The reduced risk of redirected query traffic with signed root name server data

In response to RZERC002 Recommendation 2

Consortium:

NLnet Labs

Stichting Internet Domeinregistratie Nederland (SIDN)

Authors: Willem Toorop (NLnet Labs), Yorgos Thessalonikefs (NLnet Labs), Benno Overeinder (NLnet Labs), Moritz Müller (SIDN), Marco Davids (SIDN).

Last modified: May 22, 2024

Summary

In this report we evaluate the added security benefits that a DNSSEC signed `root-servers.net` zone would have for resolvers on the internet.

At the time this document is published, the authoritative root zone name server data — including the addresses for the individual root server identifiers[1] — is contained in the `root-servers.net` zone. This zone is served by the root servers and is not signed with the Domain Name System (DNS) Security Extensions (DNSSEC). An attacker that is able to provide alternative IP addresses for the root server identifiers, by spoofed (priming) responses, can fool a resolver into taking addresses under the control of the attacker to be authoritative for the root zone. As a result, the attacker can *redirect all queries* from that resolver for the root to a rogue root server under the control of the attacker. In section The impact of redirected query traffic, we show that an attacker that controls a rogue root server can potentially take over the entire domain name space and can view all queries and alter all unsigned data undetected. DNSSEC signed root zone name server data would provide a verifiable statement of the addresses belonging to the root server identifiers and could prevent this attack scenario.

In this report we evaluate how much the risks of redirected query traffic would be reduced if the root zone name server data would be signed with DNSSEC. In section DNSSEC signed root server addresses in the priming response we show that the addresses conveyed in priming responses cannot be protected and that they need to be queried for directly. In section Reduced risk based on the 2023 DITL data we express reduced risk as the number of authoritatively acquired root server identifier addresses (that can be validated when signed) in proportion to the non-authoritatively acquired in use by resolvers as seen in the DNS Operations Analysis and Research Center's (DNS-OARC) 2023 Day In The Life of the Internet (DITL) data set. Based on that data we estimate a 1.2% overall reduced risk of redirected query traffic by signed root name server data.

Based on our measurements and analysis we recommend resolver software to acquire addresses for the root server identifiers authoritatively. Resolvers that acquire the addresses authoritatively will already have *reduced risks* of redirected query traffic even *without DNSSEC* signed root zone name server data, and will have completely *eliminated the risk with DNSSEC* signed root zone name server data (provided the resolver is DNSSEC validating). Finally, as opposed to other mitigations to reduce the risk by improving the security of the transactions to the root service, DNSSEC validating the (signed) root server IP addresses from direct queries, is the only mitigation that also protects against on-path attacks.

[1] This report follows the lexicon and terminology as given in RSSAC026v2: RSSAC Lexicon and RFC9499: DNS Terminology

Table of Contents

Introduction

The root server system is a fundamental part of the DNS and a critical part in the functioning of the Internet. The security of the root service impacts the entire domain name space.

The root service is provided by the root servers. Resolvers learn the names of the root server identifiers and their associated IP addresses (at which the root servers provide the root service) at startup through priming queries and regularly after startup. The addresses of the root server identifiers are currently unsigned, and are returned unsigned in priming queries. An adversary that manages to alter the addresses in priming responses, can view all queries and alter all unsigned parts of responses from potentially the entire domain name space, without detection.

Taking into consideration that DNSSEC signed addresses for the root server identifiers may help remedy this issue, the Root Server System Advisory Committee (RSSAC) has proposed alternative naming schemes for the root server system, all in which the root zone name server data is signed with DNSSEC (see RSSAC028: Technical Analysis of the Naming Scheme Used For Individual Root Servers[2]). The key recommendation from the report was that: “no immediate changes should be made to the current naming scheme until further studies have been conducted”. One of the further studies called for was to “understand the current behavior of DNS resolvers and how each naming scheme discussed in this document would affect these behaviours.” The same consortium of (NLnet Labs and SIDN) that is responsible for this report, took on this study which resulted in the RSSAC028 Implementation study report[3].

The Root Zone Evaluation Review Committee (RZERC) discussed DNSSEC signing of the authoritative root zone name server data, as contained in the `root-servers.net` zone, in response to a proposal from the Root Zone Maintainer (RZM) representative. The RZERC002: Recommendations Regarding Signing Root Zone Name Server Data[4] report states RZERC’s position and contains two recommendations: The first recommends ICANN to “conduct the further studies called for in Recommendation 2 of RSSAC028,” which we, the consortium, conducted and published in the RSSAC028 Implementation study report[3]. The second recommendation from RZERC002 is quoted here verbatim:

> “ **Recommendation 2: The RZERC recommends that ICANN org further explore the cost / benefit tradeoffs and risks of signed root zone name server data. Do the risks of redirected query traffic outweigh the risks of increased operational complexity?**

This document reports on the explorational study into the benefit of signed root zone name server data, with the aim of assisting in answering the question whether “the risks of redirected query traffic outweigh the risks of increased operational complexity.”

---

[2] https://www.icann.org/en/system/files/files/rssac-028-03aug17-en.pdf
[3] https://www.icann.org/en/system/files/files/rssac028-implementation-study-report-27sep23-en.pdf
[4] https://icann.org/iana_rzerc_docs/447-rzerc002-recommendations-regarding-signing-root-zone-name-server-data-v-final

Research question and Scope

The authors do not feel they are in a position to be able to estimate the risks of increased operational complexity of signing the root zone name server data, but we do feel we can assist in answering the question by evaluating the reduced risk of redirected query traffic from signed root zone name server data. To this end we formulate the following research questions:

1. Does signed root zone name server data reduce the risks of redirected query traffic?
2. To what extent?

With some boundary conditions:

- Only considering recursive resolvers to root name servers and not considering other relationships, such as stub to resolver, forwarding resolvers, etc.
- Only with currently deployed resolver software

In this report we name a few open source resolver software implementations with respect to certain features and behaviors. The implementations we name are: BIND from Internet Systems Consortium, Knot Resolver from CZ.NIC labs, PowerDNS Recursor from PowerDNS and Unbound from NLnet Labs. Other resolver software is not considered in this report.

Redirected query traffic

What is meant by “redirected query traffic” in the second recommendation of RZERC002, is clarified in the fourth paragraph of section 2 of RZERC002:

> “ If an attacker fools a resolver into thinking that a server that the attacker controls is authoritative for the root zone, that attacker could view all queries from the resolver to the root system, and could alter all unsigned parts of responses without detection.

This attack is feasible because (quoting section 2 of RZERC002) “In the root zone, the address records (A and AAAA) conveying the root server IP addresses are non-authoritative and thus are not signed.” Consequently “DNSSEC validates the correctness of the data, but not whether or not it came from the correct server.” Furthermore, the authoritative zone for the address records for the root server IP address, `root-servers.net`, is also not DNSSEC signed, “as its signatures are not needed to validate the signed root[5].”

The impact of redirected query traffic

An attacker who manages to redirect query traffic to the root can, for most resolvers, extend the attack and also redirect query traffic for zones that are delegated from the root, regardless of whether those zones are DNSSEC signed. The attack can then be further extended to also include delegations from those zones and so on. The attacker can ultimately potentially view all queries *for the entire domain name space* and alter all unsigned parts of responses without detection.

[5] See https://www.iana.org/dnssec/archive/launch-faq

The root cause for this is that the name server (NS) resource record (RR) sets (RRsets) and glue returned in referral responses, that tell the resolver where the authoritative name server for the zone can be reached, are not DNSSEC signed[6] and can be altered without detection. This is further explained and elaborated upon in the coming subsubsections.

Infrastructure data

Recursive resolvers use DNS data to establish which authoritative name server to query for a zone. This data consists of the names of the name servers for that zone and the IP addresses associated with those names. The names of the name servers are given in the resource data (RDATA) of an NS RRset with the name of the zone as owner name. The IP addresses are provided through the A and AAAA RRsets with the IP addresses for those names. In this report we refer to DNS data that is used by recursive resolvers to do their job as *Infrastructure data*.

