Network Working Group Z. Hu
Internet-Draft L. Zhu
Intended status: Standards Track J. Heidemann
Expires: August 16, 2015 USC/Information Sciences Institute
A. Mankin
D. Wessels
Verisign Labs
P. Hoffman
VPN Consortium
February 12, 2015

TLS for DNS: Initiation and Performance Considerations
draft-hzhwm-dprive-start-tls-for-dns-01

Abstract

This document offers an approach to initiating TLS for DNS: use of a dedicated DNS-over-TLS port, and fallback to a mechanism for upgrading a DNS-over-TCP connection over the standard port (TCP/53) to a DNS-over-TLS connection. Encryption provided by TLS eliminates opportunities for eavesdropping on DNS queries in the network, such as discussed in RFC 7285. In addition, this document discusses performance considerations to minimize overheads from using TCP and TLS with DNS, pertaining to both approaches.

Status of this Memo

Copyright Notice

1. Introduction

Today, nearly all DNS queries ([RFC1034] and [RFC1035]) are sent unencrypted, which makes them vulnerable to eavesdropping by an attacker that has access to the network channel, reducing the privacy of the querier. Recent news reports have elevated these concerns, and ongoing efforts are beginning to identify privacy concerns about DNS ([I-D.ietf-dprive-problem-statement]).

Prior work has addressed some aspects of DNS security, but there has been little work till recently on privacy between a DNS client and server. DNS Security Extensions (DNSSEC, [RFC4033]) provide _response integrity_ by defining mechanisms to cryptographically sign zones, allowing end-users (or their first-hop resolver) to verify replies are correct. DNSSEC by intention does not protect request and response privacy. Traditionally, either privacy was not considered a requirement for DNS traffic, or it was assumed that network traffic was sufficiently private, however these perceptions are evolving due to recent events [RFC7285].

DNSCurve [draft-dempsky-dnscurve] defines a method to add confidentiality to the link between DNS clients and servers; however, it does so with a new cryptographic protocol and does not take advantage of an existing standard protocol such as TLS. ConfidentialDNS [draft-wijngaards-confidentialdns] and IPSECA [draft-osterweil-dane-ipsec] use opportunistic encryption to offer privacy for DNS queries and responses. Finally, others have suggested DNS-over-TLS. Unbound DNS software [unbound] includes a DNS-over-TLS implementation. The present document goes beyond past DNS-over-TLS discussions by providing two modes of initiation for DNS-over-TLS, use of a well-known port, TBD, and use of a negotiation mechanism in an established connection.

This document describes a protocol that is resilient in environments affected by differing middle box concerns. The port-based initiation of TLS is very straightforward, but might be blocked by firewalls or be unwelcome to some DNS client or server implementations. If port-

based initiation of TLS fails, there is an upgrade-based negotiation mechanism to enable DNS clients and servers to upgrade an existing DNS-over-TCP connection to a DNS-over-TLS connection, analogous to upgrade mechanisms in other uses of TLS, such as STARTTLS [RFC2595] used in SMTP [RFC3207], IMAP [RFC3501] and POP [RFC1939], to name just a few of many. But like those, the upgrade-based approach has middle box considerations, particularly downgrade attacks, as discussed in Section 2.4.

The protocol described here works for any DNS client to server communication using DNS-over-TCP. In particular, the same protocol can be used for stub-recursive and recursive-authoritative communications. We expect implementation initially between stubs and recursives.

In specifying TLS negotiation for DNS, this document defines only the protocol extensions that are needed. It does not describe how DNS clients might validate server certificates or specify trusted certificate authorities. Solutions for certificate authentication are currently outside the scope of this document.

1.1. Reserved Words

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 RFC 2119 [RFC2119].

2. Protocol Changes

The only changes required for port-based DNS-over-TLS are those optimizing TCP and TLS performance discussed in the following. The DNS protocol itself is unchanged.

