DNSOP Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 7, 2015

W. Wijngaards
NLnet Labs
G. Wiley
VeriSign, Inc.

March 6, 2015

Confidential DNS
draft-wijngaards-dnsop-confidentialdns-03

Abstract

This document offers opportunistic encryption to provide privacy for DNS queries and responses.

Requirements Language

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].

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 http://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 September 7, 2015.

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.

1. Introduction

The privacy of the Question, Answer, Authority and Additional sections in DNS queries and responses is protected by the confidential DNS protocol by encrypting the contents of each section. The goal of this change to the DNS protocol is to make large scale monitoring more expensive, see [draft-bortzmeyer-dnsop-dns-privacy] and [draft-koch-perpass-dns-confidentiality]. Authenticity and integrity may be provided by DNSSEC, this protocol does not change DNSSEC and does not offer the means to authenticate responses.

Confidential communication between any pair of DNS servers is supported, both between iterative resolvers and authoritative servers and between stub resolvers and recursive resolvers.

The confidential DNS protocol has minimal impact on the number of packets involved in a typical DNS query/response exchange by leveraging a cacheable ENCRYPT Resource Record and an optionally cacheable shared secret. The protocol supports selectable cryptographic suites and parameters (such as key sizes).

The client fetches an ENCRYPT RR from the server that it wants to contact. The public key retrieved in the ENCRYPT RR is used to encrypt a shared secret or public key that the client uses to encrypt the sections in the DNS query and which the name server uses to encrypt the DNS response.

As this is opportunistic encryption, the key is (re-)fetched when the exchange fails or after the TTL expires. If the key fetch fails or the encrypted query fails, communication in the clear is performed.

The server advertises which crypto suites and key lengths may be used in the ENCRYPT RR, the client then chooses a crypto suite from this list and includes that selection in subsequent DNS queries.

The key from the server can be cached by the client, using the TTL specified in the ENCRYPT RR, the IP address of the server distinguishes keys in the cache. The server may also cache shared secrets and keys from clients.

The optional authenticated mode of operation uses two mechanisms, one for authoritative and one for recursive servers, that fetch the public key for the server and sign it with DNSSEC. For authoritative

servers, the key is included in an extra DS record in the parent's delegation. For recursive servers the key is at the reverse IP address location.

2. ENCRYPT RR Type

The RR type for confidential DNS is ENCRYPT, type TBD (decimal). The presentation format is:

. ENCRYPT [flags] [algo] [id] [data]

The flags, algo and id are unsigned numbers in decimal and the data is in base-64. The wireformat is: one octet flags, one octet algo, one octet id and the remainder of the rdata is for the data. The type is class independent. The domain name of the ENCRYPT record is '.' (the root label) for hop-by-hop exchanges.

In the flags the least two bits are the usage value. The other flag bits MUST be sent as zeroes, and the receiver MUST ignore RRs that have other flag bits set.

- PAD (usage=0): the ENCRYPT contains padding material. Algo and id are set to 0. Its data length varies (0-63 octets), and may contain any value. It is used to pad packets to obscure the packet length. Append such records to make the DNS message for queries and answers a whole multiple of 64 bytes.

- KEY (usage=1): the ENCRYPT contains a public or symmetric key. The algo field gives the algorithm. The id identifies the key, this id is copied to ENCRYPT type RRS to identify which key to use to decrypt the data. The data contains the key bits.

- RRS (usage=2): encrypted data. The data contains encrypted resource records. The data is encrypted with the selected algorithm and key id. The data contains resource records in DNS wireformat [RFC1034], with a domain name, type, class, ttl, rdatalength and rdata.

- SYM (usage=3): the ENCRYPT contains an encrypted symmetric key. The contained, encrypted data is rdata of an ENCRYPT of type KEY and has the symmetric key. The data is encrypted with the algorithm and id indicated. The encrypted data encompasses the flags, algo, id, data for the symmetric key.

The ENCRYPT RR type can contain keys. It uses the same format as the DNSKEY record [RFC4034] for public keys. algo=0 is reserved for future expansion of the algorithm number above 255. algo=1 is RSA, the rdata determines the key size. algo=2 is AES, aes-cbc, size of

the rdata determines the size of the key.

3. Server and Client Algorithm

If a clients wants to fetch the keys for the server from the server, it performs a query with query type ENCRYPT and query name '.' (root label). The reply contains the ENCRYPT (or multiple if a choice is offered) in the answer section. These ENCRYPTs have the KEY usage.

If a client wants to perform an encrypted query, it sends an unencrypted outer packet, with query type ENCRYPT and query name '.' (root label). In the authority section it includes an ENCRYPT record of type RRS. This encrypts a number of records, the first is a query-section style query record, and then zero or more ENCRYPTs of type KEY that the server uses to encrypt the reply. If the client wants to use a symmetric key, it omits the KEYs, and instead includes an ENCRYPT of type SYM in the authority section. The ENCRYPT of type RRs then follows after the SYM and can be encrypted with the key from that SYM.

If a server wants to encrypt a reply, it also uses the ENCRYPT type. The reply looks like a normal DNS packet, i.e. it has a normal unencrypted outer DNS packet. Because the query name and query type have been encrypted, the outer packet has a query name of '.' and query type of ENCRYPT and the reply has an ENCRYPT type RRS in the answer section. The reply RRs have been encrypted into the data of the ENCRYPT record. The RRS data starts with 10 bytes of header; the flags and section counts.

