CODE UNIVERSITY OF APPLIED SCIENCES

Analysis of DNSSEC Adoption at Internet Scale

Bachelor Thesis

Name:	Felix Wotschofsky
Address:	[redacted]
Email:	felix.wotschofsky@code.berlin
Study Program:	Bachelor of Science (B.Sc.) - Software Engineering
Semester:	Fall Semester 2024
Enrollment Number:	[redacted]
Supervisor One:	Prof. Dr. Peter Ruppel
Supervisor Two:	Prof. Dr. Adam Roe
Date:	November 28, 2024

Declaration

I hereby confirm that I have written the thesis titled "***Analysis of DNSSEC Adoption at Internet Scale***" by myself, without contributions from any sources other than those cited in the text and bibliography. All graphics, drawings and images not created by me but included in this thesis, along with any uses of generative AI have been fully and accurately referenced.

Furthermore, I confirm that neither this work nor parts of it have been previously or concurrently used as an assessment submission in other courses or in other examination proceedings.

In the table below, I list the generative AI tools used in creating my thesis, as well as a description of the purpose and extent of the use of each tool

AI Tool Used	**Purpose of use and description of extent of use**
Supermaven	Autocomplete and chat in code editor
Anthropic Claude	Generating BibTeX entries for websites, assistance with LaTeX syntax, and writing SQL queries
OpenAI GPT	Grammar and spelling correction ("Proofreading")

Location, Date: ______________ Student signature: ______________

Abstract

DNSSEC is a security extension to the Domain Name System (DNS) and serves as a mechanism for ensuring the integrity of records using a cryptographic approach. Since its standardization in 2005, however, adoption has been lackluster.

This thesis presents a survey of 171 million domain names across all TLDs and their adoption, as determined by querying Cloudflare's recursive resolver. Additional metadata, such as registrar and nameserver, was also collected. Domain names were primarily sourced from scraping public certificate transparency logs.

Across the entire dataset, the adoption rate of DNSSEC is 5.93%. Survey results are further segmented by TLD, registrar, nameserver, and others. The largest registrar, GoDaddy, sees only a 0.29% adoption rate, while 28.12% of domains registered with Squarespace (fka. Google Domains), have adopted the standard. The 100 most important domains, as defined by Cloudflare Radar, even show a below-average adoption rate of 5%.

The generally low adoption rate can be attributed to a lack of incentives for domain owners and registrars, combined with an unintuitive adoption path. In addition, DNSSEC is considered a risk to availability by some.

Keywords: Domain Name System, DNS Security, DNSSEC Survey, Certificate Transparency, Internet Security

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The DNS (Domain Name System) protocol is a fundamental part of the internet. While computers can target each other using IP addresses, those aren't easy to remember. DNS solves this issue by mapping and translating customizable names into IP addresses or other miscellaneous information. The domain name `example.com`, for instance, translates to the IP address `93.184.215.14`.

However, because the DNS protocol itself is unencrypted and distributed, a system for ensuring trust is needed. DNSSEC is a set of internet standards amending the original DNS specification (Domain Name System Security Extensions) with the goal of providing such a trust mechanism.

1.2 Understanding DNS

Domain names are separated into zones. For instance, `www.example.com` consists of the zones `.` (root zone), `com.`, `example.com.`, and `www.example.com.`. Every zone either has its own nameservers or relies on those of a parent zone for answering queries. To resolve a record, a recursive resolver[1] queries the nameservers of each respective zone, starting at the root zone, receiving either the answer to the query or a pointer to a child zone and its respective nameservers.

Zones can hold different record types. For example, `A` is used for IPv4 addresses, and `TXT` is used for miscellaneous values.

1.3 Understanding DNSSEC

DNSSEC provides a mechanism for validating the integrity of DNS records using public/private key cryptography. Sets of records grouped by type (`RRSets`) are signed using a private zone key. Signatures are sent alongside record responses, while the public zone key is distributed by the signing zone through the sepa-

[1] https://www.akamai.com/glossary/what-is-recursive-dns

rate `DNSKEY` record. The parent zone holds and distributes a hash of the public key through `DS` records, thereby vouching for the key's authenticity. When verifying a response, the resolver fetches the keys and hashes before checking the signature. This validation process is repeated for every zone up to the root zone.

In order to correctly adopt DNSSEC, a domain owner needs to generate a signing key pair for their zone, sign the records, and inform the parent zone through their registrar about the hashes of the keys used. While the first steps are generally handled automatically by the nameserver provider, the latter step may require manual effort.

1.4 Hypothesis

Based on various past works, it is hypothesized that current adoption rates for the DNSSEC standard are low. This is to be verified across a large dataset, which is representative of the entirety of the internet. Additional metadata is to be collected for segmentation, as well as for finding patterns and potential explanations for the adoption results.

2 Literature Review

2.1 Original Problem

The DNS protocol itself is unencrypted, making it susceptible to various attacks where an attacker provides a forged response. Recursive resolvers offer an especially large attack surface, as they usually use aggressive caching to improve performance, thus distributing malicious responses to multiple clients. This attack is referred to as a "cache poisoning" attack. Most notably, off-path (spoofing) and Man-in-the-Middle (MitM) attacks pose a threat to unprotected DNS requests [HS13].

In an off-path attack, the attacker sends a spoofed response to the victim without knowing the original request. Therefore, there is no need to intercept the original request. TCP's basic countermeasures, like using changing ports for responses, are easily circumvented through knowledge of the victim's operating system or through strategic guessing [GH14].

For a MitM attack, the attacker intercepts the request, thereby knowing exactly where to send the response, as well as any transaction identifiers specified by the resolver [HS13]. DNSSEC provides a defense mechanism against this attack using cryptographic signatures [HS14].

2.2 Specification

DNS, as introduced in RFC 1035 [87], lacks any security measures. The DNSSEC specification was first proposed as RFC 2065 [3K97] in January 1997 and refined in RFC 2535 [3rd99]. In March 2005, the set of RFC 4033 [Ros+05a], RFC 4034 [Ros+05c], and RFC 4035 [Ros+05b] reintroduced DNSSEC and were standardized.

Later additions introduced modifications for increased security. For example, RFC 6605 [HW12] introduced Elliptic Curve as a long-term replacement for RSA, and RFC 5155 [Are+08] introduced the `NSEC3` record as a less transparent successor to the `NSEC` record. RFC 9364 [Hof23] provides a full overview of relevant documents.

2.3 Existing Analysis of Adoption

In 2017, [Wan17] used zone enumeration by hash-breaking results from the NSEC3 record to find domains and survey them for DNSSEC adoption. Since zone enumeration is an unintended and, in certain cases, problematic side effect, it will be discussed further in 2.4. Across a dataset of 6.4 million, the paper effectively found two groups of TLDs: high adoption rate TLDs with >45% (.nl, .se, .cz, .no) and low adoption rate TLDs with <1% (.com, .net, .org, .de). These numbers generally align with self-reported numbers from Verisign, the registry for .com and .net [Ver24b], as well as other reports [SID24]. Verisign has since seen a rise to >4% for both TLDs [Ver24b].

Besides the number of domains supporting DNSSEC, the number of validating resolvers is also a very important metric, as only validating resolvers can take advantage of the security benefits. Statistics from APNIC [APN24] show greatly differing ratios of validating/non-validating DNS resolvers across countries. For instance, 99.22% of requests from Saudi Arabia, but only 0.06% of requests from China are validating DNSSEC. Results for other large markets are India with 64.11%, the US with 37.17%, Germany with 80.58%, and France with 34.55%.

2.4 Challenges and Problems

With DNSSEC, the response payload for DNS queries grows significantly because a cryptographic signature needs to be included alongside the records. Besides increased operating costs, this also leads to increased effectiveness of DNS amplification DDoS attacks. The attack exploits that even without DNSSEC, DNS responses are larger than their queries. The attacker sends a query to a resolver using a spoofed IP address to have the response sent to the victim instead, thereby amplifying the attack bandwidth [RSP14] [Clo24b]. While not exclusive to it, the ANY query type is especially susceptible to this attack since it already has a large response payload [Too+21]. The attack could be partially combated by reducing the amount of data sent in response to an ANY query, as proposed in RFC 8482 [Abl+19].

The NSEC record, introduced as part of DNSSEC, is intended for explicit *denial* of the existence of a record. It is needed since no signature can be created from the values of a non-existent record [Are+08]. However, this record has the major downside of exposing all zones in plain text. This vulnerability is referred to as "zone enumeration". The successor, NSEC3 [Are+08], uses SHA-1 hashes instead. The use of rainbow tables[2] is prevented through salting [DNS24a]. This makes enumeration harder, but not impossible, since nowadays SHA-1 is not considered to be a strong hash anymore and can therefore be cracked through brute force with reasonable effort. According to the authors of the proposal of vcelak-nsec5-08, the vulnerability can be mitigated using verifiable random functions [Gol+16]. The proposal, however, has been inactive since 2019 and was never standardized.

