Large-scale DNS and DNSSEC data sets for network security research

Roland van Rijswijk-Deij[1,2], Anna Sperotto[1], and Aiko Pras[1]

[1]Design and Analysis of Communication Systems (DACS), University of Twente, Enschede, The Netherlands
{r.m.vanrijswijk,a.sperotto,a.pras}@utwente.nl
[2]SURFnet bv, Utrecht, The Netherlands

Abstract

The Domain Name System protocol is often abused to perform denial-of-service attacks. These attacks, called DNS amplification, rely on two properties of the DNS. Firstly, DNS is vulnerable to source address spoofing because it relies on the asynchronous connectionless UDP protocol. Secondly, DNS queries are usually small whereas DNS responses may be much larger than the query. In recent years, the DNS has been extended to include security features based on public key cryptography. This extension, called DNSSEC, adds integrity and authenticity to the DNS and solves a serious vulnerability in the original protocol. A downside of DNSSEC is that it may further increase the potential DNS has for amplification attacks. This disadvantage is often cited by opponents of DNSSEC as a major reason not to deploy the protocol. Until recently, however, ground truth about how serious an issue this can be was never established. This technical report describes the data sets obtained during a study [1] we carried out to establish this ground truth. We make these data sets available as open data under a permissive Creative Commons license. We believe these data sets have a lot of value beyond our research. They, for example, allow characterisations of EDNS0 implementations, provide information on IPv6 deployment (presence or absence of `AAAA` records) for a large number of domains in separate TLDs, etc.

Keywords: DNS, DNSSEC, DDoS, amplification attack, reflection attack, measurements, denial-of-service, attack, network security

1 Introduction

The Domain Name System (DNS) protocol is a favourite among attackers to abuse for denial-of-service attacks. DNS is based on UDP and is thus susceptible to source address spoofing. This property enables the use of DNS in reflection attacks, where the attacker forges a request to a DNS server in which he puts a spoofed source address, the address of the victim. The DNS server will then send the response to the victim. And because, in general, DNS responses are larger than DNS queries, the attacker will also achieve what is called amplification. This means that an attacker can achieve a large attack volume while

only investing a small amount of attack traffic. Figure 1 shows this attack schematically.

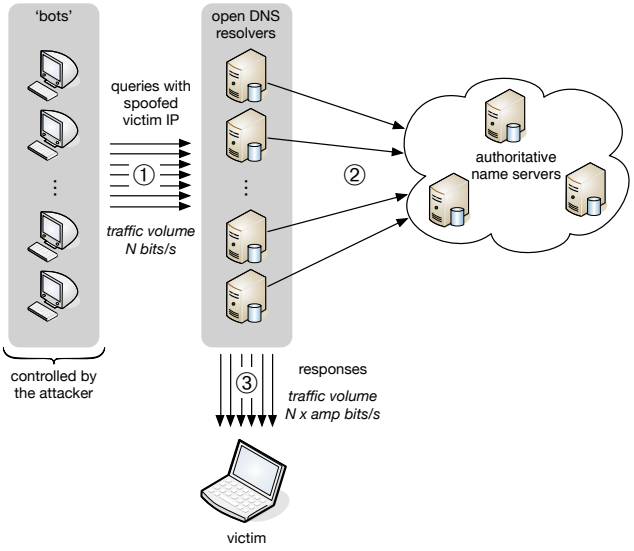


Figure 1: DNS amplification attack

Since a number of years a major overhaul of the DNS is underway, the introduction of the DNS Security Extensions (DNSSEC). DNSSEC enhances the security of the DNS by introducing authenticity and integrity into the protocol. This is achieved by digitally signing DNS data. This means that DNS answers that use DNSSEC are significantly larger than those that use regular DNS, potentially making DNSSEC a powerful vector for performing denial-of-service attacks. Opponents of DNSSEC frequently cite this property of the protocol as a reason not to deploy the technology. There had, however, never been a large scale study to assess the severity of this problem.

