Internet Engineering Task Force (IETF)
Request for Comments: 7646
Category: Informational
ISSN: 2070-1721

P. Ebersman
Comcast
W. Kumari
Google
C. Griffiths
Nominet
J. Livingood
Comcast
R. Weber
Nominum

September 2015

Definition and Use of DNSSEC Negative Trust Anchors

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. This document defines Negative Trust Anchors (NTAs), which can be used to mitigate DNSSEC validation failures by disabling DNSSEC validation at specified domains.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7646.

Table of Contents

1. Introduction and Motivation 3
   1.1. Definition of a Negative Trust Anchor 3
   1.2. Motivations for Negative Trust Anchors 4
        1.2.1. Mitigating Domain Validation Failures 4
        1.2.2. Improving End-User Experience 4
        1.2.3. Avoiding Switching to a Non-validating Resolver 5
2. Use of a Negative Trust Anchor 5
   2.1. Applicability of Negative Trust Anchors 6
3. Managing Negative Trust Anchors 7
   3.1. Alerting Users to Negative Trust Anchor Use 7
4. Removal of a Negative Trust Anchor 7
5. Comparison to Other DNS Misconfigurations 8
6. Intentionally Broken Domains 8
7. Discovering Broken Domains 9
8. Security Considerations 11
9. References 11
   9.1. Normative References 11
   9.2. Informative References 12
Appendix A. Configuration Examples 13
  A.1. NLnet Labs Unbound 13
  A.2. Internet System Consortium (ISC) BIND 14
  A.3. Nominum Vantio 14
Acknowledgements 15
Authors' Addresses 15

1. Introduction and Motivation

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 they 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.

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

Use of an NTA 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 NTA to keep DNSSEC validation enabled and still reach the misconfigured domain.

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 [RFC2119].

1.1. Definition of a Negative Trust Anchor

Trust anchors are defined in [RFC5914]. A trust anchor is used by a validating caching resolver as a starting point for building the authentication chain for a signed DNS response. By way of analogy, NTAs stop validation of the authentication chain. Instead, the validator treats any upstream responses as if the zone is unsigned and does not set the Authentic Data (AD) bit in responses it sends to clients. Note that this is a behavior and not a separate resource record. This NTA 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 an 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. Motivations for Negative Trust Anchors

1.2.1. Mitigating Domain Validation Failures

A domain name can fail validation for two general reasons: a legitimate security failure (e.g., due to an attack or compromise of some sort) or as a result of misconfiguration on the part of a zone 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 pay special attention to Section 4.2 of [RFC6781], which pertains to key rollovers, as these 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.

1.2.2. Improving End-User Experience

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 about 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 an NTA 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.

The following text from [RFC4033] is worth noting: "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.

1.2.3. Avoiding Switching to a Non-validating Resolver

As noted in Section 1.2.2, 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, and 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 DNSSEC validation 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 NTA.

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 Top-Level Domain (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 an NTA 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 an NTA 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.

2.1. Applicability of Negative Trust Anchors

An NTA MUST only be used for a limited duration. Implementors SHOULD allow the operator using the NTA to set an end time and date associated with any NTA. 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 an NTA MUST NOT be automatic.

Finally, an NTA 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, an NTA 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 NTA also SHOULD NOT affect names in another branch of the tree (such as example.net). In another example, an NTA 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 NTAs have proven useful during the early stages of DNSSEC adoption, domain owners are ultimately responsible for managing and ensuring that 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 resource record (RR) or a hash of a public key as in a DS RR.

Different DNS validators may have different configuration names for an NTA. 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 MAY issue a warning or informational message when this occurs, so that operators are not surprised when this happens.

3.1. Alerting Users to Negative Trust Anchor 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. Therefore, implementors should consider transparently disclosing NTAs that 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 an NTA is in place. Such disclosures should optimally include both the data and time that the NTA was put in place and when it was removed.

4. Removal of a Negative Trust Anchor

As explored in Section 8, 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 NTA is in place, until such validation is again successful. NTAs MUST expire automatically when their configured lifetime ends. The lifetime SHOULD NOT exceed a week. There is limited experience with

what this value should be, but at least one large vendor has documented customer feedback suggesting that a week is reasonable based on expectations of how long failures take to fix or to be forgotten. Operational experience may further refine these expectations.

Before removing the NTA, 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, an NTA 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 Start of Authority (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), 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 [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 NTAs should not add an NTA 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 a result of an operator error instead of an attack is not trivial, and the examples here should be vetted by an experienced professional before making the decision to implement an NTA.

One of the key things to look for when looking at a DNSSEC broken domain is consistency and history. Therefore, it 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 and associated answers (this is often referred to as a "passive DNS database"). Another invaluable tool is DNSViz (http://dnsviz.net), which also stores DNSSEC-related data historically. The drawback here is that you need for it to have tested 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. At the time of this writing, several tools that do this are available, including:

- DNSViz (http://dnsviz.net)
- Verisign DNSSEC debugger (http://dnssec-debugger.verisignlabs.com)
- Zonemaster (http://www.zonemaster.fr, https://github.com/dotse/zonemaster)

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. This is an incomplete list, and it is expected that additional tools will be developed over time to aid in troubleshooting DNSSEC issues.

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 caused by an operator rather than by 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 EDNS0 client subnet [CLIENT-SUBNET] is 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:

- 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.

- DNSKEYs in a 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 tool to show problems in the chain. If you debug it 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 the following. 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. Security Considerations

End-to-end DNSSEC validation will be disabled during the time that an NTA is used. In addition, the NTA may be in place after the 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 an NTA is removed. In addition, an NTA 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 applications that require DNSSEC for a domain will perform their own validation, so this should not be a major issue.

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>.

[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>.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <http://www.rfc-editor.org/info/rfc5914>.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, DOI 10.17487/RFC6781, December 2012, <http://www.rfc-editor.org/info/rfc6781>.

9.2. Informative References

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

[CLIENT-SUBNET] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", Work in Progress, draft-ietf-dnsop-edns-client-subnet-03, August 2015.

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

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

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

[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/>.

Appendix A. Configuration Examples

The section contains example configurations to achieve NTA functionality for the zone foo.example.com.

Note: These are simply examples -- name server 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 [Unbound-Config] 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 NTAs without restarting the name server.

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 [Unbound-Config].

A.2. Internet System Consortium (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.
```

Acknowledgements

Several people made contributions to this document and/or played an important role in the development and evolution of it. In some cases, this included performing a detailed review 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, Christer Holmberg, Shane Kerr, Murray Kucherawy, Rick Lamb, Marc Lampo, Ted Lemon, Scott Rose, A. Schulze, Wendy Seltzer, Antoin Verschuren, Paul Vixie, Patrik Wallstrom, Nick Weaver, W.C.A. Wijngaards, and Suzanne Woolf.

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

Authors' Addresses

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

Email: ebersman-ietf@dragon.net

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States

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

Chris Griffiths
Nominet
Minerva House
Edmund Halley Road
Oxford Science Park
Oxford OX4 4DQ
United Kingdom

Email: cgriffiths@gmail.com
URI: http://www.nominet.org.uk/

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

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

Ralf Weber
Nominum

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