Clients and servers negotiate upgrade-based DNS-over-TLS by setting a bit in the Flags field of the EDNS0 [RFC6891] OPT meta-RR. The "TLS

OK" (TO) bit is defined as the second bit of the third and fourth bytes of the "extended RCODE and flags" portion of the EDNS0 OPT meta-RR, immediately adjacent to the "DNSSEC OK" (DO) bit [RFC4033]:

```
                +0 (MSB)                +1 (LSB)
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
      0: |   EXTENDED-RCODE      |       VERSION         |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
      2: |DO|TO|                   Z                     |
         +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```


2.1. Use by DNS Clients

DNS clients first try port-based DNS-over-TLS. If that connection fails, they try upgrade-based DNS-over-TLS.

2.1.1. Port-Based DNS-over-TLS for Clients

DNS clients SHOULD first try using port-based DNS-over-TLS by establishing the TCP connection to the dedicated port TBD (number to be defined in Section 4).

Stub resolvers do not change their recursive resolvers often. A slight delay in failing to establish a port-based DNS-over-TLS connection is probably minor relative to the benefit of encrypted DNS queries and responses. The stub resolver should give a reasonable amount of time for the recursive resolver to start the TLS setup, such as a few seconds.

It SHOULD be an implementation and/or local determination as to whether to attempt TLS via the dedicated port first and then fall back to STARTTLS use, or to choose some other order of attempts and fallbacks.

2.1.2. Sending Queries for Upgrade-Based DNS-over-TLS

Setting the TO bit in queries sent using UDP transport has no protocol meaning. However, the client MAY set the TO bit when using UDP transport. The server MUST ignore the TO bit when receiving UDP transport.

DISCUSSION: community advice is sought on this. The advantage of allowing a client to send UDP on TO is that servers can collect information on deployment (as happened with the DO bit). The disadvantage is that a meaningless bit (TO over UDP) might cause confusion, and some middleboxes might not pass a UDP query with the TO bit set.

DNS clients set the TO bit in the initial query sent to a server using TCP transport to signal their desire that the TCP connection be upgraded to TLS. DNS clients SHOULD NOT set the TO bit on queries when using TCP or TLS transport because doing so has no meaning in this protocol.

Since the motivation for upgrade-based DNS-over-TLS is to preserve privacy, DNS clients SHOULD use an initial (unprotected) query that reveals no private information in the initial TO=1 query to a server. To provide a standard "dummy" query, it is RECOMMENDED to send the initial query with RD=0, QNAME="STARTTLS", QCLASS=CH, and QTYPE=TXT

("STARTTLS/CH/TXT") analogous to administrative queries already in widespread use [RFC4892].

After sending the initial TO=1 query using TCP transport, DNS clients MUST wait for the initial response before sending any subsequent queries over the same TCP connection.

2.1.3. Receiving Responses for Upgrade-Based DNS-over-TLS

A DNS client that receives a response using UDP transport that has the TO bit set handles that response as usual. It MAY record the server's support for DNS-over-TLS and use that information as part of its server selection algorithm in the case where multiple servers are available to service a particular query.

A DNS client that has sent the TO bit using TCP transport and receives a response to its initial query that has the TO bit set MUST immediately initiate a TLS handshake using the procedure described in [RFC5246]. (Note that this document does not yet deal with what happens when the TLS handshake does not succeed.)

DISCUSSION: are there any cases in which a DNS client that sent TO on

DNS-over-TCP and receives TO in the initial response from the server would not initiate the TLS handshake? Is there any reason for this to be SHOULD rather than MUST?

A DNS client that receives a response to its initial query using TCP transport that has the TO bit clear MUST not initiate a TLS handshake and SHOULD utilize the existing TCP connection for subsequent queries. DNS clients SHOULD remember server IP addresses that don't support upgrade-based DNS-over-TLS, including TLS handshake failures, and not request DNS-over-TLS from them for reasonable period (such as one hour per server).

