Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 11, 2019

V. Bertola
Open-Xchange

March 10, 2019

Recommendations for DNS Privacy Client Applications
draft-bertola-bcp-doh-clients-00

Abstract

This document presents operational, policy and security considerations for the authors and publishers of client applications that choose to implement DNS resolution through any of the protocols that provide private, encrypted connections between the application itself and the DNS resolver. As these protocols, depending on implementation choices and deployment models, may impact the Internet significantly at the architectural, legal and policy levels, the document records the current consensus on how these protocols should be used by applications, especially user-facing applications meant for mass usage by non-technical consumers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 11, 2019.

Table of Contents

1. Introduction

As a reaction to growing concerns about widespread "pervasive monitoring" activities, the IETF declared these practices to be an attack [RFC7258] and started work to promote the encryption of the transport of information across the Internet.

The Domain Name System [RFC1034] is a fundamental element of the Internet, as almost any online activity starts with one or more DNS queries, which can be used to track the services and the content that the user is accessing, and even to redirect or disallow such access; DNS traffic deriving from the activity of human beings constitutes sensitive and valuable personal information. Section 2.4.1 of [I-D.bortzmeyer-dprive-rfc7626-bis] describes the risks created by the unencrypted transmission of such information.

To mitigate these risks, two new standards have been developed to encrypt the transport of DNS queries and replies between the user's machine and the recursive resolver: DNS-over-TLS [RFC7858] and DNS-over-HTTPS [RFC8484]. The adoption of these protocols is still limited, but early deployments, especially of the latter, have raised a number of issues that pertain to consolidation and centralization, other privacy risks, various cases of content control, network security and management, applicable jurisdiction and more.

These issues, if not addressed, could outweigh the benefits deriving from the encryption of DNS transport. Some of them can be addressed at the server side of the connection; they are the subject of [I-D.ietf-dprive-bcp-op]. However, some of these issues derive from the behaviour of client applications that implement the protocols and use them to query the DNS as necessary for their activities.

This document presents the best practices that address these issues at the client side of the connection and that, as far as possible, could allow an uncontroversial and positive deployment of encrypted DNS transport protocols.

2. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

3. Architectures for Name Resolution Services

Traditionally, the DNS name resolution service for ordinary Internet users has been generally provided by a recursive DNS resolver on the local network ("local resolver"), supplied by the same ISP that is providing the user with Internet access, often through automated configuration mechanisms such as DHCP [RFC2131].

More recently, public DNS resolvers ("remote resolvers"), provided on an Internet scale by a few big operators, have been gaining ground; users have been able to set them as the resolver for all their applications at once, by changing a configuration item in their devices.

Thus, the current architecture for mainstream DNS resolution services relies on the following assumptions:

1. All the applications on the same device use the same resolver, through a library provided by the operating system;

2. By default, the device will automatically discover and use the resolver provided by the local network;

3. The user is in charge of deciding which resolver will be used, either by silently accepting the default, or by setting a specific resolver in the device configuration.

Early deployments of DNS-over-TLS have generally followed the same model; in the first mobile operating system that has implemented support for the protocol, the system will try to discover whether the resolver configured in the operating system supports DNS-over-TLS connections, and if so, it will automatically upgrade the connection to the new protocol while continuing to use the same resolver; otherwise, it will keep using the unencrypted connection.

Early deployments of DNS-over-HTTPS have followed a different model. The first major Web browser releasing support for DNS-over-HTTPS has announced the intention to follow a service architecture based on different assumptions:

1. Each application will use its own resolver, bypassing the library provided by the operating system;

2. No automatic discovery of resolvers on the local network is performed;

3. The application is in charge of determining which resolver will be used, which could be different than the one configured in the operating system, and can direct this choice by selecting and providing one as the default for all users globally, or even by limiting the user's choice to a list of resolvers vetted by the application maker.

In the rest of the document, we will refer to this new architecture as "application-level name resolution", and to the traditional one as "network-level name resolution". More specifically, we will refer to point 3 of the above list as "application-level resolver selection".

