Workgroup: dnsop
Internet-Draft:
draft-ietf-dnsop-dnssec-validator-
requirements-07
Published: 13 November 2023
Intended Status: Informational
Expires: 16 May 2024
Authors: D. Migault E. Lewis D. York
Ericsson ICANN Internet Society

Recommendations for DNSSEC Resolvers Operators

Abstract

The DNS Security Extensions (DNSSEC) defines a process for validating received data and assert them authentic and complete as opposed to forged.

While DNSSEC comes with some complexity, at least for non expert, that complexity is mostly abstracted by the resolver. As result, running a resolver with DNSSEC enabled only requires the operator to only follow a very limited set of recommendations. This document provides such recommendations.

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 https://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 16 May 2024.

Table of Contents

1. Introduction
2. Overall View of DNSSEC Validating Resolver
3. Time Recommendations
4. Trust Anchor Related Recommendations
5. Negative Trust Anchors Related Recommendations
6. Cached RRset Recommendations
7. Cryptography Deprecation Recommendations
8. Invalid Reporting Recommendations
9. Transport Recommendations
10. Secure Transport Recommendations
11. Clarification over the DNSSEC and the TLS Trust Models
12. IANA Considerations
13. Security Considerations
14. Acknowledgment
15. References
  15.1. Normative References
  15.2. Informative References
Authors' Addresses

1. Introduction

A DNS resolver is a service that locates and returns information pertaining to a query issued by some other service, such as an application. By its nature, a DNS resolver is inquisitive, susceptible to misleading information it may receive. To address this, DNS Security (DNSSEC) extensions [RFC9364] were defined to provide authenticity and integrity to responses, as well as to provide an authenticated notice for data that does not exist.

A DNS resolver operator is an organization or individual that runs a DNS resolver. The resolver may be for a small set of relying parties, for a large but bounded collection of customers, or it may be operated with no restriction on who or what may make use of it. To enhance the value of the service, a DNS resolver operator implements various security controls, including the use of DNSSEC validation.

Operating DNSSEC validation involves making use of digital signatures generated by a DNSSEC signer. Besides the simple cryptographic process of validating digital signatures there are a number of checks required due to the nature of the DNS protocol. A well-written DNSSEC

validating resolver will faithfully implement the DNSSEC processes needed, leaving an operator to manage a few items.

The items that an operator needs to attend to are:

*Absolute time

*Trust anchors (positive and negative)

*Monitoring of the service, including abuse

This document will list recommended actions for DNSSEC validating resolver operators that will help achieve the goals of DNSSEC validation. First, the goals ought to be stated.

The primary goal of any operations endeavor is to provide a service within service level agreements intended to make relying parties happy with the performance of the service. For DNSSEC, this breaks into two parts, one of accurately achieving the protections offered by DNSSEC, the other, to avoid DNSSEC from accidentally being an impediment.

The recommendations will focus on preparation of the elements of a DNSSEC validating resolver, as described earlier in the diagram. In particular, there are recommendations related to service monitoring, time source, Trust Anchor Manager/Store, and DNS Resolver. The recommendations are categorized as at initialization, during runtime, and upon demand.

2. Overall View of DNSSEC Validating Resolver

The purpose of a DNS resolver is to perform DNS resolutions on behalf of DNS client. The DNS resolution of a specific FQDN (e.g. www.example.com) requires the resolver to interact (via DNS resolutions) with a chain of authoritative servers. However, a resolver also caches the responses it receives which limits the resolver sending duplicated requests and triggers resolutions only when necessary.

With DNSSEC enabled, the resolver is able to verify data origin authentication and data integrity, and so ensures the response returned by the resolver is trustworthy.

Figure 1 illustrates the case with the resolution of the FQDN www.example.com. The DNS client sends a DNS request for the IP address of www.example.com to the resolver (1). If the response is not in the cache of the resolver, then it proceeds to the resolution. For simplicity, we suppose in our example that the resolver already knows the authoritative server of www.example.com. As a result, it only need to send a request to that authoritative server (2). We also

assume the DNS client has not explicitly requested the resolver to disable the DNSSEC validation. That DNS client may be DNSSEC aware or not, but the resolver performs the DNSSEC validation. When the resolver sends the DNS request to the authoritative server of example.com (2), it indicates the support of DNSSEC. In that case, if the zone example.com is signed, the authoritative server includes the necessary sets for the resolver to cryptographically validate the response (3). The resolver validates the response and returns the IP address to the client (4). Depending on the client's request, the server may indicate that the information is trustworthy or include the DNSSEC information so the client can re-validate the response. If the resolver is unable to validate the information an error is returned to the client (4).

