

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 29, 2014

S. Bortzmeyer  
AFNIC  
April 27, 2014

DNS privacy considerations  
draft-bortzmeyer-dnsop-dns-privacy-02

Abstract

This document describes the privacy issues associated with the use of the DNS by Internet users. It is intended to be mostly an analysis of the present situation, in the spirit of section 8 of [RFC6973] and it does not prescribe solutions.

Discussions of the document should take place on the dns-privacy mailing list [dns-privacy].

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 October 29, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect 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.

## Table of Contents

1. Introduction . . . . .	2
2. Risks . . . . .	4
2.1. The alleged public nature of DNS data . . . . .	4
2.2. Data in the DNS request . . . . .	4
2.3. Cache snooping . . . . .	5
2.4. On the wire . . . . .	6
2.5. In the servers . . . . .	7
2.5.1. In the resolvers . . . . .	8
2.5.2. In the authoritative name servers . . . . .	8
2.5.3. Rogue servers . . . . .	9
3. Actual "attacks" . . . . .	9
4. Legalities . . . . .	9
5. Security considerations . . . . .	9
6. Acknowledgments . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	10
Author's Address . . . . .	12

## 1. Introduction

The Domain Name System is specified in [RFC1034] and [RFC1035]. It is one of the most important infrastructure components of the Internet and one of the most often ignored or misunderstood. Almost every activity on the Internet starts with a DNS query (and often several). Its use has many privacy implications and we try to give here a comprehensive and accurate list.

Let us start with a small reminder of the way the DNS works (with some simplifications). A client, the stub resolver, issues a DNS query to a server, the resolver (also called caching resolver or full resolver or recursive name server). For instance, the query is "What are the AAAA records for www.example.com?". AAAA is the qtype (Query Type) and www.example.com the qname (Query Name). To get the answer, the resolver will query first the root nameservers, which will, most of the times, send a referral. Here, the referral will be to .com nameservers. In turn, they will send a referral to the example.com nameservers, which will provide the answer. The root name servers, the name servers of .com and those of example.com are called authoritative name servers. It is important, when analyzing the privacy issues, to remember that the question asked to all these name servers is always the original question, not a derived question. Unlike what many "DNS for dummies" articles say, the question sent to

the root name servers is "What are the AAAA records for www.example.com?", not "What are the name servers of .com?". So, the DNS leaks more information than it should.

Because the DNS uses caching heavily, not all questions are sent to the authoritative name servers. If the stub resolver, a few seconds later, asks to the resolver "What are the SRV records of \_xmpp-server.\_tcp.example.com?", the resolver will remember that it knows the name servers of example.com and will just query them, bypassing the root and .com. Because there is typically no caching in the stub resolver, the resolver, unlike the authoritative servers, sees everything.

Almost all the DNS queries are today sent over UDP, and this has practical consequences if someone thinks of encrypting this traffic (some encryption solutions are typically done for TCP, not UDP).

I should be noted to that DNS resolvers sometimes forward requests to bigger machines, with a larger and more shared cache, the forwarders. From the point of view of privacy, forwarders are like resolvers, except that the caching in the resolver before them decreases the amount of data they can see.

Another important point to keep in mind when analyzing the privacy issues of DNS is the mix of many sort of DNS requests received by a server. Let's assume the eavesdropper want to know which Web page is visited by an user. For a typical Web page displayed by the user, there are three sorts of DNS requests:

Primary request: this is the domain name that the user typed or selected from a bookmark or choosed by clicking on an hyperklink. Presumably, this is what is of interest for the eavesdropper.

Secondary requests: these are the requests performed by the user agent (here, the Web browser) without any direct involvment or knowledge of the user. For the Web, they are triggered by included content, CSS sheets, JavaScript code, embedded images, etc. In some cases, there can be dozens of domain names in a single page.

Tertiary requests: these are the requests performed by the DNS system itself. For instance, if the answer to a query is a referral to a set of name servers, and the glue is not returned, the resolver will have to do tertiary requests to turn name servers' named into IP addresses.

For privacy-related terms, we will use here the terminology of [RFC6973].

## 2. Risks

This draft focuses mostly on the study of privacy risks for the end-user (the one performing DNS requests). Privacy risks for the holder of a zone (the risk that someone gets the data) are discussed in [RFC5936]. Non-privacy risks (such as cache poisoning) are out of scope.

### 2.1. The alleged public nature of DNS data

It has long been claimed that "the data in the DNS is public". While this sentence makes sense for an Internet wide lookup system, there are multiple facets to data and meta data that deserve a more detailed look. First, access control lists and private name spaces notwithstanding, the DNS operates under the assumption that public facing authoritative name servers will respond to "usual" DNS queries for any zone they are authoritative for without further authentication or authorization of the client (resolver). Due to the lack of search capabilities, only a given qname will reveal the resource records associated with that name (or that name's non existence). In other words: one needs to know what to ask for to receive a response. The zone transfer qtype [RFC5936] is often blocked or restricted to authenticated/authorized access to enforce this difference (and maybe for other, more dubious reasons).

Another differentiation to be applied is between the DNS data as mentioned above and a particular transaction, most prominently but not limited to a DNS name lookup. The fact that the results of a DNS query are public within the boundaries described in the previous paragraph and therefore might have no confidentiality requirements does not imply the same for a single or a sequence of transactions. A typical example from outside the DNS world: the Web site of Alcoholics Anonymous is public, the fact that you visit it should not be.

### 2.2. Data in the DNS request

The DNS request includes many fields but two of them seem specially relevant for the privacy issues, the qname and the source IP address. "source IP address" is used in a loose sense of "source IP address + may be source port", because the port is also in the request and can be used to sort out several users sharing an IP address (CGN for instance).

