DNSSEC Case Study

Emily Gao, Maggie Van Nortwick, Rahul Toppur

What we'll be covering:

- Problem & objectives
- DNSSEC overview: benefits & adoption
- Methodology
- Obstacles
- TLD findings
- delv & whois overview
- Evaluation plan & metrics
- Data visualizations
- Conclusions
- Remaining Steps

The Problem

- DNSSEC: great but not mandatory
- Uploading DS records is tricky for sites
  - getting DS record to parent zone requires passing it through registry
  - domain owners can’t access registry directly, need to go through registrar organization
- Registrar must have methods for domain owners to upload DS records
  - If not, can’t provide DS records to their parent zones
  - domains can’t support DNSSEC even if they want to
- Thus both the domain and the registrar need to enable it

Our Objective

- Gather metrics to perform a case study on the adoption of DNSSEC
  - Restrict collection of data to the top 1 million domains
- Gain greater insight into emerging trends and DNSSEC adoption based on the following factors:
  - Whether DNSSEC is fully, partially, or not supported for a given domain
  - Country associated with each domain
  - Site categorization (shopping, finance, entertainment)
  - Buckets based on site popularity ranking

DNSSEC Benefits

- DNS on its own is vulnerable
  - Spoofing, Cache-poisoning, Man-in-the-middle, Kaminsky attacks
- DNSSEC helps protect against these
  - uses public key cryptography and digital signatures
  - verifies DNS message authenticity between zones
- Guarantees data origin authentication
  - Checking keys and signatures proves data is from the zone it says its from
- Guarantees data integrity protection
  - Verifying signatures proves data hasn’t been modified in transit

Methodology

- Starting with Tranco 1M
- Issuing delv queries to get DNSSEC data
- Filtering domains to get generic TLDs
  - Running whois queries for appropriate domains
- Creating a master dataset
  -

```
site,rank,tld,tld_type,country,delv_result
google.com,1,com,gTLD,US,0
youtube.com,2,com,gTLD,US,0
facebook.com,3,com,gTLD,US,0
microsoft.com,4,com,gTLD,US,0
twitter.com,5,com,gTLD,US,0
```

- Repeat process for site-categorization specific domains
- Create visualizations and make conclusions

Obstacles

- 1M is a big number!
- It takes time to run delv and whois queries
  - Necessary to make use of multithreading to speed up this process
  - Had to break up data among us and make use of multiple cores on khoury servers
- Whois database limits the number of requests you can send
  - Necessary to sleep for a short period in between each query
- Normalizing inconsistencies and data cleaning
  - Country codes

Top-Level Domains

- Types:
  - Country code TLDs:
    - 321,625 sites of 1M
  - General TLDs:
    - Generic TLDs, generic-restricted TLDs, sponsored TLDs
    - 677,233 sites of 1M
  - Other:
    - Infrastructure TLDs (ARPA), test TLDs, TLDs with non-latin characters, onion
    - 1142 sites of 1M
- We only made whois queries for sites in the general TLD category

delv

```
DELV(1)                                  BIND9                                  DELV(1)

NAME
       delv - DNS lookup and validation utility

SYNOPSIS
       delv [@server] [-4] [-6] [-a anchor-file] [-b address] [-c class] [-d level] [-i] [-m]
            [-p port#] [-q name] [-t type] [-x addr] [name] [type] [class] [queryopt...]

       delv [-h]

       delv [-v]

       delv [queryopt...] [query...]

DESCRIPTION
       delv (Domain Entity Lookup & Validation) is a tool for sending DNS queries and validating the
       results, using the the same internal resolver and validator logic as named.

       delv will send to a specified name server all queries needed to fetch and validate the
       requested data; this includes the original requested query, subsequent queries to follow CNAME
       or DNAME chains, and queries for DNSKEY, DS and DLV records to establish a chain of trust for
       DNSSEC validation. It does not perform iterative resolution, but simulates the behavior of a
       name server configured for DNSSEC validating and forwarding.
```