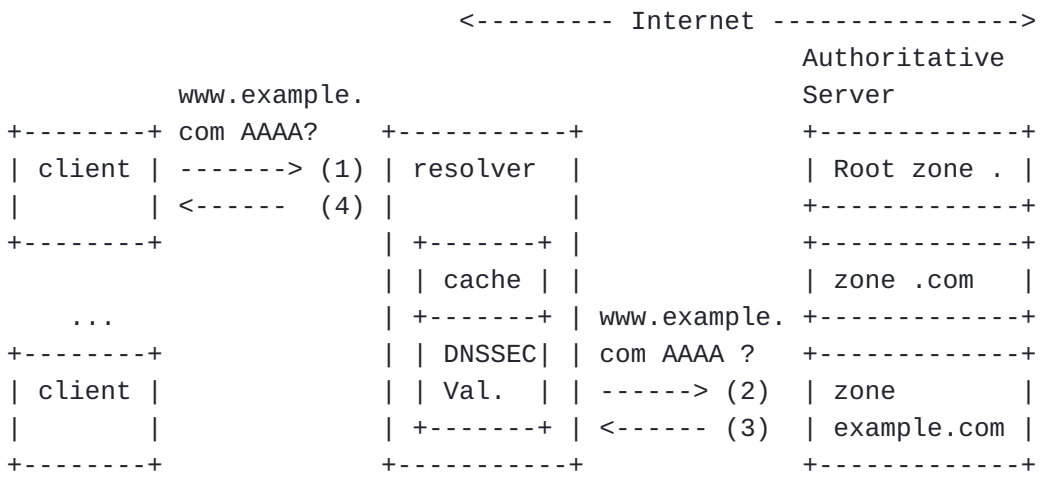

Figure 1: DNS resolution with DNSSEC validation

To help orient this document, the following schematic is offered to show some of the interrelationships among the elements of a DNSSEC validating resolver, that is to say the entity that performs DNS resolution and performs signature validation. This drawing is merely a cartoon summary, not an implementation guide.