Because denial-of-service attacks are ever increasing in size and severity and because some of the largest attacks[1] rely on DNS amplification we felt the need to establish ground truth about the potential that DNSSEC has for abuse in amplification attacks [1]. What we found is that yes, as expected, DNSSEC does increase the potential for amplification attacks. However, this was only really the case for a certain kind of DNS query (so-called `ANY` queries). Since this query type is not a regular DNS query, but rather intended for debug purposes, simple and effective measures can be taken to dampen the attack potential. If we look at regular DNS queries (such as address, or `A`, queries, which are the most frequent DNS query) then the increase in amplification that DNSSEC introduces is far smaller.

In order to measure the impact of DNSSEC on DNS amplification we performed a large scale measurement of DNSSEC-signed domains in six major top-level domains (TLDs). Our data sets cover 2.5 Million DNSSEC-signed domains (approximately 70% of the total number of signed domains) and over

[1] e.g. the 'Spamhaus attack' of 2013, `http://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet`

2.2 Million regular domains, totaling over 54GB of data. We have decided to share the data we collected during our measurements as open data. To do this, we use the SimpleWeb platform[2] established by our colleagues[2]. This report provides an overview of the data we collected and gives detailed information about the structure of the databases containing our results that are available for download.

1.1 Intended Audience

This document is intended for researchers focusing on network security and stability in general and on DNS in particular. Readers are assumed to be familiar with the workings of the DNS protocol [3] and its extensions EDNS0 [4] and DNSSEC [5, 6, 7].

2 Data Sets

2.1 Source Data

We obtained data for six major TLDs. The type of data we obtained is specified in Table 1. As the table shows, we obtained the full DNS zone file for four TLDs and we obtained partial data for two other TLDs. In the latter case we obtained a list of all DNSSEC-signed domains and a random selection of non-DNSSEC domains. The most right-hand column in the table shows the percentage of domains with DNSSEC compared to the total number of domains in the TLD.

TLD	Kind of data	Date	#domains in TLD	#DNSSEC-signed	
.com	Full zone	March 2014	113.1M	326.5k	(0.3%)
.net	Full zone	March 2014	15.2M	69.5k	(0.5%)
.org	Full zone	April 2014	10.3M	37.6k	(0.4%)
.nl	Selection	April 2014	5.4M	1696.1k	(31.2%)
.se	Full zone	March 2014	1.4M	334.9k	(24.8%)
.uk	Selection	April 2014	10.6M	10.2k	(0.1%)
total			156.0M	2.5M	(1.6%)

Table 1: Overview of source data

2.2 Collected Data

For each TLD we collected two sets of data, for DNSSEC-signed domains and for non-DNSSEC domains. Data was collected as follows:

1. For each domain we determined the set of authoritative name servers.
2. For each name server we determined the set of IPv4 and IPv6 addresses.
3. Each IP address of every name server was sent a pre-defined set of queries, the results of which comprise the data set.

We sent queries using three different modes:

[2] `http://www.simpleweb.org/wiki/Traces`

(a) using 'classic' DNS [3];

(b) using EDNS0 [4] with the maximum accepted response size set to 32768[3];

(c) using EDNS0 with DNSSEC enabled (`DO=1`) [7], again with the maximum accepted response size set to 32768.

To each IP address of each authoritative name server we sent a set of queries using all three modes described above for DNSSEC-signed domains and using modes (a) and (b) for non-DNSSEC domains. To all authoritative name servers we sent the following queries:

- `ANY` query for the domain.
- `MX` query for the domain.
- `NS` query for the domain.
- `A` query for the domain apex and the `www` and `mail` names.
- `AAAA` query for the domain apex and the `www` and `mail` names.
- `TXT` query for the domain apex and the `www` and `mail` names.

Additionally, for DNSSEC-signed domains we also sent the following DNSSEC-specific queries:

- `DNSKEY` query for the domain.
- a query for a non-existent name to trigger an `NSEC3` response.