In general, a source of challenges with DNSSEC is that it is retrofitting security measures onto an existing and established protocol. As explained by [HS14] in 2013, since responses with signatures are significantly larger, some firewalls drop these packets. This, however, may have changed since then and become a non-issue. Furthermore, for a zone to be effectively protected against MitM attacks, all dependencies from, for example, NS, CNAME, or MX records need to adopt DNSSEC as well. Otherwise, an attacker can achieve the same result by targeting the depended-on zone.

2.5 Related Technologies

Besides DNSSEC, DNSCurve and DNSCrypt are also proposals for tamper-proof DNS protocols from 2008 and 2011, respectively. However, only DNSCurve used an encrypted connection [DNS09] [DNS24b] [LGS22]. Neither of them was ever standardized.

DNSSEC ensures the integrity of DNS records but still relies on the unencrypted DNS protocol as introduced in RFC 1035 [87], thus allowing bad actors to openly inspect DNS traffic. RFC 7858 introduces DNS-over-TLS (DoT)

[2] https://en.wikipedia.org/wiki/Rainbow_table

[Hu+16], and RFC 8484 introduces DNS-over-HTTPS (DoH) [HM18]. While other encrypted DNS protocols are proposed, only the two mentioned are widely supported [Lu+19, Table 1]. For DoH, a client will often have to fall back to another — potentially unencrypted — DNS protocol to resolve the hostname of the DoH resolver [HM18, Section 10], since hostnames often are FQDNs[3] and need to be resolved themselves, making DoH only feasible for connections between a client and a known recursive resolver.

TLS on the web through HTTPS serves a similar purpose of ensuring that a user is connected to the intended server. The connecting client verifies whether the server's certificate is valid for the current domain and whether it was issued by a trusted authority. Numbers from Google Chrome usage data show that well over 90% of websites use HTTPS [Goo24b].

2.6 Conclusion and Contribution

DNSSEC was introduced over 20 years ago to combat attack vectors such as resolver cache poisoning. It is the only standardized specification serving its particular purpose. Still, many years later, it is barely adopted on the domain side, with adoption from resolvers also leaving room for improvement.

There are plenty of small-scale surveys on DNSSEC that investigate a subset of domains, such as a specific TLD or industry [Rob17] [MC17]. No up-to-date effort, however, looks at adoption rates across a large sample size while also taking additional metadata such as registrars and nameservers into consideration.

This is where this thesis adds to the discussion. The following chapters will discuss how a large set of domains was collected, surveyed, and analyzed, as well as the results of this analysis.

[3] fully qualified domain name

3 Methodology

In this chapter, we introduce the procedure used for creating the dataset on DNSSEC adoption and its analysis.

The experiment was conducted in three phases: (1) finding domains to later run the survey on, (2) surveying domains for DNSSEC adoption, and (3) analyzing the created dataset. Each phase was run separately, with a manual transfer of data from the previous phase.

3.1 Collection of Domains for Analysis

3.1.1 Certificate Transparency Logs (Scraping)

The majority of analyzed domains were collected by monitoring the Certificate Transparency Logs (CT logs) and extracting domains into a deduplicated database.

Certificates are used to establish an encrypted and trusted connection, most often over HTTPS; however, other protocols for non-web traffic may also utilize certificates. Whether the issuing authority, and by extension a certificate, can be trusted is determined through a so-called chain of trust. The issuing authority signs the new certificate using its own certificate, which, in turn, is trusted by operating systems and browser vendors. Whenever a certificate authority signs a new certificate, it reports the details of the certificate to said CT logs [Goo24a].

The certificate transparency logs were monitored from May 19, 2024, to October 9, 2024, with downtime of a couple of days over the timeframe. The resulting dataset should be rather representative of today's internet because (1) well over 90% of websites are available over HTTPS [Goo24b] and therefore require a valid certificate, and (2) authorities like Let's Encrypt push for short-lived 90-day certificates, even suggesting renewing certificates every 60 days [Aas15], thus renewing at least once during the scraping process.

Relying on issued certificates, however, also means that the results are certainly skewed toward greater adoption. This is because domain owners going through the effort of setting up DNSSEC will most likely also care about TLS.

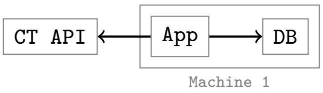

Figure 1: Scraper Architecture

The scraper for the CT logs is written in TypeScript and executed with Node.js; an open-source project[4] was used as a starting point. Updates from CT logs are received through a WebSocket-based API[5]. After extracting the apex domain (e.g., `example.com` from `www.example.com`) for every identity on the reported certificate, those values are then stored in a MongoDB database. An index on the field containing the domains is used to efficiently find already existing entries and maintain a deduplicated list.

3.1.2 Additional Sources of Domains

In addition to the domains scraped from CT logs, the following domain lists were also included in the dataset:

- Top 1 million domains from Cloudflare Radar: This list is based on requests to Cloudflare's widely used 1.1.1.1 recursive DNS resolver and ranks domains by the estimated number of users accessing a given domain. Referred to as *Cloudflare Top 1M* or *Cloudflare Top 1000* for the sublist of the highest-ranking 1,000 domains [Clo24c] [Clo24a].
- Majestic Million: This list is created by the SEO company Majestic and is ranked based on the number of subnets referring to a domain, as determined by the company's scraping efforts [Maj24].
- Alexa Traffic Rank: Alexa Internet, a company acquired by Amazon, provided a list of top websites based on proprietary estimates of site traffic and engagement. Because of the service's discontinuation in 2022, a backup from that year was used [Pet22].

[4] `https://github.com/ImLunaHey/ct-logs`
[5] `https://certstream.calidog.io/`

3.2 Surveying of Domains for Adoption

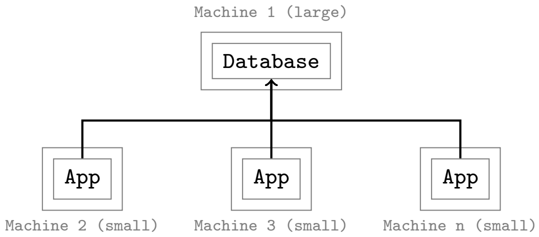

Figure 2: Survey Architecture

The survey application for collecting the details for each domain was also written in TypeScript and executed using Node.js. It was deployed in a main/worker configuration with a single server running a PostgreSQL database and multiple machines running one or multiple instances of the survey app. The database was used both as a task queue and to store the results.

For every domain, the following data points were collected:

Data Point	**Example**
Domain	`example.com`
Top-level domain	`com`
Registrar	`CloudFlare, Inc.`
Created At	`2020-01-01`
DNSSEC Status	`true`
NS DNS Records	`adam.ns.cloudflare.com`
DS DNS Records	`2371 13 2 E11336E0D71A8585AAAA...`
DNSKEY DNS Records	`256 13 oJMRESz5E4gYzS/q6XDr...`
Timestamp of Analysis	`2023-10-21T12:00:00Z`

Table 1: Collected Data Points for Each Domain

For DNS queries (`NS`, `DS`, and `DNSKEY` records), Cloudflare's 1.1.1.1 DoH DNS resolver[6] was used to minimize the number of network roundtrips compared to

[6] https://one.one.one.one/

resolving records from the authoritative nameservers directly. The DoH variant was used for its simpler integration compared to sending queries using the DNS protocol.

Registrar and creation date are taken from WHOIS. Queries were sent directly to the registries using the `whoiser` package from npm[7]. Some registries, such as those for `.de`, `.ch`, or `.es`, greatly restrict access to WHOIS information. Results from other registries, like `.br`, cannot be parsed by the `whoiser` package. In both cases, the dataset lacks the data points that are otherwise taken from WHOIS.

The status of whether a domain has adopted DNSSEC or not is taken from the `AD` bit [GW03] of 1.1.1.1's response for a query for the `A` record of a given domain. The value of the bit is determined by whether the resolver was able to verify the records (or lack thereof) using one of its supported signature algorithms[8].

Running the scraper across multiple smaller machines, compared to a single large machine, means that requests are sent from multiple IP addresses. While Cloudflare does not impose any rate limits on their resolvers, most registries impose limits of varying degrees on their WHOIS servers. For the majority of the survey, four worker machines were used in parallel. However, for inspecting domains using TLDs with very low limits, up to 160 very small machines with separate IP addresses were used. These machines were configured using a `cloud-init`[9] config.