While the document will also address the issues that derive simply from the switch to an encrypted connection, most of the issues that have been noticed during the early deployment efforts for DNS-over-HTTPS do not derive from the protocol in itself, but rather from the switch to application-level name resolution under the assumptions above, and most frequently from the adoption of application-level resolver selection.

Other possible architectures have also been identified: for example, applications could use different resolvers for each of the different servers that they have to connect to, or the servers could push DNS records to the client even before they are actually queried. As these models are still speculative, the issues that they would generate are not currently addressed by this document.

4. Issues and Recommendations

This section classifies the issues that are created by the deployment of encrypted DNS protocols and/or by the change of resolution architectures on the client side, and presents recommendations to address them.

4.1. Trust Model and User Choice

With network-level name resolution, users are in charge of the choice of the resolver used by all their applications. Advanced and educated users make full use of that opportunity, and choose to send their DNS queries to an operator they trust, either by configuring a specific resolver in the operating system or on the router that manages their home network, or as part of a broader traffic encryption and redirection service (e.g. VPNs). Other users will just rely on the local network's server; for home users, this will be provided by their Internet access provider, which they also picked, thus giving them at least some degree of trust. A different, less trusted relationship exists when users that did not configure a specific server connect to a local network which is not their own, for example in Internet cafes or hotels; in that case, they may be led to use a resolver which they do not trust.

Also, in the absence of encryption, the local network operator could still be able to track or even alter the DNS queries and replies that the user has directed to a non-local resolver. Unless this has been agreed with the user or is mandated by applicable legislation, this would be a breach of the user's trust and is a good reason to promote the adoption of encrypted DNS protocols.

With application-level name resolution, and especially with application-level resolver selection, users lose at least part of their control. Some applications may actually not give the user any choice, and just use their own resolver all the times; some others may provide configuration options, but still direct the queries to their own resolver as a default, requiring specific action for users to keep their DNS traffic going to the resolver that they already configured in the operating system.

Such a change of default behaviour would break the expectations of many users; unless appropriate communication is given and consent is acquired, both the users that explicitly configured a resolver in the operating system, and the users that want their device to use the local network's resolver, will be surprised by the application sending DNS queries to another server instead. On the other hand, for less technical users that trust the application maker to make a better choice of resolvers on their behalf, the new behaviour could be in line with their expectations.

However, also given the high sensitivity of the personal information embedded in DNS queries, it is important to ensure that it is ultimately the user, rather than any third party, that determines where their DNS queries should go, and explicitly consents to any choice of resolver that cannot be expected, or to delegating the choice to another party.

This leads to the following recommendations:

1. Users MUST have the final word on which resolver is used by each application.

2. Applications MUST NOT use a different resolver operator than the one that would be used by the operating system, unless the user has actively told the application to use a different resolver operator and has given explicit and informed consent to the change.

3. Applications MUST provide users with a configuration mechanism that allows them to tell the application to use any resolver the user wants.

4. Applications SHOULD provide users with adequate information on the location, ownership, data management policies and operational practices of each resolver, at least for the resolvers that they provide as hard-coded configuration options.

5. Applications SHOULD support, if available, easy ways for the user to direct all applications on the device to use a specific resolver and a specific protocol, without the need to configure them one by one.

For recommendation #2, it may happen that the resolver configured in the operating system does not support encrypted DNS protocols. In this case, the application SHOULD first of all try to determine whether the operator of the non-encrypted resolver that would be used by the operating system also provides any encrypted DNS resolver, through standardized discovery mechanisms such as

[I-D.ietf-doh-resolver-associated-doh]; in that case, it SHOULD use that operator's encrypted resolver instead of the unencrypted one.

For recommendation #4, the guidelines in [I-D.ietf-dprive-bcp-op] SHOULD be followed; applications could explore ways to make them more understandable to average Internet users, for example through the use of standardized visual aids. At the same time, due to the complexity and changing nature of this information, applications could simply provide links to where the operator makes it available; a standardized and automated way for resolvers to make this information available to applications would be desirable.

