DNS Resolver Recommendations

Author: Shane Kerr (Chair of the DNS Resolver Best Common Practice Task Force)
Document ID: ripe-823
Date: May 2024

Table of Contents

Terminology

- Open DNS Resolver: A DNS resolver that accepts queries from any client. Often the result of misconfiguration.
- Public DNS Resolver: A resolver intentionally configured to be an open resolver.

Introduction

What Is This Document? Who Is It For?

This document presents recommendations and best current practices for operating DNS resolvers, both public and non-public ones. It covers technical aspects of operations and provides best practice recommendations for data management, with a particular focus on user privacy, security, and resilience. The document serves as guidance for the wider Internet community, offering input to:

- Those running DNS resolver services, and
- Those who want to make informed choices between such services.

Its purpose is to provide clear guidance and promote effective practices in DNS resolver operations.

The intended audience is not the entire DNS community. Advice here is probably not useful for operators of authoritative servers, domain registrars, and so on. It is also not meant to be an introductory or educational document. There are many documents which cover the basics of DNS and the roles of organisations in it; a good overview is: Addressing the challenges of modern DNS - a comprehensive tutorial by van der Toorn et al.

The document does not consider how to measure adherence to these recommendations. So it is not intended to be used for certification, although certification created based on the principles here is possible.

How Is This Document Organised?

This document has a number of sections, and specific recommendations in each section. The intent is for each recommendation to have clear guidance at the top, and then background and discussion related to the recommendation afterwards. Each recommendation indicates whether it is mostly for operators of public DNS resolvers or for operators of any resolver.

About Recommendation Text

This is not a standards document and does not propose any way to measure compliance or interoperability. It does use words like "should" or "may be" throughout. These are meant to be interpreted in the usual English sense, and not as IETF-style RFC 2119 jargon.

System and Network Hardening

Infrastructure Considerations

Running any Internet service requires attention to the infrastructure used to operate it. This section discusses various approaches that can be used to run a DNS resolver. Everything applies to both public and non-public DNS resolvers.

Bare Metal or Public Cloud

All DNS resolver software can run either on dedicated servers (rented or colocated), or in virtualised clouds, or in a combination of those. Every approach has pros and cons. Most of these are not specific to running DNS resolvers, however, some of them are.

Running DNS resolver instances as OS level daemons on bare metal hosts.

Pros:

- Performance: Bare metal servers have direct access to the underlying hardware, and can offer superior performance/cost balance by avoiding the overhead associated with virtualisation. Moreover, you have full control over the server's configurations, down to the hardware level, which can be beneficial for performance and cost optimisation once you get the understanding of your typical workload during peak hours.

- Data Security: Since you are in control of the physical servers, there is no risk of data leakage that can occur due to vulnerabilities in multi-tenant virtualisation platforms, including CPU cache-based side-channel vulnerabilities. It could be argued that attacks targeting such issues are rare, and their impact on a DNS resolver service is low, but potential breaches may have a significant privacy impact. It is advised to evaluate this against your organisation's risk model.

- Predictability: Because there is no virtualisation layer and no "noisy neighbours" on the host, the performance of your servers is more predictable.

Cons:

- Cost of failure: If you pick hardware configuration that is not optimal for the workload of your DNS resolver, you may need to upgrade and replace hardware components afterwards. Ways to reduce this risk include renting servers instead of buying them, carrying load testing with data similar to production workloads, and providing limited beta access to the service before it fully enters the production phase.

- Scalability: Scaling up with physical servers means acquiring or renting, installing, and configuring new hardware, which will take more time than provisioning new virtual servers in a cloud environment. Moreover, most cloud environments will provide you with cluster autoscaling features, which could barely be achieved in bare metal.

- Maintenance: You will be responsible for all server maintenance tasks, including hardware issues, which can require significant effort and specific expertise.

- Redundancy: Setting up high availability and disaster recovery strategies can be more complex and time consuming compared to the cloud, where these features are often provided as value added products.

Running DNS resolver instances in containers in a public cloud.