The qname is the full name sent by the original user. It gives information about what the user does ("What are the MX records of example.net?" means he probably wants to send email to someone at example.net, which may be a domain used by only a few persons and

therefore very revealing). Some qnames are more sensitive than others. For instance, querying the A record of google-analytics.com reveals very little (everybody visits Web sites which use Google Analytics) but querying the A record of www.verybad.example where verybad.example is the domain of an illegal or very offensive organization may create more problems for the user. Another example is when the qname embeds the software one uses. For instance, \_ldap.\_tcp.Default-First-Site-Name.\_sites.gc.\_msdcs.example.org. Or some BitTorrent clients that query a SRV record for \_bittorrent-tracker.\_tcp.domain.example.

Another important thing about the privacy of the qname is the future usages. Today, the lack of privacy is an obstacle to putting interesting data in the DNS. At the moment your DNS traffic might reveal that you are doing email but not who with. If your MUA starts looking up PGP keys in the DNS [I-D.wouters-dane-openpgp] then privacy becomes a lot more important. And email is just an example, there will be other really interesting uses for a more secure (in the sense of privacy) DNS.

For the communication between the stub resolver and the resolver, the source IP address is the one of the user's machine. Therefore, all the issues and warnings about collection of IP addresses apply here. For the communication between the resolver and the authoritative name servers, the source IP address has a different meaning, it does not have the same status as the source address in a HTTP connection. It is now the IP address of the resolver which, in a way "hides" the real user. However, it does not always work. Sometimes [I-D.vandergaast-edns-client-subnet] is used. Sometimes the end user has a personal resolver on her machine. In that case, the IP address is as sensitive as it is for HTTP.

A note about IP addresses: there is currently no IETF document which describes in detail the privacy issues of IP addressing. In the mean time, the discussion here is intended to include both IPv4 and IPv6 source addresses. For a number of reasons their assignment and utilization characteristics are different, which may have implications for details of information leakage associated with the collection of source addresses. (For example, a specific IPv6 source address seen on the public Internet is less likely than an IPv4 address to originate behind a CGN or other NAT.) However, for both IPv4 and IPv6 addresses, it's important to note that source addresses are propagated with queries and comprise metadata about the host, user, or application that originated them.

### 2.3. Cache snooping

The content of resolvers can reveal data about the clients using it. This information can sometimes be examined by sending DNS queries with RD=0 to inspect cache content, particularly looking at the DNS TTLs. Since this also is a reconnaissance technique for subsequent cache poisoning attacks, some counter measures have already been developed and deployed.

#### 2.4. On the wire

DNS traffic can be seen by an eavesdropper like any other traffic. It is typically not encrypted. (DNSSEC, specified in [RFC4033] explicitly excludes confidentiality from its goals.) So, if an initiator starts a HTTPS communication with a recipient, while the HTTP traffic will be encrypted, the DNS exchange prior to it will not be. When the other protocols will become more or more privacy-aware and secured against surveillance, the DNS risks to become "the weakest link" in privacy.

What also makes the DNS traffic different is that it may take a different path than the communication between the initiator and the recipient. For instance, an eavesdropper may be unable to tap the wire between the initiator and the recipient but may have access to the wire going to the resolver, or to the authoritative name servers.

The best place, from an eavesdropper's point of view, is clearly between the stub resolvers and the resolvers, because he is not limited by DNS caching.

The attack surface between the stub resolver and the rest of the world can vary widely depending upon how the end user's computer is configured. By order of increasing attack surface:

The resolver can be on the end user's computer. In (currently) a small number of cases, individuals may choose to operate their own DNS resolver on their local machine. In this case the attack surface for the stub resolver to caching resolver connection is limited to that single machine.

The resolver can be in the IAP (Internet Access Provider) premises. For most residential users and potentially other networks the typical case is for the end user's computer to be configured (typically automatically through DHCP) with the addresses of the DNS resolver at the IAP. The attack surface for on-the-wire attacks is therefore from the end user system across the local network and across the IAP network to the IAP's resolvers.

The resolver may also be at the local network edge. For many/most enterprise networks and for some residential users the caching

resolver may exist on a server at the edge of the local network. In this case the attack surface is the local network. Note that in large enterprise networks the DNS resolver may not be located at the edge of the local network but rather at the edge of the overall enterprise network. In this case the enterprise network could be thought of as similar to the IAP network referenced above.

The resolver can be a public DNS service. Some end users may be configured to use public DNS resolvers such as those operated by Google Public DNS or OpenDNS. The end user may have configured their machine to use these DNS resolvers themselves - or their IAP may choose to use the public DNS resolvers rather than operating their own resolvers. In this case the attack surface is the entire public Internet between the end user's connection and the public DNS service.

## 2.5. In the servers

Using the terminology of [RFC6973], the DNS servers (resolvers and authoritative servers) are enablers: they facilitate communication between an initiator and a recipient without being directly in the communications path. As a result, they are often forgotten in risk analysis. But, to quote again [RFC6973], "Although [...] enablers may not generally be considered as attackers, they may all pose privacy threats (depending on the context) because they are able to observe, collect, process, and transfer privacy-relevant data." In [RFC6973] parlance, enablers become observers when they start collecting data.

Many programs exist to collect and analyze DNS data at the servers. From the "query log" of some programs like BIND, to tcpdump and more sophisticated programs like PacketQ [packetq] reference and DNSmezzo [dnsmezzo]. The organization managing the DNS server can use this data itself or it can be part of a surveillance program like PRISM [prism] and pass data to an outside attacker.