For recommendation #5, in most operating systems and devices it is yet to be discussed how to achieve the objective of letting the user set the preferred resolver and protocol in all the applications at once; the principle, however, deserves to be stated as a recommended direction for future development.

4.2. Consolidation

Currently, with network-level name resolution, DNS resolution activities are globally spread across a huge number of resolvers, possibly in the range of the hundreds of thousands or even millions. Users that keep the default of using the local network's resolver see their personal DNS queries spread across multiple servers as they move; users that configure a specific public resolver have their queries concentrated on a single resolver system, but they can pick one that they trust.

However, in many cases, the global market for user applications is much more consolidated than the global market for Internet access and for DNS resolution services. More specifically, estimates for the global browser market currently show one maker having around 60% of the market, and the first four together having over 90% of it. While these market shares may change in the future, the presence of one or a few dominant browsers has almost always been the case since the advent of mass usage of the World Wide Web.

As application-level name resolution turns the application maker into a potential gatekeeper in the choice of the resolver, there is a concern that each application maker could lead its users to the adoption of one specific resolver operator, or limit their choice to a few options that the maker has picked for its own reasons.

While it is understandable that an application maker may want to help its users make a good choice of resolver by using its technical expertise to evaluate and select a narrow set of recommended resolvers, this practice would open opportunities for misuse, as

applications could recommend resolvers not because of a fair evaluation, but because of an existing business partnership in the mutual economic interest, or even because the resolver is run directly by the application maker.

In the end, this could lead to a dramatic consolidation of global name resolution operators, with a few server systems managing the broad majority of global resolutions. This, in turn, would have several negative consequences, which will be discussed in the following sections of the document.

However, consolidation is an architectural problem in itself; the Internet was designed to be as decentralized as possible, to increase its resilience and ultimately the freedom of its users. The concern over the centralizing effect of encrypted DNS protocols has been recorded for example in [I-D.arkko-iab-internet-consolidation], section 2.5, and has to be addressed by preventing the use of a strong position in the market for a specific type of client application, especially a ubiquitous one such as browsers, to centralize DNS resolutions and establish a strong position in DNS name resolution services.

This would already be addressed by the recommendations in section 4.1, but it also leads to the following additional recommendations:

1. Applications MUST NOT prioritize any specific resolver over others unless told so by the user, or design their user interaction to lead users towards choosing a specific resolver or one of a few specific resolvers.

2. If possible, and if desired by the user, applications SHOULD provide ways to spread their DNS resolution traffic across the highest possible number of recursive resolvers. [NOTE: this recommendation is actually a placeholder, as there are drawbacks to this idea and other, better methods could be devised - this is TBD]

3. If applications would like to ensure that their users can pick a resolver that has been vetted under a set of criteria, the definition, verification and enforcement of these criteria SHOULD be deferred to an independent multi-stakeholder organization, making these criteria public and objective and giving any resolver operator from any part of the Internet a fair chance to be admitted to the list of vetted services; in any case, users MUST still be free to adopt servers outside of the vetted list if they so desire.

4.3. Namespace Fragmentation

If each application could pick a different resolver, there is a chance that different applications would receive different replies to the same DNS queries, leading to confusing user experiences, potential attack surfaces and network management and debugging problems, and in the end to the fragmentation of the Internet into non fully interoperable chunks.

The only way to prevent any occurrence of this case would be to require all applications on the same device to use the same resolver, as in the current network-level resolution model; however, such requirement would be considered too restrictive. Recommendations #2 and #3 in section 4.1 are however meant to make this case a special exception, rather than the norm, and thus mitigate this problem.

However, when allowing different applications to use different resolvers, there is a situation that would significantly endanger the stability of the Internet. Currently, as the application and the resolver are generally provided by two independent entities, and as the application cannot know in advance which resolver will be used, applications cannot rely on predetermined non-standard behaviour by a specific resolver. With application-level resolver selection, applications that enjoy a significant market share could direct users to resolvers that employ alternate DNS namespaces that they control, or start to create and remove top level domains outside of the collectively agreed policy frameworks.