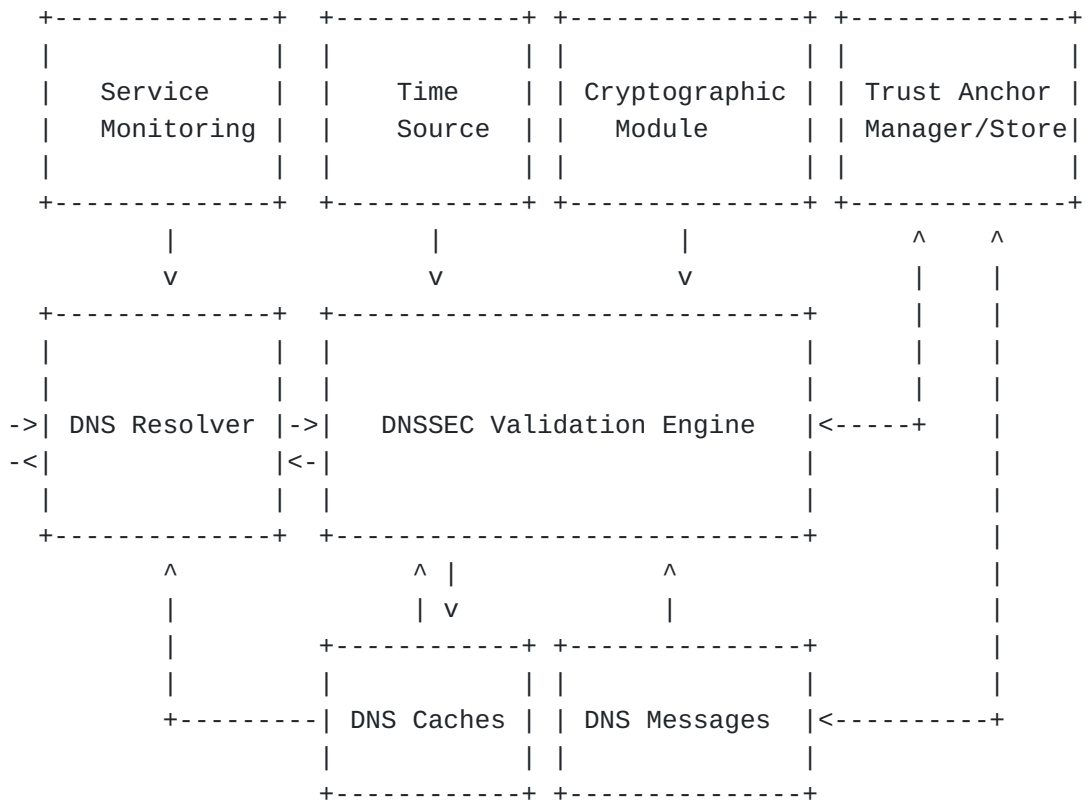

Figure 2: DNSSEC Validating Resolver Description

Across the top row are elements that an operator needs to address to properly run a DNSSEC Validating Resolver.

Service Monitoring: Enforces acceptable use policy and enables management of the service. This is a generic module for all services, here featuring some DNS and DNSSEC specific concerns.

Time Source: Provides the time. DNS has always used relative time to manage the TTL of resource record sets in caches, DNSSEC introduced the need for absolute time to thwart replay attacks and to manage the lifetime of signatures.

Cryptographic Module: implements the cryptography operations. Due to the nature of cryptography, the implementation of this module may evolve over time or at least be subject to technical refresh.

Trust Anchor Manager/Store: a module (of code) implementing functions related to the trust anchors used by the validator. This is essentially a database allowing access, monitoring of, and changes to trust anchors. The database of trust anchors used as the basis from which DNSSEC operates. This module contains the foundation upon which DNSSEC evaluates received data sets. The contents of this module are essential for proper operation. For a general-purpose validator running on a public Internet, it may

have a single entry, for the very top of the DNS name space (the root). Management of this may be left to automated processes that are carefully designed to address vulnerabilities related to automated trust management. For specific situations, the Trust Anchor Manager/Store will also manage local-policy supporting trust anchors as well. Careful operation of this module is crucial to the value of the DNSSEC validating resolver. Note that the Trust Anchor Store MAY (but not necessarily) be updated via in-band signalling [RFC5011] in which case validated message result in updating the TA Store.

DNS Resolver: The service interface offered to other services/ applications/users. This is the generic DNS resolution service, consulting the cache for hits, and seeking new answers for misses.

DNSSEC Validation Engine: This implements the DNSSEC validation process. The validation engine is assumed to faithfully implement the DNSSEC validation algorithm, relying on the information from the Time Source, Cryptographic Libraries, Trust Anchor Management/ Store as well as what is held in the local cache or gained through messages. A successful validation ensures the information is trustworthy.

DNS Caches: Include positive and negative caches. These are the ordinary caches used in DNS operations, including DNSSEC extended record types and management.

DNS Messages: Existing DNS message passing. This covers the proper implementation and operation of DNS and DNSSEC message exchanges.

3. Time Recommendations

As DNSSEC uses absolute time to temporally limit the validity of RRSIG resource records, a DNSSEC validator needs a reliable time source. If a validator can be fooled into believing that the time is a point well into the past, an incorrect RRSIG resource record may be replayed and used, or an old key whose private component has since been exposed may be able to forge a falsified answer.

The range of these recommendations include devices that do not have an embedded Real Time Clock. Such devices need to have their system clocks updated upon power up before starting the DNSSEC validator.