2.2. Use by DNS Servers

A DNS server that supports DNS-over-TLS SHOULD support port-based DNS-over-TLS, and SHOULD support upgrade-based DNS-over TLS.

2.2.1. Receiving Queries for Upgrade-Based DNS-over-TLS

A DNS server receiving a query over UDP with the TO bit ignores that bit. A DNS server receiving a query over an existing TLS connection with the TO bit ignores that bit.

A DNS server receiving an initial query over TCP that has the TO bit set MAY inform the client it is willing to establish a TLS session, as described in the next section.

A DNS server receiving subsequent queries over TCP MUST ignore the TO bit. (A client wishing to start TLS after the initial query MUST open a new TCP connection to do so.)

2.2.2. Sending Responses

A DNS server sending a response over UDP to a query that had an OPT meta-RR SHOULD set the TO bit to indicate its general support for DNS-over-TLS, as long as it is willing and able to support a TLS connection with the particular client.

A DNS server receiving an initial query over TCP that has the TO bit set MAY set the TO bit in its response. The server MUST then proceed with the TLS handshake protocol.

A DNS server receiving a "dummy" STARTTLS/CH/TXT query over TCP MUST respond with RCODE=0 and a TXT RR in the Answer section. Contents of the TXT RR are strictly informative (for humans) and MUST NOT be interpreted by the client software. Recommended TXT RDATA values are "STARTTLS" or "NO_TLS".

2.3. Established Sessions

After TLS negotiation completes, the connection will be encrypted and is now protected from eavesdropping and normal DNS queries SHOULD take place, following DNS-over-TCP framing ([RFC1035], section 4.2.2).

Both clients and servers SHOULD follow existing DNS-over-TCP timeout rules, which are often implementation- and situation-dependent. In the absence of any other advice, the RECOMMENDED timeout values are 30 seconds for recursive name servers, 60 seconds for clients of recursive name servers, 10 seconds for authoritative name servers, and 20 seconds for clients of authoritative name servers. Current work in this area may assist DNS-over-TLS clients and servers select useful timeout values [draft-wouters-edns-tcp-keepalive] [tdns].

As with current DNS-over-TCP, DNS servers MAY close the connection at any time (e.g., due to resource constraints). As with current DNS-over-TCP, clients MUST handle abrupt closes and be prepared to reestablish connections and/or retry queries. DNS servers SHOULD use the TLS close-notify request to shift TCP TIME-WAIT state to the clients. Additional requirements and guidance for optimizing DNS-over-TCP are provided by [RFC5966], [I-D.ietf-dnsop-5966bis]. As discussed in [I-D.ietf-dnsop-5966bis], TCP Fast Open [RFC7413] is of benefit.

DNS servers SHOULD enable fast TLS session resumption [RFC5077] to

avoid keeping per-client session state.

2.4. Downgrade Attacks and Middleboxes

Middleboxes [RFC3234] may be present in some networks and have been known to interfere with normal DNS resolution and create problems for DNS-over-TLS. Remarkably, downgrade attacks can affect plaintext protocols that utilize "STARTTLS" signaling in a similar way. A DNS

client attempting upgrade-based DNS-over-TLS through a middlebox, or in the presence of a downgrade attack, could have one of the following outcomes. (These outcomes are similar to those discussed in prior RFCs, such as [RFC3207].)

- o The DNS client sends a TO=1 query and receives a TO=0 response. In this case there is no upgrade to TLS and DNS resolution occurs normally, without encryption.

- o The DNS client sends a TO=1 query and receives a TO=1 response, but the middlebox does not understand the TLS negotiation. Middleboxes SHOULD clear TO in replies if they are not prepared to pass through TLS negotiation. Clients SHOULD retry DNS without TO set if negotiation fails, and then retry with TLS after a reasonable period (see Section 2.1.3).

- o The DNS client sends a TO=1 query but receives no response at all. The middlebox might be silently dropping the query due to the presence of the TO bit, when it should, in fact, ignore and pass through unknown flag bits [RFC6891]. The client SHOULD fall back to normal (unencrypted) DNS for a reasonable period (as discussed in Section 2.1.3).