Before running the survey, all domains were manually migrated from the MongoDB database used while scraping into the Postgres database. All columns except *domain* and *tld* were left as `NULL`.

Tasks were distributed to the different workers by making use of Postgres' row locking. When querying for the next unprocessed domain, a lock was applied to the returned row through `FOR UPDATE`. Currently, processing entries

[7] https://www.npmjs.com/package/whoiser

[8] https://developers.cloudflare.com/1.1.1.1/encryption/dnskey/

[9] https://cloud-init.io/

were skipped through SKIP LOCKED to ensure no task is assigned to two different workers concurrently. Full query used: SELECT domain FROM domains WHERE dnssec IS NULL LIMIT 1 FOR UPDATE SKIP LOCKED. After the surveying of a domain is finished, even if unsuccessful, the row is updated to remove the lock. An index on the dnssec column was used to help efficiently find the next unprocessed entry.

The database, running across four cores of an AMD Epyc Milan CPU at 2.4 GHz, was able to handle 10,000 workers before experiencing slowdowns. Database connections were reused across workers.

3.3 Analysis of Dataset

The data gathered, as described in 3.2, was transferred into a ClickHouse[10] database using a CSV export from Postgres and was queried from there. ClickHouse is a column-oriented SQL database, making it significantly faster at processing and aggregating large datasets compared to row-oriented databases like Postgres, at the cost of efficiently interacting with individual rows.

To aid with the segmentation of domains during analysis, helper tables containing top domains from Cloudflare Radar [Clo24c] were created. The tables contain the top 100, 500, 1,000, 5,000, 10,000, 50,000, 100,000, 500,000, and 1,000,000 domains. These subsets will be referred to as *Cloudflare Top X*, e.g., *Cloudflare Top 1000*.

The results of this analysis will be presented in the following section.

[10] https://clickhouse.com/

4 Results of Analysis

4.1 Preamble

Results in this section are split into two sets where appropriate: the full dataset and the *Cloudflare Top 1000* subset. This differentiation is interesting because a small number of sites receive a large part of the internet's traffic and therefore have a comparatively large responsibility for their users. Ahrefs found that only 0.21% of web pages receive 1001+ visits [Tim23]. While not directly indicative of traffic by domain, this still hints at the clumping of traffic around the largest properties.

The *Cloudflare Top 1000* list was specifically chosen for this, as it is based on requests to the 1.1.1.1 recursive resolver. Therefore, it not only considers domains that consumers interact with directly, such as `google.com`, but also domains used for internal purposes and infrastructure, like CDNs such as `edgesuite.net` from Akamai. This is relevant because, as pointed out in 2.4, the security efforts of the latter directly affect other domains that depend on them through, for example, `CNAME` records.

While the full dataset includes 182,333,519 domains, 11,120,941 of them were disregarded for the analysis as they were no longer registered at the time of inspection. Therefore, after purging, the dataset contains a total of 171,212,578 domains. Numbers from the *Domain Name Industry Brief (DNIB)* Quarterly Report [DNI24] suggest that the created dataset covers around half of all domains.

4.2 The Dataset in Detail

In this subsection, we inspect the dataset composition to better understand the context of the adoption rates presented in a later section.

4.2.1 Most Popular TLDs

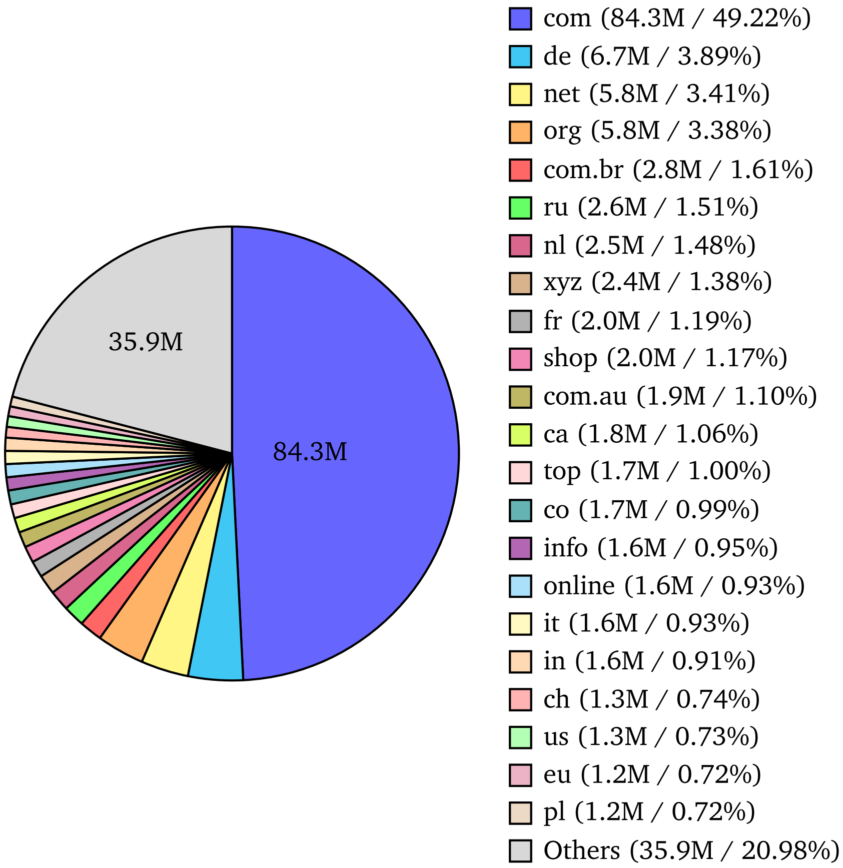

Figure 3: Distribution of TLDs; entire dataset

Total of 171,212,578 domains (entire dataset)

This figure shows the distribution of TLDs across the entire dataset. Counted in millions.

Query: Results

By far, the most popular TLD, with 84,279,284 occurrences, is `.com`, making up 49.2% of the dataset. This can be attributed to the fact that `.com` is considered the default TLD for the internet and is the one exuding the most trust.

However, a surprising appearance as the eighth most popular TLD is `.xyz`, with 2,355,157 occurrences, or 1.4%, given that it was only registered in 2014 [Int24] and therefore has no long-established trust nor any connection to a country or region with a sizeable population.

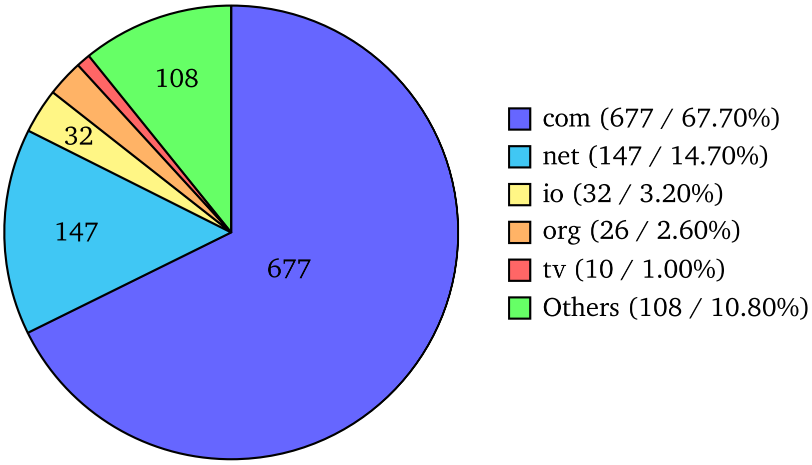

Figure 4: Distribution of TLDs; Cloudflare Top 1000

Total of 1,000 domains (*Cloudflare Top 1000* subset)

This figure shows the distribution of TLDs across the *Cloudflare Top 1000* subset.

Query: Results

Looking at the *Cloudflare Top 1000* subset, the `.com` TLD is again by far the most popular, with 677 occurrences.

Manually inspecting domains on the *Cloudflare Top 1000* list reveals that domains using the `.net` and `.io` TLDs, which are found in the second and third spots, are generally used for technical purposes rather than being directly consumer-facing. For example, the domains `edgesuite.net` (Akamai), `cloudflare.net` (Cloudflare), `cloudfront.net` (Amazon), and `fastly.net` (Fastly) are operated by CDN providers for use by their customers. `docker.io` is used for the official Docker image registry, and `nflxvideo.net` is used by Netflix for video streaming.