At initialization: a DNSSEC validator needs to be able to establish reliable time without relying on DNSSEC validation. The latter clause is needed as the initialization step is being carried out to start DNSSEC validation, it is not assumed to be up and running at this point. One way to interpret this is that a time source (Network Time Protocol) ought to be identified by a numerical IP address and not a fully qualified domain name (which would require a DNS lookup). An

operator may set its own time as for example described in [TNL] or rely on an external Time provider. In any case the system time should be retrieved from an authenticated source using Network Security Time (NST) [RFC8915]. However NST is based on TLS and as such the verification of a signature which requires an appropriated time to be be set to ensure the certificate is valid. To get a system time that enables the establishment of a TLS session, the operator may rely primarily on manual configuration or an unauthenticated NTP service such as pool.ntp.org [NTPPool] or [TimeNL]. Once that system time is set, the system time should confirm the time over a trusted source, i.e. using NST.

During runtime: a DNSSEC validator operator ought to have controls in place to monitor the current time of a validator as well as monitor the number of validation failures that can be attributed to temporal violations. Updates to the current time ought to make use of secure environments, whether secure channels for NTP or as appropriate for the installation. Updates ought to be part of an automated process, running at appropriately frequent intervals -- see [TimeNL].

Upon demand: a DNSSEC validator operator ought to be able to perform any of the runtime actions upon demand, for instance, to help diagnose a service failure.

4. Trust Anchor Related Recommendations

The TA store implements a trust model. The default trust model consists in trusting a single TA which is the KSK of the root zone. It is the recommended trust model and the default trust model of most implementations.

While not generally recommended, alternative TAs MAY be considered, for example, when the secure delegation to these RRsets may not be validated for any reasons. The trust model should at least ensure that any domain name in the DNS be covered by at least one TA. As the number of top level domains is evolving overtime, it remains safe to keep the root zone as a security entry point in order to cover the full domain name space. Upon considering TA, the resolver operator should carefully ensure that the TA meets all necessary operational criterias. This includes for example, having a bootstrapping mechanism, or having their signers committed to respect the [RFC5011] timing - at least when the operator relies on automatic updates (see below).

TAs are usually represented by a DNSKEY or DS RRset and are involved in the signature validation process to determine whether the validation is successful or not.

At initialization: The resolver operator needs to ensure the resolver can only be started with a TA store that matches the trust model and that is up-to-date. The TA needs to be retrieved securely and checked upon starting the resolver. For the default trust model, for example, [UNBOUND-ANCHOR] or [ta-fetcher] implements [RFC7958] and ensures the resolver is configured with the up-to-date TA of the root zone. Similarly, the up-to-date default trust model is very commonly implemented by the software release in which case the resolver operator may simply rely on software update.

During runtime: The resolver operator needs to ensure the TA is up-to-date. This is achieved by enabling TA to be updated automatically (via in-band or out-of-band mechanisms) as well as being able to check the status of the TA. Checking the status of the TA is expected to be performed especially when a key roll over happens or any other event associated to the TAs. TA updates are not expected to be handled manually as this introduces a potentially huge vector for configuration errors as well as potential misunderstanding of ongoing operations. Instead, the resolver operator should rely on automated procedures such as, for example, Automated Updates to DNSSEC Trust Anchors" [RFC5011] [I-D.ietf-dnsop-rfc5011-security-considerations] or software updates. As [RFC5011] is an in-band mechanism, the resolver operator is expected to understand these risks [RFC5011], Section 8. Software update or other mechanisms may also be used.

Upon demand: The resolver operator should be able to check the status of the TA within its resolvers whenever a health check is performed or when it is aware a TA roll over or any other operation affecting the TA is ongoing. This includes the TA stored in the running resolver as well as potential configuration files. The TA used by the resolver is expected to be retrieved using "Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)" [RFC8145], and may re-use similar software as those used at the initialisation to retrieve and check the expected value of the TA. The TA health check should associate a status to the TA - as defined in Section 3 of [RFC7583] - to ease the TA monitoring and potential analysis. When an unexpected (old) TA is found, the health check should evaluate if the mismatch resulted from an ongoing normal roll over, a potential emergency key roll over, failed roll over or any other envisioned cases. In any case restarting the resolver is expected to address any situation that cannot be addressed otherwise, which reinforces the recommendation to rely on TA bootstrapping mechanisms.