Sometimes, these data are kept for a long time and/or distributed to third parties, for research purposes [ditl], for security analysis, or for surveillance tasks. Also, there are observation points in the network which gather DNS data and then make it accessible to third-parties for research or security purposes ("passive DNS [passive-dns]").

#### 2.5.1. In the resolvers

The resolvers see the entire traffic since there is typically no caching before them. They are therefore well situated to observe the traffic. To summarize: your resolver knows a lot about you. The resolver of a large IAP, or a large public resolver can collect data from many users. You may get an idea of the data collected by reading the privacy policy of a big public resolver [1].

#### 2.5.2. In the authoritative name servers

Unlike the resolvers, they are limited by caching. They see only a part of the requests. For aggregated statistics ("what is the percentage of LOC queries?"), it is sufficient but it may prevent an observer to observe everything. Nevertheless, the authoritative name servers sees a part of the traffic and this sample may be sufficient to defeat some privacy expectations.

Also, the end user has typically some legal/contractual link with the resolver (he has chosen the IAP, or he has chosen to use a given public resolver) while he is often not even aware of the role of the authoritative name servers and their observation abilities.

It is an interesting question whether the privacy issues are bigger in the root or in a large TLD. The root sees the traffic for all the TLDs (and the huge amount of traffic for non-existing TLD) but a large TLD has less caching before it.

As noted before, using a local resolver or a resolver close to the machine decreases the attack surface for an on-the-wire eavesdropper. But it may decrease privacy against an observer located on an authoritative name server since the authoritative name server will see the IP address of the end client, and not the address of a big resolver shared by many users. This is no longer true if [I-D.vandergaast-edns-client-subnet] is used because, in this case, the authoritative name server sees the original IP prefix or address (depending on the setup).

As of today, all the instances of one root name server, L-root, receive together around 20 000 queries per second. While most of it is junk (errors on the TLD name), it gives an idea of the amount of big data which pours into name servers.



Many domains, including TLD, are partially hosted by third-party servers, sometimes in a different country. The contracts between the domain manager and these servers may or may not take privacy into account. But it may be surprising for an end-user that requests to a given ccTLD may go to servers managed by organisations outside of the country.

#### 2.5.3. Rogue servers

A rogue DHCP server can direct you to a rogue resolver. Most of the times, it seems to be done to divert traffic, by providing lies for some domain names. But it could be used just to capture the traffic and gather information about you. Same thing for malwares like DNSChanger[dnschanger] which changes the resolver in the machine's configuration.

### 3. Actual "attacks"

A very quick examination of DNS traffic may lead to the false conclusion that extracting the needle from the haystack is difficult. "Interesting" primary DNS requests are mixed with useless (for the eavesdropper) second and tertiary requests (see the terminology in Section 1). But, in this time of "big data" processing, powerful techniques now exist to get from the raw data to what you're actually interested in.

Many research papers about malware detection use DNS traffic to detect "abnormal" behaviour that can be traced back to the activity of malware on infected machines. Yes, this research was done for the good but, technically, it is a privacy attack and it demonstrates the power of the observation of DNS traffic. See [dns-footprint], [dagon-malware] and [darkreading-dns].

Passive DNS systems [passive-dns] allow reconstruction of the data of sometimes an entire zone. It is used for many reasons, some good, some bad. It is an example of privacy issue even when no source IP address is kept.

### 4. Legalities

To our knowledge, there are no specific privacy laws for DNS data. Interpreting general privacy laws like [data-protection-directive] (European Union) in the context of DNS traffic data is not an easy task and it seems there is no court precedent here.

### 5. Security considerations

This document is entirely about security, more precisely privacy. Possible solutions to the issues described here are discussed in [I-D.bortzmeyer-dnsop-privacy-sol] (qname minimization, local caching resolvers), [I-D.hzhwm-start-tls-for-dns] (encryption of traffic) or in [I-D.wijnngaards-dnsop-confidentialdns] (encryption also). Attempts have been made to encrypt the resource record data [I-D.timms-encrypt-naptr].

## 6. Acknowledgments

Thanks to Nathalie Boulevard and to the CENTR members for the original work which led to this draft. Thanks to Ondrej Sury for the interesting discussions. Thanks to Mohsen Souissi for proofreading. Thanks to Dan York, Suzanne Woolf, Tony Finch, Peter Koch and Frank Denis for good written contributions.

## 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.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

### 7.2. Informative References

- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.

- [I-D.vandergaast-edns-client-subnet]  
Contavalli, C., Gaast, W., Leach, S., and E. Lewis,  
"Client Subnet in DNS Requests", draft-vandergaast-edns-client-subnet-02 (work in progress), July 2013.
- [I-D.bortzmeyer-dnsop-privacy-sol]  
Bortzmeyer, S., "Possible solutions to DNS privacy issues", draft-bortzmeyer-dnsop-privacy-sol-00 (work in progress), December 2013.
- [I-D.wijngaards-dnsop-confidentialdns]  
Wijngaards, W., "Confidential DNS", draft-wijngaards-dnsop-confidentialdns-00 (work in progress), November 2013.
- [I-D.timms-encrypt-naptr]  
Timms, B., Reid, J., and J. Schlyter, "IANA Registration for Encrypted ENUM", draft-timms-encrypt-naptr-01 (work in progress), July 2008.
- [I-D.hzhwm-start-tls-for-dns]  
Zi, Z., Zhu, L., Heidemann, J., Mankin, A., and D. Wessels, "Starting TLS over DNS", draft-hzhwm-start-tls-for-dns-00 (work in progress), February 2014.
- [I-D.wouters-dane-openpgp]  
Wouters, P., "Using DANE to Associate OpenPGP public keys with email addresses", draft-wouters-dane-openpgp-02 (work in progress), February 2014.
- [dns-privacy]  
IETF, , "The dns-privacy mailing list", March 2014.
- [dnsop]  
IETF, , "The dnsop mailing list", October 2013.
- [dagon-malware]  
Dagon, D., "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority", 2007.
- [dns-footprint]  
Stoner, E., "DNS footprint of malware", October 2010.
- [darkreading-dns]  
Lemos, R., "Got Malware? Three Signs Revealed In DNS Traffic", May 2013.
- [dnschanger]  
Wikipedia, , "DNSChanger", November 2011.