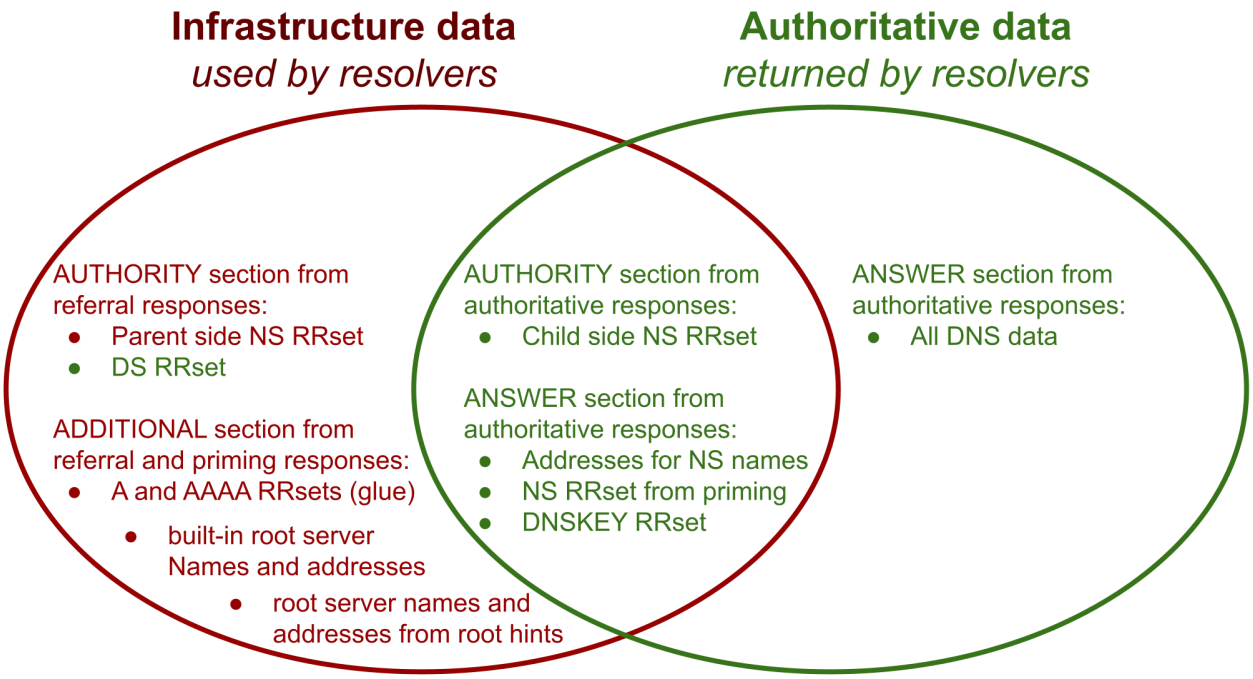

*Figure 1: Authoritative DNS data as **returned by** recursive resolvers (in the right ellipse) in relation to the DNS data that is **used by** recursive resolvers (in the left ellipse) which is not always authoritative. Note that the DS RRset is authoritative (and thus also signed) in the parent. It is returned authoritatively in referral responses.*

The NS RRset with the names of the name servers for a zone is authoritatively present at the apex of that zone. However, recursive resolvers cannot query for that authoritative NS RRset until they know how to contact the authoritative name servers for that zone. Therefore an NS RRset for the same name is also present non-authoritatively in the parent zone. This parent-side non-authoritative version of the NS RRset is returned in the Authority section of

[6] See the second to last paragraph of RFC 4033: DNS Security Introduction and Requirements, Section 12 Security Considerations, https://www.rfc-editor.org/rfc/rfc4033#page-16

referral responses (see Figure 1). Referral responses can furthermore contain A and AAAA RRsets with the IP addresses for the names in the RDATA of the NS RRset in the additional section, which is known as *glue* (see Figure 1). These glue addresses are non authoritative and can be non-authoritatively present in the parent zone. This non-authoritative data returned in referral responses is not DNSSEC signed[6] and can be altered by a machine in the middle for the parent zone without detection.

How DNSSEC delegations impact query redirection

Secure delegations are realized by a signed Delegation Signer (DS) RRsets in the parent, whose RDATA hash field matches a Secure Entry Point (SEP) DNS public key (DNSKEY) RR in the child. A referral response will include the signed parent's DS RRset and this part of the referral response cannot be altered undetected. However, the unsigned NS RRset can still be altered and used to redirect the queries to a machine under the attacker's control. The attacker machine can pass through all responses from the genuine name servers leaving all DNSSEC signed data unaltered. The attacker can still view all queries to the DNSSEC signed zone and alter all unsigned parts such as further referrals.

Insecure delegations give the redirecting attacker the ability to freely alter responses. The role of DNSSEC in insecure delegation is as follows. As with secure delegations, the NSEC or NSEC3 RRset can be passed through in referral responses unchanged. In the case of insecure delegation, these authenticated denial of existence RRsets show that there is no DS RRset on the name. The NS RRset and glue can once more be altered without detection. Redirected queries for the delegated insecure zone can then be altered freely. And as a result of the delegation being insecure, the attacker no longer needs to pass through authenticated parts in the responses for such a zone.

A consequence of the above is that, if an attacker can hijack the root service (by a spoofed priming response), it can also spoof answers for the insecure `root-servers.net zone`.

Ranked DNS data

Figure 1 above shows that infrastructure data can also be obtained authoritatively by recursive resolvers. RFC 2181: Clarifications to the DNS Specification, Section 5.4.1. Ranking data[7], specifies that recursive resolvers should replace cached RRsets with RRsets from a response if it has a higher trustworthiness. Specifically, authoritative RRsets should replace non-authoritative RRsets in the cache. The RFC provides a list with relative trustworthiness based on the source of the data. An adapted version of the ranking list is provided in Table 1 below. Each item on the original list is supplemented with a value for the trustworthiness rank, referenced in the remainder of the report. The highest ranking data has value **AAA** and the lowest ranking has value **CC**.

Values **AAA**, **AA**, **A** and **A-** are all for authoritative data. This data will have DNSSEC signatures included, when the zone for that data is DNSSEC signed. Infrastructure data coming from an authoritative source (i.e. with trustworthiness value **AAA**, **AA**, **A** or **A-**) cannot be altered without detection.

[7] https://datatracker.ietf.org/doc/html/rfc2181#section-5.4.1

AAA	Data from a primary zone file, other than glue
AA	Data from a zone transfer, other than glue
A	Authoritative data in the answer section
A-	The authority section of an authoritative answer
BBB	Glue from a zone file or zone transfer
BB	The answer section of a non-authoritative answer, and Non-authoritative data from the answer section
B	Additional information from an authoritative answer, The authority section of a non-authoritative answer, Additional information from non-authoritative answers.
CCC	Root server addresses from a root hints file, or
CC	In resolver software built-in root server addresses

*Table 1: Ranking data table from RFC 2181 Section 5.4.1.[7] The list items text is slightly adapted to prevent line breaks. The trustworthiness values (**AAA**, **AA**, **BBB**, etc.) are a new addition to the list, and the bottom two list items for built-in root server names and addresses and those values from root hints file were also not in the original table.*

As shown in section Infrastructure data, there are two versions of the NS RRset for a zone:

1. One version is present non-authoritative in the parent zone. It is included in a referral response without signatures and has trustworthiness value **B**. This is the version a recursive resolver encounters first.

2. Another version is present authoritatively in the child zone. Some authoritative name server software will include it (with signatures) in the Authority section of authoritative answers for data from that zone, and some software will not. A resolver may or may not encounter this in the Authority section for an authoritative answer. When it does, it has trustworthiness value **A-**. A machine in the middle can strip this authoritative NS RRset without detection as it is not obligatory to include and resolvers can thus not expect it.