The global uniqueness of the DNS namespace and its collective policy-making procedures should be preserved, and this leads to the following additional recommendation:

1. Applications SHOULD NOT adopt alternate DNS namespaces or promote the use of resolvers that do not rely on the global DNS root server system.

2. Applications SHOULD NOT deviate from the DNS namespace management policies, technical standards and operational practices that are defined in the relevant Internet governance venues.

Regarding recommendation #1, there may be specific use cases in which a user may want to use an alternate namespace and root server set, and applications, if they want, should be free to support them; however, these cases must be exceptions and not for mainstream usage.

4.4. Privacy

Protecting the user's privacy is the primary aim of transitioning the DNS to encrypted communications. However, risks to privacy deriving from DNS communications are not limited to the eavesdropping of unencrypted DNS communications; [I-D.bortzmeyer-dprive-rfc7626-bis] lists, in section 2.4.2, several privacy risks that still exist even when DNS communications are encrypted; and, in section 2.5.1, it describes the risks that derive from the behaviour of the resolver in use, independently from whether the connection is encrypted or not.

To address the former category of privacy risks, some have suggested that, through the use of DNS-over-HTTPS, the user's requests could be hidden within unrelated HTTPS traffic, locating DNS-over-HTTPS servers at the same IP address and port of widely used web servers. However, this method creates security and legal issues that are documented in the following sections of this document, and so it is not recommended unless the privacy gain is tantamount in respect to any other consideration, for example because of the extreme sensitivity of the online activity or because of a hostile environment that could lead to significant personal risks.

In all other cases, users that feel the need for further obfuscation of their encrypted DNS communication should rather be directed to spreading their communications across a great number of encrypted resolvers, as per recommendation #2 of section 4.2, making it harder to track the entirety of their DNS exchanges.

To address the latter category of privacy risks, the first and foremost mitigation measure is to allow each user to pick a resolver that is managed by an organization that they trust, under appropriate regulatory conditions, privacy policies and operational practices. While "privacy-friendly" applications might (and, indeed, should) help users in making this choice, there is not a unique, globally applicable agreement on what constitutes a "privacy-friendly" resolver, and it is hard to ensure that all application makers will always make disinterested choices in the pure interest of the user; so it must be the user, in the end, to decide who to entrust with their personal information. The recommendations in sections 4.1 and 4.2 thus also contribute to mitigate this type of risks.

In addition, specific technical recommendations can be made to prevent applications from adopting practices that would make it easier for the resolver operator to track and fingerprint the user, mirroring the ones that have been developed in [I-D.ietf-dprive-bcp-op] section 5:

1. Applications MUST authenticate the resolver they connect to, if they have securely discovered the information required to do so.

2. In the case of DNS-over-HTTPS, applications SHOULD NOT attach or receive HTTP cookies on the connections used for the DNS message exchange, unless there is a specific use case for cookies that has been explicitly requested by the user.

3. [to be continued]

For recommendation #1, and for DNS-over-TLS, a discussion of resolver authentication methods and possibilities can be found in [RFC8310].

For recommendation #2, additional considerations on privacy when using HTTPS as a transport mechanism for other protocols can be found in section 6.1 of [I-D.ietf-httpbis-bcp56bis].

4.5. Content Access Control

DNS name resolution services are commonly chosen as a platform to control user access to content; while there are other mechanisms in use, checking and blocking destinations at the DNS level is an effective and relatively inexpensive one. As a consequence, the choice of the resolver also determines whether the user will be able to access certain destinations on not, depending on the policies applied by the individual resolver.

Sometimes, on unencrypted DNS connections, content control policies will also be applied to DNS connections directed to other resolvers having a different policy than the local network's one, though this is made impossible by the switch to encrypted DNS connections.

The motivations, the procedures, and the types of content subject to blocking are very variable. Some of the filtering activities are meant to increase the security and health of the network; they can be aimed at non-human clients, such as bots, trying to prevent them from connecting to their command and control servers; or they can try to prevent unaware users from connecting to phishing and malware websites.