- [dnscrypt] Denis, F., "DNSEncrypt", .
- [dnscurve] Bernstein, D., "DNSCurve", .
- [packetq] , "PacketQ, a simple tool to make SQL-queries against PCAP-files", 2011.
- [dnsmezzo] Bortzmeyer, S., "DNSmezzo", 2009.
- [prism] NSA, , "PRISM", 2007.
- [crime] Rizzo, J. and T. Dong, "The CRIME attack against TLS", 2012.
- [ditl] , "A Day in the Life of the Internet (DITL)", 2002.
- [data-protection-directive] , "European directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data", November 1995.
- [passive-dns] Weimer, F., "Passive DNS Replication", April 2005.
- [tor-leak] , "DNS leaks in Tor", 2013.

## Author's Address

Stephane Bortzmeyer  
AFNIC  
Immeuble International  
Saint-Quentin-en-Yvelines 78181  
France

Phone: +33 1 39 30 83 46  
Email: bortzmeyer+ietf@nic.fr  
URI: <http://www.afnic.fr/>

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: June 20, 2014

S. Bortzmeyer  
AFNIC  
December 17, 2013

Possible solutions to DNS privacy issues  
draft-bortzmeyer-dnsop-privacy-sol-00

Abstract

This document describes some possible solutions to the DNS privacy issues described in [I-D.bortzmeyer-dnsop-dns-privacy].

Discussions of the document should currently take place on the dnsop mailing list [dnsop].

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 June 20, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect 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.

## Table of Contents

1. Introduction and background . . . . .	2
2. Possible technical solutions . . . . .	2
2.1. On the wire . . . . .	3
2.1.1. Reducing the attack surface . . . . .	3
2.1.2. Encrypting the DNS traffic . . . . .	3
2.2. In the servers . . . . .	4
2.2.1. In the resolvers . . . . .	4
2.2.2. In the authoritative name servers . . . . .	5
2.2.3. Rogue servers . . . . .	6
3. Security considerations . . . . .	6
4. Acknowledgments . . . . .	6
5. References . . . . .	6
5.1. Normative References . . . . .	6
5.2. Informative References . . . . .	7
Author's Address . . . . .	8

## 1. Introduction and background

The problem statement is exposed in [I-D.bortzmeyer-dnsop-dns-privacy]. The terminology here is also defined in this companion document.

## 2. Possible technical solutions

We mention here only the solutions that could be deployed in the current Internet. Disruptive solutions, like replacing the DNS with a completely new resolution protocol, are interesting but are kept for a future work. Remember that the focus of this document is on describing the threats, not in detailing solutions. This section is therefore non-normative and is NOT a technical specification of solutions. For the same reason, there are not yet actual recommendations in this document.

Raising seriously the bar against the eavesdropper will require SEVERAL actions. Not one is decisive by itself but, together, they can have an effect. The most important suggested here are:

qname minimization,

encryption of DNS traffic,

padding (sending random queries from time to time).

We detail some of these actions later, classified by the kind of observer (on the wire, in a server, etc). Some actions will help

against several kinds of observers. For instance, padding, sending gratuitous queries from time to time (queries where you're not interested in the replies, just to disturb the analysis), is useful against all sorts of observers. It is a costly technique, because it increases the traffic on the network but it seriously blurs the picture for the observer.

## 2.1. On the wire

### 2.1.1. Reducing the attack surface

See Section 2.2.1 since the solution described there apply against on-the-wire eavesdropping as well as against observation by the resolver.

### 2.1.2. Encrypting the DNS traffic

To really defeat an eavesdropper, there is only one solution: encryption. But, from the end user point of view, even if you check that your communication between your stub resolver and the resolver is encrypted, you have no way to ensure that the communication between the resolver and the authoritative name servers will be. There are two different cases, communication between the stub resolver and the resolver (no caching but only two parties so solutions which rely on an agreement may work) and communication between the resolver and the authoritative servers (less data because of caching, but many parties involved, so any solution has to scale well). Encrypting the "last mile", between the user's stub resolver and the resolver may be sufficient since the biggest danger for privacy is between the stub resolver and the resolver, because there is no caching involved there.

The only encryption mechanism available for DNS which is today an IETF standard is IPsec in ESP mode. Its deployment in the wide Internet is very limited, for reasons which are out of scope here. Still, it may be a solution for "the last mile" and, indeed, many VPN solutions use it this way, encrypting the whole traffic, including DNS to the safe resolver. In the IETF standards, a possible alternative could be DTLS [RFC6347]. It enjoyed very little actual deployment and its interaction with the DNS has never been considered, studied or of course implemented. There are also non standard encryption techniques like DNSCrypt [dnscrypt] for the stub resolver <-> resolver communication or DNSCurve [dnscurve] for the resolver <-> authoritative server communication. It seems today that the possibility of massive encryption of DNS traffic is very remote.

A last "pervasive encryption" solution for the DNS could be the promising [I-D.wijnngaards-dnsop-confidentialdns].