   Resolvers also encounter the authoritative version of the NS RRset when directly querying for it. The NS RRset will be authoritatively included in the Answer section of an authoritative response with trustworthiness value **A**.

   Resolvers query for the NS RRset directly either,

   1. Because they needed to resolve the data in response to a request for it, or

   2. Because they do delegation revalidation[8]

[8] See “Delegation revalidation”, https://datatracker.ietf.org/doc/draft-ietf-dnsop-ns-revalidation/

When the authoritative version of the infrastructure NS RRset for a DNSSEC zone is acquired by a validating recursive resolver, it cannot be changed by an attacker undetected. Since a resolver cannot rely upon explicit incoming requests for the authoritative version of the infrastructure NS RRset, the only reliable way for a resolver to be certain to obtain the data authoritatively is to perform delegation revalidation[8]. With delegation revalidation we mean proactively querying for authoritative infrastructure data. Similarly, glue from the additional section of referral responses (trustworthiness value **B**) cannot be protected with DNSSEC. The A and AAAA RRsets for the values of the NS RRset for a zone can also only be protected by DNSSEC when resolvers are revalidating them by querying for the authoritative (DNSSEC signed) data directly (trustworthiness value **A**). The Unbound resolver will do delegation revalidation when enabled with the harden-referral-path configuration option.

Although both the NS RRset as well as glue values around zone cuts should be insured to be consistent and remain to be so[9], in practice they may diverge. Because at least the parent side needs to lead to a working authoritative name server for the zone, for the zone to be resolvable at all, some, so called parent centric resolvers, choose to rely only on the parent-side non-authoritative versions of the infrastructure data[10]. These resolvers do not follow directions of RFC 2181 section 5.4.1.[7] and will never use authoritative infrastructure data.

Resolvers that do follow directions of RFC 2181 section 5.4.1.[7] are susceptible to GHOST domain attacks[11,12] and should take adequate counter measures, such as restricting the Time To Live (TTL) of the child side NS RRset to be equal or smaller than the TTL value of the parent side, or by revalidating the parent NS RRset regularly[8].

Conclusions on the impact of redirected query traffic

We started this section on the impact of redirected query traffic by saying that an attacker that manages to redirect query traffic to the root can ultimately potentially view all queries *for the entire domain name space* and alter all unsigned parts of responses without detection, *for most resolvers*. In subsection Infrastructure DNS data we pointed out that the attack is based on modifying the non-authoritative versions of infrastructure data. In subsection Ranking DNS data we showed how and when resolvers get the authoritative version of infrastructure data. We also showed that it can only get all of it reliably when performing delegation revalidation[8].

Thus, the impact spans to the entire domain name system only for resolvers that do not do delegation revalidation. For resolvers that *do* delegation revalidation, the attack is limited to only viewing queries to the root service and extending the redirected queries for only the delegations with infrastructure data that is not protected by DNSSEC. Delegation revalidation is not common as we will see in section Reduced risk based on the 2023 DITL data. The Unbound resolver supports delegation revalidation with the `harden-referral-path: yes` configuration option.

---

[9] See last paragraph of RFC 1034: DOMAIN NAMES - CONCEPTS AND FACILITIES, Section 4.2.2. Administrative considerations, https://www.rfc-editor.org/rfc/rfc1034#section-4.2.2

[10] See “Updating Resolver Algorithm”, https://datatracker.ietf.org/doc/draft-fujiwara-dnsop-resolver-update/

[11] Jiang, Jian, et al. "Ghost domain names: Revoked yet still resolvable." *NDSS 2012*, https://dspace.networks.imdea.org/handle/20.500.12761/708

[12] Li, Xiang, et al. "Ghost Domain Reloaded: Vulnerable Links in Domain Name Delegation and Revocation." *NDSS*. 2023, https://i.blackhat.com/Asia-23/AS-23-Li-Phoenix-Domain-Attack-wp.pdf

The risks of redirected query traffic

In the previous subsection we showed what the *impact* would be when an attacker would succeed to redirect all query traffic to the root to a machine under control of the attacker. In this subsection we will estimate the *chance* for such an attack to succeed.

The risks with (partial) on-path attackers

If the attacker is already on the path of one or more root server IP addresses, it will already see a certain amount of query traffic to the root service . The amount of query traffic it sees is the number of IP addresses the attacker is on-path for, proportional to the number of root server IP addresses that are reachable for the resolver: at the time of writing, 13 for IPv4 only resolvers, and 26 when the resolver is also IPv6 capable.

An on-path attacker that is also able to send a spoofed response to the resolver before the genuine response, can subsequently succeed in redirecting all queries to the root service to a machine under the control of the attacker by spoofing a priming response. At the time of writing, the TTL of the root NS RRset is $TTL_N = 518400$ seconds (6 days). Assuming that:

- the resolver under attack sends a root priming query every 6 days
- the resolver under attack does not revalidate the root server addresses,
- the on-path attacker has a 100% chance of returning a spoofed response

---

then, the on-path attacker will succeed to redirect query traffic to the root within

$\frac{26}{On\ path\ \#\ IPs} \cdot 518400$ seconds with an IPv6 capable resolver, or

$\frac{13}{On\ path\ \#\ IPs} \cdot 518400$ seconds with an IPv4 only resolver.

To give some concrete values, when an attacker has effectively a machine in the middle of 1 IP address, this amounts to a successful hijack of the whole root service within 156 days with an IPv6 capable resolver and 78 days with an IPv4 only resolver.

One of the preconditions above is that the victim resolver does not revalidate the root server addresses. Similarly as referral hijacks can be protected by delegation revalidation, the priming query can be too. Unbound will revalidate the root server IP addresses, when delegation revalidation is enabled. Knot Resolver always revalidates root server addresses.

At the time of writing the TTL for the authoritative A and AAAA RRsets for the root server IP addresses is $TTL_A = 3600000$ seconds (1000 hours, or 41⅔ days). Assuming that the attacker missed to spoof the initial priming response and the resolver cached all root server addresses authoritatively with a TTL of 41⅔ days, the chance of spoofing the next priming response is reduced $\frac{TTL_A}{TTL_N} = \frac{3600000}{518400} \approx 7$ times. It will take (slightly less than) 7 times longer for a complete hijack to succeed. This positive effect is reduced by the fact that most resolvers cap the TTL to a maximum. This is further elaborated upon, together with some estimates that are more likely to match reality in subsection Time to live maximum for RRsets in cache.

When the root name server data would be signed, then a resolver that is both doing root server addresses revalidating *and* is DNSSEC validating, is no longer susceptible to a query redirection attack. Such a resolver must validate the authoritative A and AAAA RRsets for the root server addresses. If it cannot do that, they are considered DNSSEC BOGUS and will be dismissed. The on-path attacker cannot extend the attack to be on-path for all root server identifiers, and is confined to just viewing all queries and modifying all the unsigned parts of responses (if it is quicker than the actual name server) only for the IP addresses for which it was already on-path.

The risks with off-path attackers

In RFC 5452: Measures for Making DNS More Resilient against Forged Answers[13], a formula is given for the combined chance of at least one successful spoofed response for an off-path brute force attack within a chosen time period:

$$P_{cs} = 1 - \left(1 - \frac{D \cdot R \cdot W}{N \cdot P \cdot I}\right)^{\frac{T}{TTL}}$$ where

$D =$ average number of identical outstanding queries of a resolver (typically 1)
$R =$ number of packets sent per second (70,000)
$W =$ window of opportunity, in seconds. (0.1)
$N =$ number of authoritative name servers (13)
$P =$ number of ports used (1)
$I =$ number distinct IDs available (65536)
$T =$ chosen time period
$TTL =$ Time To Live of the RRset (6 days)

with the chosen values for the symbols, this gives:

$$P_{cs} = 1 - \left(1 - \frac{1 \cdot 70000 \cdot 0.1}{13 \cdot 1 \cdot 65536}\right)^{\frac{\#\ days}{6}}$$

Formula 1: The combined chance of at least one successful spoofed response for an off-path brute force attack

We assume the attacker can send spoofed responses with the source IP addresses of the root name servers. We furthermore have chosen some values for the symbols which make the attack more feasible to succeed. We assume the victim resolver is IPv4, so we only have $N = 13$ authoritative name servers to spoof for. We also assume the attacker knows the destination port for the spoofed response ($P = 1$), perhaps through a side channel attack (such as for example the SAD DNS attack[14]). Also, the default value of R, the number of packets sent per second, in RFC 5452[13] was 7000 (resulting in a packet stream of 4.5 Mbit/s). We have chosen a ten-fold 70,000 packets per second resulting in a packet stream of 45 Mbit/s. We assume this traffic volume to be unnoticed by the operator of the victim resolver.

