DPRIVE
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2017
T. Reddy
D. Wing
P. Patil
Cisco
July 6, 2016

DNS over DTLS (DNSoD)
draft-ietf-dprive-dnsodtls-07

Abstract

DNS queries and responses are visible to network elements on the path between the DNS client and its server. These queries and responses can contain privacy-sensitive information which is valuable to protect. An active attacker can send bogus responses causing misdirection of the subsequent connection.

To counter passive listening and active attacks, this document proposes the use of Datagram Transport Layer Security (DTLS) for DNS, to protect against passive listeners and certain active attacks. As DNS needs to remain fast, this proposal also discusses mechanisms to reduce DTLS round trips and reduce DTLS handshake size. The proposed mechanism runs over port 853.

Table of Contents

1. Introduction

The Domain Name System is specified in [RFC1034] and [RFC1035] . DNS queries and responses are normally exchanged unencrypted and are thus vulnerable to eavesdropping. Such eavesdropping can result in an undesired entity learning domains that a host wishes to access, thus resulting in privacy leakage. DNS privacy problem is further discussed in [RFC7626] .

Active attackers have long been successful at injecting bogus responses, causing cache poisoning and causing misdirection of the subsequent connection (if attacking A or AAAA records). A popular mitigation against that attack is to use ephemeral and random source ports for DNS queries [RFC5452] .

This document defines DNS over DTLS (DNSoD, pronounced "dee-enn-sod") which provides confidential DNS communication between stub resolvers

and recursive resolvers, stub resolvers and forwarders, forwarders and recursive resolvers.

The motivations for proposing DNSoD are that

- o TCP suffers from network head-of-line blocking, where the loss of a packet causes all other TCP segments to not be delivered to the application until the lost packet is re-transmitted. DNSoD, because it uses UDP, does not suffer from network head-of-line blocking.

- o DTLS session resumption consumes 1 round trip whereas TLS session resumption can start only after TCP handshake is complete. Although TCP Fast Open [RFC7413] can reduce that handshake, TCP Fast Open is only available on a few OSs, it is not yet ubiquitous.

1.1. Relationship to TCP Queries and to DNSSEC

DNS queries can be sent over UDP or TCP. The scope of this document, however, is only UDP. DNS over TCP could be protected with TLS, as described in [RFC7858].

DNS Security Extensions (DNSSEC [RFC4033]) provides object integrity of DNS resource records, allowing end-users (or their resolver) to verify legitimacy of responses. However, DNSSEC does not protect privacy of DNS requests or responses. DNSoD works in conjunction with DNSSEC, but DNSoD does not replace the need or value of DNSSEC.

2. Terminology

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 [RFC2119] .

3. Establishing and Managing DNS-over-DTLS Sessions

3.1. Session Initiation

DNSoD MUST run over standard UDP port 853 as defined in Section 7.

The host should determine if the DNS server supports DNSoD by sending a DTLS ClientHello message. A DNS server that does not support DNSoD will not respond to ClientHello messages sent by the client. If no response is received from that server, and the client has no better round-trip estimate, the client MUST retransmit the DTLS ClientHello

according to Section 4.2.4.1 of [RFC6347]. After 15 seconds, it MUST cease attempts to re-transmit its ClientHello. The client MAY repeat that procedure in the event the DNS server upgrades to support DNSoD, but such probing SHOULD NOT be done more frequently than every 24 hours and MUST NOT be done more frequently than every 15 minutes. This mechanism requires no additional signaling between the client and server. Behavior for an unsuccessful DTLS connection is discussed in Section 6.

3.2. DTLS Handshake and Authentication

Once the DNS client succeeds in receiving HelloVerifyRequest from the server via UDP on the well-known port for DNS over DTLS, it proceeds with DTLS handshake as described in [RFC6347], following the best practices specified in [RFC7525].

DNS privacy requires encrypting the query (and response) from passive attacks. Such encryption typically provides integrity protection as a side-effect, which means on-path attackers cannot simply inject bogus DNS responses. However, to provide stronger protection from active attackers pretending to be the server, the server itself needs to be authenticated. To authenticate the server providing DNS privacy, DNS client can use the authenication mechanisms discussed in [I-D.ietf-dprive-dtls-and-tls-profiles]. This document does not propose new ideas for authentication.

After DTLS negotiation completes, the connection will be encrypted and is now protected from eavesdropping.