Another solution would be to use more TCP for the queries, together with TLS [RFC5246]. DNS can run over TCP and it provides a good way to leverage the software and experience of the TLS world. There have been discussions to use more TCP for the DNS, in light of reflection attacks (based on the spoofing of the source IP address, which is much more difficult with TCP). For instance, a stub resolver could open a TCP connection with the resolver at startup and keep it open to send queries and receive responses. The server would of course be free to tear down these connections at will (when it is under stress, for instance) and the client could reestablish them when necessary. Remember that TLS sessions can survive TCP connections so there is no need to restart the TLS negotiation each time. This DNS-over-TLS-over-TCP is already implemented in the Unbound resolver. It is safe only if pipelining multiple questions over the same channel. Name compression should also be disabled, or CRIME-style [crime] attacks can apply.

Encryption alone does not guarantee perfect privacy, because of the available metadata. For instance, the size of questions and responses, even encrypted, provide hints about what queries have been sent. (DNSCrypt uses random-length padding, and a 64 bytes block size, to limit this risk, but this raises other issues, for instance during amplification attacks. Other security protocols use similar techniques, for instance ESPv3.) Observing the periodicity of encrypted questions/responses also discloses the TTL, which is yet another hint about the queries. Non-cached responses are disclosing the RTT between the resolver and authoritative servers. This is a very useful indication to guess where authoritative servers are located. Web pages are made of many resources, leading to multiple requests, whose number and timing fingerprint which web site is being browsed. So, observing encrypted traffic is not enough to recover any plaintext queries, but is enough to answer the question "is one of my employees browsing Facebook?". Finally, attackers can perform a denial-of-service attack on possible targets, check if this makes a difference on the encrypted traffic they observe, and infer what a query was.

## 2.2. In the servers

### 2.2.1. In the resolvers

It does not seem there is a possible solution against a leaky resolver. A resolver has to see the entire DNS traffic in clear.



The best approach to limit the problem is to have local resolvers whose caching will limit the leak. Local networks should have a local caching resolver (even if it forwards the unanswered questions to a forwarder) and individual laptops can have their very own resolver, too.

One mechanism to potentially mitigate on the wire attacks between stub resolvers and caching resolvers is to determine if the network location of the caching resolver can be moved closer to the end user's computer (reducing the attack surface). As noted earlier in [I-D.bortzmeyer-dnsop-dns-privacy], if an end user's computer is configured with a caching resolver on the edge of the local network, an attacker would need to gain access to that local network in order to successfully execute an on the wire attack against the stub resolver. On the other hand, if the end user's computer is configured to use a public DNS service as the caching resolver, the attacker needs to simply get in the network path between the end user and the public DNS server and so there is a much greater opportunity for a successful attack. Configuring a caching resolver closer to the end user can also reduce the possibility of on the wire attacks.

#### 2.2.2. In the authoritative name servers

A possible solution would be to minimize the amount of data sent from the resolver. When a resolver receives the query "What is the AAAA record for www.example.com?", it sends to the root (assuming a cold resolver, whose cache is empty) the very same question. Sending "What are the NS records for .com?" would be sufficient (since it will be the answer from the root anyway). To do so would be compatible with the current DNS system and therefore could be deployable, since it is an unilateral change to the resolvers.

To do so, the resolver needs to know the zone cut [RFC2181]. There is not a zone cut at every label boundary. If we take the name www.foo.bar.example, it is possible that there is a zone cut between "foo" and "bar" but not between "bar" and "example". So, assuming the resolver already knows the name servers of .example, when it receives the query "What is the AAAA record of www.foo.bar.example", it does not always know if the request should be sent to the name servers of bar.example or to those of example. [RFC2181] suggests an algorithm to find the zone cut, so resolvers may try it.

Note that DNSSEC-validating resolvers already have access to this information, since they have to find the zone cut (the DNSKEY record set is just below, the DS record set just above).

It can be noted that minimizing the amount of data sent also partially addresses the case of a wire sniffer.

One should note that the behaviour suggested here (minimizing the amount of data sent in qnames) is NOT forbidden by the [RFC1034] (section 5.3.3) or [RFC1035] (section 7.2). Sending the full qname to the authoritative name server is a tradition, not a protocol requirement.

Another note is that the answer to the NS query, unlike the referral sent when the question is a full qname, is in the Answer section, not in the Authoritative section. It has probably no practical consequences.

#### 2.2.3. Rogue servers

Traditional security measures (do not let malware change the system configuration) are of course a must. A protection against rogue servers announced by DHCP could be to have a local resolver, and to always use it, ignoring DHCP.

### 3. Security considerations

Hey, man, the entire document is about security!

### 4. Acknowledgments

Thanks to Olaf Kolkman and Francis Dupont for the interesting discussions, specially about qname minimization. Thanks to Mohsen Souissi for proofreading.

### 5. References

#### 5.1. 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.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.
- [I-D.bortzmeyer-dnsop-dns-privacy]

Bortzmeyer, S., "DNS privacy problem statement", draft-bortzmeyer-dnsop-dns-privacy-00 (work in progress), November 2013.

## 5.2. Informative References

- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.
- [I-D.koch-perpass-dns-confidentiality]  
Koch, P., "Confidentiality Aspects of DNS Data, Publication, and Resolution", draft-koch-perpass-dns-confidentiality-00 (work in progress), November 2013.
- [I-D.vandergaast-edns-client-subnet]  
Contavalli, C., Gaast, W., Leach, S., and E. Lewis, "Client Subnet in DNS Requests", draft-vandergaast-edns-client-subnet-02 (work in progress), July 2013.
- [I-D.wijngaards-dnsop-confidentialdns]  
Wijngaards, W., "Confidential DNS", draft-wijngaards-dnsop-confidentialdns-00 (work in progress), November 2013.
- [dnsop] IETF, , "The dnsop mailing list", October 2013.
- [dagon-malware]  
Dagon, D., "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority", 2007.
- [dns-footprint]  
Stoner, E., "DNS footprint of malware", October 2010.
- [darkreading-dns]

Lemos, R., "Got Malware? Three Signs Revealed In DNS Traffic", May 2013.