4.2.2 Distribution of Registrars

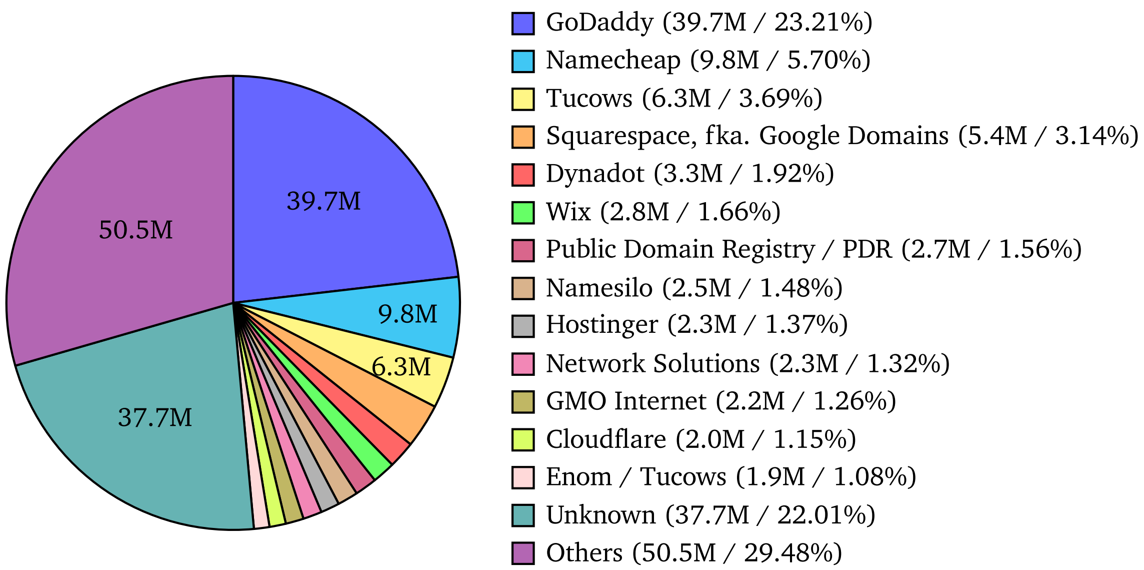

Figure 5: Distribution of Registrars; entire dataset

Total of 171,212,578 domains (entire dataset)

This figure shows the distribution of registrars as reported by WHOIS across the entire dataset. Because many registrars operate under different legal entities for different TLDs (e.g., *GoDaddy.com, LLC* and *GoDaddy Online Services Cayman Islands Ltd.*), results were aggregated based on the brand name shown in the diagram. Counted in millions.

Queries: Discovery, Results

In the dataset used, GoDaddy has by far the greatest market share. In practice, however, the company's reach is even greater since domains may be registered with subsidiaries operating under a different name and brand. For example, registrar *Mesh Digital Limited* refers to a company acquired by British hosting company Host Europe [Lüc12], which itself was later acquired by GoDaddy [Ble16].

For this diagram and further analysis throughout this thesis, only shallow matching was performed. E.g., *Squarespace Domains LLC* and *Squarespace Domains II LLC* both equate to Squarespace Domains. However, domains from *Wild West Domains, LLC* were not counted towards GoDaddy. The reason is that while two registrars may be owned by the same company, completely different external names hint at different internal procedures, and therefore, the given registrars should be treated as separate.

Registrars listed here are not necessarily selling domain names under their own brand. Tucows, for example, while also operating its own brand *Hover*[11], provides registration services for resellers like Vercel [Ver24a]. These resellers may have completely different processes and use their own authoritative nameservers, therefore setting a different path for DNSSEC adoption.

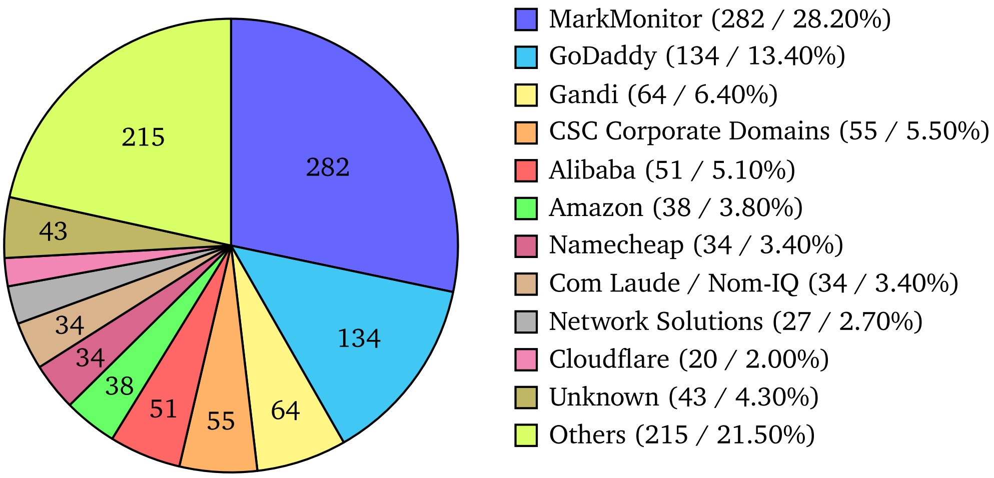

Figure 6: Distribution of Registrars; Cloudflare Top 1000

Total of 1,000 domains (*Cloudflare Top 1000* subset)

This figure shows the distribution of registrars as reported by WHOIS across the *Cloudflare Top 1000* subset. Because many registrars operate under different legal entities for different TLDs (e.g., *GoDaddy.com, LLC* and *GoDaddy Online Services Cayman Islands Ltd.*), results were aggregated based on the brand name shown in the diagram.

Queries: Discovery, Results

Cloudflare Top 1000 domains show a greatly different distribution of registrars compared to the entire dataset. Registrars like MarkMonitor, CSC Corporate Domains, and Com Laude specialize in managing high-value domains for enterprise customers. MarkMonitor is the most popular registrar within the subset, with 28.2%, and even manages 53% of *Cloudflare Top 100* domains[12]. This

[11] `https://www.hover.com/`

[12] Query: `SELECT COUNT(*) FROM domains WHERE domain IN (SELECT domain FROM top_100) AND LOWER(registrar) LIKE '%markmonitor%'`

puts the company in a very interesting position due to it holding the keys to many of the world's most trafficked domains.

Alibaba and Amazon also make a first appearance here. This can most likely be attributed to their offerings being integrated into their widely popular cloud platforms.

4.2.3 Distribution of Authoritative Nameservers

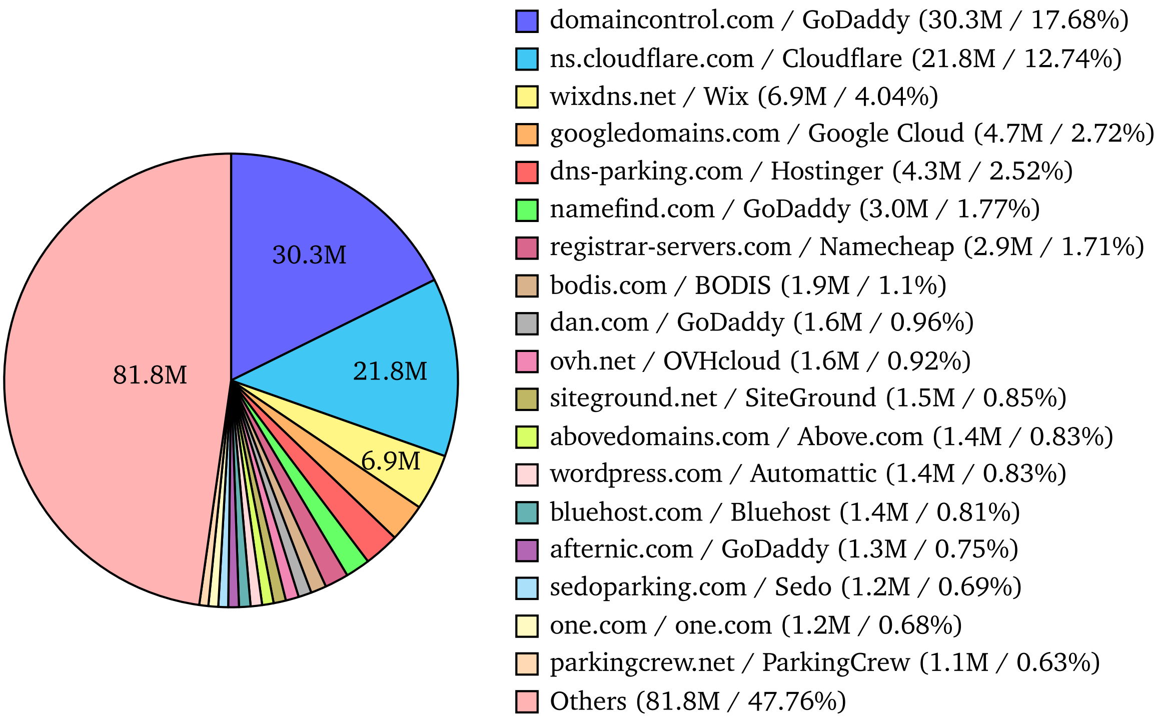

Figure 7: Distribution of Primary Authoritative Nameservers; entire dataset

Total of 171,212,578 domains (entire dataset)

This figure shows the distribution of primary authoritative nameservers across the entire dataset. Since nameservers generally use subdomains, nameservers were grouped by their parent domain, e.g., `domaincontrol.com` includes `ns00.domaincontrol.com`, `ns01.domaincontrol.com`, and others. Counted in millions.