Prior to performing the queries, we perform a `DNSKEY` query through a resolver to determine whether a domain that is supposed to be signed actually is and vice versa. Domains that should have been signed and were not were omitted from the data set, and the same applies to domains that should not have been signed but were.

Every query is attempted only once and only successful queries were recorded in the data sets. For each successful query we recorded the following data in the database:

- the query size (UDP datagram size);
- the response size (UDP datagram size);
- the amplification factor defined as $\frac{response\ size}{query\ size}$;
- the EDNS0 maximum response size advertised by the authoritative name server;
- the value of the truncation (TC) flag;
- the number of answers in the response;

[3]We chose this value to also register results that exceed the commonly used maximum response size of 4KB; we decided not to use the maximum value (65535) since we did not want to risk running into possible boundary conditions in DNS software implementations.

- the number of authority records in the response;
- the number of additional records in the response;
- the number of distinct resource record types in the response.

Data collection took place over a period of approximately 5 weeks from March 11th until April 17th 2014. The final data sets, described in Table 2 (DNSSEC) and Table 3 (non-DNSSEC), contain the results of almost half a billion queries.

TLD	#domains	#failed	#skipped	#queried	#queries	#auth ns
.com	326504	7416	471	318576	54.6 M	2550
.net	69552	2672	55	66814	11.0 M	2476
.org	37621	555	19	37024	6.7 M	2073
.nl	1696103	12304	1002	1682770	233.3 M	1316
.se	334880	8696	100	326067	43.3 M	3681
.uk	10225	314	10	9894	1.6 M	570
total	2474885	31957	1657	2441145	350.5 M	n/a

Table 2: Overview of DNSSEC data sets

TLD	#domains	#failed	#skipped	#queried	#queries	#auth ns
.com	498502	55909	2231	436593	37.6 M	27168
.net	99564	13904	355	84882	7.4 M	26396
.org	100000	11031	277	88372	7.5 M	27761
.nl	1000000	69092	6812	921441	69.3 M	31108
.se	499999	37361	149560	311871	21.5 M	23756
.uk	26131	3883	92	21858	1.6 M	7091
total	2224196	191180	159327	1865017	144.9 M	n/a

Table 3: Overview of non-DNSSEC data sets

2.3 Database Scheme

2.3.1 General information

TLD	DNSSEC database	size	non-DNSSEC database	size
.com	`com.dnssec.db`	5.9GB	`com.non-dnssec.db`	4.2GB
.net	`net.dnssec.db`	1.2GB	`net.non-dnssec.db`	0.8GB
.org	`org.dnssec.db`	0.7GB	`org.non-dnssec.db`	0.8GB
.nl	`nl.dnssec.db`	26.0GB	`nl.non-dnssec.db`	7.6GB
.se	`se.dnssec.db`	4.7GB	`se.non-dnssec.db`	2.5GB
.uk	`uk.dnssec.dn`	0.2GB	`uk.non-dnssec.db`	0.2GB

Table 4: Database files

All databases are listed in Table 4 and are SQLite databases created using SQLite version 3.7.9. They should be compatible with any SQLite version from

3.7 and up, but may also work with earlier versions of SQLite 3 as no features specific to version 3.7 were used. Every database contains two types of tables described in the next two subsections below. In these descriptions, database columns that contain data that was anonymised are marked with `[A]`. The anonymisation applied is described in the final part of this section.

2.3.2 Domain information

Each database contains the following four tables with domain information:

- `DOMAINS` – source list of domain names for the TLD used as input for the measurement.
- `DOMAINS_DONE` – unused; originally intended to store the list of domains for which data was actually obtained. This table was abandoned due to a change in measurement strategy, but the data contained in it can easily be constructed using the SQL query specified in Listing 1, where `<RESULTS_TABLE>` needs to be replaced by one of the query result table names specified in the next subsection.
- `DOMAINS_FAILED` – this table lists the domains for which the measurement software failed to obtain the authoritative name server set and corresponds to the third column in Table 2 and Table 3.
- `DOMAINS_SKIPPED` – this table lists the domains that the measurement software skipped because they contained DNSSEC data whereas the measurement to be performed was for a non-DNSSEC domain or vice versa. This table corresponds to the fourth column in Table 2 and Table 3.