Note also that [RFC8145] also enables to check how the TA roll over is performed. Such cooperation is expected to be useful and benefit the overall operation of the DNS system. The owner of the TA is able to check whether its key roll over is being performed appropriately, and if not, it may cancel it for further analysis.

5. Negative Trust Anchors Related Recommendations

When the DNSSEC resolver is not able to validate signatures because a key or DS has been published with an error, the resolver operator may temporarily disable the signature check for that key until the time the error is addressed. Negative Trust Anchor (NTA) represents the only permitted intervention in the resolving process for a resolver operator.

A signature validation failure is either an attack or a failure in the signing operation on the authoritative servers. The resolver operator is expected to confirm this offline before introducing the NTA. This is likely to happen via a human confirmation which is based on information collected during running time. The qualification of the issue as well as getting a confirmation may require some minimal expertise.

The designation of NTA might be misleading, but NTA is not expected to be part of the trust model even though the NTA belongs to the TA store.

At initialization: Similarly to TA, the resolver operator is expected to automatically configure the resolver with the NTA.

Upon demand: the resolver operator is expected to automatically determine the used NTA and handle NTA as described in [RFC7646].

At running time: The resolver operator should monitor the number of signature failures associated with each DNSKEY. These numbers are only hints and must not trigger automated insertion of NTA.

6. Cached RRset Recommendations

During runtime: A resolver operator is not expected to perform any operations over the cached RRset. A common concern resolver operator has is the consistency between the cached RRset with those published by the DNS system. Resolver operator should not implement or deploy any non standard mechanism of there own and limit themselves to enable standard options / feature provided by their implementation. [I-D.ietf-dnsop-ns-revalidation] is one of these standard mechanisms, for example that improves the synchronization between the cache of the resolver and those published on the Internet. Section 8.1 of [RFC4033] also mentions the ability by the resolver to set the upper bound of the TTL to the remaining signature validity period. This value has usually a default value set by the resolver and the resolver operator may change that default value.

Upon demand: a resolver operator may have been informed that a rogue or unwilling DNSKEY has been published. In such a situation, the

resolver operator should be able to remove the RRsets validated by the rogue DNSKEY - which may be done by flushing the full cache.

7. Cryptography Deprecation Recommendations

As mentioned in [RFC8247] and [RFC8221] cryptography used one day is expected over time to be replaced by new and more robust cryptographic mechanisms. In the case of DNSSEC signature protocols are likely to be updated over time. In order to anticipate the sunset of one of the signature schemes, a DNSSEC validator may be willing to estimate the impact of deprecating one signature scheme.

Currently, interoperability and security are enforced via cryptographic recommendations [RFC8624] that are followed by both resolvers and authoritative servers. The implementation of such guidance is ensured by the software vendors and the compliance of their releases.

At initialization: The resolver operator is expected to ensure recent software releases are used and that this release complies with the most recent cryptographic guidelines.

During runtime: a resolver operator may regularly request and monitor the signature scheme supported by an authoritative server. In addition, when a validation fails because a deprecated algorithm is used, the resolver operator should return an "Unsupported DNSKEY Algorithm" as defined in [RFC8914] to the DNS client.

Note that one inconvenience to such a strategy is that it does not let one resolver operator take advantage of more recent cryptographic algorithms. While currently not being widely used, a resolver operator may announce an authoritative server the supported signature schemes to the authoritative server [RFC6975] in the hope that future deployment of authoritative servers will be able to leverage it.

8. Invalid Reporting Recommendations

A DNSSEC validator receiving a DNS response cannot make the difference between receiving an non-secure response versus an attack. Dropping DNSSEC fields by misconfigured middle boxes, such as DS, RRRSIG is considered as an attack. A DNSSEC validator is expected to perform secure DNS resolution and as such protects its stub client. An invalid response may be the result of an attack or a misconfiguration, and the resolver operator may play an important role in sharing this information with the authoritative server or domain name owner.

At runtime: a resolver operator should monitor and report DNSSEC validation errors and inform the DNS client with "Extended DNS Errors" [RFC8914].

9. Transport Recommendations

DNSSEC validation requires that the validator is able to reliably obtain necessary records, especially DNSKEY records. This should be done at initial configuration, and tested periodically.

