Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 26, 2020

P. Sood
Google
P. Hoffman
ICANN

April 24, 2020

Upgrading Communication from Stub Resolvers to DoT or DoH
draft-pp-add-stub-upgrade-00

Abstract

This document describes methods for a DNS stub resolver to upgrade its communications with a known recursive resolver to include encrytion using DoT or DoH. This protocol is designed for the scenario where the stub resolver already has the IP address of the recursive resolver.

Other protocols under develpment address scenarios where the stub resolver wants to discover recursive resolvers that use DoT or DoH. This document does not cover such discovery.

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 October 26, 2020.

Copyright Notice

Copyright (c) 2020 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 (https://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 include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

A stub resolver (hereafter called "a stub") using traditional DNS over port 53 may wish to use encrypted communication with the recursive resolver (hereafter called "a resolver"). In such a scenario, the stub needs to know how to probe the resolver to find out if it can use encrypted communication. This document describes a mechanism for a stub that knows the IP address of the resolver to do so.

The mechanism in this document assumes that a stub wants to attempt to upgrade its communication with the resolver to either DNS-over-TLS (DoT, [RFC7858]) or DNS-over-HTTPS (DoH, [RFC8484]). It can later be extended to other secure transports for stub-to-resolver communication transports.

1.1. Definitions

In the rest of this document, the term "resolver" without qualification means "recursive resolver" as defined in [RFC8499].

Also, the term "stub" is used to mean "stub resolver".

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

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Method Overview

The pseudocode for the method is:

```
# Things the stub resolver knows
# dohCapable    Does the stub know how to do DoH
# dotCapable    Does the stub know how to do DoT
# resIP         IP address of resolver
# upgradeNoAuth Does the stub want to upgrade even if it can't
                authenticate the TLS session
# insecureOK    Does the stub want to use unauthenticated classic
                DNS if DoH/DoT upgrades fail

if dohCapable:
    start TLS session on resIP:443
    if it succeeds
        if it authenticates correctly
            https://resIP/.well-known/resolver-info
            if 200-level response
                use result to do DoH; finished
            else if 300-level response
                    follow redirect, act appropriately
            else if 400-level response
                    break
        else if upgradeNoAuth:
            https://resIP/.well-known/resolver-info
            if 200-level response
                use result to do DoH; finished
            else if 300-level response
                follow redirect, act appropriately
            else if 400-level response
                break
        else
```

```
                break
        else
            break
```



```
   if dotCapable:
       start TLS session on resIP:853
       if it succeeds
           if it authenticates correctly
               start doing DoT; finished
           else if upgradeNoAuth:
               start doing DoT; finished
           else
               break
       else
           break

   if insecureOK:
       Use unencrypted DNS on port 53
   else
       DNS transport setup failed
```

2.1. Order of Desired Protocols

The pseudocode in the previous section attempts to use DoH, DoT, and unencrypted DNS, in that order. This is done to keep the pseudocode simple while demonstrating one possible order of transport selection. A stub implementation could attempt some or all of the available DNS transports in an implementation-specific or user-defined order. For example, possible lists of transports to attempt might be:

- DoH, DoT, classic DNS
- DoT, DoH

- DoT, classic DNS

- Classic DNS

3. Method Details

3.1. Inputs to the process

The method described here requires the following information. It is listed with variable names from the pseudocode in Section 2.

resIP The IP address of resolver. This can be either an IPv4 or IPv6 address.

dohCapable Set to true if the stub knows how to be a DoH client

dohCapable Set to true if the stub knows how to be a DoT client

upgradeNoAuth Set to true the stub wants to use unauthenticated DoT or DoH if it is available. Note that using unauthenticated DoT or DoH is inherently insecure because an on-path attacker can impersonate the resolver.

insecureOK Set to true if the stub wants to keep using classic (unencrypted) DNS on port 53 if the attempt to upgrade fails. Note that setting this to false will cause further DNS queries to fail if upgrade fails.

3.2. Use of Resolver Information for DoH

[pp-add-resinfo] describes how to use HTTPS to get information about a resolver. In the protocol described in this document, the JSON returned from resolving the URI with the .well-known prefix MUST have a name/value pair with a name of "doh-uri-template". The value for this name is the URI template for the DoH service, as described in [RFC8484].

3.3. TLS Authentication

In this mechanism, the stub has an IP address of the resolver. It

does not necessarily have a domain name associated with that IP address.

In order to authenticate TLS sessions, the stub resolver must have a set of TLS trust anchors, such as those maintained by some operating systems.

If the stub has a domain name associated with the resolver's IP address, and if the resolver uses that domain name in one of the subject identifiers in its certificate during the TLS exchange, the stub can use the domain name for authentication of the TLS session.

The stub always has an IP address for the resolver. If the resolver uses the same IP address used by the stub in one of the subject identifiers in its certificate during the TLS exchange, the stub can use the IP address for authentication of the TLS session.

A resolver that uses this method to publish its information SHOULD, if possible, have a TLS certificate whose subject identifiers contain any of the IP addresses that stubs might be using for the resolver. At the time that this document is published, getting IP addresses in TLS certificates is possible, but there are only a few widely-trusted CAs that issue such certificates. [RFC8738] describes a protocol that may cause IP address certificates to become more common.

4. IANA Considerations

This document needs no IANA actions.

5. Security Considerations

The method described in this document explicitly allows a stub to perform DNS communications over traditional unencrypted, unauthenticated DNS on port 53.

The method described in this document explicitly allows a stub to choose to allow unauthenticated TLS. In this case, the resulting communication will be susceptible to obvious and well-understood attacks from an attacker in the path of the communications.

6. References

6.1. Normative References

[pp-add-resinfo]
"DNS Resolver Information Self-publication", n.d..

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

6.2. Informative References

[RFC8738] Shoemaker, R., "Automated Certificate Management Environment (ACME) IP Identifier Validation Extension", RFC 8738, DOI 10.17487/RFC8738, February 2020, <https://www.rfc-editor.org/info/rfc8738>.

Authors' Addresses

Puneet Sood
Google

Email: puneets@google.com

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org