delv

- delv is a command line tool for Domain Entity Lookup & Validation, which additionally validates DNSSEC trust chains.
- Our script utilizes the +vtrace and +multiline flags
  - vtrace shows the individual fetches/validation steps
  - multiline prints long records
- Collectively, it took us roughly 54 hours to run the script on 1M entries, as well as repeat the process on entries which fell into none of our categorization buckets.
  - The script runs delv using Google's DNS resolver (@8.8.8.8), and writes I/O files based on a csv we split into parts, sourced from Tranco
  - We ended up with approximately 3.2k domains for which delv failed to resolve

delv @8.8.8.8 +vtrace +multline nsa.gov

```
(base) rahul@debian:~ $ delv @8.8.8.8 +vtrace nsa.gov
;; fetch: nsa.gov/A
;; validating nsa.gov/A: starting
;; validating nsa.gov/A: attempting positive response validation
;; fetch: nsa.gov/DNSKEY
;; validating nsa.gov/DNSKEY: starting
;; validating nsa.gov/DNSKEY: attempting positive response validation
;; fetch: nsa.gov/DS
;; validating nsa.gov/DS: starting
;; validating nsa.gov/DS: attempting positive response validation
;; fetch: gov/DNSKEY
;; validating gov/DNSKEY: starting
;; validating gov/DNSKEY: attempting positive response validation
;; fetch: gov/DS
;; validating gov/DS: starting
;; validating gov/DS: attempting positive response validation
;; fetch: ./DNSKEY
;; validating ./DNSKEY: starting
;; validating ./DNSKEY: attempting positive response validation
;; validating ./DNSKEY: verify rdataset (keyid=20326): success
;; validating ./DNSKEY: signed by trusted key; marking as secure
```

```
;; validating gov/DS: in fetch_callback_validator
;; validating gov/DS: keyset with trust secure
;; validating gov/DS: resuming validate
;; validating gov/DS: verify rdataset (keyid=14631): success
;; validating gov/DS: marking as secure, noqname proof not needed
;; validating gov/DNSKEY: in dsfetched
;; validating gov/DNSKEY: dsset with trust secure
;; validating gov/DNSKEY: verify rdataset (keyid=7698): success
;; validating gov/DNSKEY: marking as secure (DS)
;; validating nsa.gov/DS: in fetch_callback_validator
;; validating nsa.gov/DS: keyset with trust secure
;; validating nsa.gov/DS: resuming validate
;; validating nsa.gov/DS: verify rdataset (keyid=48498): success
;; validating nsa.gov/DS: marking as secure, noqname proof not needed
;; validating nsa.gov/DNSKEY: in dsfetched
```

nsa.gov

```
(base) rahul@debian:~ $ delv @8.8.8.8 +rtrace +multiline nsa.gov
;; fetch: nsa.gov/A
;; fetch: nsa.gov/DNSKEY
;; fetch: nsa.gov/DS
;; fetch: gov/DNSKEY
;; fetch: gov/DS
;; fetch: ./DNSKEY
; fully validated
nsa.gov.                20 IN A 23.50.46.198
nsa.gov.                20 IN RRSIG A 7 2 20 (
                              20210425161554 20210422151554 41795 nsa.gov.
                              v6hyS8lD8FtZFCx+Sf8+QUS/aSq+xA18OjEpnsw0yMnr
                              WKRm+BwLiAiKhNrYX1fTkMO10N+MB70VLVnLMt4LXg6O
                              c482OYBPkQWaVPCvzQtaYC4kvJWSRqqmbn0BbbLKKtAD
                              62Wjzl77iZgSo4ah9jw3/qrer7csMzAJJEFWhCY= )
(base) rahul@debian:~ $
```