The client may lookup keys whenever it wants to. It may cache the keys for the server, using the TTL of those ENCRYPT records. It should also cache failures to lookup the ENCRYPT record for some time. If the client fails to look up the ENCRYPT records it MUST fall back to unencrypted communication (this is the opportunistic encryption case). The result of an encrypted query may also be timeouts, errors or replies with mangled contents, in that case the client MUST fall back to unencrypted communication (this is the opportunistic encryption case).

If some middlebox removes the ENCRYPT from the authority section of an encrypted query, the query looks like a . ENCRYPT lookup and likely a reply with ENCRYPTs of type KEY is returned instead of the encrypted reply with an ENCRYPT of type RRS, and again the client does the unencrypted fallback (this is the opportunistic encryption case). If the server has changed its keys and does not recognize the keys in an encrypted query, it should return an ENCRYPT record of type PAD with no data. A server may decide it does not (any longer) have the resources for encryption and reply with SERVFAIL to

encrypted queries, forcing unencrypted fallback (this is the opportunistic encryption case). Keys for unknown algorithms should be ignored by the client, if no usable keys remain, fallback to insecure (this is for both opportunistic and authenticated).

The client may cache the ENCRYPT of type SYM for a server together with the symmetric secret, this is better for performance, as public-key operations can be avoided for repeated queries. The server may also cache the ENCRYPTs of type SYM with the decoded secret, associating a lookup for the rdata of the SYM record with the decoded secret, avoiding public-key operations for repeated queries. This is why the SYM record is sent separately in the authority section in queries (it is identical and can be used for cache lookups).

Key rollover is possible, support the old key for its TTL, while advertising the new key, for the servers. For clients, generate a new public or symmetric key and use it.

4. Authenticated Operation

The previous documented the opportunistic operation, where deployment is easier, but security is weaker. This documents options for authenticated operation. The client selects if encryption is authenticated, opportunistic, or disabled in its local policy (configuration).

The authentication happens with a DNSSEC signed DS record that carries the key for confidential DNS. This removes a full roundtrip from the connection setup cost. The DS has hash type TBDhashtype, that is specific for confidential DNS. The DS record carries a flag byte and the public key (in DNSKEY's wireformat) in its rdata. This means that the confidential DNS keys are acquired with a referral to the zone and are secured with DNSSEC.

Because the key itself is carried, the probe sequence can be omitted and an encrypted query can be sent to the delegated server straight away. The nameservers for that zone then MUST support using that key for encrypting packets. The servers have the same key with authenticated mode, where with the opportunistic mode, every server could have its own key.

Validators do not know or support the DS with ENCRYPT hash type, those validators ignore them and continue to DNSSEC validate the zone. Validators that support the new hash type should use them to encrypt messages and use the remaining DS records to DNSSEC validate the zone.

This changes the opportunistic encryption to authenticated

encryption. The fallback to insecure is still possible and this may make deployment easier. The one byte at the start of the base64 data, in its least significant bit, signals if fallback to insecure is allowed (value 0x01). That gives the zone owner the option to enable fallback to insecure or if it should be disabled. The remainder of the DS base64 data contains a public key in the same format as when sent in the rdata of ENCRYPT KEY. The type of the key is in the key type field of this DS record. With fallback to insecure disabled and the keys authenticated the confidential DNS query and response should be fully secure (i.e. not 'Opportunistically' secure).

With fallback to insecure disabled, queries fail instead of falling back to insecure. This means no answer is acquired, and DNS lookups for that zone fail because the security failed.

The DS method works for authority servers. Recursors need another method. The client looks up reverse-of-recursors-IP.arpa ENCRYPT and gets the keys signed with DNSSEC from there (type ENCRYPT KEY lookup). If there is no dnssec secure answer with a key, the opportunistic key exchange is attempted. Do this for DNSSEC-insecure answers, if there is no trust anchor, or when no such name and ENCRYPT are present. If it is dnssec bogus, then authentication failed and it is not possible to communicate with the server (with the authenticated communication mode selected by the client).

5. IANA Considerations

An RR type registration for type ENCRYPT with number TBD and it references this document [[to be done when this becomes RFC]].

A DS record hash type is registered TBDhashtype that references this document. It is for the confidential DNS public key, acronym ENCRYPT.

6. Security Considerations

Opportunistic encryption can be configured. Opportunistic encryption has many drawbacks against active intrusion, but it works against pervasive passive surveillance, and thus it improves privacy.

With authentication (if selected by the client) the key is secured with DNSSEC.

This technique encrypts DNS queries and answers, but other data sources, such as timing, IP addresses, and the packet size can be observed. These could provide almost all the information that was encrypted.

7. Acknowledgments

Roy Arends

8. Normative References

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

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

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

Authors' Addresses

Wouter Wijngaards
NLnet Labs
Science Park 140
Amsterdam 1098 XH
The Netherlands

EMail: wouter@nlnetlabs.nl

Glen Wiley
VeriSign, Inc.
Reston, VA
USA

EMail: gwiley@verisign.com