This means the validator must ensure it is configured so that the UDP and TCP transports, and DNS resolver components, are compatible with the network paths that the majority of DNS queries traverse - which includes compatibility between DNS and transport parameters with the Maximum Transmission Unit (MTU).

In other words, make sure that:

1. DNS UDP bufsize (EDNS parameter) is set to a value compatible with network MTUs the queries and responses will encounter. If the validator advertises a bufsize >> MTU, responses with the IPv4 Don't Fragment (DF) bit set whose size R where MTU < R <= bufsize exceeds the MTU will be dropped by the router with MTU < R.

2. The validator's OS TCP configuration has its advertised Maximum Segment Size (MSS) set to a value compatible with network MTUs the queries and responses will encounter.

*Having an advertised MSS set to a value < MTU ensures that Path MTU Discovery is not required

*If PMTUD fails for any reason, or if the server responding does not maintain or use PMTUD, and advertised MSS > MTU at any point in the path, TCP may encounter problems caused by IP fragmentation and reassembly.

*This is particularly relevant if there are any NAT type devices in the path, as those may not properly handle fragmentation and reassembly

*If all TCP segments are smaller than the path MTU, TCP will work reliably.

The avoidance of fragmentation in order to address known fragmentation-related security issues with DNS (leading to cache poisoning, for example) has resulted in the need to set the DF bit on UDP. Validators will need to ensure their local environment can reliably get any critical DNSSEC records (notably DNSKEY) over UDP, or reliably get responses with TC=1 if overly large responses cannot be sent over UDP due to answers not fitting within the advertised bufsize payload. Validators also need to ensure TCP works if it is needed, for the same situations.

10. Secure Transport Recommendations

An operator should consider secure transport as a complementary element of its trust model. Such resolvers are usually designated as encrypted resolvers and their presence is usually discovered by the DNS client via a discovery mechanisms. That discovery mechanism may require the resolver to respond to specific DNS requests.

In many cases, anoperator enables DNSSEC to ensure the cached RRset have not been modified on-path and corresponds to those published by the owner of the zone. To ensure RRsets are protected against on-path modification from the resolver to the DNS client, the DRO may enable a secure transport such as DNS over TLS (DoT) [RFC7858], DNS over HTTPS [RFC8484] (DoH) or DNS over QUIC (DoQ) [RFC9250]. The TLS termination may be supported by the running resolver software or via a TLS front end and TLS should follow its own TLS recommendations [RFC9325].

When an operator sets a resolver over a secure transport, this resolver does not operates anymore on port 53 and an operator may wonder how DNS client will be able to discover that new service and more importantly if there are implications on its operations. A DNS operator should consider how the DNS client will discover the encrypted resolver. In many cases, the DNS client are able to discover the encrypted resolver using [I-D.ietf-add-ddr] in which case, the resolver needs to be configured to respond to special requests. It is also possible to provision directly the DNS client with the encrypted resolver via specific mechanisms such as DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers [I-D.ietf-add-dnr] or 3GPP TS 24.008 mechanisms.

11. Clarification over the DNSSEC and the TLS Trust Models

While the document is essentially focused on the security implemented by DNSSEC, it also mentions the combination of TLS and DNSSEC. DNSSEC is essentially a protocol focused on authenticating the DNS data, but is not addressing confidentiality of the data nor the privacy of the user requesting that data. TLS provides authentication and confidentiality of a communication.

As a result, TLS is necessary wherever privacy is needed. In the current model a DNS client is using TLS to protect the communication with the resolver. If that resolver is trusted and believed to host trustworthy RRsets - that is unmodified RRsets validated by the DNSSEC - the TLS communication enables the DNS client to trust the RRSets without performing a DNSSEC validation. If that resolver is expected to perform some operations agreed by the DNS client that may change the RRsets, DNSSEC cannot be performed by the DNS client and TLS is used to carried these trusted RRsets to the DNS client. On the

other hand, if the DNS client does not fully trust the resolver, than the DNS client must authenticate the received RRsets with DNSSEC. In that case, TLS is only providing privacy protection.