[dnsschanger]

Wikipedia, , "DNSChanger", November 2011.

[dnscrypt]

Denis, F., "DNSEncrypt", .

[dnscurve]

Bernstein, D., "DNSCurve", .

[prism]

NSA, , "PRISM", 2007.

[crime]

Rizzo, J. and T. Dong, "The CRIME attack against TLS", 2012.

[ditl]

, "A Day in the Life of the Internet (DITL)", 2002.

[data-protection-directive]

, "European directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data", November 1995.

[passive-dns]

Weimer, F., "Passive DNS Replication", April 2005.

[tor-leak]

, "DNS leaks in Tor", 2013.

#### Author's Address

Stephane Bortzmeyer  
AFNIC  
Immeuble International  
Saint-Quentin-en-Yvelines 78181  
France

Phone: +33 1 39 30 83 46  
Email: bortzmeyer+ietf@nic.fr  
URI: <http://www.afnic.fr/>

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 5, 2015

Z. Hu  
L. Zhu  
J. Heidemann  
USC/Information Sciences  
Institute  
A. Mankin  
D. Wessels  
Verisign Labs  
July 4, 2014

Starting TLS over DNS  
draft-hzhwm-start-tls-for-dns-01

Abstract

This document describes a technique for upgrading a DNS TCP connection to use Transport Layer Security (TLS) over standard ports. Encryption provided by DNS-over-TLS eliminates opportunities for eavesdropping of DNS queries in the network. The proposed mechanism is backwards compatible with clients and servers that are not aware of DNS-over-TLS.

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 January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect 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

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 ([draft-bortzmeyer-dnsop-dns-privacy]).

Prior work has addressed some aspects of DNS security, but none addresses privacy between a DNS client and server using standard protocols. 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 however does nothing to protect request or 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.

More recently, DNSCurve [draft-dempsey-dnscurve] defines a method to provide link-level confidentiality and integrity between DNS clients and servers. However, it does so with a new cryptographic protocol and so does not take advantage of TLS. ConfidentialDNS [draft-wijnngaards-confidentialdns] and IPSECA [draft-osterweil-dane-ipsec] use opportunistic encryption to provide privacy for DNS queries and responses. However, it is unclear how a client can locate an RR specific to its first-hop resolver. Finally, others have suggested DNS-over-TLS. Recent work suggests DNS-over-TLS ([draft-bortzmeyer-dnsop-privacy-sol]), and the Unbound DNS software [unbound] includes a DNS-over-TLS implementation. However, neither defines methods to negotiate TLS use over an existing connection; unbound instead requires DNS-over-TLS to run on a different port.

The mechanism described in this document enables DNS clients and servers to upgrade an existing DNS-over-TCP connection to a DNS-over-TLS connection. It is analogous to STARTTLS [RFC2595] used in SMTP [RFC3207], IMAP [RFC3501] and POP [RFC1939].

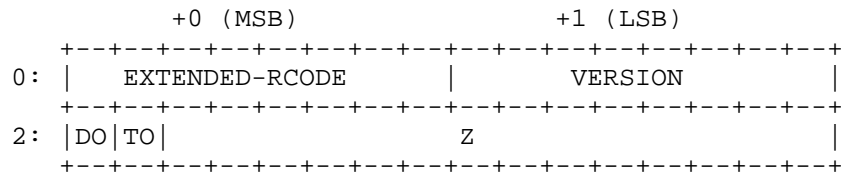
This document defines only the protocol extensions necessary to support TLS negotiation. It does not describe how DNS clients might validate server certificates or specify trusted certificate authorities. Solutions for certificate authentication are outside the scope of this document.

### 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

Clients and servers indicate their support for, and desire to use, 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]:



### 2.1. Use by DNS Clients

#### 2.1.1. Sending Queries

DNS clients MAY set the TO bit in queries sent using UDP transport to signal their general ability to support DNS-over-TLS. Clients which get no response to UDP TO=1 queries SHOULD retransmit them without the TO bit set.

DNS clients MAY 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 MUST NOT set the TO bit on subsequent queries when using TCP or TLS transport (to avoid ambiguity).

Since the motivation for DNS-over-TLS is to preserve privacy, DNS clients SHOULD use a 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.

#### 2.1.2. Receiving Responses

A DNS client that receives a response using UDP transport that has the TO bit set MUST handle 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 receives a response to its initial query using TCP transport that has the TO bit set MUST immediately initiate a TLS handshake using the procedure described in [RFC5246].

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 DNS-over-TLS (including TLS handshake failures) and SHOULD NOT request DNS-over-TLS from them for reasonable period. (We suggest 1 hour, or when the client discovers a new resolver.)

### 2.2. Use by DNS Servers

#### 2.2.1. Receiving Queries

A DNS server receiving a query over UDP MUST ignore the TO bit.

A DNS server receiving a query over an existing TLS connection MUST ignore the TO 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 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.

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.

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 DNS-over-TLS through a middlebox, or in the presence of a downgrade attack, could have one of the following outcomes (as discussed in prior RFCs [RFC3207]):

1. 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.
2. The DNS client sends a TO=1 query and receives a TO=1 response, but the TLS handshake fails because the server's certificate cannot be authenticated. In this case the client SHOULD close the established connection and fall back to unencrypted DNS for a reasonable period (as discussed in Section 2.1.2).
3. 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 2.1.2).
4. 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 2.1.2).