Queries: Discovery, Results

The nameservers listed in the figure are all either offered for free with domain orders from the respective provider or offered by domain parking and resale services. Included nameservers are generally provisioned automatically by the registrar, with a default configuration deemed suitable.

Unsurprisingly, given their large market share as a registrar, many domains use GoDaddy's nameservers. Furthermore, domains registered with subsidiaries like Host Europe use `domaincontrol.com` by default as well.

While Cloudflare also offers registration services, the majority of domains using its nameservers are registered elsewhere, as can be seen from the significantly lower number of registrations in 4.2.2. This can certainly be attributed to their attractive free CDN and security offerings[13].

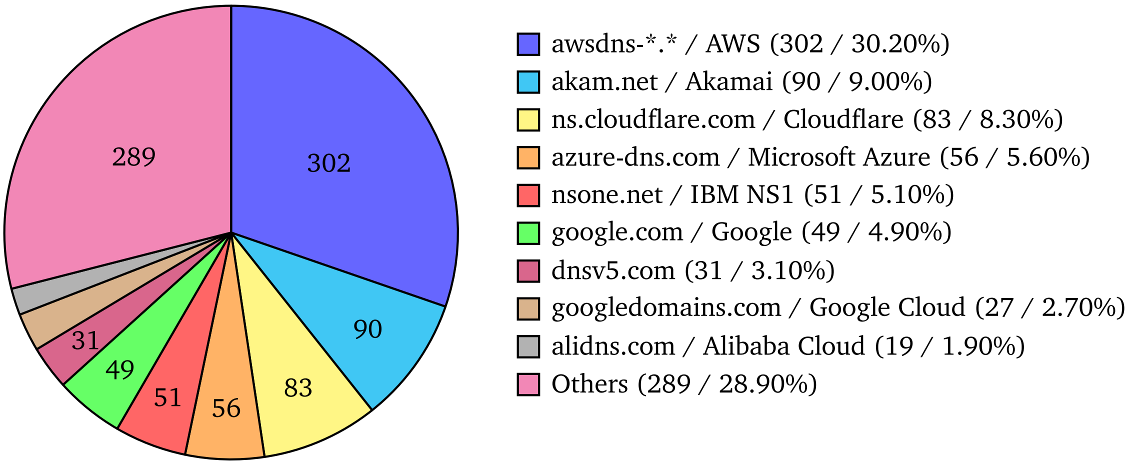

Figure 8: Distribution of Primary Authoritative Nameservers; Cloudflare Top 1000

Total of 1,000 domains (*Cloudflare Top 1000* subset)

This figure shows the distribution of primary authoritative nameservers across the *Cloudflare Top 1000* subset. Since nameservers generally use subdomains, nameservers were grouped by their parent domain, e.g., `akam.net` includes `a1-71.akam.net`, `a12-64.akam.net`, and others. Furthermore, AWS uses both multiple apex domains *and* multiple subdomains for their nameservers, like `ns-156.awsdns-19.com` and `ns-1002.awsdns-61.net`. These are all grouped as `awsdns-*.*`.

Queries: Discovery, Results

Unlike most other domains, domains in *Cloudflare Top 1000*, with only a few exceptions, choose premium nameservers separate from their registrar. Therefore, these nameservers had to be manually configured instead of being automatically configured by the registrar.

[13] `https://blog.cloudflare.com/cloudflares-commitment-to-free/`

4.2.4 Domain Registrations by Year

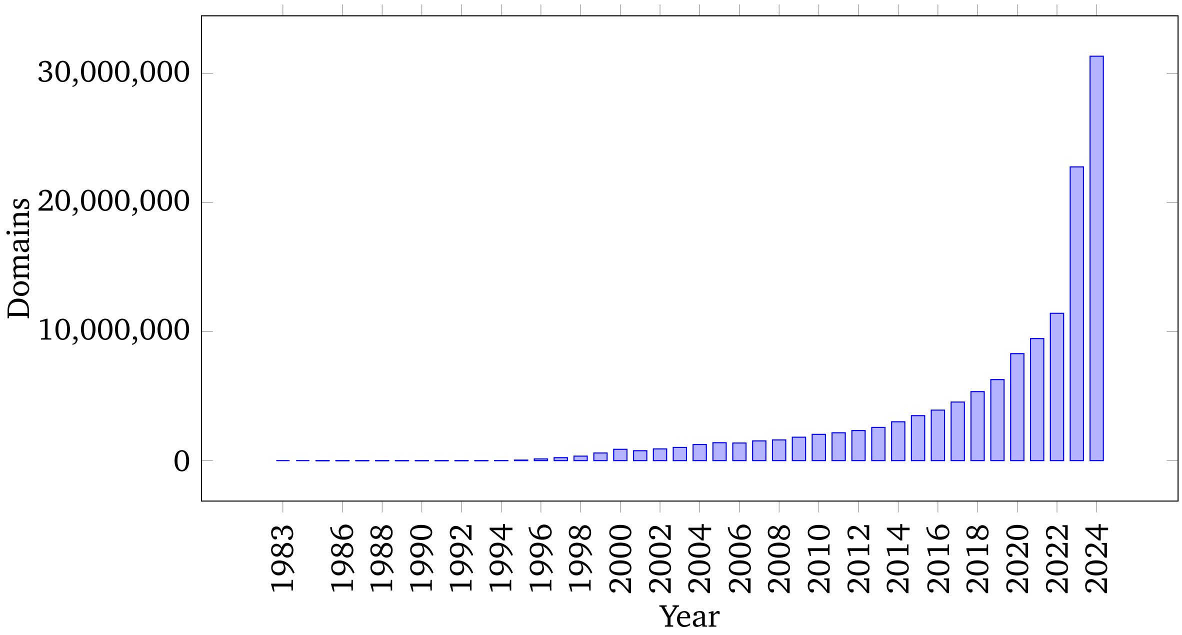

Figure 9: Distribution of Original Domain Creation Years

Total of 132,935,495 domains

This figure shows the distribution of creation years as reported by WHOIS across the entire dataset. 132,935,495 domains are shown; 38,277,097 were excluded because no valid creation date could be extracted from WHOIS.

Query: Results

A large chunk of the domains in the dataset was registered in recent years. In 2024 and 2023, a known total of 37,780,864 domains were registered, making up 41% of domains with known creation dates. Therefore, these domains will be using modern default configurations from their respective registrars, which can be assumed to be more likely to support DNSSEC or adopt it by default.

4.3 Adoption Rates

In this subsection, we look at the adoption rates of DNSSEC and break down the results by various categories.

4.3.1 Overall Adoption

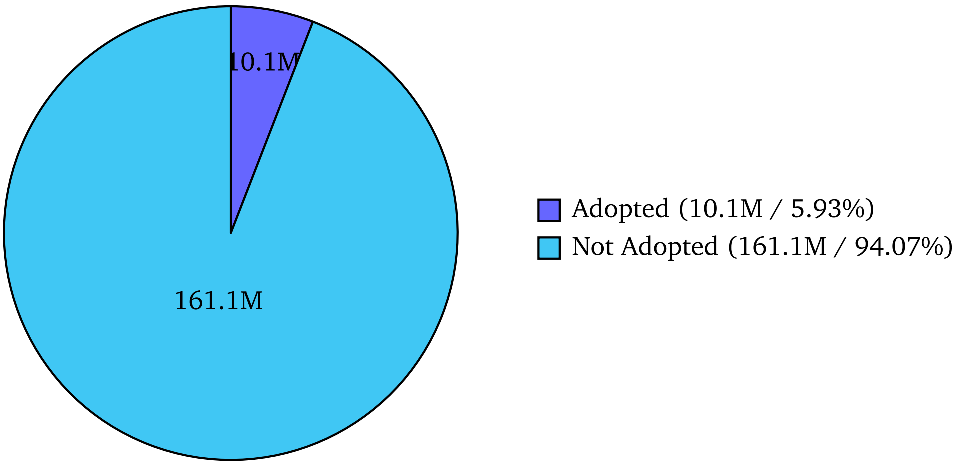

Figure 10: Overall DNSSEC Adoption

Total of 171,212,578 domains (entire dataset)

This figure shows the distribution of domains across the entire dataset by whether they have adopted DNSSEC or not. Counted in millions.