With all assumptions and (favorably) chosen values, according to the formula there is a 39% chance of success within 1 year, and a 64% chance of success within 2 years. Selecting slightly less pessimistic values, will reduce the risk. For example, with 1024 destination ports to guess ($P = 1024$), the risk is reduced to a 0.3% chance of success within 6 years.

---

[13] RFC 5452: Measures for Making DNS More Resilient against Forged Answers, https://www.rfc-editor.org/rfc/rfc5452
[14] See https://www.saddns.net/

Similarly to the on-path attack, the risk is significantly reduced when the victim resolver is revalidating the addresses of the root name servers. Changing the TTL symbol in the formula to the TTL for the authoritative A and AAAA RRsets for the IP addresses of the root name servers (41⅔ days at the time of writing), will result in a 6.8% chance of a successful attack within 1 year, and 13% within 2 years with $P = 1$. $P = 1024$ changes that to 0.04% within 6 years.

Also similarly to the on-path attack, when the root name server data would be DNSSEC signed, a resolver would no longer be susceptible to the redirected query attack, provided the resolver is revalidating the IP addresses of the root name server and DNSSEC validating.

Time to live maximum for RRsets in cache

One of the assumptions in the calculations above is that the resolver under attack caches RRsets in the priming response for the in the response given TTL values. In practice many resolvers cap long TTL values and this is in fact recommended by specification[15]. The recommended maximum TTL value to cache RRsets is 7 days, and this is also the default value at which the BIND resolver caps the TTL. The Unbound resolver, Knot Resolver and PowerDNS Recursor all cap at 1 day.

To have a better sense of the real world risks and how much revalidation would reduce that (with unsigned root name server data), we performed RIPE Atlas[16] measurements to determine at which values the TTL is capped by resolvers. We assume that it will reflect sufficiently close the state for all resolvers. The measurements consist of DNS queries for a RRset with the same TTL as the RRsets in `root-servers.net` (41⅔ days, almost 6 weeks). The query has a random left label in order to minimize the effect of different probes using the same resolver and returning earlier cached responses.

The perspective for our risk calculations were resolvers, and not the probability of users of resolvers being victim to query redirection. We calculate the risk of an attacker being successful to redirect queries from a *victim resolver* by spoofed responses from the authoritative side (or the querying side) of the resolver. To establish counting unique resolvers as seen from the authoritative side, we targeted a special for the purpose crafted authoritative name server responding to A queries with the IPv4 address of the querying resolver, and AAAA queries with the IPv6 address of the querying resolver. To establish counting all the IPv4 and all the IPv6 addresses used by the resolvers, we scheduled two measurements. One using a name that is only resolvable on IPv4[17], and one that is only resolvable on IPv6[18].

---

[15] See RFC 8767: Serving Stale Data to Improve DNS Resiliency, Section 4: Standards Action, https://www.rfc-editor.org/rfc/rfc8767#name-standards-action

[16] RIPE Atlas is an Internet data collection system by the Réseaux IP Européens Network Coordination Centre (RIPE NCC). It is a global network of devices, called probes, that actively measure the Internet. Anyone can access the results of public measurements. RIPE Atlas users can perform customized measurements. See: https://atlas.ripe.net/landing/about/

[17] https://atlas.ripe.net/measurements/70926548/

[18] https://atlas.ripe.net/measurements/70928145/

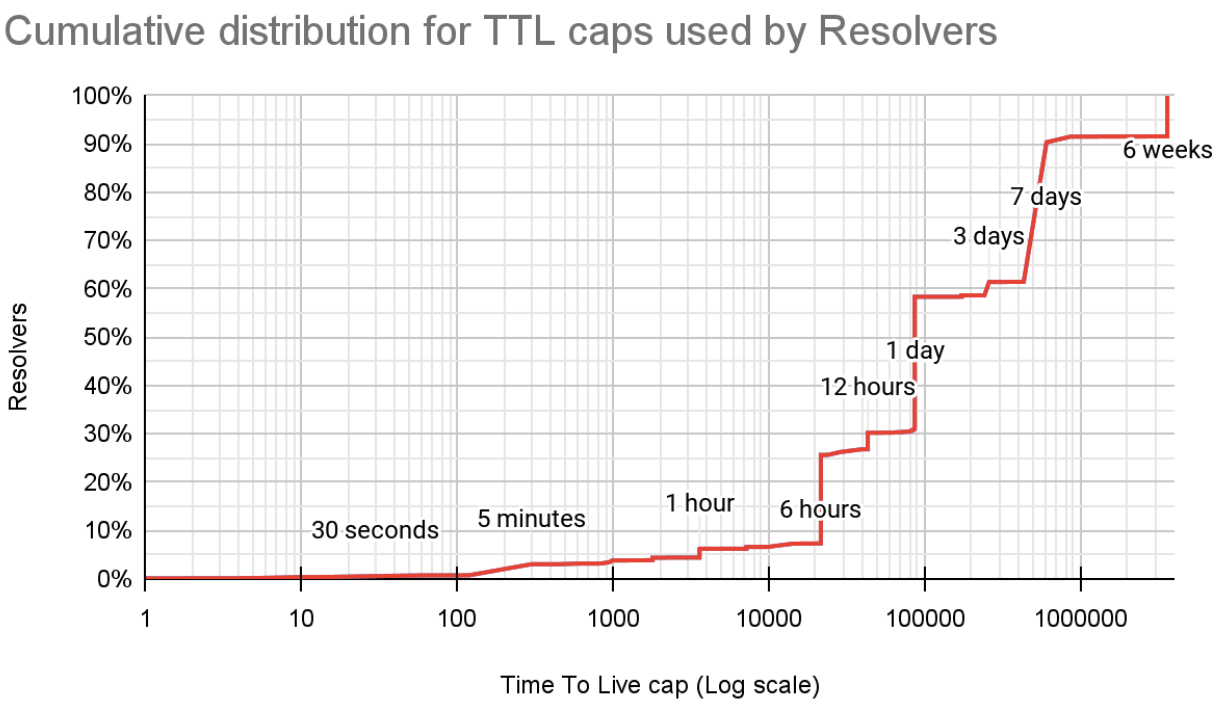

Figure 2: Cumulative distribution for TTL caps used by unique (authoritative side) IP addresses of resolvers as in use on RIPE Atlas.

The measurements ran roughly from 11:00 UTC on the 8th of May 2024 for 48 hours until 11:00 UTC on the 10th of May 2024. In this period 12,060 probes participated resulting in 16,371 unique IPv4 and 11,039 unique IPv6 resolver IP addresses (as perceived from the authoritative).

Resolvers as used by RIPE Atlas probes may be forwarding to other resolvers that may themselves be forwarding to other resolvers. The capping value in the response will be the one of the resolvers with the lowest capping value in the chain. To minimize these effects we took per resolver IP address at the highest capping value seen in responses.