yahoo.com

```
(base) rahul@debian:~ $ delv @8.8.8.8 +rtrace +multiline yahoo.com
;; fetch: yahoo.com/A
;; fetch: com/DS
;; fetch: ./DNSKEY
;; fetch: yahoo.com/DS
;; fetch: com/DNSKEY
;; fetch: yahoo.com.dlv.isc.org/DLV
;; fetch: dlv.isc.org/DNSKEY
; unsigned answer
yahoo.com.              3200171710 IN A 74.6.143.25
yahoo.com.              3200171710 IN A 74.6.143.26
yahoo.com.              3200171710 IN A 74.6.231.20
yahoo.com.              3200171710 IN A 74.6.231.21
yahoo.com.              3200171710 IN A 98.137.11.163
yahoo.com.              3200171710 IN A 98.137.11.164
(base) rahul@debian:~ $
```

gembook.jp

```
(base) rahul@debian:~ $ delv @8.8.8.8 +rtrace +multiline gembook.jp
;; fetch: gembook.jp/A
;; resolution failed: SERVFAIL
(base) rahul@debian:~ $
```

delv

- We decided on splitting domains up into three categories: fully verified, partially verified, and not verified at all
  - fully verified DNS trust chains resulted in a ‘fully validated’ message
  - partially verified chains result in ‘unsigned answer’
  - fully unverified trust chains result in SERVFAIL messages in stderr
- A few of our queries returned resolution failures through timeouts, usually when checking at the SLD phase
  - We decided not to place them in the partial validation or non-supported category

whois

```
WHOIS(1)                          Debian GNU/Linux                          WHOIS(1)

NAME
       whois - client for the whois directory service

SYNOPSIS
       whois  [  {  -h  |  --host  }  HOST  ]  [  {  -p  |  --port  }  PORT  ]  [  -abBcdGHKlLmMrRx ]
       [ -g SOURCE:FIRST-LAST ] [ -i ATTR[,ATTR]... ] [ -s SOURCE[,SOURCE]... ] [ -T TYPE[,TYPE]... ]
       [ --verbose ] OBJECT

       whois -q KEYWORD

       whois -t TYPE

       whois -v TYPE

       whois --help

       whois --version

DESCRIPTION
       whois searches for an object in a RFC 3912 database.

       This  version  of  the  whois  client tries to guess the right server to ask for the specified
       object. If no guess can be made it will connect to whois.networksolutions.com for NIC  handles
       or whois.arin.net for IPv4 addresses and network names.
```

whois

- Whois protocol allows queries to WHOIS databases, which contain a listing of all registered domains
- Queries are either thick or thin
  - Thin: Data sufficient to identify sponsoring registrar and status of registration
  - Thick: Maintains registrant's contact information
- Used python-whois (Python wrapper) to issue queries
  - Opens a socket and queries the server directly, instead of going through any intermediate web services

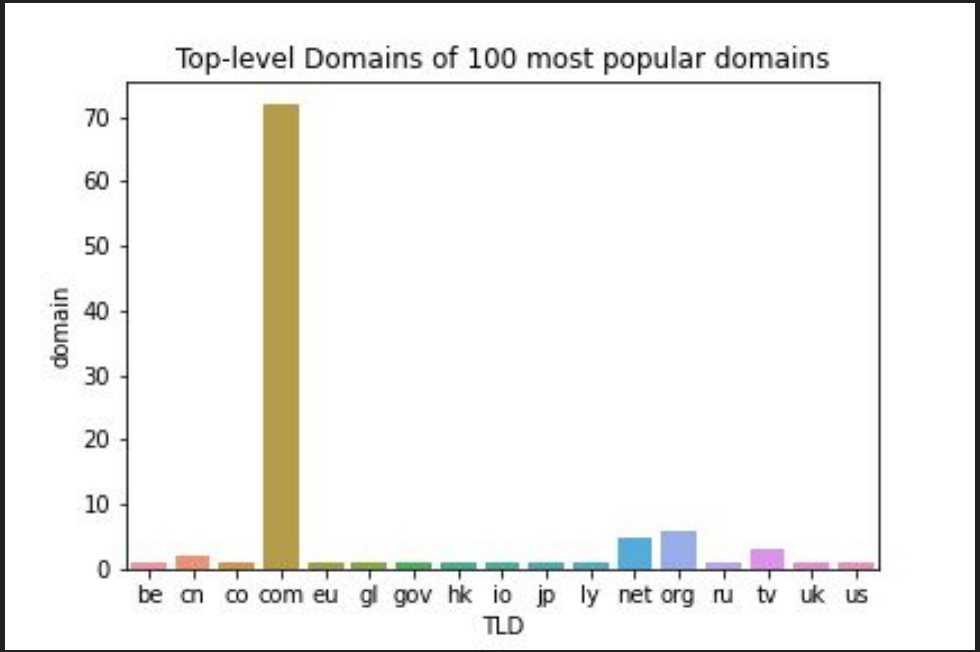
```
Python 3.8.3 (default, Jul  2 2020, 16:21:59)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import whois
>>> q = whois.query("yahoo.com")
>>> q.__dict__["registrant_country"]
'US'
```