Query: Result

Across the entire dataset, only 10,148,003 domains or 5.93% have adopted DNSSEC. Given that DNSSEC was standardized 19 years ago in 2005, this is an incredibly low number.

4.3.2 Adoption by Domain Popularity

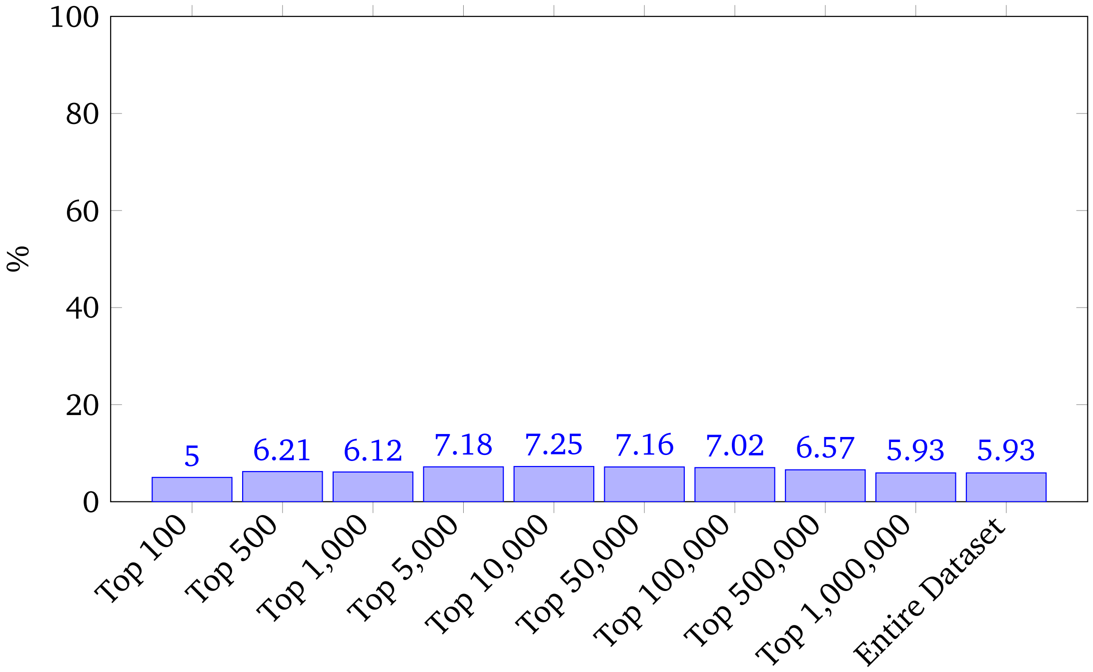

Figure 11: DNSSEC Adoption by Domain Popularity

This figure shows the adoption rates of DNSSEC by subsets as defined by *Cloudflare Top* lists. Domains from smaller subsets are also included in larger ones, e.g., *Cloudflare Top 100* is included in *Cloudflare Top 500*. *Entire Dataset* considers all 171,212,578 domains.

Query: Results

Regardless of a domain's popularity, adoption rates differ only slightly, ranging from 5.00% to 7.25%. A peak is visible in the *Cloudflare Top 10,000* subset, with adoption rates falling off to either side.

Out of the top 100 domains, only 5 domains support DNSSEC: `cloudflare-dns.com` (Cloudflare's DoH service), `cloudflare.com`, `dns.google` (Google's DoH service), `one.one` (marketing site for 1.1.1.1), and `taboola.com`. Out of these, only the last one is operated by a company providing any services related to DNS.

4.3.3 Adoption by TLD

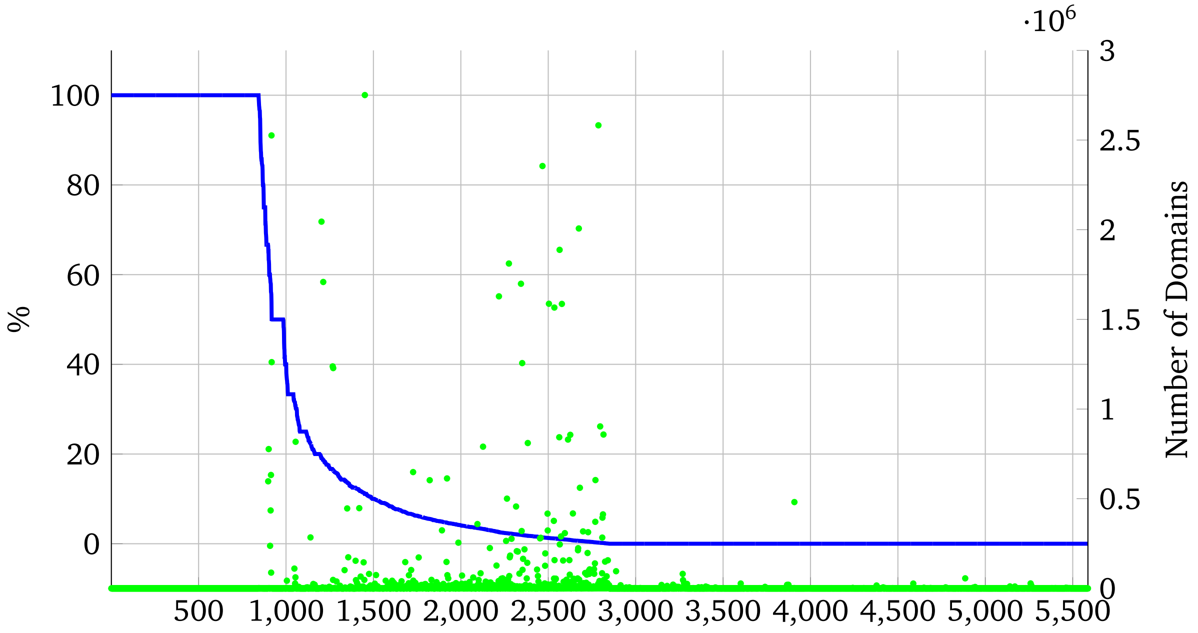

Figure 12: TLDs by Adoption Rate; entire dataset

Total of 5,587 TLDs and 171,212,578 domains

This figure shows the average adoption rates for every TLD in the entire dataset (blue line). Additionally, the number of domains for every given TLD is marked (green dots). For better readability, dots for TLDs with over 3M domains are not shown.

Query: Results

Out of the 5,587 unique TLDs in the dataset, 843 have a perfect DNSSEC adoption rate of 100%. Most of them are corporate TLDs that only have a single domain registered under them, usually `nic.tld`. Another 2,739 have an adoption rate of 0.00%. These, however, show varying total domain counts of up to 481,354 for `.ir`. Most popular TLDs are located between the two extremes, with clumping towards the lower end.

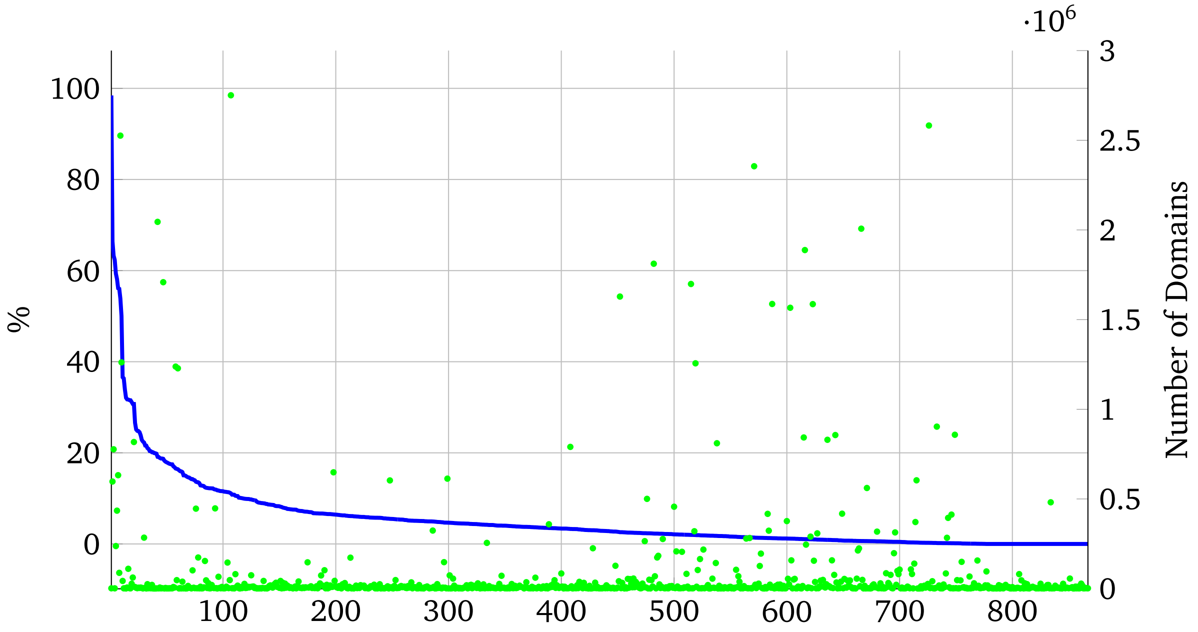

Figure 13: TLDs by Adoption Rate; min. 1,000 domains

Total of 867 TLDs and 170,902,587 domains

This figure shows the average adoption rates for every TLD with at least 1,000 domains in the dataset (blue line). Additionally, the number of domains for every given TLD is marked (green dots). For better readability, dots for TLDs with over 3M domains are not shown.

