davids, Brian Dickson, Patrik Falstrom,
Tony Finch, Chris Ganster, Olafur Gudmundsson, Peter Hagopian, Wes
Hardaker, Paul Hoffman, Shane Kerr, Murray Kucherawy, Rick Lamb, Marc
Lampo, Scott Rose, Ted Lemon, Antoin Verschuren, Paul Vixie, Patrik
Wallstrom, W.C.A. Wijngaards, Nick Weaver

Edward Lewis, Evan Hunt, Andew Sullivan and Tatuya Jinmei provided
especially large amounts of text and / or detailed review.

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, March 2005.

[RFC5914]   Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor
            Format", RFC 5914, June 2010.

[RFC6781]   Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC
            Operational Practices, Version 2", RFC 6781, December
            2012.

10.2. Informative References

[Alexa]   Alexa, an Amazon.com Company, "Alexa "The top 500 sites on
            the web. "", , May 2015, <http://www.alexa.com/topsites>.

[Disclosure-Example]
            Comcast, "faa.gov Failing DNSSEC Validation (Fixed)",
            Comcast , February 2013,
            <http://dns.comcast.net/index.php/entry/
            faa-gov-failing-dnssec-validation-fixed>.

[Measuring-DNSSEC-Validation-of-Website-Visitors]
            Mens, J., "Is my Web site being used via a DNSSEC-
            validator?", July 2012, <http://jpmens.net/2012/07/30/
            is-my-web-site-being-used-via-dnssec-validator/>.

[Netalyzr]
            Weaver, N., Kreibich, C., Nechaev, B., and V. Paxson,
            "Implications of Netalyzr's DNS Measurements", Securing
            and Trusting Internet Names, SATIN 2011 SATIN 2011, April
            2011, <http://conferences.npl.co.uk/satin/presentations/
            satin2011slides-Weaver.pdf>.

[Unound-Configuration]
            Wijngaards, W., "Unbound: How to Turn Off DNSSEC", June
            2010, <http://unbound.net/documentation/
            howto_turnoff_dnssec.html>.

Appendix A. Configuration Examples

   The section contains example configurations to achieve Negative Trust
   Anchor functionality for the zone foo.example.com.

   Note: These are simply examples - nameserver operators are expected
   to test and understand the implications of these operations. Note
   also that some of available implementations may not implement all
   recommended functionality in this document. In that case it is
   advisable to request the developer or vendor of the implementation to
   support the missing feature, rather than start using the incomplete
   implementation.

A.1. NLNet Labs Unbound

   Unbound lets us simply disable validation checking for a specific
   zone by adding configuration statements to unbound.conf:

   server:
           domain-insecure: "foo.example.com"

   Using the 'unbound-control' command one can add and remove Negative
   Trust Anchors without restarting the nameserver.

      Using the "unbound-control" command:
           list_insecure   list domain-insecure zones
           insecure_add zone   add domain-insecure zone
           insecure_remove zone   remove domain-insecure zone

   Items added with the "unbound-control" command are added to the
   running server and are lost when the server is restarted. Items from
   unbound.conf stay after restart.

   For additional information see [Unound-Configuration]

A.2. ISC BIND

   Use the "rndc" command:

```
   nta -dump
            List all negative trust anchors.
   nta [-lifetime duration] [-force] domain [view]
            Set a negative trust anchor, disabling DNSSEC validation
            for the given domain.
            Using -lifetime specifies the duration of the NTA, up
            to one week. The default is one hour.
            Using -force prevents the NTA from expiring before its
            full lifetime, even if the domain can validate sooner.
   nta -remove domain [view]
            Remove a negative trust anchor, re-enabling validation
            for the given domain.
```

A.3. Nominum Vantio

   **

   negative-trust-anchors

   Format: name

   Command Channel: view.update name=world negative-trust-
   anchors=(foo.example.com)

   Command Channel: resolver.update name=res1 negative-trust-
   anchors=(foo.example.com)

   Description: Disables DNSSEC validation for a domain, even if the
   domain is under an existing security root.