Evaluation Plan and Metrics

- Top-level domains for the 100 most popular sites
- World heatmap for DNSSEC adoption
- DNSSEC adoption per top 25 TLDs
- DNSSEC adoption per site categorization
- Overall percentage of sites that have fully validated vs. partially validated certificate chains
- DNSSEC adoption based on popularity rank
    - Sites ranked 1-100, 100-1000, etc.

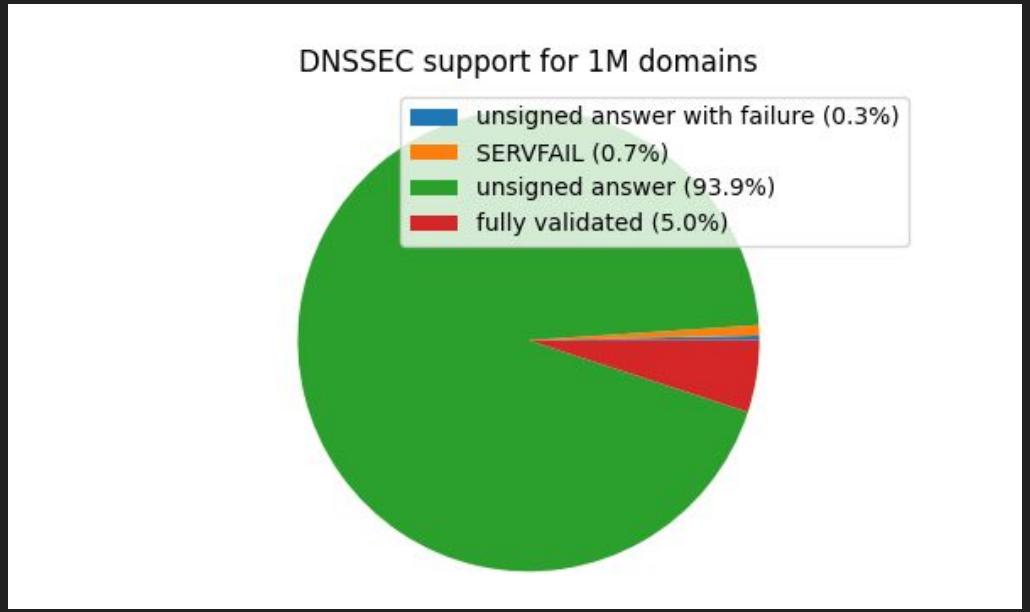