3.3. Established Sessions

In DTLS, all data is protected using the same record encoding and mechanisms. When the mechanism described in this document is in effect, DNS messages are encrypted using the standard DTLS record encoding. When a user of DTLS wishes to send an DNS message, it delivers it to the DTLS implementation as an ordinary application data write (e.g., SSL_write()). To reduce client and server workload, clients SHOULD re-use the DTLS session. A single DTLS session can be used to send multiple DNS requests and receive multiple DNS responses.

DNSoD client and server can use DTLS heartbeat [RFC6520] to verify that the peer still has DTLS state. DTLS session is terminated by the receipt of an authenticated message that closes the connection (e.g., a DTLS fatal alert).

```
      Client                                            Server
      ------                                            ------

      ClientHello             -------->


                              <-------    HelloVerifyRequest
                                            (contains cookie)

      ClientHello             -------->
      (contains cookie)
      (empty SessionTicket extension)
                                                 ServerHello
                                 (empty SessionTicket extension)
                                                Certificate*
                                          ServerKeyExchange*
                                         CertificateRequest*
                              <--------      ServerHelloDone

      Certificate*
      ClientKeyExchange
      CertificateVerify*
      (ChangeCipherSpec)
      Finished                -------->
                                              NewSessionTicket
                                            (ChangeCipherSpec)
                              <--------              Finished


      DNS Request             --------->

                              <---------  DNS Response
```

Message Flow for Full Handshake Issuing New Session Ticket

4. Performance Considerations

To reduce number of octets of the DTLS handshake, especially the size of the certificate in the ServerHello (which can be several kilobytes), DNS client and server can use raw public keys [RFC7250] or Cached Information Extension [I-D.ietf-tls-cached-info] . Cached Information Extension avoids transmitting the server's certificate and certificate chain if the client has cached that information from a previous TLS handshake.

Since pipelined responses can arrive out of order, clients MUST match responses to outstanding queries on the same DTLS connection using

the Message ID. If the response contains a question section, the client MUST match the QNAME, QCLASS, and QTYPE fields. Failure by clients to properly match responses to outstanding queries can have serious consequences for interoperability ([RFC7766] , Section 7).

It is highly advantageous to avoid server-side DTLS state and reduce the number of new DTLS sessions on the server which can be done with [RFC5077] . This also eliminates a round-trip for subsequent DNSoD queries, because with [RFC5077] the DTLS session does not need to be re-established.

Compared to normal DNS, DTLS adds at least 13 octets of header, plus cipher and authentication overhead to every query and every response. This reduces the size of the DNS payload that can be carried. DNS client and server MUST support the EDNS0 option defined in [RFC6891] so that the DNS client can indicate to the DNS server the maximum DNS response size it can reassemble and deliver in the DNS client's network stack. The client sets its EDNS0 value as if DTLS is not being used. The DNS server must ensure that the DNS response size does not exceed the Path MTU. The DNS server must consider the amount of record expansion expected by the DTLS processing when calculating the size of DNS response that fits within the path MTU. Path MTU MUST be greater than equal to [DNS response size + DTLS overhead of 13 octets + padding size ([RFC7830]) + authentication overhead of the negotiated DTLS cipher suite + block padding (Section 4.1.1.1 of [RFC6347]]. If the DNS server's response were to exceed that calculated value, the server sends a response that does fit within that value and sets the TC (truncated) bit. The client, upon receiving a response with the TC bit set and wanting to receive the entire response, establishes a DNS-over-TLS [RFC7858] connection to the same server, and sends a new DNS request for the same resource record.

DNSoD puts an additional computational load on servers. The largest gain for privacy is to protect the communication between the DNS client (the end user's machine) and its caching resolver.

5. Anycast

DNS servers are often configured with anycast addresses. While the network is stable, packets transmitted from a particular source to an anycast address will reach the same server that has the cryptographic context from the DNS over DTLS handshake. But when the network configuration changes, a DNS over DTLS packet can be received by a server that does not have the necessary cryptographic context. To encourage the client to initiate a new DTLS handshake, DNS servers SHOULD generate a DTLS Alert message in response to receiving a DTLS packet for which the server does not have any cryptographic context.

Upon receipt of an un-authenicated DTLS alert, the DTLS client validates the Alert is within the replay window (Section 4.1.2.6 of [RFC6347]). It is difficult for the DTLS client to validate that the DTLS alert was generated by the DTLS server in response to a request or was generated by an on- or off-path attacker. Thus, upon receipt of an in-window DTLS Alert, the client SHOULD continue re-transmitting the DTLS packet (in the event the Alert was spoofed), and at the same time it SHOULD initiate DTLS session resumption. When the DTLS client receives authenticated DNS response from one of those DTLS sessions, the other DTLS session should be terminated.