Appendix B. Document Change Log

   [RFC Editor: This section is to be removed before publication]

   -08 to -09

   o Clarified that an NTA MUST take precedece over a positive, local
      TA.

   -07 to -08

   o Added some cleanup from Paul Hoffman and Evan Hunt.

   o Some better text on how to make Unbound do this, provided by
      W.C.A. Wijngaards.

   -06 to -07

o Addressed a large number of comments from Paul Hoffman, Scott Rose
   and some more from Jinmei.

-05 to -06

o A bunch of comments from Tony Finch.

-04 to -05

o A large bunch of cleanups from Jinmei. Thanks!

o Also clarified that if there is an NTA at foo.bar.baz.example, and
   a positive *trust anchor* at bar.baz.example, the most specific
   wins. I'm not very happy with this text, any additional text
   gratefully accepted...

-03 to -04:

o Addressed some comment from an email from Jinmei that I had
   missed. Turns out others had made many of the same comments, and
   so most had already been addressed.

-02 to -03:

o Included text from Ralph into Appendix B

o A bunch of comments from Andrew Sullivan ('[DNSOP] negative-trust-
   anchors-02" - Mar 18th)

o Updated keywords

-01 to -02:

o Gah! I forgot to run spell check. And I type like a chimpanzee
   with bad hand-eye coordination...

-00 to -01:

o Stole chunks of text from Ed Lewis' mailing list "tirade" :-)

o New rndc usage text from Evan.

o Deleted the (already resolved) open issues from Appendix C, moved
   the unresolved issues into github, resolved them!

o Clarification that automated removal is best removal method, and
   how to implement (Evan Hunt)

o Clarify that an NTA is not a RR (Rick Lamb)

o Grammar fixes.

Ind-07 - WG-00:

o Simply updated name to reflect WG doc.

Individual-00: First version published as an individual draft.

Individual-01: Fixed minor typos and grammatical nits. Closed all
open editorial items.

Individual-02: Simple date change to keep doc from expiring.
Substantive updates planned.

Individual-03: Changes to address feedback from Paul Vixie, by adding
a new section "Limited Time and Scope of Use". Changes to address
issues raised by Antoin Verschuren and Patrik Wallstrom, by adding a
new section "Intentionally Broken Domains" and added two related
references. Added text to address the need for manual investigation,
as suggested by Patrik Falstrom. Added a suggestion on notification
as suggested by Marc Lampo. Made several additions and changes
suggested by Ralf Weber, Wes Hardaker, Nick Weaver, Tony Finch, Shane
Kerr, Joe Abley, Murray Kucherawy, Olafur Gudmundsson.

Individual-04: Moved the section defining a NTA forward, and added
new text to the Abstract and Introduction per feedback from Paul
Hoffman.

Individual-05: Incorporated feedback from the DNSOP WG list received
on 2/17/13 and 2/18/13. This is likely the final version before the
IETF 86 draft cutoff date. Updated references to RFC6781 to RFC6781,
per March Davids.

Individual-06: Added more OPEN issues to continue tracking WG
discussion. No changes in the main document - just expanded issue
tracking.

Individual-07: Refresh document - needs revision and rework before
IETF-91. Planning to add more contributors.

o Using github issue tracker - go see https://github.com/wkumari/
   draft-livingood-dnsop-negative-trust-anchors for more details.

o A bunch of readability improvments.

   o Issue: Notify the domain owner of the validation failure -
      resolved.

   o Issue: Make the NTA as specific as possible - resolved.

Authors' Addresses

   Paul Ebersman
   Comcast
   One Comcast Center
   1701 John F. Kennedy Boulevard
   Philadelphia, PA 19103
   US

   Email: ebersman-ietf@dragon.net

   Chris Griffiths

   Email: cgriffiths@gmail.com

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

   Email: warren@kumari.net
   URI: http://www.google.com

   Jason Livingood
   Comcast
   One Comcast Center
   1701 John F. Kennedy Boulevard
   Philadelphia, PA 19103
   US

   Email: jason_livingood@cable.comcast.com
   URI: http://www.comcast.com

   Ralf Weber
   Nominum

   Email: Ralf.Weber@nominum.com
   URI: http://www.nominum.com