Short Paper: On Deployment of DNS-based Security Enhancements

Pawel Szalachowski and Adrian Perrig

ETH Zurich, Switzerland

Abstract. Although the Domain Name System (DNS) was designed as a naming system, its features have made it appealing to repurpose it for the deployment of novel systems. One important class of such systems are security enhancements, and this work sheds light on their deployment. We show the characteristics of these solutions and measure reliability of DNS in these applications. We investigate the compatibility of these solutions with the Tor network, signal necessary changes, and report on surprising drawbacks in Tor's DNS resolution.

1 Introduction

DNS is one of the most successful Internet infrastructures. It is a naming system for resources over the Internet, and its most prominent use is to translate human-readable names to IP addresses. Currently, this hierarchical and distributed system is a core infrastructure of the Internet, and over the years the availability and reliability of standard DNS operations have increased [17]. Although DNS is primarily (and was designed as) a system for name resolution, due to its success and flexibility it is used by various, not initially intended, applications. One family of such applications are various security enhancements. These systems are particularly difficult to deploy [16], as different actors are reluctant to deploy and invest in a security-dedicated infrastructure. Due to low cost, well-understood operations and administration, and its ubiquity, DNS seems like an ideal environment to support deployment of new security enhancements. Thus, it is naturally appealing to protocol designers to repurpose the DNS infrastructure, rather than designing and deploying a new one. For those reasons, DNS is currently being employed by various security enhancements. As a consequence, new systems rely on the infrastructure designed decades ago. Therefore, it is necessary to investigate how robust and applicable the infrastructure is for these use cases. The essence of the new uses is to transport additional information using DNS, however, there exist indications that such a transport can be unreliable.

In this work we make the following contributions: ***1***) investigate the use of DNS-based security enhancements, ***2***) study DNS reliability for these applications, ***3***) check the compatibility of the enhancements if the DNS resolution occurs over Tor.

2 Background

DNS Resolution is a process of translating human-readable domain names to IP addresses. It is conducted through the DNS infrastructure, namely *DNS clients*, *resolvers*, and *servers*. To resolve a domain name, e.g., `www.a.com`, a client initiates the process by querying its resolver, which in turn contacts one of the *DNS root servers* (root

servers' IP addresses are fixed and known to resolvers). The root server returns an address of a *DNS authoritative server* for the `com` domain. Then, the resolver queries the `com` authoritative server to find an authoritative server for `a.com`, which finally is queried about `www.a.com`. The `a.com` authoritative server returns the IP address(es) of `www.a.com`. The lengthy resolution process is usually shortcut by using cached information.

DNS allows to associate various information with domain names. Information is encoded and delivered within *resource records* (RRs) with dedicated types, e.g., `A` and `AAAA` RRs map domain names to IPv4 and IPv6 addresses, respectively, `NS` RRs indicate authoritative servers, while `TXT` RRs can associate an arbitrary text. DNS responses can contain multiple RRs of the queried type. It is also possible to translate IP addresses into domain names (to this end `PTR` RRs are used).

DNS deploys UDP as a default transport protocol, however, for responses larger than 512 bytes a *failover* mechanism is introduced. Larger responses are truncated to fit 512 bytes and marked by a `truncated` flag. Resolvers receiving a truncated response query the server again via TCP to obtain the complete response. (Clients can increase the limit by signaling the maximum UDP response size they can handle [18].)

DNS Resolution (Un)Reliability. Although DNS is reliable for its major application (i.e., translating names to IP addresses), the reliability for other applications is questionable. For instance, many of DNS clients, resolvers, and servers are realized as non-compliant implementations [3]. It was reported [9] that a significant fraction of all clients (2.6%) and a large fraction of resolvers (17%) cannot perform the UDP-to-TCP failover. This behavior limits clients ability to receive responses larger than 512 bytes. Another potential issue [3] is caused by network environments, where devices can handle only unusually small Maximum Transmission Unit (MTU) packets, thus introducing IP fragmentation decreasing the reliability of the DNS resolution. DNS traffic is also a subject to traffic analysis, and some middleboxes manipulate DNS responses [7, 19]. It is believed that some non-standard RRs are discriminated by non-compliant implementations or/and network devices. For instance, some experiments [12, 19] indicate that `A` RRs are more reliable than `TXT` RRs.

