Abstract

This document defines a mechanism under which a client can request that an upstream recursive resolver perform DNS filtering on behalf of a client-requested policy. This may be done, for example, under a subscription model, where the client wishes not to get redirected to domains known to host malware or malicious content. This request is sent as an EDNS0 option with every DNS request, or potentially just to the first DNS request in a stream when using DNS over TLS, DNS over DTLS or DNS over DOH for example.

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 http://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 March 29, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
1. Introduction

[DOCUMENT STATUS NOTE: this specification is VERY INCOMPLETE and is at the stage of "discuss whether this is a good or bad idea in general", and not at the stage of "your processing steps are broken" or, worse "you mispelled misspelled". Keep reading for further background.]

1.1. Purpose of this document

Right now, the DNS ecosystem is being used in a multitude of ways that are intricately bound together based on its evolution over time. DNS resolvers today are acting as both a DNS resolution service, as originally intended, and as a control point by offering filtering (and rewriting) services on behalf of the client, the ISP, and policies imposed by enterprises/organizations and governments. One significant issue that has arisen under some proposed deployment architectures for [DOH] in which Applications Doing DNS (in the ADD pseudo-WG) may bypass traditional DNS resolvers within ISPs, alleviating those ISPs from offering DNS-based filtering and protection services.

This document is an attempt to see if those two roles can be safely severed, so users in an [DOH] world can select a resolver that best suits their resolution policies and separately select filtering policies that best suit their access requirements.
There are many other ways such a policy transmission feature could be implemented. DNS real-time blacklist (DNSRBL) like techniques could be used, ISPs could publish policy pointers under the DNS reverse tree, DoH clients could publish policies within HTTP headers (limiting its use to just DoH), ... I selected the one below as the "most out of the box" to promote thinking, not because I expect it to be the best option. Specifically, I have doubts that public large scale DoH providers will want to memorize large numbers of published policy lists (and hence, DNSRBL may actually be a better choice).

[There are other ways to implement what is described below, but I wanted to pick a more novel idea to promote wider thinking than "use an RBL like pointer" or "use a HTTPS header for just DOH because that's really what triggered the filtering discussions in the first place."]

1.2. Real Introduction

DNS today provides a distributed name resolution database that serves as the basis for many technologies, and is the starting point for nearly all communication that occurs on the Internet. Because of this, it frequently serves as a filtering mechanism by Network Providers who which to deploy DNS filtering or data modification technologies, for better or worse. As DNS is pushing further into being encrypted from client to recursive resolver by technologies such as [DNSTLS] and [DOH], clients are increasingly using encrypted communication to DNS resolvers that may have different or no filtering policies and mechanisms (protective or otherwise), than intended by the networking configuration distributed from their Internet Service Provider (ISP) or other access point. This document puts the selection of a selective DNS filtering service back in the hands of the user, since DNS centralization threatens to remove client ability to do so.

Specifically, this document defines a mechanism under which a client can request that its upstream recursive resolver perform DNS filtering on behalf of a client-requested policy. This is may be done, for example, under a subscription model, where the client wishes not to get redirected to domains known to host malware or malicious content. This request is sent as an EDNS0 [EDNS0] option with every DNS request, or potentially to just the first DNS request in a stream when using DNS over TLS, DNS over DTLS or DNS over DOH for example.

One could argue that clients could accomplish these goals by simply using a different resolver. However, this specifications allows decoupling of resolvers and filtering such that a default resolver configured in an operating system or application can still use a
system-level configured filtering mechanism acting independently of
resolution. A client can then select the best resolver to support
resolution services which can be independent from the best source of
malicious content or other filtering.

1.3. Requirements notation

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]

2. Extension Overview

2.1. Extension Packet Format

The EDNS0 option format for passing a Filter Request (FR) list to the
upstream DNS resolver using the following format:

0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
0: |                            OPTION-CODE                        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2: |                           OPTION-LENGTH                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
4: / FILTER-NAME ...                                            /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

The FILTER-NAME field is a normally encoded DNS NAME that is expected
to point to a publicly published DNS record from the filtering
service a client wishes to make use of. Details of this record are
documented in Section 3.

XXX: better text for normally encoded, and compression, etc

3. Filter Record Overview

Filtering services that wish to publish a DNS domain filter list may
publish a DNS record containing a URI from which a resolver may fetch
the current filter list. This published name MUST be of type TXT and
MUST begin with _dnsfilter but otherwise may be published at any
point in the DNS tree. Multiple records SHOULD be considered as
alternate fetch points and recursive resolvers supporting this
specification should fetch the first one available and then continue
with the steps outlined in Section 5.

Example:

_dnsfilter.example.com 86400 IN TXT "https://dnsfilter.example.org/"
The name ".dnsfilter.example.com" may then be referred to by clients in the FR extension packet.

4. ISP signalling

ISPs offering filtering service to their clients may signal suggested filtering lists to their clients via ... DHCP? (because starting one fight in this document wasn’t enough)

Maybe a DNS request hosted by the dhcp configured resolver?

(aka: ideas welcome here.)

5. Resolver processing

Recursive resolvers supporting this specification should perform the following steps upon receiving a request with a FILTER-NAME option.

1. If the recursive resolver does not support filtering, it should process the DNS request as normal and return an Extended DNS Error (EDE) error of "filteringNotSupported" along with the response. Stop.

2. If the FILTER-NAME is not currently in its cached set of DNS filters, it should attempt to resolve the name pointed to by the FILTER-NAME record. The list of returned URLs should attempt to be fetched, and the first successful download should be stored in a filter cache along with the FILTER-NAME and the cache length returned by the URL server [XXX: what’s the HTTP field; I forget]. If no URL can be successfully retrieved, then the resolver should continue to process the DNS request without applying a filter and return an EDE error of "filteringUnavailable".

3. The filter list returned by the URL must be of type text/plain, and must be a simple list of domain names that are to be blocked as requested. Names encode in the list MUST domain names, as encoded in printed zone-format names including any required internationalization support. The names MUST not include a leading or trailing dot. For simplicity, no wild-carding is supported and a prefix of "." is assumed. Partial end-matches MUST NOT but considered a match. For example, a domain "horrible.football.example.org" will match a filter entry of "football.example.com" but MUST NOT match an entry of "ball.example.org". See Section 5.1 for an example of what a filter list may look like. If the client’s request matches a filter in the requested filter list, a response is sent to the
client with an REFUSED RCODE and a EDE error code of "errfiltered (18)".

4. The resolver should continue normal resolution of the client’s request.

XXX: should we add a ‘stop’ or ‘continue’ on error bit to the EDNS0 option?

5. Example Filter List

An example filter list might include the following name list:

eample.com
malware.example.org
notforchildren.subdomain.example.org
eample
[need an internationalization example here]

Note that the last example matches everything under the ‘example’ TLD

6. Security Considerations

Modification, addition or removal of the EDNS0 option by device-in-the-middle attackers may cause unintended consequences for clients hoping to apply (or avoid) filtering. It is advisable that DNS requests that make use of this option send it over an authenticated transport such as [DNSTLS] or [DOH].

Similarly, providers of DNS FILTERING lists SHOULD published their FILTER-NAME within a DNSSEC signed zone. They SHOULD offer (and require) URLs that make use of protected transports, such as [HTTPS].

7. IANA Considerations

This document adds two new EDE codes to the EDE (xxx: ref) specification: filteringNotSupported and filteringUnavailable.

8. Informative References


Appendix A. Acknowledgments

peeps that help out will go here

Author’s Address

Wes Hardaker
USC/ISI

Email: ietf@hardakers.net