Pros:

- Scalability: Clouds excel at scaling applications. You can scale up and down rapidly based on load, which is important for a DNS resolver that needs to handle variable query loads. In case of regional or geographically distributed resolvers, in every region where the resolver would be deployed, daily periodicity is likely to be observed, for example peak hour is likely to occur around 19:00 local time, and off-peak hours may begin at around 01:00-03:00. In a situation like that, using cluster autoscaling features and tools, you can run less instances in the night and more instances throughout the day, which may help to optimise your cloud hosting costs.

- Fault Tolerance and High Availability: Most clouds have built-in strategies, features, and products for handling node failures, which can increase your service's availability.

- Deployment and Management: Cloud providers offer built-in methods to deploy and manage applications, which can simplify operations and reduce the likelihood of human errors if your infrastructure management department is already familiar with these tools.

- Cost: While this largely depends on your specific usage, cloud services can sometimes be more cost-effective than managing your own physical servers, especially when you consider the total cost of ownership, including power, cooling, and maintenance.

Cons:

- Performance: The virtualisation layer of public clouds can impact performance. While this certainly could be mitigated through scaling the number of virtual hosts, the cost would also increase accordingly.

- Complexity: Advanced cloud technologies are complex systems which come with a steep learning curve. Without prior experience, properly configuring and managing a cloud-based compute cluster can be challenging.

- Cost Variability: While the cloud can be cheaper, it can also be more expensive if not properly managed. Costs can rise unexpectedly based on traffic. Make sure to always set some limits on how much may be spent on hosting in the cloud control panel, and to set up notifications to be sent to you when these thresholds are about to be triggered.

- Multi-tenancy Risks: In a public cloud environment, the "noisy neighbour" problem could potentially affect your service's performance. Additionally, even though cloud providers take steps to isolate tenant environments, vulnerabilities could potentially expose sensitive data (see the previous section for a detailed explanation).

Additional considerations:

- In today's environments, Kubernetes and Terraform are sometimes used as a substitute for cloud APIs when it comes to production services' management. When running a DNS resolver in a Kubernetes cluster on top of a public cloud environment, all the pros and cons of the public cloud apply; basically, Kubernetes becomes your public cloud provider. If you have significant prior experience running services in Kubernetes in production, you may successfully replicate your experience with the DNS resolver software. Otherwise, we would advise against Kubernetes in this case.

- The only reason we may find to run a DNS resolver in a Kubernetes cluster on top of self-hosted dedicated servers is when you have significant hands-on experience with Kubernetes and it is natural for you to manage applications this way. Otherwise, running DNS resolver daemons in containers brings little, if any, benefit. Autoscaling features are not available to you in this case, and neither horizontal nor vertical pod autoscaling is of any use, because DNS resolver software typically scales in-host by itself just fine.

- When designing a cluster of resolvers for autoscaling, keep in mind that newly spawned resolver machines would need to populate resolver cache first before they are fully useful. Your DNS resolver software may provide cache replication mechanisms.Otherwise, it is safe to overprovision clusters somewhat under heavy load, and discarding excessive instances once all the caches are populated and the average load of a compute instance decreases. In addition, it may be worthwhile to consider sharing cache data between instances.

- It is always advised to prefer environments your infrastructure management team is familiar with.

Software Considerations

Open Source

Choose any well-maintained DNS software you are comfortable using.

Regardless of which software you choose, ensure you have somewhere to go for support. In the case of open source software, consider providing financial support to ensure continued development. Some open source maintainers take donations, while others offer support contracts.

There are both open source and proprietary implementations of DNS resolver software. Mixing these is also possible, for example, by using proprietary extensions with open source software or deploying open source software modified in-house.

General observations:

- Software licensing is orthogonal to software security. Neither is proprietary software less secure on principle nor are contributions by "unknown" developers more of a risk in open source.

Benefits of open source:

- Open source allows for inspection, independent auditing, and troubleshooting.
- Open source can avoid vendor lock-in.
- Open source can aid internet standards development.
- Widely-deployed open source implementations allow proponents of standards drafts to contribute proof of concept implementations without permission or cooperation of vendors.