The cumulative distribution for the highest seen TTL capping values per unique resolver IP address is shown in Figure 2:

- 58% of the resolvers cap at a TTL of 1 day or less
- 8% of the resolvers do not cap, or at least not below the `root-servers.net` TTL

This means that the above calculated risks for off- and on-path attacks are only correct for 8% of the resolvers. The risk of falling victim to an redirected query attack from a spoofed priming response is for the majority of resolvers on RIPE Atlas significantly higher. An attacker that is on-path of 1 root server's IPv4 address and can deliver responses faster than the actual root server has for 58% of the resolvers a chance to redirect all traffic to the root service within 13 days (for an IPv4 only resolver). The off-path attack (with the same for the attacker favorable chosen parameters) has the same chance of success in 6 times shorter periods for those 58% of resolvers. 39% chance of success in 2 months, and 64% in 4 months.

We pointed out that revalidation of the root server address RRsets will eliminate risk for redirected queries when root name server data would be DNSSEC signed, but that it also already helps when root name server data would not be DNSSEC signed. This is still the case for TTL capping resolvers against off-path attackers. This stems from the fact that to provide alternative addresses for all the root servers, 26 responses need to be spoofed, which is considerably harder than to equip false root server addresses for all the root server identifiers at once in the priming response.

Revalidation does not help against on-path attackers for resolvers that cap at a TTL lower than the non-authoritative root server addresses in the priming response (TTL_N). We expressed the reduced risk as the reduced rate of opportunity to provide a spoofed priming response; it will take $\frac{TTL_A}{TTL_N}$ times longer for the opportunity for a spoofed priming response to succeed. This "weight" (W_A) needs to be adjusted to take the minimum of the cap value, and authoritative and non-authoritative TTLs: $\frac{\min(TTL_A, TTL_{cap})}{\min(TTL_N, TTL_{cap})}$.

When applying that to the TTLs from our measurement, we find 7 distinct values (see Table 2). Taking the average from that distribution, we can say that it will take 1.6 times longer for a complete hijack to succeed for an on-path attack without signed root name server data.

percentage of resolvers	$W_A = \frac{\min(TTL_A, TTL_{cap})}{\min(TTL_N, TTL_{cap})}$
61.4%	1
28.9%	1.1666…
1.2%	1.666…
8.4%	6.9444…
average weight: $\overline{W_A} \approx$	1.6

Table 2: Reduced risk weight from the authoritative proportional to the non-authoritative TTLs

Conclusions on redirected query traffic

In section The impact of redirected query traffic we evaluated how a successful redirected query traffic attack for the root can potentially be extended to span the entire domain name space. We concluded that for resolvers that do delegation revalidation[8] and are also DNSSEC validating, the impact is limited to only viewing queries to the root service and extending the redirected queries for only the delegations with infrastructure data that is not protected by DNSSEC. We pointed out that the Unbound resolver supports delegation revalidation, but disabled by default.

In section The risks of redirected query traffic we evaluated the chance for a resolver to fall victim to a redirected query traffic attack for the root. We evaluated both (partial) on-path as well as off-path attacks. We showed that resolvers, that revalidate the root server IP addresses after a priming response, have a significantly smaller chance to fall victim to the attack; and, that the chance would be reduced to zero for root server address revalidating resolvers that are also DNSSEC validating. We pointed out that Knot Resolver will always revalidate root server addresses, and that the Unbound resolver has support for it, but disabled by default.

DNSSEC signed root zone name server data

In this section we address some aspects of signed root zone name server data. First we mention that there currently already is a DNSSEC signed statement which includes the root server IP addresses, albeit not the authoritative version of the data, in the form of a Message Digest for the root zone. In section DNSSEC signed root server addresses in the priming response we will show that, at the time of writing, there is no added security or reduced risk of redirected query traffic by including DNSSEC signatures in the priming response.

Message Digest for the root zone

Since December 6, 2023[19], the root zone contains a DNSSEC signed cryptographic message digest[20], in the form of a ZONEMD RR, over the whole zone. The digest covers all data including all non-authoritative data such as the A and AAAA RRsets for the IP addresses of the root server identifiers, as well as the NS RRsets and glue that make up the delegations.

The root zone can be served local to a Resolver[21]. The root zone can be transferred by AXFR over TCP from several root server operators as well as from DNS servers at ICANN[22]. Several recursive resolver software implementations can natively transfer the root zone and DNSSEC validate and verify the ZONEMD RRset[23]. This resolver software will prime from its local copy of the root zone whose integrity and origin authenticity has been verified. It is not susceptible to redirected query traffic attacks. Not just for the root servers, but also redirected query attacks on the infrastructure data for the delegations is protected by this mechanism.

Revalidating resolvers that revalidate infrastructure data whose authoritative version is not DNSSEC protected may still be susceptible to query redirection attacks on those (unsigned) authoritative responses. However, authoritative responses are the hardest to spoof, because of the high ranking (**A)** and the restriction on the source for such data (see Table 1). Furthermore, for a full redirected query attack in which all name servers of a delegation are redirected, all the individual authoritative responses need to be spoofed, which is also significantly harder than spoofing all infrastructure data at once in a single spoofed referral response.

DNSSEC signed root server addresses in the priming response

During the RSSAC028 Implementation study[3], several alternative naming schemes used for individual root servers (from the RSSAC028 report[2]) were evaluated. We reference the naming schemes by the section number in which they are described in RSSAC028. Naming schemes 5.3 and 5.3.1 “In-zone NS Names” and naming schemes 5.6 and 5.6.1 “Single Shared Label for All Operators” (Figure 3) all have the A and AAAA RRsets for the root server IP addresses authoritatively within the root zone.

---

[19] See https://lists.dns-oarc.net/pipermail/dns-operations/2023-December/022388.html
[20] See RFC 8976: Message Digest for DNS Zones, https://www.rfc-editor.org/rfc/rfc8976.html
[21] RFC 8806: Running a Root Server Local to a Resolver, https://www.rfc-editor.org/rfc/rfc8806
[22] See RFC 8806, Appendix A: Current Sources of the Root Zone https://rfc-editor.org/rfc/rfc8806#section-appendix.a
[23] At the time of writing the recursive resolver software that supports both RFC 8806 and RFC 8976 is BIND since version 9.19, PowerDNS Recursor since version 4.7.0 and Unbound since version 1.13.2.

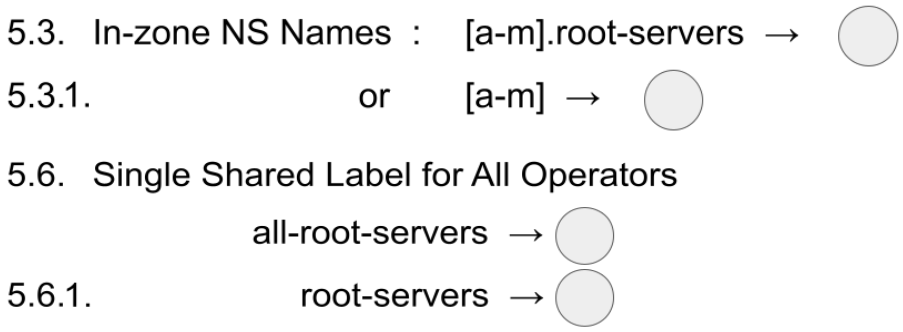

Figure 3: Naming schemes from RSSAC028 which have the A and AAAA RRsets for the root server IP addresses authoritatively within the root zone.

We observed (during the RSSAC028 Implementation study) that some authoritative name server software in use on the root servers, does include the DNSSEC signatures for the additional address records in the priming response, and other root name server software does not[24]. However the presence or absence of signatures did not influence resolver behavior.