Each of the four tables listed above has the same schema outlined in Table 5.

Col#	Name	Type	Description
1	`id`	`INTEGER`	Unique identifier for the domain used in all domain information and result tables
2	`domain [A]`	`VARCHAR(255)`	The domain name
3	`dnssec`	`BOOLEAN`	Set to 0 if the domain is non-DNSSEC and set to 1 for DNSSEC-signed domains

Table 5: Domain information table schema

Listing 1: Querying for domains for which data was obtained

```
SELECT domain FROM DOMAINS WHERE id IN
        (SELECT distinct domain_id FROM <RESULTS_TABLE>);
```

2.3.3 Query results

For each query type measured (outlined in Section 2.2) there are three tables: one for plain DNS queries, one for EDNS0 queries and one for EDNS0 with DNSSEC enabled queries. This means that the following result tables are present in each database:

- A_RESPONSES_REGULAR
- A_RESPONSES_EDNS0_PLAIN
- A_RESPONSES_EDNS0_DNSSEC
- AAAA_RESPONSES_REGULAR
- AAAA_RESPONSES_EDNS0_PLAIN
- AAAA_RESPONSES_EDNS0_DNSSEC
- MX_RESPONSES_REGULAR
- MX_RESPONSES_EDNS0_PLAIN
- MX_RESPONSES_EDNS0_DNSSEC
- NS_RESPONSES_REGULAR
- NS_RESPONSES_EDNS0_PLAIN
- NS_RESPONSES_EDNS0_DNSSEC
- TXT_RESPONSES_REGULAR
- TXT_RESPONSES_EDNS0_PLAIN
- TXT_RESPONSES_EDNS0_DNSSEC
- ANY_RESPONSES_REGULAR
- ANY_RESPONSES_EDNS0_PLAIN
- ANY_RESPONSES_EDNS0_DNSSEC
- NSEC_RESPONSES_EDNS0_DNSSEC
- DNSKEY_RESPONSES_EDNS0_DNSSEC

Note that for DNSSEC-specific queries we did not measure plain DNS or EDNS0 without DNSSEC, hence these tables are not present.

Each result table has the same schema outlined in Table 6 on page 10 of this report.

2.3.4 Anonymisation

As was already mentioned in the introduction to this section, certain values in the data set have been anonymised. This applies to all domain names in the data sets as well as all IP addresses of authoritative name servers in the data set. These values were anonymised for several reasons:

- Anonymisation was a condition of the TLDs from which source data was obtained for publishing the data sets.
- Anonymisation prevents abuse of the data sets for creating lists of DNSSEC-signed domains that are especially suited for abuse in amplification attacks.
- The privacy of domain name holders is protected.
- The privacy of DNS operators is protected.

To anonymise the data, the following algorithm was applied:

```
anon_value = hash(secret-salt | actual_value)
```

The *SHA-256* hash algorithm [8] was used as the hashing function, a fresh random 512-bit salt generated using a cryptographically secure random number generator was used for each separate result database. This also prevents users of the data sets from combining the results from multiple sets to find authoritative name servers that serve zones in more than one TLD. The salt values were retained and are stored in a secure location such that we can identify individual domain names and authoritative name servers in the set upon the request of fellow researchers. Note that there is no efficient way to reverse anonymisation, thus we will consider such requests on a case-by-case basis and reserve the right to deny the request at our discretion.

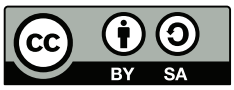
2.4 Collection Software

We have decided not to release the software used to collect the data sets into the public domain for the moment as we intend to use this software for a follow-up project. If you are interested in using our software for your research, please contact one of the authors of this report and state your interest in doing so. The software is written in ANSI C and depends on `libldns` from NLnet Labs[4], SQLite 3.7 or up and requires POSIX threads. It should run on any modern Linux, UNIX or BSD system without modification.