Drawbacks of open source:

- Both open source and proprietary software require skilled maintenance, which has costs. Proprietary licensed software orappliances typically come with license fees to cover these. In contrast, open source licenses decouple usage by operators from monetary compensation to developers. It is up to operators to consider the financial sustainability of continued maintenance of the open source DNS software they depend upon.

Please also consider deploying different software implementations to ensure diversity, as discussed in the diversity section below.

Networking Considerations

IPv4 and IPv6

If available, both IPv4 and IPv6 must be deployed.

Large parts of the authoritative DNS are only accessible via IPv4, so the resolver must be able to originate IPv4 queries. Authoritative DNS that is only accessible via IPv6 is very rare.

Depending on the connectivity of clients, a resolver may be IPv4-only, IPv6-only, or support IPv4 and IPv6.

Addressing

Using multiple IP addresses for the service address should be considered.

Using two or more IPv4 addresses and two or more IPv6 addresses from different RIRs will allow resilience in failure at an RIR, either governance, security, or technical. Note that support for multiple addresses for recursive resolvers varies and some clients perform poorly if any address does not respond normally.

There is no need to pick an IPv4 address with all octets the same, like 2.2.2.2 or 11.11.11.11.

Publishing a list of back-end addresses used for resolving should be considered.

Publishing a list of back-end addresses used for resolving can be useful for other network and DNS operators (for example, geo-IP location, making sure data is getting to correct places, and so on).

High Availability

This can be considered in terms of local and global scope.

Local Scope

Inside a single location/region, such as an office, campus, or small ISP network, the main availability concern is that a resolver is always reachable. Client systems can be configured with multiple resolver addresses, but the failover behaviour of stub resolvers to a second address can be painful. Ideally the primary address is highly available and such fallback rarely required. How much effort is put into ensuring this is true should probably scale in line with the number of users, or sensitivity of the clients using that resolver to delayed resolution.

There are several ways to promote high availability of an individual resolver address, such as dedicated load balancing equipment, or network techniques like VRRP, or IP anycast. These generally have in common a pool of recursive servers and the means to direct queries to them when a health check has determined them to be capable of answering those queries.

Dedicated free or commercially produced, hardware or software load balancing solutions are available. These typically own the resolver IP address and forward queries to the currently available instances of a pool of recursive servers.

VRRP enables a technique to make the resolver IP address available on multiple servers, often used to provide automatic failover between two. A pool of recursive servers using this technique must reside in the same broadcast domain.

IP anycast in the local scope typically involves a pool of recursive servers advertising a route to a shared resolver IP address into a routing protocol. This can be configured in failover or load-sharing configurations. A load sharing configuration typically requires network equipment able to balance traffic to a destination over equal cost paths (ECMP). A pool of recursive servers using this technique can be distributed in different parts of the network.

Global Scope

The same concerns as for local service availability are present in the global scope, with the added issue that DNS resolution over long distances may be slow. Practically speaking, only multiple resolver addresses, or IP anycast are useful strategies here. The motivations for finding better failover solutions than multiple resolver addresses have been covered above.

IP anycast in the global scope means routing the same IP prefix to more than one location. This can provide effective solutions for failover and, when optimally configured for routing client queries to the topologically least distant recursive server location. IP anycast in the global scope requires the use of globally routable prefixes. If a separate prefix is to be used for anycasting, usually this means a /24 in IPv4 and a /48 in IPv6, as those are the smallest sizes that will be widely propagated in BGP. A common practice is to use a covering prefix (/23 in IPv4 or /47 in IPv6) for fallback, and a more-specific prefix (/24 or /48) for the traffic. The more-specific prefix can then be withdrawn to send traffic to a backup site; this will happen automatically if the site is disconnected from routing.

RFC 7094 discusses anycast architecture in detail, including references to various other RFC which discuss anycast in general and to DNS in particular.

RFC 4786 discusses operation of anycast services.