3 Security Enhancements Employing DNS

We focus our study on two families of security enhancements that can benefit from a robust DNS infrastructure, namely email and TLS PKI enhancements. The main reason why DNS infrastructure can be appealing for these technologies is that both email and TLS PKI are domain based. As the DNS lookup usually precedes the email exchange or TLS connection establishment, the client can obtain some relevant information before the connection setup. Additionally, such DNS-based information pre-fetching does not violate the privacy, as no additional third party is contacted (DNS servers are contacted anyways). A security assumption for these schemes is that an adversary cannot control DNS entries of targeted domains.

3.1 Email

SPF [10] enables domains to make assertions (in DNS) about hosts that are authorized to originate email for that domain. When an email is received by an email exchanger, it parses the domain name from the email's `From` address field, and queries the DNS to

check whether the sender is authorized to send email. This mitigates spam and phishing emails that abuse the `From` field. SPF mainly uses `TXT` RRs, although a dedicated `SPF` RR was introduced.

Sender ID [1] is an anti-spoofing proposal based on SPF. The main difference is that it aims in verifying the sender address displayed to an email client (the `From` field and the address displayed by email clients can differ). Such an address is introduced as a Purported Responsible Address (PRA) [15]. By setting a special `TXT` or `SPF` record, a domain can specify if only SPF should be verified, or both SPF and PRA, or PRA only.

DKIM [4] is an email authentication protocol based on signatures. A domain publishes RR with its public key. Next, the domain's outbound email server signs sent emails with the corresponding private key. An inbound email server, after receiving a signed email, extracts its origin domain name (via the `From` field) and performs a DNS lookup to obtain the domain's public key used to verify the email. Usually, DKIM is executed by email servers rather than email clients (i.e., authors and recipients). Public keys are stored in `TXT` RRs, and to obtain a key of `a.com`, `_dkim.a.com` is queried. DKIM protects emails from modification, however, the scheme can be bypassed by an active adversary by simply stripping the DKIM headers.

DMARC [11] is a comprehensive system that allows an email-originating organization to express domain-level policies for email management. A policy can specify how emails should be validated and how receivers should handle validation failures. Additionally, DMARC policies can be used to implement a reporting system (i.e., to report on actions performed under a policy). DMARC deploys SPF and DKIM, and domain owners can specify which of those mechanisms (or both) should be used to validate their emails. DMARC uses `TXT` RRs to store policies, and the RRs are associated with domain names prepended with the `_dmarc.` prefix, e.g., `_dmarc.a.com`.

3.2 TLS PKI Enhancement

DANE [8] allows domains to specify their key(s) or key(s) of Certificate Authorities (CAs) they trust. To this end, a domain publishes a special DNS entry with its public key(s) or public key(s) of trusted CA(s). DANE introduces a new `TLSA` RR. The scheme relies on DNSSEC, requiring that the RRs be signed with the domain's DNSSEC key. DANE records are created per service, thus a DANE query encodes a transport protocol, and a port number used. For instance, keys of a HTTPS server running at `www.a.com` can be checked by querying `_443._tcp.www.a.com`. Such a flexible mechanism allows to use DANE for all services that deploy TLS.

CAA [6] aims to provide trust agility and remove a single point of failure from the TLS PKI. Specifically, it allows a domain to specify (in DNS) CA(s) authorized to issue certificates for the domain. This simple procedure can prevent the two following threats: (i) compromised CA: a CA that is not listed by a domain cannot issue a valid certificate for the domain, (ii) identity spoofing: a benign CA can refuse certificate issuance if it is not listed by the domain. CAA introduces new `CAA` RRs, which do not have to be protected via DNSSEC, although it is recommended.