Other filtering activities are meant to make some content inaccessible because it violates the law in the user's and/or the resolver's jurisdiction, or because of court rulings that mandate the block. In authoritarian countries, content may be blocked to prevent criticism of the ruling entity, while in democratic countries it can be blocked to defend the safety and rights of some parties and prevent violence and attacks on democracy. Destinations may also be made inaccessible, independently from their content, for lack of

compliance with any applicable regulation, such as requirements for licenses or the payment of taxes in the user's country.

The opinions on the appropriateness of these practices are highly influenced by local culture, history and socio-political environments, as well as by each individual's set of values, and it is thus impossible to make blanket recommendations on whether and when they should be forbidden, allowed, or actively supported; the only possible recommendation is to follow the applicable regulations.

In some cases, however, users actually demand DNS-based content control services to shape their own Internet access: for example, families that want to make sure that their children using the Internet from home cannot access inappropriate websites; businesses that want to restrict the way their employees can use the Internet in the office during working hours.

This leads to the following recommendations:

1. Applications MUST NOT prevent users from using any DNS-based content control service that they freely choose to adopt.

2. Applications SHOULD comply with any legislation and legal ruling on content control that applies to them in the jurisdiction where they are being used.

Further considerations on the relationship between applications and applicable legislation will be made in section 4.7.

4.6. Security and Network Management

Network management and network security practices often rely on checks and policies implemented at the DNS level, through the monitoring and mangling of the DNS queries that the users of the local network send to the local resolver. For example, these practices include checking for any unusual patterns in DNS traffic to detect devices that have been infected with malware; blocking queries for known botnet command-and-control servers to make it impossible for bots to communicate with them and take orders; implementing a content control list of web destinations that the network administrator considers dangerous; associating certain names to different IP addresses, some of which may be private, depending on the client and on its topological location; creating names in special use or non-standard top level domains and making them resolvable only from the inside of the local network.

Especially the use of local names and "split horizon" configurations is a widely used security practice in corporate network environments;

using a resolver located outside of the network would break this mechanism and at the same time leak information that would be very valuable to an attacker.

These practices rely on forcing all the users of the local network to use the local resolver, rather than a remote public resolver; this requirement is generally enforced through one or more of the following practices:

a. Making sure that all devices that are connected to the local network are configured to use the local resolver, disallowing the users from modifying this configuration entry;

b. Blocking access to remote resolvers at the edge of the local network, for example through firewalls and blocks by destination IP address and/or port;

c. Examining, and if necessary amending, all DNS queries and replies in transit towards remote resolvers.

The switch to encrypted DNS connections makes practice c) generally impossible; the switch to application-level resolver selection also makes a) impossible, unless the network administrator can prevent the installation of any application on all the devices connected to the network, which is not easily feasible in most cases. Finally, the implementation of "mixed mode" DNS-over-HTTPS, obfuscating the traffic within ordinary HTTPS connections to widely used web hosts, would make also practice b) impossible.

So, the deployment of encrypted DNS over dedicated connections disables c), but still leaves a) and b) as options to network administrators; however, the deployment of DNS-over-HTTPS in mixed mode disables all three practices and leaves the network administrators powerless; additionally, it even provides an undetectable data exfiltration vector for malicious applications that are trying to steal confidential information from the local network and forward it to the outside.

While disabling control by the local network administrator might actually be a positive intent in very specific cases, disabling all these security practices at once is dangerous and inappropriate in most cases. It is especially problematic on private networks that host sensitive or commercially valuable information, as they need to be able to provide some degree of connectivity to the Internet while scrutinizing and limiting its usage to guarantee security.