In general, clients that attempt TLS and fail can either fall back on unencrypted DNS, or wait and retry later, depending on their privacy requirements.

3. Performance Considerations

DNS-over-TLS incurs additional latency at session startup. It also requires additional state (memory) increased processing (CPU).

1. Latency: Compared to UDP, DNS-over-TCP requires an additional round-trip-time (RTT) of latency to establish the connection. The TLS handshake adds another two RTTs of latency. Clients and servers should support connection keepalive (reuse) and out-of-order processing to amortize connection setup costs. Moreover, TLS connection resumption can further reduce the setup delay.

2. State: The use of connection-oriented TCP requires keeping

additional state in both kernels and applications. TLS has marginal increases in state over TCP alone. The state requirements are of particular concerns on servers with many clients. Smaller timeout values will reduce the number of concurrent connections, and servers can preemptively close connections when resources limits are exceeded.

3. Processing: Use of TLS encryption algorithms results in slightly higher CPU usage. Servers can choose to refuse new DNS-over-TCP clients if processing limits are exceeded.

A full performance evaluation is outside the scope of this specification. A more detailed analysis of the performance implications of DNS-over-TLS (and DNS-over-TCP) is discussed in a technical report [tdns].

4. IANA Considerations

This document defines a new bit ("TO") in the Flags field of the EDNS0 OPT meta-RR. At the time of approval of this draft in the standards track, as per the IANA Considerations of RFC 6891, IANA is requested to reserve the second leftmost bit of the flags as the TO bit, immediately adjacent to the DNSSEC DO bit, as shown in Section 2.

IANA is requested add the following value to the "Service Name and Transport Protocol Port Number Registry" registry. That registry is populated by expert review [RFC6335], and such a review will be requested if this document progresses. It would be desirable to be assigned port 54 upon completion of review.

Service Name	DNS-over-TLS
Transport Protocol(s)	TCP
Assignee	IESG
Contact	TBD
Description	DNS query-response protocol run over TLS
Reference	This document

5. Security Considerations

The goal of this proposal is to address the security risks that arise because DNS queries may be eavesdropped upon, as described above. There are a number of residual risks that may impact this goal.

1. There are known attacks on TLS, such as person-in-the-middle and protocol downgrade. These are general attacks on TLS and not specific to DNS-over-TLS; we refer to the TLS RFCs for discussion of these security issues.

2. Any protocol interactions prior to the TLS handshake are performed in the clear and can be modified by a man-in-the-middle attacker. For this reason, clients MAY discard cached information about server capabilities advertised prior to the start of the TLS handshake.

3. As with other uses of STARTTLS-upgrade to TLS, the mechanism specified here is susceptible to downgrade attacks, where a person-in-the-middle prevents a successful TLS upgrade. Keeping track of servers known to support TLS (i.e., "pinning") enables clients to detect downgrade attacks. For servers with no connection history, clients may choose to refuse non-TLS DNS, or they may continue without TLS, depending on their privacy requirements.

4. This document does not propose new ideas for certificate authentication for TLS in the context of DNS. Several external methods are possible, although each has weaknesses. The current Certificate Authority infrastructure [RFC5280] is used by HTTP/TLS [RFC2818]. With many trusted CAs, this approach has recognized weaknesses [CA_Compromise]. Some work is underway to partially address these concerns (for example, with certificate pinning [certificate_pinning], but more work is needed. DANE [RFC6698] provides mechanisms to root certificate trust with DNSSEC. That use here must be carefully evaluated to address potential issues in trust recursion. For stub-to-recursive resolver use, certificate authentication is sometimes either easy or nearly impossible. If the recursive resolver is manually configured, its certificate can be authenticated when it is configured. If the recursive resolver is automatically configured (such as with DHCP [RFC2131]), it could use DHCP authentication mechanisms [RFC3118]).