Log-based approaches are recent PKI enhancements that introduce publicly-verifiable logs. The most prominent example is CT [13], whose goal is to make all certificates issued by CAs visible. To this end, every certificate is submitted to a log, which returns a signed *promise* that the certificate will be logged. Then, in every TLS connection

a client receives a certificate accompanied with the logging promise. However, it is important to verify whether the promise was met, and to do so the client has to obtain a proof from the log that given certificate indeed was logged. Laurie et al. propose [14] that clients ask a special CT-supported DNS server for such a proof. An advantage of this scheme is that DNS requests are sent via a local resolver, thus the CT DNS server (and the log) cannot identify the client, but only his resolver (usually run by his ISP).

4 Current State of Deployment

First, we investigate deployment characteristics of the enhancements. In particular, we focus on factors that can influence reliability of DNS as a transport (i.e., RRs used and response sizes). To this end, we conduct a measurement of the hundred thousand most popular domains of the Internet (according to the Alexa list: `http://www.alexa.com/topsites`). For each domain name we queried for RRs that implement a given functionality. We queried for DANE's RRs specific to HTTPS, i.e., `_443._tcp.`, and `_443._tcp.www.` prepended to a queried domain name. We omitted log-based mechanism, as no scheme is combined with DNS yet (up to our knowledge).

Mechanism	RR(s) Queried	Successful Responses	Response Size (B) min	med	avg	max
SPF	TXT	53365 (53.37%)	25	148	185	3138
	SPF	4182 (4.18%)	27	122	144	1606
Sender ID	TXT	1766 (1.77%)	56	303	333	1285
	SPF	98 (0.10%)	79	234	247	538
DKIM	TXT	5049 (5.05%)	49	64	97	1007
DMARC	TXT	7361 (7.36%)	35	133	140	1003
DANE	TLSA	48 (0.05%)	80	88	96	182
CAA	CAA	15 (0.02%)	58	106	106	269

Table 1: Measured scale of deployment and response sizes.

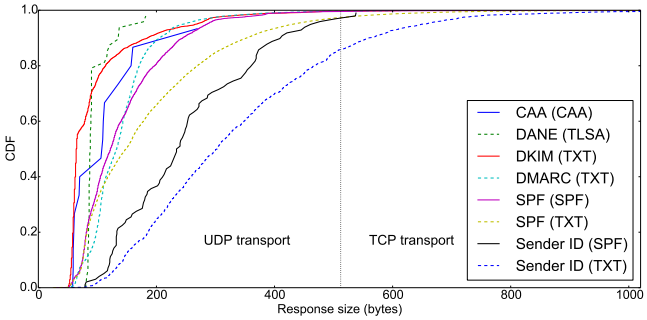

Fig. 1: CDF of the measured response sizes.

Table 1 presents the measured scale of deployment with the response size characteristics, while Figure 1 presents a CDF of the measured response sizes. As depicted, `TXT` RRs dominate, constituting about 94% of all successful responses. It is mainly due to well-established deployment of the mail enhancements (SPF mainly). Although, new RR types (like `SPF`) were introduced, the operators clearly prefer to rely on older `TXT` RRs. PKI enhancements do not have significant deployment, which is probably caused by their relative immaturity (e.g., SPF was introduced in 2006, while DANE and CAA

in 2012 and 2013, respectively). Another finding is that most of the responses fit the limit of 512 bytes. An exception are responses including Sender ID's data (approximately 15% of them exceed the limit).

5 Reliability of DNS

To investigate how reliable DNS is for the security enhancements, we conducted a series of experiments using RIPE Atlas (`https://atlas.ripe.net/`), the largest publicly available global testbed for network measurements. RIPE Atlas is a network of hardware devices, called *probes*, used for active Internet measurements. It supports DNS measurements, and provides good geographic coverage [2]. Through the measurements we wanted to answer the two following questions:

1. Are `TXT` RRs discriminated (dropped or manipulated) by some DNS clients/resolvers or network devices?
2. How reliable is DNS in transporting UDP responses larger than 512 bytes?