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. If the problem of middleboxes and threat of downgrade attacks is too serious, the IETF might consider allocating a dedicated port for DNS-over-TLS [RFC6335].

### 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.

#### 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 of opportunistic TLS (connections without CA validation, [draft-hoffman-uta-opportunistic-tls]) may be relevant to DNS-over-TLS.

## 6. Acknowledgments

We would like to thank Stephane Bortzmeyer, Brian Haberman, Paul Hoffman, Kim-Minh Kaplan, Bill Manning, George Michaelson, Eric Osterweil and Glen Wiley for reviewing this Internet-draft, and to 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.
- [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>>.
- [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.
- [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.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.
- [certificate\_pinning]  
OWASP, "Certificate and Public Key Pinning", <[https://www.owasp.org/index.php/Certificate\\_and\\_Public\\_Key\\_Pinning](https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning)>.
- [draft-bortzmeyer-dnsop-dns-privacy]  
Bortzmeyer, S., "DNS Privacy issues", draft-bortzmeyer-dnsop-dns-privacy-01 (work in progress), November 2013, <<http://tools.ietf.org/html/draft-bortzmeyer-dnsop-dns-privacy-01>>.
- [draft-bortzmeyer-dnsop-privacy-sol]  
Bortzmeyer, S., "Solutions to DNS privacy issues", draft-bortzmeyer-dnsop-privacy-sol-00 (work in progress), December 2013, <<http://tools.ietf.org/html/draft-bortzmeyer-dnsop-privacy-sol-00>>.
- [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-hoffman-uta-opportunistic-tls]  
Hoffman, P., "Opportunistic Encryption Using TLS", draft-hoffman-uta-opportunistic-tls-00 (work in progress), February 2014, <<http://tools.ietf.org/html/draft-hoffman-uta-opportunistic-tls-00>>.
- [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-wijnngaards-confidentialdns]  
Wijnngaards, W., "Confidential DNS", draft-wijnngaards-dnsop-confidentialdns-00 (work in progress), November 2013, <<http://tools.ietf.org/html/draft-wijnngaards-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](mailto: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





Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: May 16, 2014

P. Koch  
DENIC eG  
November 12, 2013

Confidentiality Aspects of DNS Data, Publication, and Resolution  
draft-koch-perpass-dns-confidentiality-00

Abstract

This document describes aspects of DNS data confidentiality in the light of recent IETF discussions on pervasive monitoring. It focuses on potential information leaks rather than prescribing methods of mitigation.

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 May 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect 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 Domain Name System (DNS) [RFC1034] [RFC1035] is the Internet's primary name lookup system. It consists of a publication aspect, represented by authoritative name servers providing access to DNS data covering parts of the DNS tree in units of zones, and a resolution aspect. The latter consists of applications that initiate DNS requests, DNS stub resolvers and DNS full resolvers (sometimes also called recursive resolvers or recursive name servers). Resolvers might be chained using a forwarding mechanism. In today's reality, there is a variety of intercepting DNS proxies and other middle boxes which are currently out of scope but may be addressed in future versions of this memo.

Threats to the DNS are described in [RFC3833] and have been addressed by DNSSEC [RFC4033] [RFC4034] [RFC4035], both to the extent that data origin authentication is concerned. Confidentiality was not a DNSSEC design goal, although in subsequent discussion that eventually led to the specification and deployment of NSEC3 [RFC5155], confidentiality of zone content was a major issue.

### 1.1. The alleged public nature of DNS data

It has long been claimed that "the data in the DNS is public". While this sentence makes sense for an Internet wide lookup system, there are multiple facets to data and meta data that deserve a more detailed look. First, access control lists and private name spaces notwithstanding, the DNS operates under the assumption that public facing authoritative name servers will respond to "usual" DNS queries for any zone they are authoritative for without further authentication or authorization of the client (resolver). A DNS query consists of QNAME, QCLASS and QTYPE. Due to the lack of search capabilities, only a given QNAME will reveal the resource records associated with that name (or that name's non existence). In other words: one needs to know what to ask for to receive a response. The zone transfer QTYPE [RFC5936] is often blocked or restricted to authenticated/authorized access to enforce this difference (and maybe for other, more dubious reasons).

Another differentiation to be applied is between the DNS data as mentioned above and a particular transaction, most prominently but not limited to a DNS name lookup. The fact that the results of a DNS query are public within the boundaries described in the previous paragraph and therefore might have no confidentiality requirements does not imply the same for a single or a sequence of transactions. Any transaction has meta data associated with the query data, e.g., a source address and a timestamp.

## 1.2. Disclaimer

The practices listed in this document appear only to support an informed discussion. Their presence (or absence) does not imply any form of support, engagement, applicability, appropriateness, fitness, or stance on legal status.

## 2. DNS Element walk through

This section will address the specific confidentiality issues of various elements of the DNS ecosystem. We will start at the authoritative servers, leaving the provisioning side out of scope, cover the resolution and recursive resolvers and finally address DNS queries at large and packet capturing.

### 2.1. Authoritative Name Servers

DNS zone data is published by authoritative name servers. Starting at the primary master, zone data is transferred in full (AXFR) or increments (IXFR) to secondary servers along the XFR dependency graph. The zone data thereby is inevitably revealed to any of the authoritative servers. Some zones, including the DNS root zone, are deliberately published by methods other than DNS AXFR.

While client as well as server authentication and data integrity are usually achieved by TSIG [RFC2845], there is no DNS protocol feature that provides zone transfer confidentiality. However, VPNs or other private arrangements are occasionally used. [RFC2182] is the most recent IETF document potentially dealing with this issue.