Noting that even [RFC7258], in section 2, stresses that "Making networks unmanageable to mitigate pervasive monitoring is not an

acceptable outcome", and calls for "an appropriate balance" between encryption and network management needs, the following recommendations, in addition to those in section 4.1., represent such balance:

1. Applications SHOULD NOT adopt the practice of hiding their encrypted DNS traffic in ways that prevent the local network administrator from isolating it, monitoring it (without breaking the encryption) and, if desired, blocking it; exceptions to this principle MUST be motivated by a clear and compelling use case which cannot be addressed otherwise, and their use MUST be limited to such use case. [Note: of course I am sure that many people will want to discuss this - the idea is to not stop this from happening when it is really useful, but also restrict its usage to when it is really useful, leaving network admins with the possibility to block external encrypted resolvers if they want, without starting a technical "arms race".]

2. [to be continued]

Further reasons supporting recommendation #1 can be found in sections 4.4 and 4.7.

4.7. Jurisdiction

The current prevailing practice of using local resolvers, located topologically near to the user, generally ensures that the resolver and the user fall under the same jurisdiction. This allows the country managing such jurisdiction to apply rules and policies to the user's Internet access by acting upon the resolver, recommending or mandating actions to the ISP that runs the resolver.

The use of remote resolvers, in most cases located in a different country and under a different jurisdiction than the user, has already provided a way for users to bypass their local laws. Many countries have tolerated this loss of sovereignty because it only affected a minority of users, possibly smarter technical users that could have anyway found other ways to bypass national rules applied at the resolver level, such as running their own recursive resolver. Other countries have instead engaged in enforcing their Internet access rules at other levels, such as by requiring firewalls at each connection between any national network and the broader Internet and filtering out destinations by IP address, or requiring the ISPs to apply similar blocks on each user's Internet connectivity.

In [RFC7754], the IETF has recommended the use of filtering at the endpoints, rather than inside the network, to address issues that require access control to Internet destinations; but filtering at the

edges is much harder to set up and to enforce for countries, and the same RFC agrees that "a hybrid approach that combines endpoint-based filtering with network filtering may prove least damaging". In this regard, making resolver-level law-mandated filtering policies impossible is anyway likely to push more countries to mandate heavier, more damaging filtering practices at the IP address level.

From the user's viewpoint, a change in jurisdiction may be beneficial or damaging, depending on which issues are most important for the user and on the applicable legislation in the user's and, if different, in the resolver's country. Users located in jurisdictions that restrict their rights may actively seek to use a resolver in a different jurisdiction to bypass the restrictions. Users that enjoy a high degree of rights in their country, such as extended protections for privacy and network neutrality, may reject the use of a resolver in another jurisdiction not to lose such protections.

It is out of scope for the IETF to decide whether the policies and laws of any country should be supported or opposed. By leaving as much control as possible to the user, as recommended in section 4.1, each user is allowed to decide whether to seek the use of a resolver in a different jurisdiction or stay within their own, bearing the responsibility for any legal consequence that might derive from that decision; and it is important that such a decision is not taken lightly and especially is not taken by the application without telling the user, as the user could otherwise unwillingly end up in legal troubles.

At the same time, while individual users and individual application makers may have different views and follow other practices, it is inappropriate for the IETF as a whole to promote the deployment of technologies in a way that is explicitly designed to undermine any country's sovereignty and jurisdiction. This is another reason to support recommendation #1 in section 4.6 and recommendation #2 in section 4.5. More generally, the following recommendations can be devised:

1. Applications SHOULD clearly inform the user whenever the resolver that it is going to be employed lies under a jurisdiction different than the one of the user.

2. Applications SHOULD NOT adopt a resolver located under a jurisdiction different than the user's one, unless the user has explicitly consented to such change of jurisdiction, and unless the user's jurisdiction allows the use of resolvers located in a different country.

4.8. Disaster Recovery

Remote resolvers rely on global Internet connectivity to be able to serve users from every part of the world. However, there may be cases in which long distance Internet connectivity is so poor to make the use of remote resolvers ineffective, or has been greatly reduced or even completely severed by the effects of natural disasters, upstream legal and contractual issues or any other special situation.

In these cases, it is important that applications can continue working if connectivity to a local resolver can be established, as the resolver, even with global connectivity issues, can still provide access to much needed local resources.