6. Usage

Using DNS privacy with an authenticated server is most preferred, DNS privacy with an unauthenticated server is next preferred, and plain DNS is least preferred. This section gives a non-normative discussion on common behaviors and choices.

An implementation MAY attempt to obtain DNS privacy by contacting DNS servers on the local network (provided by DHCP) and on the Internet, and make those attempts in parallel to reduce user impact. If DNS privacy cannot be successfully negotiated for whatever reason, the client can do three things, in order from best to worst for privacy:

1. refuse to send DNS queries on this network, which means the client cannot make effective use of this network, as modern networks require DNS; or,

2. use opportunistic security, as described in [RFC7435] or,

3. send plain DNS queries on this network, which means no DNS privacy is provided.

Heuristics can improve this situation, but only to a degree (e.g., previous success of DNS privacy on this network may be reason to alert the user about failure to establish DNS privacy on this network now). Still, the client (in cooperation with the end user) has to decide to use the network without the protection of DNS privacy.

7. IANA Considerations

This specification uses port 853 already allocated in the IANA port number registry as defined in Section 6 of [RFC7858].

8. Security Considerations

The interaction between a DNS client and DNS server requires Datagram Transport Layer Security (DTLS) with a ciphersuite offering confidentiality protection and guidance given in [RFC7525] must be followed to avoid attacks on DTLS. DNS clients keeping track of servers known to support DTLS enables clients to detect downgrade attacks. To interfere with DNS over DTLS, an on- or off-path attacker might send an ICMP message towards the DTLS client or DTLS server. As these ICMP messages cannot be authenticated, all ICMP errors should be treated as soft errors [RFC1122] . For servers with no connection history and no apparent support for DTLS, depending on their Privacy Profile and privacy requirements, clients may choose to (a) try another server when available, (b) continue without DTLS, or (c) refuse to forward the query. Once a DNSoD client has established a security association with a particular DNS server, and outstanding normal DNS queries with that server (if any) have been received, the DNSoD client MUST ignore any subsequent normal DNS responses from that server, as all subsequent responses should be encrypted. This behavior mitigates all possible attacks described in Measures for Making DNS More Resilient against Forged Answers [RFC5452] .

A malicious client might attempt to perform a high number of DTLS handshakes with a server. As the clients are not uniquely identified by the protocol and can be obfuscated with IPv4 address sharing and with IPv6 temporary addresses, a server needs to mitigate the impact of such an attack. Such mitigation might involve rate limiting handshakes from a certain subnet or more advanced DoS/DDoS techniques beyond the scope of this paper.

9. Acknowledgements

Thanks to Phil Hedrick for his review comments on TCP and to Josh Littlefield for pointing out DNSoD load on busy servers (most notably root servers). The authors would like to thank Simon Josefsson, Daniel Kahn Gillmor, Bob Harold, Ilari Liusvaara, Sara Dickinson, Christian Huitema, Stephane Bortzmeyer and Geoff Huston for discussions and comments on the design of DNSoD.

10. References

10.1. Normative References

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

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <http://www.rfc-editor.org/info/rfc4033>.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <http://www.rfc-editor.org/info/rfc5077>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <http://www.rfc-editor.org/info/rfc5452>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <http://www.rfc-editor.org/info/rfc6520>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <http://www.rfc-editor.org/info/rfc6891>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <http://www.rfc-editor.org/info/rfc7525>.

[RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830, DOI 10.17487/RFC7830, May 2016, <http://www.rfc-editor.org/info/rfc7830>.

10.2. Informative References

[I-D.ietf-dprive-dtls-and-tls-profiles]
Dickinson, S., Gillmor, D., and T. Reddy, "Authentication and (D)TLS Profile for DNS-over-(D)TLS", draft-ietf-dprive-dtls-and-tls-profiles-02 (work in progress), June 2016.

[I-D.ietf-tls-cached-info]
Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", draft-ietf-tls-cached-info-23 (work in progress), May 2016.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <http://www.rfc-editor.org/info/rfc7250>.

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <http://www.rfc-editor.org/info/rfc7413>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

[RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <http://www.rfc-editor.org/info/rfc7626>.

[RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <http://www.rfc-editor.org/info/rfc7766>.

[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, <http://www.rfc-editor.org/info/rfc7858>.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com