The first question is motivated by the importance of TXT RRs (see §3) and by the common belief that a significant fraction of `TXT` RRs is not transported correctly (probably due to its non-standard type). We investigate the second question to verify how the 512 bytes limit for UDP DNS responses is enforced by the DNS infrastructure. This question is important as the previous work indicates that the TCP support at DNS resolvers is incomplete [9], thus it is risky to rely on the failover mechanism. (Note, that RIPE Atlas does not expose an option to check whether a probe's DNS client/resolver correctly handles responses with the `truncated` flag set.)

In order to conduct the measurements, we launched an authoritative DNS server, and prepared it with DNS responses of the following sizes:

494 bytes : the size is below the 512 bytes limit, but it can handle most of the current responses (see Figure 1). We investigated transport over `A` and `TXT` RRs, to verify whether `TXT` RRs are discriminated (while compared to `A` RRs).

1005 bytes : responses with this size allow us to investigate how robust the DNS infrastructure is, when the UDP response size limit is exceeded. This size is also below the standard MTUs (i.e., about 1500 bytes).

1997 bytes : by responses with this size, we want to investigate how exceeding the standard MTU influences DNS transport.

Our DNS server was configured not to set the `truncated` flag, and in the RIPE Atlas setting we set the acceptable response size to 4096 bytes. We scheduled measurements on the RIPE Atlas at the end of August 2016. We assigned all 9270 connected probes to query our DNS server. For response sizes of 1005 and 1997 bytes we investigated only `TXT` RRs. Depending on the queried target, the following number of probes have responded: 8952 for queried `A` and `TXT` RRs sent in 494 bytes responses, 8934 for 1005 bytes responses, and 7990 for 1997 bytes responses. Note, that each probe could respond with multiple DNS responses.

In Table 2 we present the obtained results. As probes can use the same, popular resolvers, beside the absolute number of responses, we also present results for unique resolutions, where a unique resolution is defined as a triple: number of RRs within a response, response size, and resolver's address. The successful results are divided into responses that were received with the exact size served (by the authoritative DNS server), and larger responses (resolvers add other information that is relevant to the

			Successful Resolutions			Failed Resolutions			
	Test	**Total**	**Total**	**Exact**	**Larger**	**Total**	**Error**	**Empty**	**Truncated**
All Responses	A	16570	15468	15356	112	1102	867	189	46
	494B	100%	93.35%	92.67%	0.68%	6.65%	5.23%	1.14%	0.28%
	TXT	16570	15460	15343	117	1110	892	206	12
	494B	100%	93.30%	92.60%	0.71%	6.70%	5.38%	1.24%	0.07%
	TXT	16553	13480	936	12544	3073	1504	1155	414
	1005B	100%	81.44%	5.65%	75.78%	18.56%	9.09%	6.98%	2.50%
	TXT	13727	7286	29	7257	6441	2360	3617	464
	1997B	100%	53.08%	0.21%	52.87%	46.92%	17.19%	26.35%	3.38%
Unique Responses	A	7452	6625	6526	99	827	633	166	28
	494B	100%	88.90%	87.57%	1.33%	11.10%	8.49%	2.23%	0.38%
	TXT	7447	6618	6516	102	829	638	181	10
	494B	100%	88.87%	87.50%	1.37%	11.13%	8.57%	2.43%	0.13%
	TXT	7938	6222	450	5772	1716	922	636	158
	1005B	100%	78.38%	5.67%	72.71%	21.62%	11.62%	8.01%	1.99%
	TXT	6887	3741	19	3722	3146	1252	1652	242
	1997B	100%	54.32%	0.28%	54.04%	45.68%	18.18%	23.99%	3.51%

Table 2: Measured reliability of DNS.

query, like addresses of authoritative servers). Failed resolutions are divided into three categories. First, the fraction of resolution errors is presented. These are errors such as a DNS resolver that could not be found, or a failed connection. Then, we present empty DNS responses (i.e., number of answers equals zero). The last category shows the number of truncated responses, i.e., responses with fewer number of RRs than expected or/and shorter payload of the response.