### 2.2. DNS Name Resolution

Since the communication between an application and the local resolver or between the local (stub) resolver and a full recursive resolver is rarely authenticated, DNS queries can and have been redirected. This has mostly been done with the malicious intent to inject forged responses, but could also be used as a man-in-the-middle (MITM) attack to learn a particular system's DNS queries and the response content.

The same queries (and responses) could be captured on the wire, even on the way to (and from) the correct, intended full resolver. Usually it has been assumed that the DNS resolution would not add additional intelligence given that subsequent communication would most likely reveal more than the DNS lookup. However, with recent suggestions to encrypt, say, web (HTTP) and mail (SMTP) connections, the DNS information could be of increased interest, disclosing otherwise unavailable information.

Operators of recursive resolvers could collect and examine queries directed to their systems.

The content of resolvers can reveal data about the clients using it. This information can sometimes be examined by sending DNS queries with RD=0 to inspect cache content, particularly looking at the DNS TTLs. Since this also is a reconnaissance technique for subsequent cache poisoning attacks, some counter measures have already been developed and deployed.

### 2.3. DNS Queries

DNS queries are initiated by an application handed over to a stub resolver, sometimes involving a host dependent name caching mechanism that is out of scope of this document. They consist of a QNAME, QCLASS and QTYPE, a DNS query ID and other parameters at the IP or transport layer. Among those are an IP source address, an IP ID and a source port number [RFC5452]. While some of these parameters have received increased attention due to their significance for DNS response spoofing mitigation, they do not contribute to confidentiality and may in fact deliver additional intelligence by supporting correlation of multiple queries from one system or even a single process or application at the same source. This is sometimes used in resolver software fingerprinting or behavioural analysis.

The source address in a DNS query is necessary to direct the response, but it may help to identify the requesting entity, be that a system, a process or an end user. For recursive resolvers it is sometimes argued that the size of the population 'behind' that resolver contributes to the noise. However, a private extension [I-D.vandergaast-edns-client-subnet] exists that will disclose the source address, or some prefix of the source address to the receiver, usually an authoritative name server.

The QNAME itself will be an existing or a non existing domain name. With reference to the earlier discussion of the public (or not) nature of DNS data, the response may reveal information. More importantly, due to the use of search paths [RFC1535] the QNAME may also disclose information relative to the querying entity:

```
_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs.example.org.
```

For parts of the domain name tree that more deeply enjoy the hierarchic nature of the DNS, like the IPv6 reverse delegation [RFC3596] or ENUM [RFC6116], the query name itself, asked for at a particular time, may disclose related, either ongoing or subsequent communication. This is partly due to the fact that the DNS treats the QNAME in full all the time.

Attempts have been made to encrypt the resource record RDATA [I-D.timms-encrypt-naptr].

## 2.4. DNS Packet Capturing

Both ephemeral and long term DNS captures have become DNS operational practice [DITL1] [DITL2]. Taking these packet traces usually occurs close to the authoritative servers, packets being captured on the wire, but under the control of the endpoint operator.

Initially designed to reconstruct DNS zone content from query response data, passive DNS [FW2005] has evolved into a widely used tool. These traces are usually sourced by on the wire traffic between recursive resolver and authoritative server.

## 3. Security Considerations

This document does not define a new protocol. It deals with confidentiality issues of the current DNS protocol and operations.

## 4. IANA Considerations

This document does not propose any new IANA registry nor does it ask for any allocation from an existing IANA registry.

## 5. Acknowledgements

This document was inspired by discussion with Wouter Wijngaards and Alexander Mayrhofer. Stephane Bortzmeyer and Nathalie Boulvard raised the issue of packet captures at a CENTR workshop. Jonathan Spring triggered some thoughts on the same topic.

## 6. References

### 6.1. 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.

- [RFC2182] Elz, R., Bush, R., Bradner, S., and M. Patton, "Selection and Operation of Secondary DNS Servers", BCP 16, RFC 2182, July 1997.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.
- [RFC3833] Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.
- [RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

## 6.2. Informative References

- [DITL1] CAIDA, "A Day in the Life of the Internet (DITL)", 2011.
- [DITL2] DNS-OARC, "DITL Traces and Analysis", 2013.
- [FW2005] Weimer, F., "Passive DNS Replication", FIRST 17, April 2005.
- [I-D.timms-encrypt-naptr] Timms, B., Reid, J., and J. Schlyter, "IANA Registration for Encrypted ENUM", draft-timms-encrypt-naptr-01 (work in progress), July 2008.
- [I-D.vandergaast-edns-client-subnet] Contavalli, C., Gaast, W., Leach, S., and E. Lewis, "Client Subnet in DNS Requests", draft-vandergaast-edns-client-subnet-02 (work in progress), July 2013.
- [RFC1535] Gavron, E., "A Security Problem and Proposed Correction With Widely Deployed DNS Software", RFC 1535, October 1993.

- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, January 2009.
- [RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, March 2011.

#### Appendix A. Document Revision History

This section is to be removed should the draft be published.

\$Id: draft-koch-perpass-dns-confidentiality.xml,v 1.1 2013/11/12  
09:29:48 pk Exp \$

##### A.1. Initial Document

First draft

##### Author's Address

Peter Koch  
DENIC eG  
Kaiserstrasse 75-77  
Frankfurt 60329  
DE

Phone: +49 69 27235 0  
Email: pk@DENIC.DE



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.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

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.

- o 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.
- o 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.
- o 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.
- o 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](mailto:wouter@nlnetlabs.nl)

Glen Wiley  
VeriSign, Inc.  
Reston, VA  
USA

EMail: [gwiley@verisign.com](mailto:gwiley@verisign.com)