Visualizations

TLDs for the Tranco 100 most popular domains



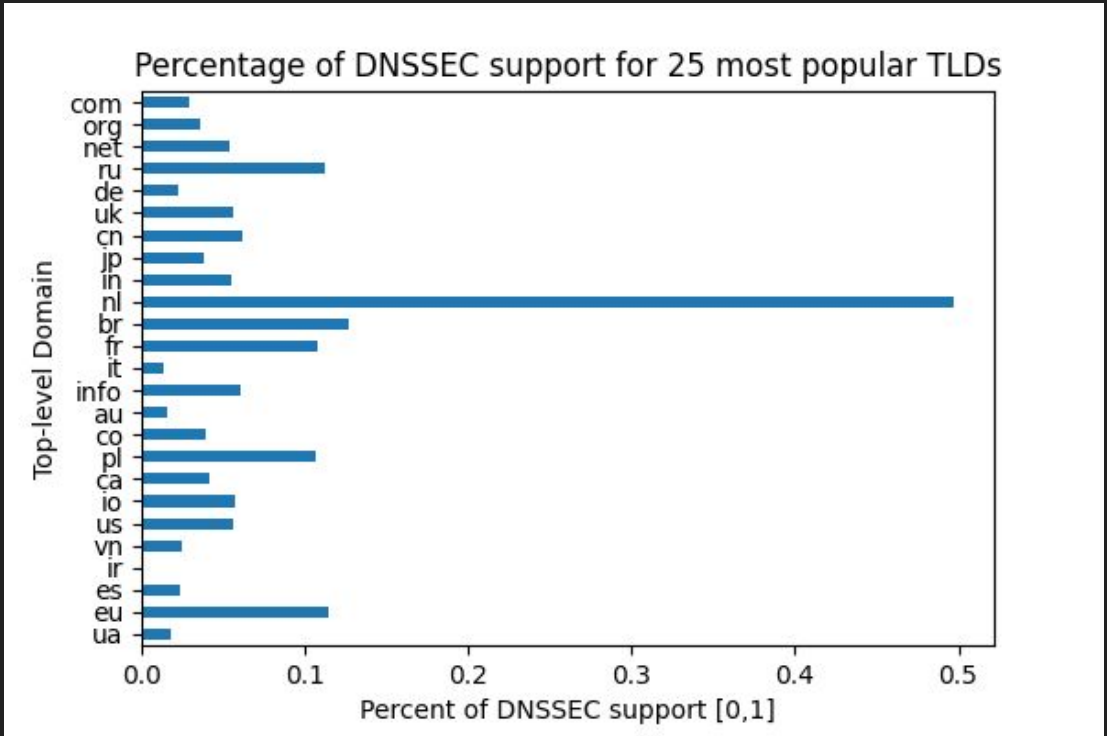
.com is the most popular TLD by far, making up over 70% of the top 100 domains

DNSSEC support for Tranco 1M



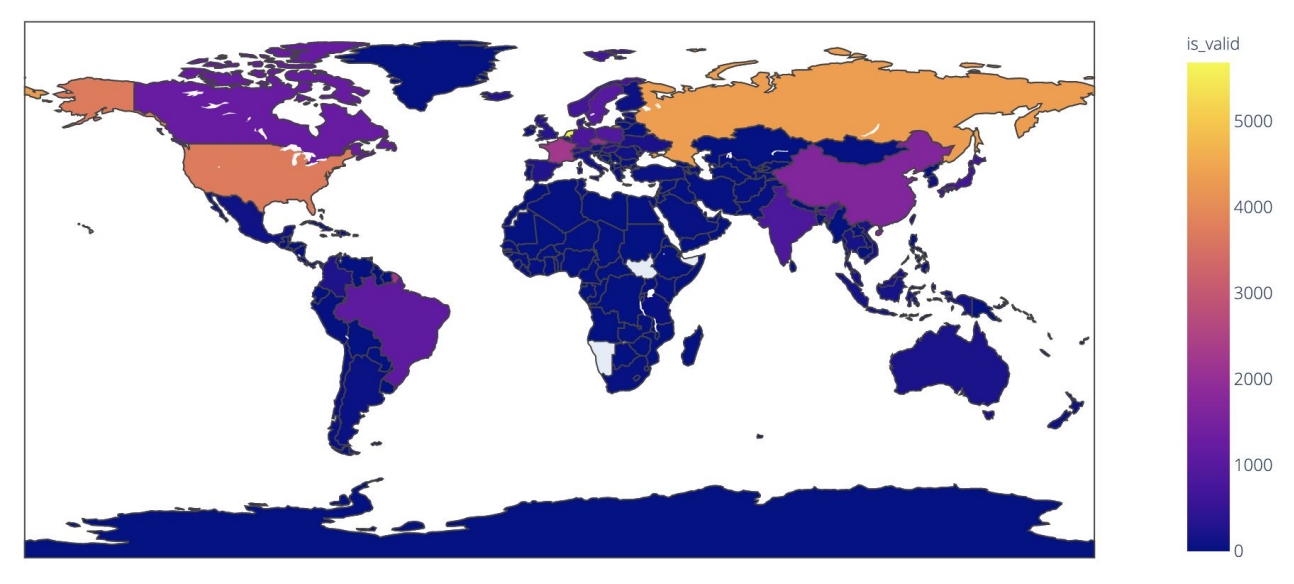
Most sights have some partial DNSSEC support, but only 5% are fully validated. Only 1% returned an error or had no DNSSEC support