Our first observation is that for the 494 bytes long responses there is only a negligible difference between reliability of A-only responses versus TXT-only responses. Secondly, the results show that UDP responses with size above the 512 bytes limit increase the failure rate from 6.70% to 18.56% (all responses) and from 11.13% to 21.62% (unique responses). Taking into consideration the results about failing TCP support, it might be more effective to use UDP with increased size instead of TCP. Lastly, the largest responses investigated (1997 bytes) are successfully delivered only in about 50% of all cases. That is probably caused by MTU issues, as common MTUs over the Internet are about 1500 bytes. We also observe, that resolvers enlarge responses usually when they are large already.

Although RIPE Atlas is an ideal open testbed for such tests, it introduces some biases. Probes are plug-and-forget devices, thus an owner may be not aware that DNS resolution at his/her probe does not work properly (this could explain the large fraction of DNS errors even for the smallest responses investigated). Moreover, probes are usually installed by network-savvy users like research institutions, Internet operators, hobbyists, and the probe distribution (based on their ASes) is heavy-tailed [2].

6 Tor and Security Enhancements

Tor [5] is the most popular software and infrastructure for enabling anonymous communication over the Internet. It is an onion routing protocol, where an *encryption circuit* is selected by the Tor client software. DNS querying over Tor is also anonymous and conducted by an *exit node* of the circuit (this node will forward traffic to destinations).

The DNS resolution in Tor is restricted only to A, AAAA, and PTR RRs. This obviously limits the deployment of DNS-supported security enhancements in Tor. It is especially important for the PKI enhancements, as they assume clients to participate in the protocol (the mail enhancements are deployed mainly by the mail infrastructure).

In this section, we investigate whether the supported RRs can be used to implement DNS-supported enhancements (for instance, one could convey information on a series of A or AAAA RRs). We measured DNS resolution over Tor, using our authoritative server, that was also configured as a Tor Linux client (i.e., the server queried itself through the Tor network, as presented in Figure 2). For every set of queries, a new Tor circuit was selected, and we conducted 15000 such resolutions. We investigated how reliable Tor is in resolving requests for the supported RRs (i.e., A, AAAA, and PTR). We checked PTR queries for both, IPv4 and IPv6 addresses.

The first observation is that all asked resolvers limited DNS responses only to a single RR. This limits ways the supported RRs can be used to encode some additional data (e.g., single A query can return only four bytes). Table 3 presents the fraction of successfully resolved requests. As depicted, A queries are resolved slightly more reliably than PTR queries for IPv4 addresses, which in turn are less reliable for IPv6 addresses. The results also show, that although AAAA RRs are supported, they are resolved correctly only for 23% of requests (probably, only nodes supporting IPv6 resolve them).

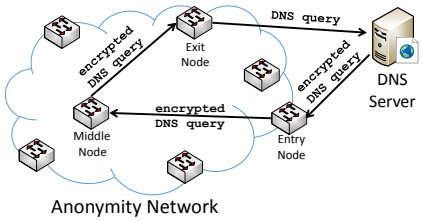

Fig. 2: Tor-based measurement scenario.

A	PTR (IPv4)	PTR (IPv6)	AAAA
99.78%	99.22%	98.89%	23.05%

Table 3: Fraction of successful resolutions (i.e., single RR returned) depending on type.

Surprisingly, we observed that some resolvers fail to return any response when the response from the authoritative server is large (but still below 512 bytes). To further investigate this phenomena, we prepared responses with A RRs with different sizes. We then measured when requests are processed successfully (by success we mean a response to the client that contains a single RR, although many were served). The results (see Table 4) show that reliability of DNS resolution decreases with the response size. Only 38% of all resolutions succeeded at all with 494 bytes long responses served.