Query: Results

When filtering out all TLDs with <1,000 domains, all TLDs with a perfect adoption rate of 100% disappear. Only `.bank` comes close with 98.45%. The tail end of TLDs with 0.00% adoption rates still prevails but shrinks down to only 91 TLDs.

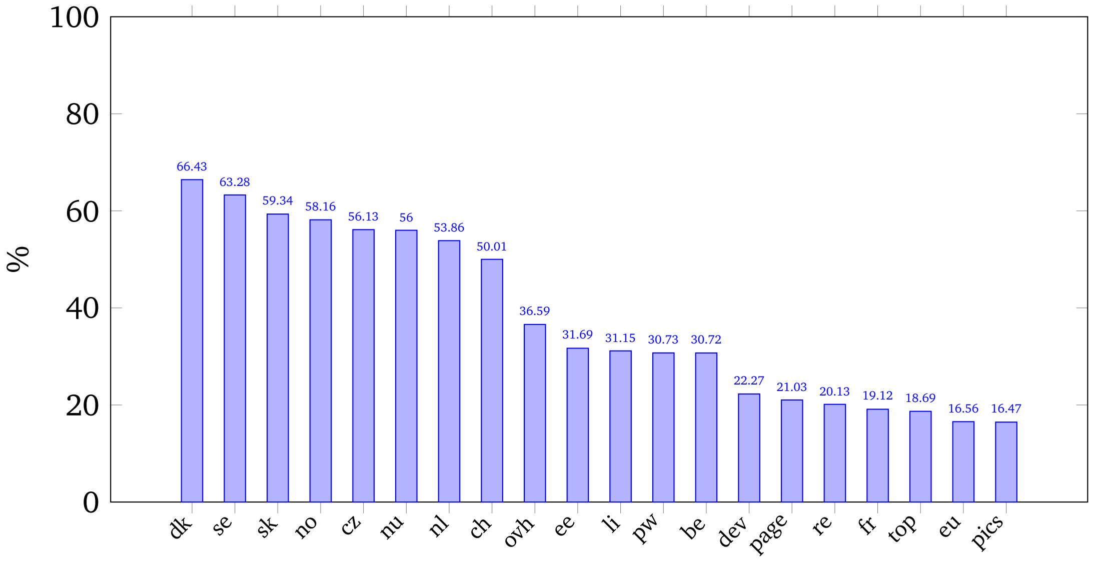

Figure 14: Top TLDs by Adoption Rate; min. 10,000 domains

171,212,578 domains (entire dataset) considered

This figure shows the adoption rates of DNSSEC for the 20 TLDs with the highest adoption rates and at least 10,000 domains across the entire dataset.

Query: Results

Zooming in further, the subset of TLDs with the highest adoption rate out of all TLDs with ≥ 10,000 domains registered displays three clear plateaus around 55%, 32%, and 20%.

The majority of TLDs here are country code top-level domains (ccTLDs), as indicated by the length of 2 characters, all of which are from European countries. The only exception is `.nu`, which is operated by The Swedish Internet Foundation, according to the Government of Niue, without any valid permission [NIC24].

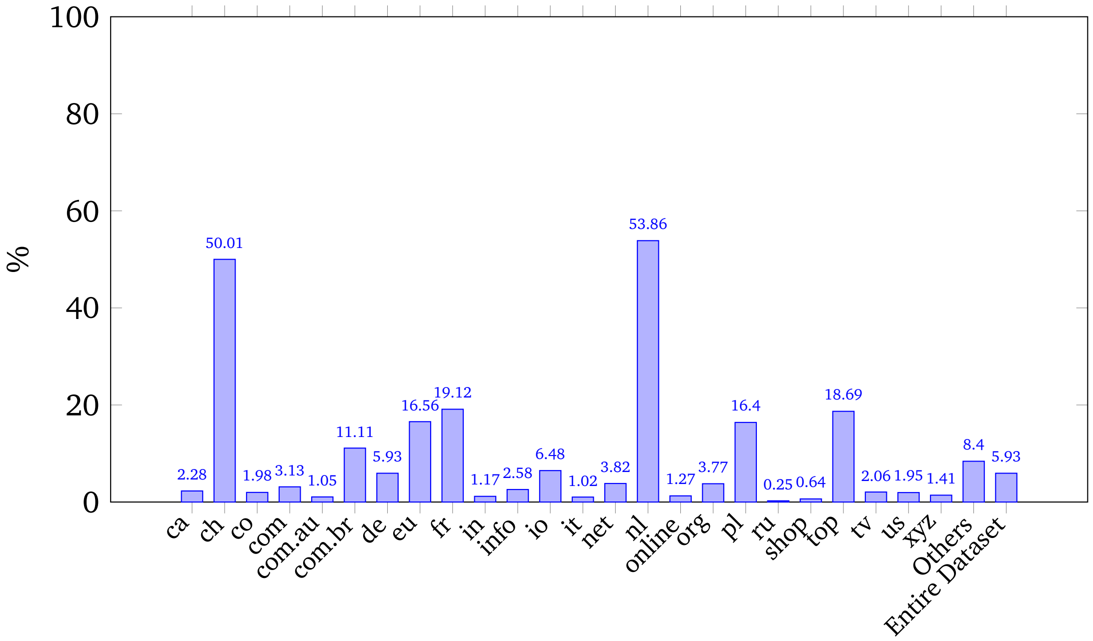

Figure 15: DNSSEC Adoption by TLD; most popular TLDs

171,212,578 domains (entire dataset) considered

This figure shows the adoption rates of DNSSEC by their TLD across the entire dataset. Included are the most popular TLDs as listed in 4.2.1. *Others* shows the average adoption rate for TLDs not explicitly listed, *Entire Dataset* shows the average for all domains. Both are calculated across domains without grouping by TLD.

Query: Results

The majority of TLDs shown here have a below-average adoption rate. Only .ch and .nl stand out as having a >50% adoption rate. A handful of TLDs such as .eu, .fr, .pl, and .top show a greatly above-average yet still low adoption rate.

Also worth mentioning is that TLDs not explicitly listed (see *Others*) combined show an above-average adoption rate.

