Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2021

P. Sood
Google
P. Hoffman
ICANN

June 30, 2020

Upgrading Communication from Stub Resolvers to DoT or DoH
draft-pp-add-stub-upgrade-02

Abstract

This document describes methods for a DNS stub resolver to upgrade its communications with a known recursive resolver to include encryption 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

Copyright Notice

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. It is assumed that the IP address was received insecurely, such as through DHCP.

The method 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]). The method is basically to use a DNS request as defined in [I-D.pp-add-resinfo] to get information about whether the resolver supports DoT or DoH. The method 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. Using RESINFO Responses for Upgrade

This document defines two entries for the IANA DNS Resolver Information Registry that is defined in [I-D.pp-add-resinfo].

2.1. Contacting This Resolver Using DoH

The "doh-templates" name is used to specify the URI template or templates that can be used by the stub resolver for DoH queries. The value MUST be an array of URI templates. Each element of the array in the value is a JSON string. The host part of the URI template MUST be an IP address.

[[For future: maybe drop the "MUST be an IP address" restriction and say that it can be either an IP address or host name.]]

The array in the value can be empty, which indicates that the resolver does not offer DoH service. An empty array and the absence of a name/value pair for "doh-templates" have identical meanings.

The value of "doh-templates" is an array of strings instead of just one string because a resolver might have more than one IP address or URL paths. The order of the elements in the array has no meaning; that is, the array could instead be considered a set.

[[This section needs to be updated to handle DoH over HTTP/3. These updates then need to be reflected in Section 3.]]

2.2. Contacting This Resolver Using DoT

The "dot-ports" name is used to specify the port(s) that can be used by the stub resolver for DoT queries. The value MUST be an array of port numbers. Each element of the array in the value is a JSON number.

The value of "dot-ports" is an array of numbers instead of just one number because a resolver might support DoT on more than one port. The order of the elements in the array has no meaning; that is, the array could instead be considered a set.

The array in the value can be empty, which indicates that the resolver does not offer DoT service. An empty array and the absence of a name/value pair for "dot-ports" have identical meanings.

[[For future: maybe add "dot-hostnames" to enable authentication.]]

2.3. Examples

A resolver has two IP addresses, 192.0.2.222 and 203.0.113.77. It offers DoH service, and offers DoT service on the default port. It's response to the RESINFO query might be either one of:

```
{ "dot-ports": [ 853 ], "doh-templates":
   [ "https://203.0.113.77//dns-query{?dns}",
   "https://192.0.2.222//dns-query{?dns}" ] }
```

A resolver does not offer DoH service, but does offer DoT service on the default port. It's response to the RESINFO query might be either one of:

```
{ "dot-ports": [ 853 ], "doh-templates": [] }
```

or

```
{ "dot-ports": [ 853 ] }
```

3. 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
```

[[Need to fix dohCapable to deal with DoH templates that point to resolvers other than the one queried.]]

```
   if dohCapable:
       send a DNS query of resolver-info.arpa/IN/RESINFO
       if there is a non-empty "doh-templates" name in the response:
           for each template in the name/value pair:
               start TLS session on resIP, port from DoH template
               if it succeeds
                   if it authenticates correctly
                       resolve the URI template
                       if 200-level response
                           use result to do DoH; finished
                       else if 300-level response
                               follow redirect, act appropriately
                       else if 400-level response
                               continue
                   else if upgradeNoAuth:
                       resolve the URI template
                       if 200-level response
                           use result to do DoH; finished
                       else if 300-level response
                           follow redirect, act appropriately
                       else if 400-level response
                           continue
                   else
                       continue
               else
                   continue
           # no DoH template worked

   if dotCapable:
       send a DNS query of resolver-info.arpa/IN/RESINFO
       if there is a non-empty "dot-ports" name in the response:
           for each port in the name/value pair:
               start TLS session on resIP and the port number
               if it succeeds
                   if it authenticates correctly
                       start doing DoT; finished
                   else if upgradeNoAuth:
                       start doing DoT; finished
                   else
                       continue
               else
                   continue
           # no DoT port worked

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

3.1. 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

4. Method Details

4.1. Inputs to the Process

The method described here requires the following information. It is listed with variable names from the pseudocode in Section 3.

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

dotCapable 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.

[[Add some possible implementation examples. Here's one.]]

For example, if an OS implementation's design is "just try TLS on port 853 of the current resolver", resIP is the resolver address, dohCapable is false, dotCapable is true, and upgradeNoAuth is set to true.

4.2. 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.

5. IANA Considerations

This document defines two entries for the IANA DNS Resolver Information Registry that is defined in [I-D.pp-add-resinfo].

5.1. Registration for doh-templates in the IANA DNS Resolver Information Registry

Name: doh-templates

Value type: Array of strings

Specification: This document, Section 2.1

5.2. Registration for dot-ports in the IANA DNS Resolver Information Registry

Name: dot-ports

Value type: Array of numbers

Specification: This document, Section 2.2

6. 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.

7. References

7.1. Normative References

[I-D.pp-add-resinfo]
Sood, P. and P. Hoffman, "DNS Resolver Information Self-publication", draft-pp-add-resinfo-01 (work in progress), May 2020.

[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>.

7.2. 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