7 Conclusions

Our study confirms that DNS-based security enhancements should respect the conservative limit of 512 bytes for responses, as robustness of DNS transport can be influenced by many uncontrollable factors. Fortunately, the limit is sufficient for about 95% of all

61B	**110B**	**158B**	**254B**	**366B**	**398B**	**430B**	**462B**	**478B**	**494B**
1 RRs	4 RRs	7 RRs	13 RRs	20 RRs	22 RRs	24 RRs	26 RRs	27 RRs	28 RRs
99.77%	99.77%	99.77%	99.77%	99.23%	99.16%	98.10%	92.87%	91.27%	38.36%

Table 4: Fraction of successful resolutions (i.e., single A RR returned) depending on the response size (from the authoritative server).

received responses. However, our study does not confirm the common belief that TXT RRs are being discriminated. Our work identifies DNS resolution in Tor as an interesting subject for future work, as we found it surprising and inconsistent: resolvers fail to return large responses, slightly differently handle PTR RRs for IPv4 and IPv6 addresses, AAAA RRs are officially supported, but in practice are resolved only by 23% of all resolvers. We also observe that restricting other RRs (especially PKI-related, like TLSA) will actually decrease security of end users. Hence, to fulfill Tor's mission (i.e., *"to allow people to improve their privacy and security on the Internet"*) the developers should consider supporting DNS-based security enhancements.

Acknowledgment

We gratefully acknowledge support from ETH Zurich and from the Zurich Information Security and Privacy Center (ZISC). We thank Brian Trammell and the anonymous reviewers, whose feedback helped to improve the paper.

References

1. E. Allman and H. Katz. SMTP Service Extension for Indicating the Responsible Submitter of an E-Mail Message. RFC 4405, 2006.
2. V. Bajpai, S. J. Eravuchira, and J. Schönwälder. Lessons learned from using the RIPE Atlas platform for measurement research. *SIGCOMM CCR*, 2015.
3. A. Buddhdev. Testing your Resolver for DNS Reply Size Issues. https://goo.gl/gU7mNu, 2009.
4. D. Crocker, T. Hansen, and M. Kucherawy. DomainKeys Identified Mail (DKIM) Signatures. RFC 6376, 2011.
5. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.
6. P. Hallam-Baker and R. Stradling. DNS Certification Authority Authorization (CAA) Resource Record. RFC 6844, 2013.
7. S. Hätönen, A. Nyrhinen, L. Eggert, S. Strowes, P. Sarolahti, and M. Kojo. An experimental study of home gateway characteristics. In *ACM IMC*, 2010.
8. P. Hoffman and J. Schlyter. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, 2012.
9. G. Huston. A Question of DNS Protocols. https://goo.gl/d8kwCK, 2013.
10. S. Kitterman. Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208, 2014.
11. M. Kucherawy and E. Zwicky. Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489, 2015.
12. A. Langley. Why not DANE in browsers. https://goo.gl/0kVppI, 2015.
13. B. Laurie, A. Langley, and E. Kasper. Certificate Transparency. RFC 6962, 2013.
14. B. Laurie, P. Phaneuf, and A. Eijdenberg. Certificate transparency over DNS. https://goo.gl/PoLkmu, 2016.
15. J. Lyon. Purported Responsible Address in E-Mail Messages. RFC 4407, 2006.
16. M. Nikkhah, C. Dovrolis, and R. Guérin. Why didn't my (great!) protocol get adopted? In *HotNets*, 2015.
17. V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. Impact of configuration errors on DNS robustness. In *SIGCOMM CCR*, 2004.
18. P. Vixie. Extension Mechanisms for DNS (EDNS0). RFC 2671, 1999.
19. N. Weaver, C. Kreibich, B. Nechaev, and V. Paxson. Implications of Netalyzrs DNS measurements. In *SATIN*, 2011.