Generally

Operators of a globally scoped recursive service are encouraged to also adopt the local scope recommendations in each of the locations where the service is provisioned.

Though the above deals with the shortcomings of reliance on stub resolver failover between a list of addresses those recommendations should not be seen as an exclusive alternative. Multiple resolver addresses, where each is provisioned using differing failover strategies, can provide a resolver of last resort and further improved resilience.

Ingress Filtering

Ingress filtering to follow BCP 38 should be deployed.

DNS normally uses UDP traffic, which makes it a common vector of both reflection and amplification attacks. To minimise the amount of spoofed traffic that a resolver responds to, the network should be configured as recommended in BCP 38.

RPKI Sign Advertised Routes

Route Advertisements should be signed using RPKI.

Using RPKI to sign any route advertisements - either toward authoritative servers or toward DNS clients - is straightforward to do and will reduce the impact of BGP (Border Gateway Protocol) misconfigurations and some BGP hijacking attempts.

RPKI validation is also possible, although the effort is greater. It is possible that the hosting provider or the transit provider for your service validates BGP; asking and making this part of your selection criteria is reasonable.

(D)DoS Measures

Denial-of-Service (DoS) attacks, both distributed (DDoS) and not are a threat to any Internet service. Network operators for a service providing any DNS service must be prepared for large amounts of attack traffic.

In addition to attacks on the service itself, a resolver may be used both as an attack reflector and as an attack amplifier.

Active monitoring of network and service usage, careful logging, and a security team that is able to respond to problem reports is necessary. Mitigation techniques will include filtering or rate-limiting traffic, both on the authoritative and client side of the resolver.

Capacity Planning

Server Capacity

If using a model that is easy to scale (cloud based, or Kubernetes based, or similar), then getting server capacity correct is largely a question of budgeting. If using a less-flexible model (bare metal for example), then under-estimating will mean problems delivering service.
Hardware performance varies widely, as does operating system and resolver performance. Some lab testing will be necessary to estimate the number of systems needed.

Network Capacity

Since DNS is mostly UDP-based, it is often easy to generate large amounts of spoofed traffic to and from DNS servers. DNS traffic is small compared to application traffic (videos and other content), but still significant. Authoritative server operators often build their networks and servers to handle 10 times their normal load. Recursive server operators may need to do the same. When the service only accepts traffic from IP addresses that cannot be spoofed (for example users within a network that operated by the same company) this can be reduced, for example to three times normal load. To estimate expected load, the best approach is to examine historical usage for the actual expected users of the system.

Resilience

System Diversity

Operators should consider whether to use different software implementations to provide service. This allows continued operation if a critical vulnerability is found in one implementation, by shifting traffic to other implementations.

Placing resolvers and control systems in different physical locations will allow continued operation in the event of a disaster or other problem that impacts a single location. In addition, ensuring diverse connectivity to other networks will prevent single points of failure on the network side. Ensuring network diversity may take some care, as it is not always obvious what fate is shared between any given path; this may be physical, virtual, or organisational, and my sometimes be hidden.

Security

In addition to the DNS-specific security considerations, normal security best practices for any Internet service should be followed,including updating software updated regularly, patching software as soon as possible for any known security vulnerabilities, following CERT announcements and so on.

Certification

It may be useful or required for an organisation to obtain specific certifications, such as ISO or SOC 2 Type 2. These can be government-defined or industry-defined. For end users there is typically not much direct value, but business customers will often look for services that are operated by organisations meeting such standards. Audits or other reports about this may be published, see for example certifications and compliance resources.

DNS Configuration Knobs

The DNS is an old protocol that has a lot of settings that can be tweaked. This section reviews these and provides recommendations on which should be used for a DNS resolver.

DNSSEC Validation

DNSSEC validation should be enabled.

For: All DNS resolver operators.

DNSSEC validation is the best way to ensure that the answers from the owner of the domain name being queried are returned.

The root KSK must be updated when it changes. While RFC 5011 defines an automated way to do this, a DNS resolver operator will probably either manage this trust anchor directly or have it updated via OS updates.