DNSSEC adoption per TLD



General TLDs such as .com have pretty low DNSSEC support, while country code TLDs, such as .nl, more commonly had higher levels of support.

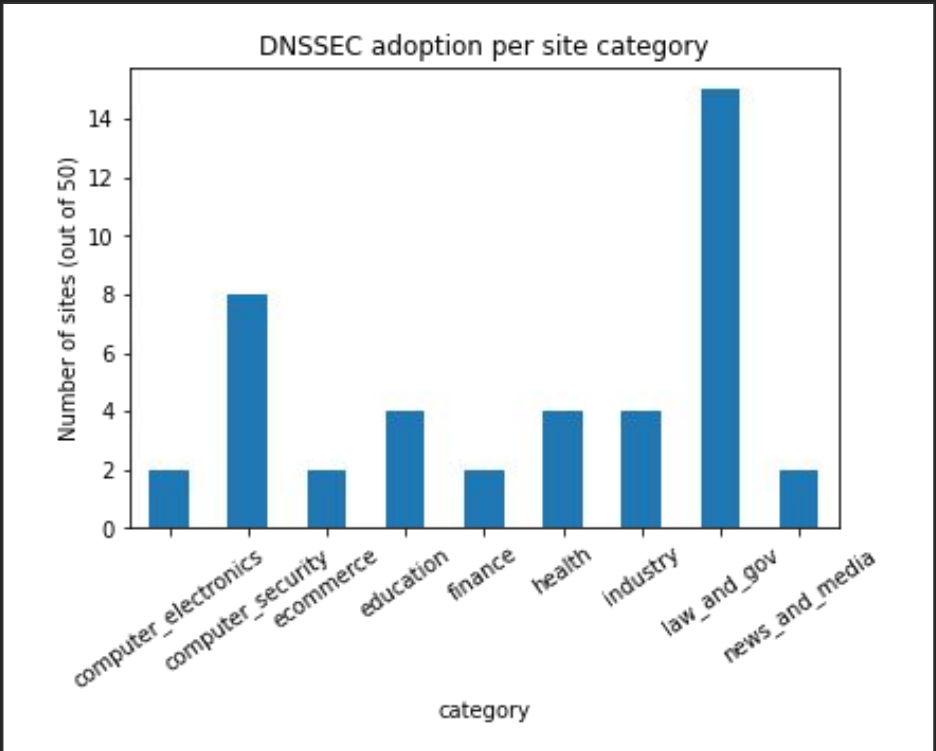
DNSSEC adoption per country



This figure uses country code TLDs and whois query data to determine country of origin for a domain.