The trust in a DNS resolver depends on multiple factors, but one significant concern is the ability of the DRO to perform a man in the middle attack and change on the fly the RRsets without the stub client being aware of it. Confidential computing [I-D.arkko-dns-confidential] may be one way to address such attacks. Another concern, related to privacy, is the ability of a resolver to track a certain user and log every sites requested by the user. Confidential Computing [I-D.arkko-dns-confidential] or oblivious DNS [RFC9230] are means to address such issues.

A model where TLS would be used to protect the transactions between the DNS client and the authoritative server is unlikely in a near future for scalability reasons. A compromise to this model may consists in having the TLS communications between the resolvers and the authoritative servers. In such scenario, the privacy requirement might be questionable as the resolver aggregates the traffic of multiple DNS clients which may be considered to provide sufficient privacy. However, a model involving a communication composed of multiple TLS segments is only trusted if all involved nodes are trusted (DNS client, resolver, authoritative server). In practice, it is unlikely to have all these intermediary nodes trusted, in which case trust will rely on DNSSEC.

12. IANA Considerations

There is no IANA consideration for this document.

13. Security Considerations

Security consideration of DNSSEC operations are described in [RFC6781]. However, most DNSSEC operations are performed by the owner of the zone as opposed to the operator of the resolver. Operators largely relies on the software vendor as well as automated procedure implemented by the software vendor as to limit potential manual intervention from the operator. This section emphases on operator related security considerations.

Regarding time inaccuracy, a RRSIG is only valid between the inception and expiration time. As a result, when the time is outside this period validation is disabled, and this could be used by an attacker to disable validation for example to poison the cache.

Trust anchors are the root of the trust in DNSSEC and potentially, an attacker being able to provide a rogue trust anchor is potentially able to hijack any RRset below that Trust Anchor. On the other hand, mishandling Trust Anchor is likely resulting in a validator unable to

validate most of the traffic under the TA. The use of a common trust model shared by many operators and implemented by multiple vendors reduces these risks.

Negative trust anchors by definition disable validation, and as such must be handled very cautiously by the operator. This could be used by an attacker, for example, to disable validation and poison the resolver.

Using weak cryptography reduces the strength in the trust implemented by DNSSEC as it relied on cryptographic signatures. A weak cryptographic algorithm may be used by an attacker to forge a signature. However, there is little action the resolver operator can actually do, as the cryptographic algorithms to be used are defined by the owner of the zone or the RRSet.

The resolver operator is operating one part of the DNS system. While an operator operates independently, it is believed that communication between the different actors involved in the DNS system will enhance the global resiliency of the system. As a result, this document encourages the operators to provides some information to the stub client when a signature validation fails. It also encourages the operators to authorize third parties to request what trust anchor or more generally DNSKEY are being used, so concerned party may be able to contact the her if needed. This is expected, for example when a authoritative server is performing a key roll over to check the update has been performed properly before removing the old key. The same considerations for communications also holds between resolver operators as well as with software vendors.

As the software used for the DNSSEC validator is not immune to bugs [ENT] and may become vulnerable independently of how it is operated, it is recommended an operator relies on different vendors.

14. Acknowledgment

The need to address DNSSEC issues on the resolver occurred during multiple discussions including among others Ted Lemon, Ralph Weber, Normen Kowalewski, Mikael Abrahamsson, Jim Gettys, Paul Wouters, Joe Abley, Michael Richardson, Vladimír Čunát, James Gannon, Andrew McConachie, Peter Thomassen, Florian Obser, Brian Dickson and Christian Huitema.

We also appreciated the support of the DNSOP chairs Tim Wicinski, Suzanne Woolf and Benno Overeinder.

15. References

15.1. Normative References

[I-D.ietf-add-ddr] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", Work in Progress, Internet-Draft, draft-ietf-add-ddr-10, 5 August 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-add-ddr-10>.

[I-D.ietf-add-dnr] Boucadair, M., Reddy.K, T., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", Work in Progress, Internet-Draft, draft-ietf-add-dnr-16, 27 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-add-dnr-16>.

[I-D.ietf-dnsop-ns-revalidation] Huque, S., Vixie, P. A., and R. Dolmans, "Delegation Revalidation by DNS Resolvers", Work in Progress, Internet-Draft, draft-ietf-dnsop-ns-revalidation-04, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-ns-revalidation-04>.