RFC 9364 provides a lot of useful information, and links to further documents about DNSSEC. However, operators usually do not need to know the details, and can simply ensure that DNSSEC validation is enabled in their software.

DNS resolver software that does not support DNSSEC validation should be avoided.

DNS Transport Protocols

UDP and TCP must be supported.

For: All DNS resolver operators.

UDP is what most clients use, and TCP is necessary for DNS answers that are too large for a single UDP packet.

RFC 7766 explains why TCP is necessary in more detail.

Packet Fragmentation Avoidance

Servers should be configured to avoid fragmentation.

For: ALL DNS resolver operators.

Packet fragmentation can cause issues with DNS over UDP, especially over IPv6. These issues can be minimised by choosing implementations that set IP options to avoid this, and by taking care with EDNS0 message sizes.

Recommendations are available in IP Fragmentation Avoidance in DNS over UDP.

Encrypted DNS

At least one of DNS-over-TLS (DoT), DNS-over-HTTPS (DoH), and DNS-over-QUIC (DoQ) should be supported.

For: All DNS resolver operators.

DoT, DoH, and DoQ are different technologies that all provide an encrypted channel between the resolver and the authoritative server. DoT is the oldest, and provides encrypted DNS using TLS. DoH uses HTTP over TLS as a way to transmit queries and answers, and is widely supported by web browsers. DoQ is the newest, and provides advanced features such as separate streams for each query, avoiding the "head of line" blocking problem common with all protocols layered on top of TCP (such as DoT and DoH).

- DoT
  - RFC 7858
- DoH
  - RFC 8484
- DoQ
  - RFC 9250

Discovery of DNS Designated Resolvers.

There are new mechanisms that allow DNS clients to use DNS records to discover encrypted DNS configurations. Resolvers should publish DNS records to assist clients finding encrypted resolvers.

- Discovery of Designated Resolvers
  - RFC 9462

QNAME Minimisation

QNAME minimisation should be enabled.

For: All DNS resolver operators.

Using QNAME minimisation, a resolver does not send the full name that it is trying to resolver to authoritative servers higher in the DNS hierarchy. So, for example, when querying "atlas.ripe.net" the servers for ".net" would only be asked for "ripe.net". This improves privacy for the end user querying the name.

RFC 7816 covers QNAME minimisation.

Aggressive NSEC Caching

Aggressive NSEC caching may be enabled.

For: Public DNS resolver operators.

"Aggressive NSEC caching", meaning negative caching based on NSEC and NSEC3 values, can reduce traffic greatly. It is important to protect against random subdomain attacks.

This style of caching takes advantage of the way that NSEC and NSEC3 records cover a range of names in a zone. A DNS resolver can know that a query falls within such a range without sending any further queries, by remembering the NSEC or NSEC3 records that is has seen as answers to earlier queries.

Aggressive NSEC caching is almost always a good idea. However enabling this is less important for DNS resolver operators who have a close relationship with users, since they can stop attacks by blocking users or otherwise directly dealing with the source of abusive queries.

RFC 8198 describes negative caching in detail.

ANY Queries

ANY queries responses should be limited.

For: All DNS resolver operators.

Public or large-scale resolvers should be exceptionally careful with queries of type ANY, which return all records at a given name. If a resolver replies with all of the records cached for a given type, the response can be much larger than for a single record type. Strict limits should be enforced on volumes of such queries to prevent amplification abuse, or truncation should be applied to prevent spoofed redirections.

RFC 8482 describes several approaches to limiting ANY responses.

Local Root

Local root should be used.

For: Public DNS resolver operators.

Running a local root has several benefits, but it is an additional component to maintain. For public DNS resolver operators this is definitely worth the cost, but other resolver operators may choose to simply send all queries to the well-distributed root name servers.

RFC 8806 describes local root, including several example configurations.

In the future it will be possible to use ZONEMD to validate the copy of the root zone obtained before using it. This is currently available for the root zone.

RFC 8976 describes ZONEMD.