Ongoing discussion and development of opportunistic TLS (connections without CA validation, [RFC7435]) may be relevant to DNS-over-TLS.

6. Acknowledgments

We would like to thank Stephane Bortzmeyer, Brian Haberman, Kim-Minh Kaplan, Bill Manning, George Michaelson, Eric Osterweil, Glen Wiley,

John Dickinson, and Sara Dickinson for reviewing this Internet-draft, and Nikita Somaiya for early work on this idea.

Work by Zi Hu, Liang Zhu, and John Heidemann in this paper is partially sponsored by the U.S. Dept. of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement number FA8750-12-2-0344, and contract number D08PC75599.

7. References

7.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5966] Bellis, R., "DNS Transport over TCP - Implementation Requirements", RFC 5966, August 2010.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.

7.2. Informative References

[CA_Compromise]
Infosec Island Admin, "CA Compromise", January 2012, <http://www.infosecisland.com/blogview/19782-Web-Authentication-A-Broken-Trust-with-No-Easy-Fix.html>.

[I-D.ietf-dnsop-5966bis]
Dickinson, J., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", draft-ietf-dnsop-5966bis-00 (work in progress), December 2014.

[I-D.ietf-dprive-problem-statement]
Bortzmeyer, S., "DNS privacy considerations", draft-ietf-dprive-problem-statement-00 (work in progress), October 2014.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION

4rev1", RFC 3501, March 2003.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4892] Woolf, S. and D. Conrad, "Requirements for a Mechanism Identifying a Name Server Instance", RFC 4892, June 2007.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)

Protocol: TLSA", RFC 6698, August 2012.

[RFC7285] Alimi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014.

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, December 2014.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, December 2014.

[certificate_pinning] OWASP, "Certificate and Public Key Pinning", 2014, <https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning>.

[draft-dempsky-dnscurve] Dempsky, M., "DNSCurve", draft-dempsky-dnscurve-01 (work in progress), August 2010, <http://tools.ietf.org/html/draft-dempsky-dnscurve-01>.

[draft-osterweil-dane-ipsec] Osterweil, E., Wiley, G., Mitchell, D., and A. Newton,

"Opportunistic Encryption with DANE Semantics and IPsec: IPSECA", draft-osterweil-dane-ipsec-00 (work in progress), February 2014, <http://tools.ietf.org/html/draft-osterweil-dane-ipsec-00>.

[draft-wijngaards-confidentialdns]
Wijngaards, W., "Confidential DNS", draft-wijngaards-dnsop-confidentialdns-00 (work in progress), November 2013, <http://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-00>.

[draft-wouters-edns-tcp-keepalive]
Wouters, P. and J. Abley, "The edns-tcp-keepalive EDNS0 Option", draft-wouters-edns-tcp-keepalive-00 (work in progress), October 2013, <http://tools.ietf.org/html/draft-wouters-edns-tcp-keepalive-00>.

[tdns] Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., and N. Somaiya, "T-DNS: Connection-Oriented DNS to Improve Privacy and Security", Technical report ISI-TR-688, February 2014, <Technical report, ISI-TR-688, ftp://ftp.isi.edu/isi-pubs/tr-688.pdf>.

[unbound] NLnet Labs, Verisign labs, "Unbound", December 2013, <http://unbound.net/>.

Authors' Addresses

Zi Hu
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1133
Marina del Rey, CA 90292
USA

Phone: +1 213 587-1057
Email: zihu@usc.edu

Liang Zhu
USC/Information Sciences Institute

4676 Admiralty Way, Suite 1133
Marina del Rey, CA 90292
USA

Phone: +1 310 448-8323
Email: liangzhu@usc.edu

John Heidemann
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
USA

Phone: +1 310 822-1511
Email: johnh@isi.edu

Allison Mankin
Verisign Labs
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200
Email: amankin@verisign.com

Duane Wessels
Verisign Labs
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200
Email: dwessels@verisign.com

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org