DPRIVE T. Reddy
Internet-Draft D. Wing
Intended status: Standards Track P. Patil
Expires: May 27, 2016 Cisco
November 24, 2015

DNS over DTLS (DNSoD)
draft-ietf-dprive-dnsodtls-03

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 2
1.1. Relationship to TCP Queries and to DNSSEC 3
2. Terminology 3
3. DTLS session initiation, Polling and Discovery 3
4. Performance Considerations 4
5. Established sessions 5
6. Anycast 6
7. Downgrade attacks 7
8. IANA Considerations 7
9. Security Considerations 8
9.1. Authenticating a DNS Privacy Server 8
10. Acknowledgements 9
11. References 9
11.1. Normative References 10
11.2. Informative References 12
Authors' Addresses 13

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 [I-D.bortzmeyer-dnsop-dns-privacy].

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 for stub resolvers, recursive resolvers, iterative resolvers and authoritative servers.

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 not yet available in commercially-popular operating systems.

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 [I-D.ietf-dprive-dns-over-tls].

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. DTLS session initiation, Polling and Discovery

Many modern operating systems already detect if a web proxy is interfering with Internet communications, using proprietary mechanisms that are out of scope of this document. After that mechanism has run (and detected Internet connectivity is working), the DNSoD procedure described in this document should commence. This timing avoids delays in joining the network (and displaying an icon indicating successful Internet connection), at the risk that those initial DNS queries will be sent without protection afforded by DNSoD.

DNSoD MUST run over standard UDP port 853 as defined in Section 8. A DNS server that supports DNSoD MUST listen for and accept DTLS packets on a designated port 853.

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. The client MUST use timer values defined in Section 4.2.4.1 of [RFC6347] for retransmission of ClientHello message and if no response is received from the DNS server. After 15 seconds, it MUST cease attempts to re-transmit its ClientHello. If the DNS client receives a hard ICMP error [RFC1122], it MUST immediately cease attempts to re-transmit its ClientHello. Thereafter, the client MAY repeat that procedure in the event the DNS server has been upgraded 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.

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.

Multiple DNS queries can be sent over a single DTLS session and the DNSoD client need not wait for an outstanding reply before sending the next query. The existing Query ID allows multiple requests and responses to be interleaved in whatever order they can be fulfilled by the DNS server. This means DNSoD reduces the consumption of UDP port numbers, and because DTLS protects the communication between a DNS client and its server, the resolver SHOULD NOT use random ephemeral source ports (Section 9.2 of [RFC5452]) because such source port use would incur additional, unnecessary DTLS load on the DNSoD server. When sending multiple queries over a single DTLS session, clients MUST take care to avoid Message ID collisions. In other words, they MUST not re-use the DNS Message ID of an in-flight query.

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 handle without IP fragmentation. If the DNS sever's response exceeds the EDNS0 value, the DNS server sets the TC (truncated) bit. On receiving a response with the TC bit set, the client establishes a DNS-over-TLS 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. Implementing DNSoD on root servers is outside the scope of this document.

5. 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

6. 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, as usual (Section 4.1.2.6 of [RFC6347]). It is difficult for the DTLS client to validate 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.

7. Downgrade attacks

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:

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.

8. IANA Considerations

IANA is requested to add the following value to the "Service Name and Transport Protocol Port Number Registry" registry in the System Range. The registry for that range requires IETF Review or IESG Approval [RFC6335] and such a review has been requested using the Early Allocation process [RFC7120] for the well-known UDP port in this document.

Service Name	domain-s
Transport Protocol(s)	UDP/TCP
Port	853
Assignee	IESG
Contact	dwing@cisco.com
Description	DNS query-response protocol runs over DTLS and TLS
Reference	This document

9. 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. 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].

The DNS Fragment extension does not impact security of DTLS session establishment or application data exchange. DNS Fragment provides fragmentation and reassembly of the encrypted DNS payload.

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.

9.1. Authenticating a DNS Privacy Server

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, the DNS client needs to be configured with the names or IP addresses of those DNS privacy servers. The server certificate MUST contain DNS-ID (subjectAltName) as described in Section 4.1 of [RFC6125]. DNS names

and IP addresses can be contained in the subjectAltName entries. The client MUST use the rules and guidelines given in section 6 of [RFC6125] to validate the DNS server identity.

This could be implemented by adding the certificate name to the /etc/resolv.conf file, such as below:

```
nameserver 8.8.8.8
certificate google-public-dns.google.com
nameserver 208.67.220.220
certificate resolver.opendns.com
```

For DNS privacy servers that don't have a certificate trust chain (e.g., because they are on a home network or a corporate network), the configured list of DNS privacy servers can contain the Subject Public Key Info (SPKI) fingerprint of the DNS privacy server (i.e., a simple whitelist of name and SPKI fingerprint). The public key is used for the same reasons HTTP pinning [RFC7469] uses the public key. Raw public key-based authentication mechanism defined in [RFC7250] can be also used to authenticate the DNS server.

This could be implemented by adding the SPKI fingerprint to the /etc/resolv.conf file, such as below (line split for Internet Draft formatting):

```
nameserver 192.168.1.1
sha256 : "d6qzRu9zOECb90Uez27xWltNsj0e1Md7GkYYkVoZWmM="
```

The only algorithm considered at this time is "sha256", i.e., the hash algorithm SHA256 [RFC6234]; additional algorithms may be allowed for use in this context in the future. The quoted-string is a sequence of base 64 digits: the base64-encoded SPKI Fingerprint [RFC4648].

10. 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 and Sara Dickinson for discussions and comments on the design of DNSoD.

11. References

11.1. 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>.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <http://www.rfc-editor.org/info/rfc4492>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <http://www.rfc-editor.org/info/rfc4648>.

[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>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, DOI 10.17487/RFC5288, August 2008, <http://www.rfc-editor.org/info/rfc5288>.

[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>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <http://www.rfc-editor.org/info/rfc6125>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <http://www.rfc-editor.org/info/rfc6234>.

[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, DOI 10.17487/RFC6335, August 2011, <http://www.rfc-editor.org/info/rfc6335>.

[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>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <http://www.rfc-editor.org/info/rfc7120>.

[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>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <http://www.rfc-editor.org/info/rfc7469>.

[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>.

11.2. Informative References

[I-D.bortzmeyer-dnsop-dns-privacy] Bortzmeyer, S., "DNS privacy considerations", draft-bortzmeyer-dnsop-dns-privacy-02 (work in progress), April 2014.

[I-D.ietf-dprive-dns-over-tls] Zi, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "DNS over TLS: Initiation and Performance Considerations", draft-ietf-dprive-dns-over-tls-01 (work in progress), October 2015.

[I-D.ietf-tls-cached-info] Santesson, S. and H. Tschofenig, "Transport Layer Security (TLS) Cached Information Extension", draft-ietf-tls-cached-info-20 (work in progress), October 2015.

[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>.

[RFC3749] Hollenbeck, S., "Transport Layer Security Protocol Compression Methods", RFC 3749, DOI 10.17487/RFC3749, May 2004, <http://www.rfc-editor.org/info/rfc3749>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <http://www.rfc-editor.org/info/rfc4821>.

[RFC6928] Chu, J., Dukkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, <http://www.rfc-editor.org/info/rfc6928>.

[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>.

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