DNS Cookies

Interoperable DNS Cookies may be supported.

For: Public DNS resolver operators.

DNS cookies provide some improved security over plain UDP, and are designed to be more lightweight than TCP. If more than one server is responding for a given IP address, then the Server Secret must be shared by all servers, and the answer must be constructed in a consistent manner by all server implementations.

Since client-side support for DNS cookies is not very widespread, and since managing server-side secrets involves some work, the costs may outweigh the benefits for some non-public DNS resolver operators.

RFC 7873 describes DNS cookies, and RFC 9018 standardises shared DNS cookies.

TTL Recommendations

TTL limits may be adjusted.

For: All DNS resolver operators.

Software typically defaults to a maximum stored TTL of one or two days. A lower TTL will mean removing rarely-used records that have long TTL, and should not have much operational impact from a CPU or network point of view.

It is possible to set a minimum TTL in many implementations. This is a violation of the DNS protocol, although may be useful to reduce load from records with very low TTL (less than five seconds).

Note that software may set different maximum and minimum TTL independent of the results that the DNS resolver returns. That may have a significant impact on queries as well, but DNS resolver operators cannot influence that.

TTL-based Record Pre-Fetch

TTL record pre-fetch should be enabled when available.

For: All DNS resolver operators.

Some DNS resolvers have the ability to look up a record before it has expired from cache, in order to refresh the value and extend the TTL. This way there is never a time when the records are missing from the cache. This is not currently standardised, but a form of this was proposed in the IETF as DNS Hammer. We recommend turning this feature on if available.

EDNS Client Subnet (ECS)

ECS may be enabled.

For: All DNS resolver operators.

EDNS Client Subnet (ECS) allows the resolver to include information about the IP address of the client querying it when sending messages to authoritative servers. This may allow authoritative servers to provide different answers which are more appropriate for the client. However, ECS will increase the amount of cache space required by resolvers, may reduce DNS performance, and may have privacy implications.

A resolver operator that has clients that are limited to a specific region may see no benefit. A resolver operator that has a widely distributed anycast network may not have much benefit from ECS, since the locations that initiate the query will be close to the client. But a resolver operator that answers client queries only from a few locations, and expects clients to come from a wide area, may provide better service for end-users by supporting ECS. EDNS client subnet is described in RFC 7871, an informational RFC.

Extended DNS Errors

Extended DNS errors should be enabled.

For: All DNS resolver operators.

DNS traditionally provides very broad error reporting, SERVFAIL being the most common. This makes diagnosing and fixing problems difficult. Extended DNS errors provide extra information about failures, for example expired DNSSEC signatures. They also allow resolver operators to report administrative reasons for DNS failures, such as blocks due to legal requirements.

RFC 8914 defines extended DNS errors.

Negative Trust Anchors

Negative Trust Anchors may be deployed.

For: All DNS resolver operators.

Negative Trust Anchors (NTA) allow a resolver operator to handle a case where an authoritative server has a DNSSEC problem and becomes inaccessible. They basically disable DNSSEC checking for a domain. When this is warranted is difficult to know with certainty, and will usually requires some manual checking.

Since DNSSEC validation is now widespread, DNSSEC failures on the authoritative side will impact many resolvers.

Because of these reasons this document does not recommend NTA, but also does not recommend that a deployment avoid NTA if it makes sensefor that environment.

NTA are documented in RFC 7646.

DNS Error Reporting

DNS error reporting may be enabled.

For: All DNS resolver operators.

DNS error reporting is a way for resolver operators to let authoritative operators know about problems in authoritative servers or zones. It provides little direct value for the resolver operators, but over time should improve the overall quality of the DNS ecosystem. It is neither widely deployed nor standardised, but hopefully will be both soon. Resolver operators are encouraged to enable DNS error reporting when it is available.

DNS error reporting is proposed in DNS Error Reporting.

Trust Anchor Reporting

Trust anchor reporting should be enabled.

For: All DNS resolver operators.