The names of the root server identifiers have a name that is in the DNS hierarchy below the root itself. There is no indication if that name is below a delegation or not. There is no way for a resolver to tell if the names for the root server identifiers are below a delegation or authoritatively present within the root zone. Therefore, resolvers cannot expect signatures because there is no indication of whether the RRsets are authoritatively in the response or not. An adversary spoofing a priming response can therefore include false A and AAAA RRsets in the additional section without DNSSEC signatures without consequences for any naming scheme in RSSAC028.

```
.       IN     NS     .
.       IN     RRSIG  NS 8 0 ...
.       IN     A      198.41.0.4
  .     IN     A      170.247.170.2
.       IN     A      192.33.4.12
; the other records in the A RRset...
.       IN     RRSIG  A 8 0 ...
.       IN     AAAA   2001:503:ba3e::2:30
.       IN     AAAA   2801:1b8:10::b
.       IN     AAAA   2001:500:2::c
; the other records in the AAAA RRset...
.       IN     RRSIG  AAAA 8 0 ...
```

Single Shared (root) Label for all Operators naming scheme in which it is certain that the RRsets for the root server addresses are authoritatively present with the root zone

We have come up with a naming scheme with which the resolver can assume that the additional addresses are authoritatively within the root zone. This scheme, similar to 5.6 and 5.6.1, has a Single Shared Label for All Operators, where the label is also the empty (root) label.

[24] See section Priming responses properties on page 33 of the RSSAC028 Implementation study report, https://icann.org/en/system/files/files/rssac028-implementation-study-report-27sep23-en.pdf#h.iww5bv476nuk

Provoking direct root server address queries

In the previous sections we saw that the only way resolvers can benefit from signed root name server data, is by direct queries for the authoritative A and AAAA RRsets for the root server IP addresses. In this section we will evaluate if those direct queries can in any way be provoked from the root servers. In section Returning fewer addresses in the priming response we will show that provoking more direct queries with fewer additional addresses in the priming response is not feasible because of divergent behavior of the different resolver software. In section Reducing the TTL of the non-authoritative address RRsets we cover that reducing the TTLs of the non-authoritative address RRsets in the priming response does not provoke direct queries with the resolver software we tested either.

Returning fewer addresses in the priming response

In the RSSAC028 Implementation study[3] we observed priming responses from different authoritative name server software in response to root priming queries sent with a variety of query parameters from the different resolver software we evaluated. These responses sometimes had all the IP addresses for all the root server identifiers, sometimes a few and sometimes no additional addresses. Table 3 shows follow-up root server address queries behavior for the tested resolver software in response to those absent in the priming response.

tested resolver software	*version(s)*	*behavior to no, partial or completely missing root server addresses in the additional section of the priming response*
PowerDNS Recursor	4.0.9, 4.1.15, 4.2.1, 4.7.5, 4.8.4	Never query root server addresses
Unbound	1.5.10, 1.6.8, 1.7.3, 1.8.3, 1.9.6, 1.13.0, 1.14.0, 1.17.1	Query for missing root server addresses
BIND	9.9.11, 9.10.8, 9.11.6, 9.12.4, 9.13.7, 9.14.10, 9.15.8, 9.16.41, 9.18.15, 9.19.13	Query all root server addresses, but only when there were none in the additional section of the priming response
BIND	9.9.11	Queries root server addresses when the names of the root server identifiers (RSI) differ from the ones built-in or from hints
Knot Resolver	5.5.3 and 5.6.0	Always query for all root server addresses

Table 3: Follow-up queries for the authoritative versions of the A and AAAA RRset for the IP addresses of the root servers in response to their absence in the additional section of the priming response

Only Unbound queries for missing root server addresses. BIND only queries for all of them if none were present. However, provoking direct queries by leaving out all of them is not a deployable option since the tested versions of PowerDNS Recursor never query for them. Worse, in section Missing additional addresses of Group I and II name servers on page 38 of the RSSAC Implementation study report[25], we can see that the tested versions of PowerDNS Recursor fail resolving alltogether after having received a priming response without any additional addresses.

Reducing the TTL of the non-authoritative address RRsets

We briefly executed another experiment on the Resolver testbed developed and used for the RSSAC028 Implementation study. The experiment had lower TTLs (With TTL value 1) for the non authoritative versions of the A and AAAA RRsets for the IP addresses for the root server identifiers for the current naming scheme. We did not observe any direct queries for the authoritative versions of the A and AAAA RRsets from any of the resolver software in the duration of the experiment (±20 seconds). We conclude that reducing the TTL of the non-authoritative address RRsets is not a feasible method to provoke direct queries.

Reduced risk based on the 2023 DITL data

In the previous sections we saw that DNSSEC validating resolvers could benefit from DNSSEC signed authoritative A and AAAA RRsets for the root server IP addresses, in the sense that those authoritatively learned IP addresses are genuinely belonging to a root server identifier and that traffic for that root server identifier cannot be redirected by spoofing a fake IP address. We can thus express the reduced risk for redirected query traffic, as the number of authoritatively acquired root name server addresses (that can be validated when signed) in proportion to the non-authoritatively acquired root name server addresses. This can be done for a single and also for a set of resolvers.

We also saw in section The risks of redirected query traffic that most resolvers do not normally query for all of the authoritative root server addresses on their own initiative. We evaluated different ways to let resolvers learn them by server-side measures in sections DNSSEC signed root server addresses in the priming response and Provoking direct root server address queries, but to no avail. Therefore we can only determine reduced risk of redirected query traffic with signed root name server data, based on the authoritatively acquired root server IP addresses by resolvers' own initiative as observed in real-life packet captures..

The Day in the Life of the Internet (DITL) is a data collection initiative which is coordinated annually by the DNS-OARC. DNS operators of significant zones (such as the root zone) run packet captures over the same 48 hour period and contribute them to the DITL data set. DITL data is made available under the terms of the OARC Data Sharing Agreement[26].

[25] https://icann.org/en/system/files/files/rssac028-implementation-study-report-27sep23-en.pdf#h.5w64x9ibbh1w
[26] https://www.dns-oarc.net/files/agreements/oarc-datashare.pdf

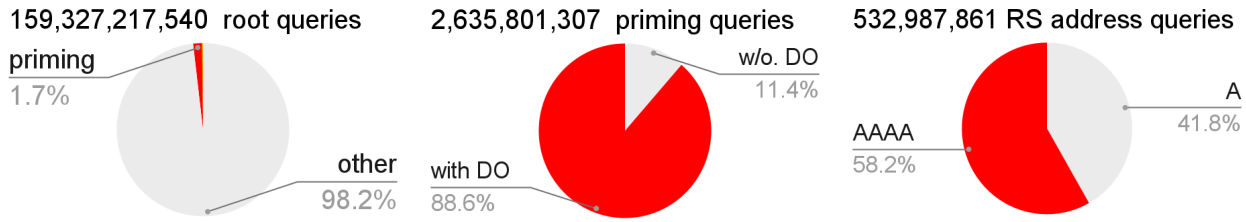

Figure 4: Counted root server queries from the DITL 2023 data set. The charts show from left to right: all root queries, priming queries and Root Server (RS) address queries (with DO)

In 2023, nine of the twelve root server operators contributed query traffic captured from roughly the 11th of April 2023 11:00 UTC till the 13th of April 13:00 UTC. In this period, 1.7% of the queries in those captures were priming queries.

The `root-servers.net` zone contains 13 A RRsets and 13 AAAA RRsets, each RRset consisting of a single resource record. At startup, resolvers have 26 addresses for the root servers (either built-in or from a hints file), which are non-authoritative and not DNSSEC validated. Since the addresses conveyed in the priming response are also non-authoritative and not DNSSEC validatable, all resolvers still have 26 unvalidated non-authoritative addresses for the root servers immediately after receiving the first priming response, regardless of how many addresses were conveyed in the priming response. Some resolvers will then query some addresses authoritatively.

The `root-servers.net` zone is also served by the root servers. The root server DITL captures also contain the queries for the A and AAAA RRsets for the root server IP addresses. The rightmost pie chart in Figure 4 shows the counts for those queries that also had the DO bit set. The authoritatively acquired addresses have a longer TTL (TTL_A) than the non-authoritative ones from the priming response (TTL_N). Assuming that most resolvers will prefer and retain authoritatively acquired RRsets[27], the protective effect of the authoritative RRsets will last $\frac{TTL_A}{TTL_N}$ longer and their rate of appearance (in the captures) will be $\frac{TTL_A}{TTL_N}$ less frequent in proportion to cache occurrence. Their weight compared to the non-authoritative RRsets is $\frac{TTL_A}{TTL_N}$. But only if resolvers would adhere to those TTLs.