[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>.

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

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

[RFC7583] Morris, S., Ihren, J., Dickinson, J., and W. Mekking, "DNSSEC Key Rollover Timing Considerations", RFC 7583, DOI 10.17487/RFC7583, October 2015, <https://www.rfc-editor.org/info/rfc7583>.

[RFC7646] Ebersman, P., Kumari, W., Griffiths, C., Livingood, J., and R. Weber, "Definition and Use of DNSSEC Negative Trust Anchors", RFC 7646, DOI 10.17487/RFC7646, September 2015, <https://www.rfc-editor.org/info/rfc7646>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[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>.

[RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <https://www.rfc-editor.org/info/rfc8221>.

[RFC8247] Nir, Y., Kivinen, T., Wouters, P., and D. Migault, "Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8247, DOI 10.17487/RFC8247, September 2017, <https://www.rfc-editor.org/info/rfc8247>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[RFC8624] Wouters, P. and O. Sury, "Algorithm Implementation Requirements and Usage Guidance for DNSSEC", RFC 8624, DOI 10.17487/RFC8624, June 2019, <https://www.rfc-editor.org/info/rfc8624>.

[RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <https://www.rfc-editor.org/info/rfc8914>.

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <https://www.rfc-editor.org/info/rfc8915>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/info/rfc9250>.

[RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security

(DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

15.2. Informative References

[ENT] Levigneron, V., "ENT was here !!!", n.d., <https://indico.dns-oarc.net/event/25/contributions/403/attachments/378/647/AFNIC_OARC_Dallas.pdf>.

[I-D.arkko-dns-confidential] Arkko, J. and J. Novotny, "Privacy Improvements for DNS Resolution with Confidential Computing", Work in Progress, Internet-Draft, draft-arkko-dns-confidential-02, 2 July 2021, <https://datatracker.ietf.org/doc/html/draft-arkko-dns-confidential-02>.

[I-D.ietf-dnsop-rfc5011-security-considerations] Hardaker, W. and W. A. Kumari, "Security Considerations for RFC5011 Publishers", Work in Progress, Internet-Draft, draft-ietf-dnsop-rfc5011-security-considerations-13, 16 July 2018, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-rfc5011-security-considerations-13>.

[NTPPool] "The NTP Pool Project", n.d., <https://www.ntppool.org/en/use.html>.

[RFC6975] Crocker, S. and S. Rose, "Signaling Cryptographic Algorithm Understanding in DNS Security Extensions (DNSSEC)", RFC 6975, DOI 10.17487/RFC6975, July 2013, <https://www.rfc-editor.org/info/rfc6975>.

[RFC7958] Abley, J., Schlyter, J., Bailey, G., and P. Hoffman, "DNSSEC Trust Anchor Publication for the Root Zone", RFC 7958, DOI 10.17487/RFC7958, August 2016, <https://www.rfc-editor.org/info/rfc7958>.

[RFC9230] Kinnear, E., McManus, P., Pauly, T., Verma, T., and C.A. Wood, "Oblivious DNS over HTTPS", RFC 9230, DOI 10.17487/RFC9230, June 2022, <https://www.rfc-editor.org/info/rfc9230>.

[ta-fetcher] Davies, J. S. K., "DNSSEC Trust Anchor Fetcher", n.d., <https://github.com/iana-org/get-trust-anchor>.

[TimeNL] "TimeNL Public NTP service", n.d., <https://time.nl/index_en.html>.

[TNL] Hesselman, M. D. C., "TimeNl comes of age", n.d., <https://www.sidnlabs.nl/en/news-and-blogs/timenl-comes-of-age>.

[UNBOUND-ANCHOR] "unbound-anchor - Unbound anchor utility", n.d., <https://nlnetlabs.nl/documentation/unbound/unbound-anchor/>.

Authors' Addresses

Daniel Migault
Ericsson
8275 Trans Canada Route
Saint Laurent, QC 4S 0B6
Canada

Email: daniel.migault@ericsson.com

Edward Lewis
ICANN
United States of America

Email: edward.lewis@icann.org

Dan York
Internet Society
United States of America

Email: york@isoc.org