Some small countries such as Netherlands have high percentages of fully validated sites, otherwise the countries with the most support tend to be larger ones i.e. Russia, China, India, USA, Brazil, Canada

DNSSEC adoption per site category



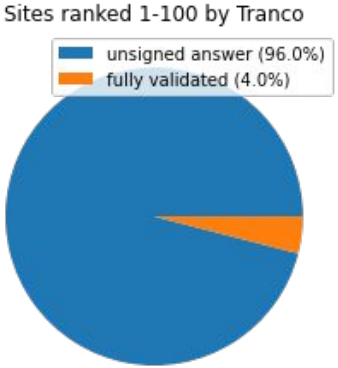
We've sourced these categorizations from similarweb.com

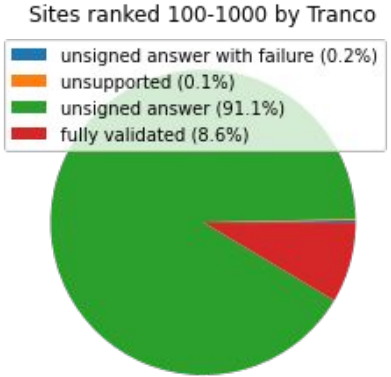
The category with the most DNSSEC support is predictably law and government, followed by computer security

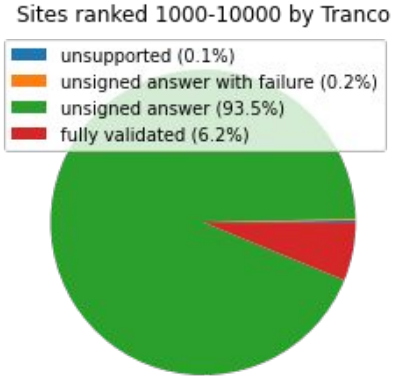
Finance, e-commerce, news and media are surprisingly low

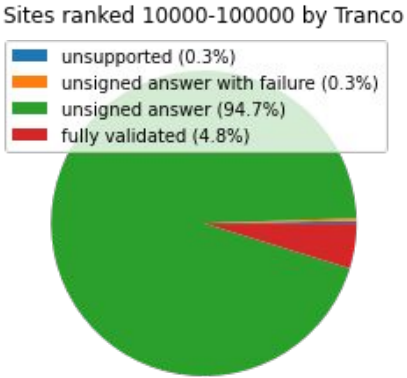
Health, education, and heavy industry somewhere in the middle

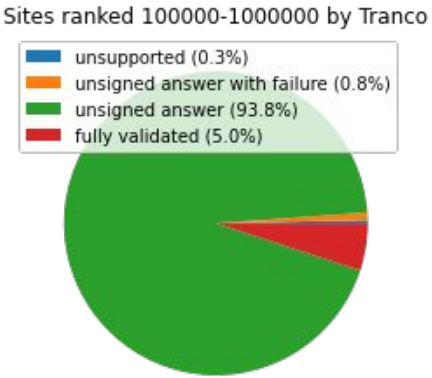
Ranking buckets from Tranco List











Majority of all sites for all buckets was unsigned answer

Top 100 sites had smaller percent of fully validated sites compared to every other bucket

Top 100-1000 sites had the highest percent of fully validated

Percentage of unsupported or unsigned answer with failure minimal for all buckets, but became more significant for buckets of less popular sites

Conclusions

- Full support for DNSSEC is low, but partial support relatively common
- Amongst centralized TLDs such as those for countries (not general TLDs), DNSSEC support is often a bit higher
    - Speculation: nation-wide infrastructure for DNSSEC more effective than decentralized infrastructure like in general TLDs such as .com
- Top sites have better DNSSEC support but only marginally
- Law and government site category had highest DNSSEC support, followed by computer security site category
    - Predictable, but we’re glad they’re keeping up

Remaining Steps

- ***Organize*** our GitHub repository better and clean up our code!
- Submit our deliverables!

References

[1] https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en
[2] https://blog.apnic.net/2017/12/06/dnssec-deployment-remains-low/
[3] https://tld-list.com/free-downloads
[4] https://gist.github.com/derlin/421d2bb55018a1538271227ff6b1299d
[5] https://www.ip2location.com/free/country-information
[6] https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains#Types
[7] https://www.similarweb.com/category/
[8] https://tranco-list.eu/

Thanks for watching!