Trust anchor reporting is a way for resolver operators to convey their DNSSEC trust anchor configuration to the operator of the zone that it is for. For most resolvers this is only the root zone. This information is intended to be used during a root KSK rollover to ensure that it is safe to proceed. In the future ICANN is planning an algorithm roll for the root KSK, and this information could be helpful. Resolver operators are encouraged to enable trust anchor reporting.

RFC 8145 covers trust anchor reporting, in both possibilities available.

Name Server Identification

Servers should be configured to identify themselves.

For: All DNS resolver operators.

Large resolver operations, especially publicly available resolvers, should support an in-band method of discovery that is obvious to permit users to discover what node has answered their query. This improves troubleshooting significantly, and may be useful for research and testing purposes. NSID (Name Server Identifier) is ideal for this, though also ,”CH TXT id.server” support is also reasonable. Geographic hints should be provided in this data, though specific host data is optional for arrays of servers in clusters. IATA codes have traditionally been used for naming points-of-presence, though this is at the discretion of the operator.

RFC 5001 describes NSID.

RFC 4892 describes name server identification in general, and documents the pre-NSID approaches.

Privacy, Filtering, Transparency

Privacy & Anonymity

DNS resolver operators are advised to apply RFC 8932.

"Recommendations for DNS Privacy Service Operators" as follows:

1. Its operational and policy guidance related to DNS encrypted transports and data handling, by applying all "Threat mitigations"(thereby by meeting its level of "minimally compliant") and additionally by applying the "Optimizations" on EDNS Client Subnet listed in section 5.3.1.

2. Its framework on a recursive operator privacy statement, by publishing a privacy statement on their website covers all topics in Section 6. (See for example Quad9's privacy page).

Logging Considerations

In addition to the logging recommendations from RFC 8932, operators should consider the following:

1. Third party access to personal data: Resolver operators may receive third party requests for information they have logged that relates to users, including IP addresses, queries and meta data. Resolvers should only comply with such requests when balancing legitimate third party interest with the user's fundamental rights, including rights to privacy. Usage information can be personal data, Personally Identifable Infomation (PII) or similarly regulated under the privacy laws applicable to the users, operator or third party, revealing a person's health, lifestyle or other personal preferences (profiling). For example, logging information that documents a user resolving a website for alcoholics anonymous may relate to the health of a person behind an IP address.

2. Data security: DNS resolver operators should take appropriate technical and organisational measures to protect logging information that relates to users.

Advertisement Policy

If there is any advertising from the service, the policy should be published as well as how it can potentially affect the users' privacy.

Filtering and Blocking

Block Lists

Resolvers can be directed to block or modify answers in various ways. Blocklists may be provided by governments, communities, or other parties (for example security firms).

Response Policy Zone (RPZ) allows a way to both document specific modifications that resolvers will make to DNS answers, and send the rules to resolvers. This allows updates to occur very quickly. If RPZ or some other high-speed blocking technology is used, the parties supplying these sources must be highly trusted, as changes to blocklists will usually immediately impact user queries. RPZ is not standardised, but there is an IETF draft.

Legal Blocking

Legal requests and blocking and filtering laws: DNS resolver operators should not filter content and block access to web services. When the local law requires blocking, and the law applies to the resolver, the resolver should transparently disclose a list of blocked websites and services, when possible (disclosing such a list may not be allowed by law or regulation). Similarly, the resolver should disclose the source of such block lists, when possible.

If it is not possible to disclose the source of blocklists, operators should try to be as transparent as possible about how they receive those blocklists, based on what criteria, and how they mitigate errors and false positives. Disclosing which organisations operators interact with, how they liaise, and so on, can help users understand the impact on the service provided.

If possible, resolvers should provide information about blocked responses via the Extended DNS Error with the Blocked, Censored, Filtered, or Prohibited code - whichever applies best - along with a text why the response was blocked, censored, filtered, or prohibited.

RFC 8914 provides information about the meanings of the different codes.