This leads to the following recommendation:

1. Applications, when using a remote resolver, SHOULD monitor the availability of sufficient connectivity to it, and SHOULD prompt the user to switch temporarily to a local resolver, if available, when such connectivity is not available.

4.9. User Support

Functioning DNS resolution is a requirement to make Internet connectivity work, and, when it does not, most users have a hard time discerning the specific technical factor that prevents it from working. As they generally acquire their Internet connectivity from a local ISP, they will thus contact the ISP's support desk whenever the connectivity does not work, regardless of the actual cause. However, if the application is using a remote resolver, the ISP's support desk will not be able to know whether such resolver is experiencing operational issues or applying policies that make the user's desired action fail.

It is thus important that application makers, remote resolver operators and ISPs cooperate to allow users to get support on their Internet access problems. For what regards applications, this leads to the following recommendations:

1. Applications SHOULD make it easy for users to determine whether any failure that they experience in the use of the application can be attributed to a failure in DNS resolution.

2. Applications SHOULD cooperate with remote resolver operators to direct users that experience problems due to resolver failures to the support service of the resolver operator.

3. Applications SHOULD provide easy ways for their users to retrieve the information on the resolver currently in use, so that they can pass on this information to whoever is helping them to restore their connectivity.

5. Security Considerations

The use of encrypted DNS protocols is beneficial to security as it prevents unauthorized third parties from altering the DNS queries and replies, but it also creates new security risks by disrupting a number of existing and commonly used practices for network security, and by providing, under certain conditions, a mechanism for data exfiltration from within a network through the submission of appropriately designed DNS queries.

More detailed security considerations can be found in section 4.6 of this document.

6. Privacy Considerations

The use of encrypted DNS protocols in itself is beneficial to privacy as it prevents eavesdropping of the connection. However, the issues created by the deployment model can lead to an overall loss of privacy, for example because the user is led to adopt a resolver operator that offers less privacy protection than the one they are currently using, or that is located under a jurisdiction that offers less privacy protection.

Issues specifically related to privacy, and recommendations to address them, are discussed in section 4.4 of this document.

7. Human Rights Considerations

[to be written]

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgements

[to be written]

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[I-D.arkko-iab-internet-consolidation]
Arkko, J., Trammell, B., Nottingham, M., Huitema, C., Thomson, M., and J. Tantsura, "Considerations on Internet Consolidation and the Internet Architecture", draft-arkko-iab-internet-consolidation-00 (work in progress), October 2018.

[I-D.bortzmeyer-dprive-rfc7626-bis]
Bortzmeyer, S. and S. Dickinson, "DNS Privacy Considerations", draft-bortzmeyer-dprive-rfc7626-bis-02 (work in progress), January 2019.

[I-D.ietf-doh-resolver-associated-doh]
Hoffman, P., "Associating a DoH Server with a Resolver", draft-ietf-doh-resolver-associated-doh-02 (work in progress), March 2019.

[I-D.ietf-dprive-bcp-op]
Dickinson, S., Overeinder, B., Rijswijk-Deij, R., and A. Mankin, "Recommendations for DNS Privacy Service Operators", draft-ietf-dprive-bcp-op-01 (work in progress), December 2018.

[I-D.ietf-httpbis-bcp56bis]
Nottingham, M., "Building Protocols with HTTP", draft-ietf-httpbis-bcp56bis-08 (work in progress), November 2018.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7754] Barnes, R., Cooper, A., Kolkman, O., Thaler, D., and E. Nordmark, "Technical Considerations for Internet Service Blocking and Filtering", RFC 7754, DOI 10.17487/RFC7754, March 2016, <https://www.rfc-editor.org/info/rfc7754>.

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

[RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <https://www.rfc-editor.org/info/rfc8310>.

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

Author's Address

Vittorio Bertola
Open-Xchange
Via Treviso 12
Torino 10144
Italy

Email: vittorio.bertola@open-xchange.com
URI: https://www.open-xchange.com