3 Conditions of Use

Use of the data sets described in this document and specified explicitly in Table 4 is subject to the terms of the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license that can be found at:

`http://creativecommons.org/licenses/by-sa/4.0/`



If you use the data sets for research purposes and want to publish your results, citing this technical report fulfills the requirements of the license. We appreciate any feedback on the data sets and would like to hear how you have used our data for your research.

4 Acknowledgements

We would like to thank Siôn Lloyd (Nominet, `.uk`), Marco Davids (SIDN, `.nl`) and Patrik Wallström (IIS, `.se`) for helping out with DNS data for their respective TLDs and granting us the right to publish anonymised versions of the measurement data we collected for their top-level domains. The authors would also like to thank VeriSign for sharing data on `.com` and `.net` and the Public Interest Registry (PIR) for sharing data on `.org`.

Part of this work has been supported by the EU-FP7 FLAMINGO Network of Excellence Project (318488) and by the GigaPort3 programme funded by the Dutch Economic Structure Enhancing Fund (FES).

References

[1] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. DNSSEC and its potential for DDoS attacks - a comprehensive measurement study. In *Submitted to the Internet Measurement Conference 2014 (IMC 2014)*, 2014.

[2] Rafael R.R. Barbosa, Ramin Sadre, Aiko Pras, and Remco van de Meent. Simpleweb/University of Twente Traffic Traces Data Repository. Technical report, University of Twente, 2010.

[4] `https://www.nlnetlabs.nl/projects/ldns/`

[3] Paul Mockapetris. RFC 1035 - Domain Names - Implementation and Specification, 1987.

[4] J. Damas, M. Graff, and P. Vixie. RFC 6891 - Extension Mechanisms for DNS (EDNS(0)), 2013.

[5] R. Arends, R. Austein, M. Larson, Massey, D., and S. Rose. RFC 4033 - DNS Security Introduction and Requirements, 2005.

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

[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. RFC 4035 - Protocol Modifications for the DNS Security Extensions, 2005.

[8] D. Eastlake and T. Hansen. RFC 4634 - US Secure Hash Algorithms (SHA and HMAC-SHA), 2006.

Col#	Name	Type	Description
1	`domain_id`	`INTEGER`	Corresponds to the domain ID (`id` column) from the domain information tables
2	`auth_ns [A]`	`VARCHAR(255)`	The IPv4 or IPv6 address of the authoritative name server from which this result was obtained
3	`qname`	`VARCHAR(255)`	The query name (only for `A`, `AAAA` and `TXT` queries), can be empty, `www` or `mail`
4	`rcode`	`INTEGER`	The DNS response code (`RCODE`) returned in the response
5	`edns0_supported_size`	`INTEGER`	The EDNS0 supported size reported by the authoritative name server in the `OPT` record in the additional section (EDNS0 queries only)
6	`query_size`	`INTEGER`	The query size on the wire (UDP datagram size) as reported by `libldns` (see also Section 2.4)
7	`response_size`	`INTEGER`	The response size on the wire (UDP datagram size) as reported by `libldns`
8	`amplification`	`DOUBLE`	The amplification factor defined as $\frac{\texttt{response\_size}}{\texttt{query\_size}}$
9	`truncated`	`BOOLEAN`	Set to `true` if the response was truncated (`TC` flag set), `false` otherwise
10	`ans_count`	`INTEGER`	The number of answers in the response (see Section 4.1 of [3])
11	`aut_count`	`INTEGER`	The number of answers in the authority section of the response (see Section 4.1 of [3])
12	`add_count`	`INTEGER`	The number of answers in the additional section of the response (see Section 4.1 of [3])
13	`distinct_rr_types`	`INTEGER`	The distinct number of different resource record types in the response

Table 6: Result table schema