In section Time to live maximum for RRsets in cache we saw that most resolvers have a maximum for the TTL value in cache and will cap the TTLs of an RRsets to that maximum when storing them. The weight of the authoritative TTL needs to take that maximum into account and take the minimum of the cap value and the observed authoritative and non-authoritative TTLs: $W_A = \frac{\min(TTL_A, TTL_{cap})}{\min(TTL_N, TTL_{cap})}$. Table 2 shows the distribution of weight values observed with the resolvers on RIPE Atlas[17]. We can take the average overall weight to come to an overall reduced risk assessment.

[27] See section Ranking DNS data

We also take the IPv6 capability of resolvers along in our reduced risk assessment. We assume that resolvers that are IPv6 capable are also IPv4 capable and can use 13 IPv4 + 13 IPv6 = 26 root server IP addresses, while resolvers that are IPv4 only can only use the 13 IPv4 addresses. We assume that the number of IPv6 only resolvers is negligible. The fraction of IPv6 capable resolvers is taken from the same RIPE Atlas measurements that we used to generate Table 2. We measured 16,371 unique IPv4 and 11,039 unique IPv6 addresses with which resolvers, in use by the probes on RIPE Atlas, query the authoritative name servers. Assuming that the IPv6 resolvers are also IPv4 capable, the fraction of IPv6 capable resolvers is $q_{46} = \frac{11039}{16371} \approx 0.67$.

Taking all the aforementioned aspects into consideration we come to the following formula:

$$Q_{prot.} = W_A \cdot \frac{r_A}{r_N} \quad \text{where} \quad W_A = \frac{\min(TTL_A, TTL_{cap})}{\min(TTL_N, TTL_{cap})},$$

Weight of authoritative root server address RRsets compared to the non-authoritative from priming

$$\overline{W}_A \cdot \frac{r_A}{r_N} \approx 1.2\% \qquad \overline{W}_A \approx 1.6$$

The average overall weight of authoritative root server address RRsets compared to the non-authoritative

$TTL_A =$ TTL of the authoritative root server address RRsets

$TTL_N =$ TTL of the non-authoritative root server address RRsets

$TTL_{cap} =$ Value to which resolvers will cap the TTL in cache

$r_A = n_a + q_{46} \cdot n_{aaaa}$
Number of usable authoritative addresses from queries

$r_N = n_p \cdot q_4 \cdot 13 + n_p \cdot q_{46} \cdot 26$
Number of usable non-authoritative addresses, built-in, from hints and from priming responses

$n_p =$ Number of priming queries

$n_a =$ Number of A queries

$n_{aaaa} =$ Number of AAAA queries

$q_{46} =$ Fraction of IPv6 capable resolvers *(≈ 0.67)*

$q_4 =$ Fraction of IPv4 only resolvers = 1 - q_{46}

Formula 2: The reduced risk of redirect query traffic with signed DNSSEC name server data, expressed as the number of authoritatively acquired root server address RRsets in proportion to the non-authoritatively acquired root server address RRsets in cache

Applying the query counts from the root server DITL data 2023 to Formula 2 with the average overall weight of authoritative root server address RRsets (from the distribution in Table 2), results in a **1.2% overall reduced risk of redirected query traffic by signed root name server data** for the period in which the DITL traffic was captured.

The risk reduction mentioned above is not equally distributed. As can be seen in Figure 4, 11.4% of the resolvers did not send the EDNS DO flag. We can assume that they are not DNSSEC capable, will not DNSSEC validate the authoritative RRsets and will not contribute to reduced risk of redirected queries.

We also saw 1,725,832 `root-servers.net` NS queries (NS_A) in the DITL data. We consider it likely that these resolvers were revalidating the root server IP addresses, as both Knot Resolver and Unbound query for it when revalidating them. Assuming that only Knot Resolver and Unbound revalidate root server address RRsets, then we do not need to amplify the prominence of the authoritative RRsets since TTL_{cap} (1day) is below both TTL_A and TTL_N and $W_A = 1$. $\frac{NS_A}{n_p} \approx 0.1\%$ resolvers which are root server IP address revalidating (r_R) would be completely protected against query redirection with signed name server data.

With the remaining $88.6\% - 0.1\% = 88.5\%$ resolvers that did send a priming query with EDNS DO flag (n_{DO}), reduced risk ($Q^i_{prot.}$) is for each weight (W^i_A) in each row (i) in Table 2:

$Q^i_{prot.} = W^i_A \cdot \frac{r_A}{n_{DO} \cdot q_4 \cdot 13 + n_{DO} \cdot q_{46} \cdot 26}$, which results in the reduced risk distribution as shown in Table 4 below.

percentage of population	*revalidating root server addresses*	*DNSSEC validating*	*weight of authoritative RRsets (W_A)*	***reduced risk***
11.4%				0 %
54.4%		✓	1	0.9%
25.6%		✓	1.1666…	1.0%
1.0%		✓	1.666…	1.4%
7.5%		✓	6.9444…	5.9%
0.1%	✓	✓		100 %

Table 4: Distributed risk reduction by percentage of resolver population with certain properties

Conclusions

This report contains a response to the second recommendation of the RZERC002 report:

> Recommendation 2: The RZERC recommends that ICANN org further explore the cost / benefit tradeoffs and risks of signed root zone name server data. Do the risks of redirected query traffic outweigh the risks of increased operational complexity?

We restricted the question in the recommendation to evaluating and quantifying the reduced risk of redirected query traffic from signed root zone name server data, by formulating two research questions in subsection Research question and Scope.

Based on the notion that authoritatively acquired signed A and AAAA RRsets are not susceptible to a redirected query traffic attack when the victim resolver is DNSSEC validating (see section The risks of redirected query traffic), we expressed the reduced risk as the number of authoritatively acquired root server address RRsets in proportion to the non-authoritatively acquired root server address RRsets (in section Reduced risk based on the 2023 DITL data). We further noted (in section DNSSEC signed root server addresses in the priming response) that the authoritative root server address RRsets can *only* be obtained by the resolver directly querying for them. In section Reduced risk based on the 2023 DITL data, we analyzed 2023 DITL data for those queries and arrived at 1.2% overall reduced risk. This means that if the overall risk for redirected query traffic would be P, the reduced risk would be $P \cdot (1 - 0.012)$.

In the 2023 DITL data we also saw 0.1% of resolvers revalidating the `root-servers.net NS RRset`, which was considered an indication of the resolver revalidating all root server IP addresses. These resolvers would no longer be susceptible to redirected query traffic with signed root name server data. Knot Resolver already revalidates all root server addresses by default. The Unbound resolver can be enabled to do it with a configuration option.

With this, we can answer our two research questions:

1. Does signed root zone name server data reduce the risks of redirected query traffic?

   It does for resolvers that *query* for the authoritative root server address RRsets

2. To what extent?

   We measured 1.2% reduced risk based on the 2023 DITL data.
   We assume the actual reduced risk to be close to this value.

And, coming back to the question posed in RZERC002 Recommendation 2:

> Do the risks of redirected query traffic outweigh the risks of increased operational complexity?

We do not feel we are in a position to estimate the operational complexity of signing `root-servers.net`, but we do notice that since very few resolvers depend on the full set of authoritative root server address RRsets (0.1%), there would also be limited risk of resolvers breaking because of signed address RRsets. It may be an opportunity to deploy a DNSSEC signed `root-servers.net` now and establish a strong case for resolvers to start revalidating the root server addresses.