Community governance of blocklists: Blocklists, if mandatory, have to be audited and assessed by third parties and there should be a right to appeal for those blocked. The Internet community can vet the blocklists from time to time to avoid blocking access to websites that are mistakenly blocked. During crisis - when mistakes can have drastic effects on accessing a critical service - preferably filtering and blocking should not be used.

Opt-in/Opt-out Mechanisms

End users may choose to use a DNS resolver that filters specific kinds of traffic. For example, they may wish to avoid potential malware websites. Or resolver operators may be required to default to filtering but allowed for to provide an unfiltered DNS resolver service.

Depending on the specific requirements, a resolver service may publish different IP addresses and what type of filtering applies to each address. It is also possible to perform client authentication and authorisation, using IP-based authentication, TSIG keys, or client-side TLS certificates.

Transparency

Public DNS resolver operators should publish transparency reports to build user trust in their adherence to policies and practices. This goes beyond our advise to apply RFC 8932, section 6.2.

A common frequency is once a year. The reports inform the public about disclosure of user information and removal of content required by law enforcement and other government agencies.

Transparency reports should (to the extent that the law allows) indicate which government agencies and law enforcement agencies request access on what basis.

It should also be clear from the transparency reports what kind of data has been requested and if content removal and content blocking have been requested. Categories of data include: Content Data, Basic Subscriber Data, Other Non-Content Data and Content Blocking.

Negative Trust Anchor Reporting

Negative Trust Anchors (NTA) are discussed in the previous section on DNS configurations. If NTAs are present in the resolver, they should be published with as much detail as possible about them. This includes reasons for insertion, dates of activation and expected removal dates, or a published review date or cycle for when NTAs should be actively examined for deletion if such fine-grained information cannot be shared.

NTAs are equivalent to a security fault, and may even be more significant than a block event as they remove expected trust behavior with limited signal of that trust downgrade ("limited" because few if any clients care about those response bits changing).

Human Rights Considerations

DNS resolvers can opt for declaring their understanding of their responsibilities regarding human rights from the Universal Declaration of Human Rights. As an example of a public DNS resolver operator, Quad9 mentions rights to freedoms without distinction made on the basis of country, no interference with privacy, the right to freedom of opinion and expression, the right to peaceful assembly, and the right to freely participate in the cultural life of the community.

See Quad9's Human Rights Considerations for the full statement.

It also invokes other human rights related solutions other than UDHR such as Articles 8 and 9 of Resolution 42/15 of the United Nations Human Rights Council on the right to privacy in the digital age of 26 September 2019 more directly define the responsibilities of the private sector toward the furtherance of human rights in modern terms.

They also follow the Guidelines for Human Rights Protocol and Architecture Consideration of the Human Rights Protocol Considerations Research Group at Internet Research Task Force.

The latest version of the IRTF Guidelines for HRPC may be considered for all network operators.

Provenance and Rationale

There is increasing concern that large open DNS resolvers will become centralised points of DNS operations on the Internet. In order to address this, the European Commission issued the DNS4EU proposal. However, such an initiative could lead to centralised guidance or regulation which might interfere with the decentralised way the Internet infrastructure works - including the DNS. See for reference the RIPE NCC Open House discussion on this topic.

Rather than attempting to respond to the European Commission proposal or organise specific DNS resolver deployments, the RIPE community has decided that it is best able to provide advice and guidance. The RIPE Community is well positioned to provide a set of Best Current Practices that operators of Open DNS resolvers will be encouraged to subscribe to.

Document History

24 January 2023
DNS Resolver Best Common Practice Task Force is formed.

8 March 2023
The task force begins work on recommendations.

26 November 2023
Draft sent to DNS working group.

19 February 2024
Revised draft sent to the task force with feedback from the DNS Working Group.

17 March 2024
Revised draft sent to the task force based on its feedback.

29 March 2024
Final version sent to the RIPE Chair and RIPE NCC.

1 May 2024
Published as ripe-823.

Acknowledgements

The author would like to thank all members of the DNS Resolver Best Common Practice Task Force for their contributions. We would also like to thank Ralf Weber, Petr Špaček, Dave Knight and Peter Thomassen for their substantial commentary and review.