Further reducing the risks

More direct queries for the authoritative root server address RRsets would further reduce the risk of redirected query traffic, including while root name server data remains unsigned (as shown in sections The risks with (partial) on-path attackers and The risks with off-path attackers). Unfortunately resolvers cannot reliably be provoked by the root servers to do more direct queries as we pointed out in section Provoking direct root server address queries. Some resolvers always send direct queries for all the root server address RRsets. We named this resolver behavior *revalidation of the root server address RRsets*. The Knot resolver software does this by default, and Unbound can be configured to do so.

Revalidating root server address RRsets reduces the risk of redirected query traffic, but as we pointed out in section The impact of redirected query traffic, the *extent* of a successful redirected query traffic attack would be significantly limited when resolvers would do *delegation revalidation*[8]. The Unbound resolver can be configured to do delegation revalidation.

Revalidation of the root server addresses is a subset of delegation revalidation, in the sense that the priming query is already a revalidation of the built-in root server names (or from a hints-file). Revalidation of the root server addresses would complete it.

- The higher up in the DNS tree (the closer to the root) revalidation is done, the more positive impact revalidation has.
- If more resolvers would revalidate the root server address RRsets, then the risk for redirected query traffic will be further reduced. Already with unsigned root name server data.

Alternatively, a root zone local to the resolver with a verified and validated ZONEMD RR (as discussed in section Message Digest for the root zone), would provide protection similarly strong to the combination of revalidating the root server IP addresses and the delegations of the top level domains. In the 2023 DITL data we observed 4,174,339 transfers of the root zone, 0.2% in proportion to the total number of priming queries.

What if all resolvers would revalidate the root server address RRsets

If all resolvers would revalidate the root server address RRsets, then the amount of traffic to the root service will increase. With the current naming scheme of `root-servers.net`, There will be 26 extra revalidating root server address queries and 1 extra revalidating `root-servers.net` NS query per priming response. These RRsets have overall on average W_A times longer TTL, so their rate should be divided by this factor. There will also be an extra `.net` NS query with, at the time of writing, the same TTL as the non-authoritative version. With n_r the total number of queries to the root and $n_{addr.} = 753{,}598{,}915$ the total number direct root server address queries (also the ones without DO flag), the increase of traffic will be: $\frac{n_r - n_{addr.} + \frac{27 \cdot n_p}{W_A} + n_p}{n_r} \approx 30\%$, based on DITL data from 2023.

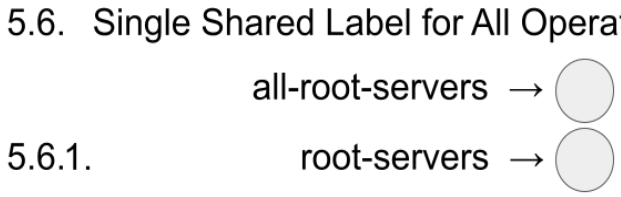

Figure 5: Naming scheme 5.6 from RSSAC028

Naming schemes with fewer root server address RRsets, such as with naming scheme 5.6 from RSSAC028[2], "Single Shared Label for All Operators", cause less traffic increase. With this scheme there are just 2 address RRsets, and no further NS RRsets to revalidate. The traffic increase will be: $\frac{n_r - n_{addr.} + \frac{2 \cdot n_p}{W_A}}{n_r} \approx 1.7\%$.

Other mitigations

Revalidating the root server address RRsets with signed name server data, together with a root zone local to the resolver with verified and validated ZONEMD, are the only measures that would eliminate the risk of redirected query traffic for the individual resolver doing it, and are the only measures that also work for on-path attackers. Alternatively the risk of redirected query traffic can be reduced (for off-path attackers only) by better protecting the DNS transaction.

DNS-Cookies[28] are a lightweight DNS transaction security mechanism that provides limited protection to DNS servers and clients against a variety of denial-of-service and amplification / forgery or cache poisoning attacks by off-path attackers. When properly configured, all name servers that share the same IP address (in an anycast set), should also share the same secret for generating and validating server cookies. If that is the case, then DNS Cookies help victims of reflection amplification attacks by preferring to service returning clients (that can show a valid server cookie), opposed to clients from which the source IP may be spoofed. But even if not properly configured, DNS Cookies, add 64 bits of entropy in the client cookie to be verifiable by the querier.

DNS Cookies are enabled by default in the BIND name server software since version 9.11.0 (october 2016). Many resolvers already send DNS Cookies, and some root servers already respond to DNS Cookies. In the 2023 DITL data set we observed 234,305,425 priming queries (8.9% of all priming queries) to contain DNS Cookies. From those queries 10,579,831 (0.4% of all priming queries) had a server cookie and enjoyed increased security with 64 extra bits of entropy protecting the transaction.

[28] See RFC 7873: Domain Name System (DNS) Cookies, https://rfc-editor.org/rfc/rfc7873 and RFC 9018: Interoperable Domain Name System (DNS) Server Cookies, https://rfc-editor.org/rfc/rfc9018

Other considerations

So far in this report we have only considered securing the IP addresses associated with a root server identifier, but the IP addresses themselves can also be hijacked in the routing system. DNSSEC only protects the relation between the root server identifiers and the IP addresses, but does not help against routing hijacks. One way to defend against route hijacking is to engage in the Resource Public Key Infrastructure (RPKI) and to have cryptographically signed Route Origin Authorizations (ROAs) associating the autonomous system numbers (ASNs) with the prefixes that contain the IP addresses on which the root name servers provide their service.

Acknowledgements

Many thanks to Jennifer Bryce, Matt Larson and Paul Hoffman for their patience, support and positive feedback. Thanks to Paul Hoffman for pointing out the relevance of revalidation already in an early stage of this work. The discussion we had during the IETF119 in Brisbane surrounding the re-evaluation of delegations and what infrastructure data is used by resolvers, were instrumental to further sharpen our thinking about it. Many thanks for sharing their insights to Mark Andrews, David Blacka, Manu Bretelle, Paul Hoffman, Shumon Huque and Ralf Weber. Thanks to the Root Server Operations participants for letting us present an early, slightly rough, version of this study. Also, many thanks to the root operators and the DNS-OARC for coordinating the collecting and making available of root server packet captures through the DITL data collection initiative. The DITL data was vital for this research. Without it we would not have been able to put a number to *the reduced risk of redirect query traffic with signed root name server data*.

Appendix A - 2023 DITL data query counts

This appendix contains the absolute values of the variables that are used in the formulas in this report. These mainly concern query counts from packet captures for 10 of the 13 root server identifiers, that were captured from roughly the 11th of April 2023 11:00 UTC till the 13th of April 13:00 UTC and made available in the 2023 DITL data set.

What	*Symbol used for the variable*	*Amount*
Total queries to the root	n_r	159,327,217,540
Total priming queries	n_p	2,635,801,307
Total priming queries with DO flag	n_{DO}	2,336,201,765
Total root server address A queries with DO flag	n_A	222,669,199
Total root server address AAAA queries with DO flag	n_{AAAA}	310,318,662
Total root server address queries without DO flag		220,611,054
Total root server address queries	$n_{addr.}$	753,598,915
Total `root-servers.net` NS queries	NS_A	1,725,832
Total root zone transfers (. AXFR)		4,174,339
Total queries with a DNS Cookie		19,876,582,988
Total queries with a DNS Server Cookie (more secure)		3,076,739,818
Total priming queries with a DNS Cookie		234,305,425
Total priming queries with a DNS Server Cookie (more secure)		10,579,831
TTL of the non-authoritative `root-servers.net` RRsets	TTL_N	518,400
TTL of the authoritative `root-servers.net` RRsets	TTL_A	3,600,000
Number of (authoritative side) resolver IPv6 addresses[29]		11,039
Number of (authoritative side) resolver IPv4 addresses[30]		16,371
IPv6 capable resolvers	$q_{46} = \frac{10897}{16180} \approx$	0.67
IPv4 only resolvers	$q_4 = 1 - q_{46} \approx$	0.33

[29] From RIPE Atlas measurement: https://atlas.ripe.net/measurements/70928145/
[30] From RIPE Atlas measurement: https://atlas.ripe.net/measurements/70926548/