4.3.4 Adoption by Registrar

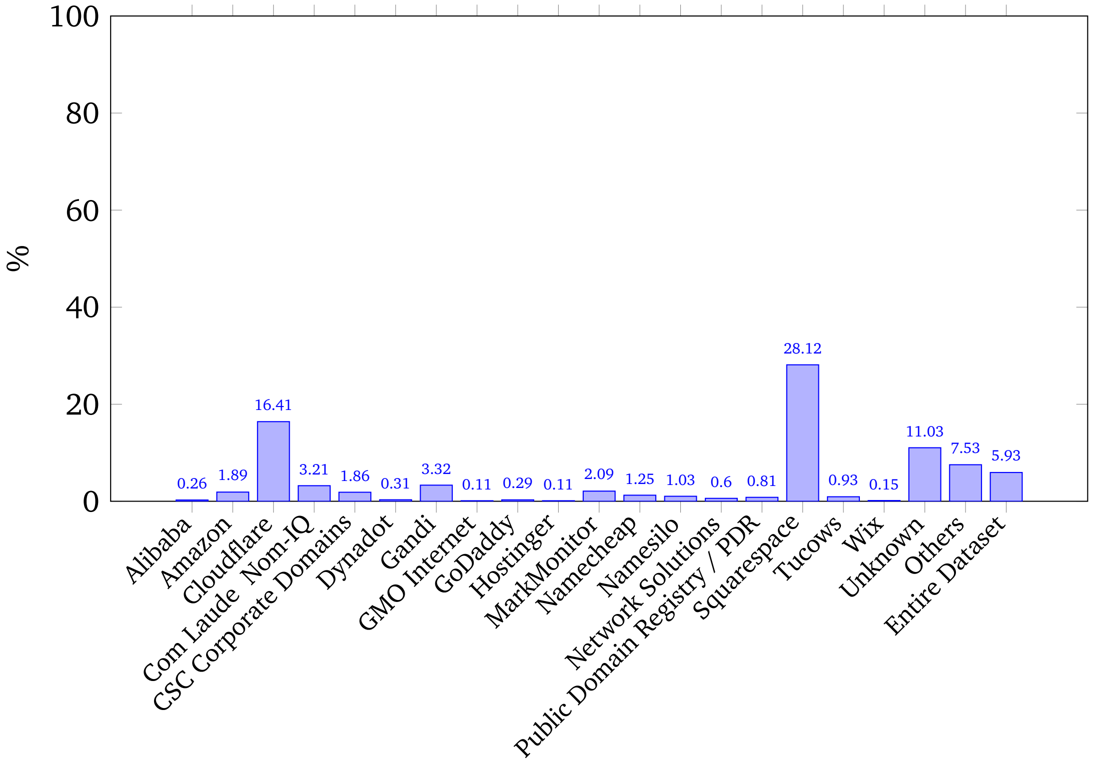

Figure 16: DNSSEC Adoption by Registrar

171,212,578 domains (entire dataset) considered

This figure shows the adoption rates of DNSSEC across the entire dataset grouped by registrars as reported by WHOIS. Included are the most popular registrars as listed in 4.2.2. *Others* shows the average adoption rate for registrars not explicitly listed, *Entire Dataset* shows the average for all domains. Both are calculated across domains without grouping by registrar.

Query: Results

Out of the most popular registrars, only *Cloudflare* and *Squarespace* show above-average adoption rates of 16.41% and 28.12%, respectively. Both registrars cater to a technically educated audience, which is more likely to enable DNSSEC. At the other end of the spectrum, registrars like *GoDaddy* or *Wix* target non-technical customers, which manifests in a very low adoption rate for domains registered with these companies.

Most notably, however, the "Unknown" set of registrars shows a rather high adoption rate of 11.03%, which is almost double the average.

4.3.5 Adoption by Authoritative Nameserver

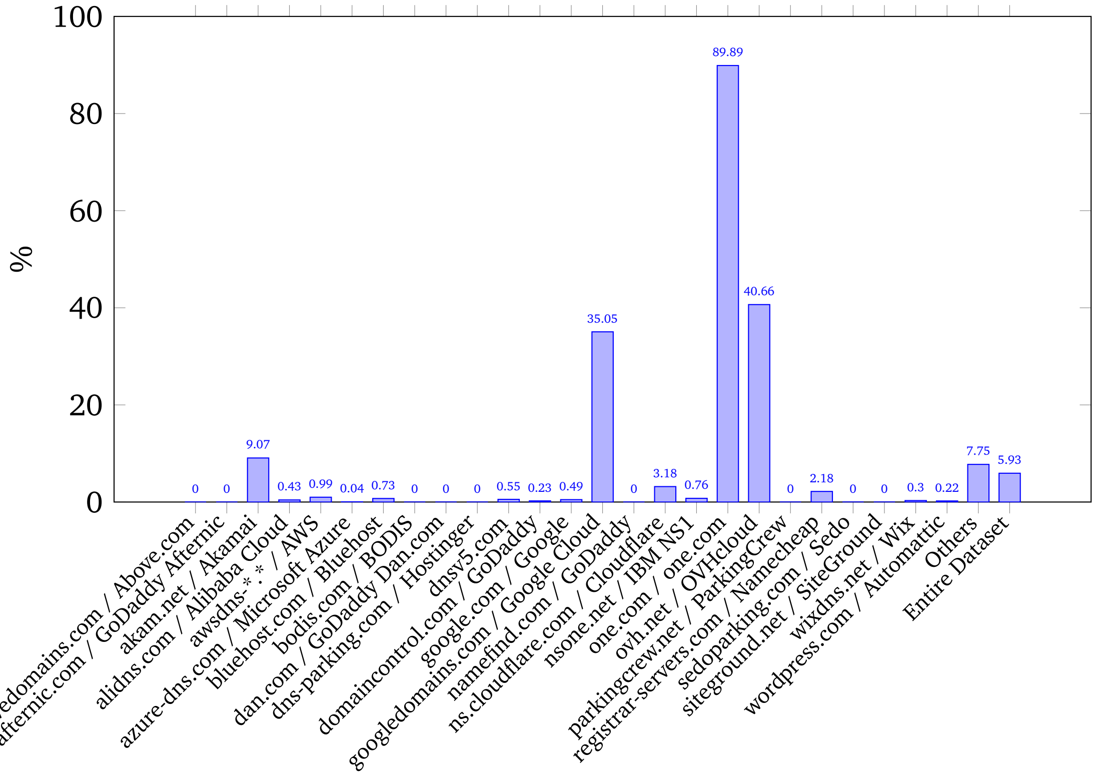

Figure 17: DNSSEC Adoption by Primary Authoritative Nameserver

171,212,578 domains (entire dataset) considered

This figure shows the adoption rates of DNSSEC across the entire dataset grouped by authoritative nameserver providers. Included are the most popular providers as listed in 4.2.3. Domains may be included twice if using multiple DNS providers. *Others* shows the average adoption rate for nameservers not explicitly listed, *Entire Dataset* shows the average for all domains. Both are calculated across domains without grouping by nameserver.

Query: Results

The included nameservers from *Squarespace* (using Google Cloud), *one.com*, and *OVHcloud* show by far the greatest adoption rates. Meanwhile, domain parking providers *Above.com*, *GoDaddy*, *BODIS*, *ParkingCrew*, and *Sedo* all show a 0.00% adoption rate.

Out of the nameserver providers chosen by *Cloudflare Top 1000* domains (see 4.2.3), only *Akamai* stands out for having an above-average adoption rate. *Cloudflare* and *Google Cloud* show slightly higher and high adoption rates respectively, but unlike Akamai, are chosen by high-traffic domains and low-traffic domains alike.

4.3.6 DNSSEC Adoption by Registration Year

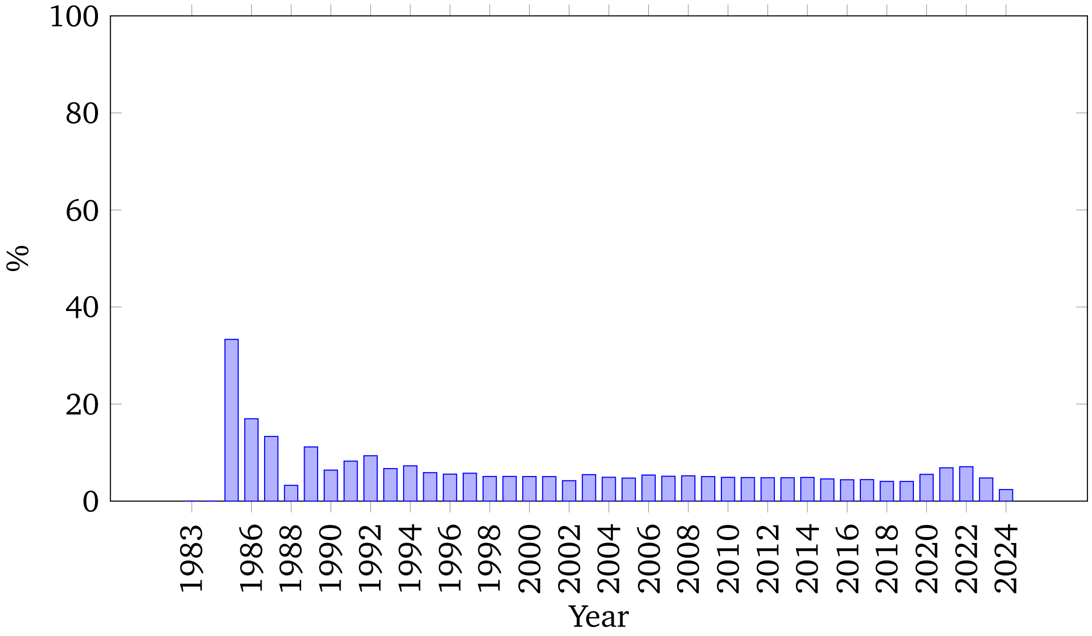

Figure 18: DNSSEC Adoption by Original Domain Creation Year

Total of 132,935,495 domains

This figure shows the adoption rates of DNSSEC across the entire dataset grouped by creation years as reported by WHOIS. 132,935,495 domains are shown; 38,277,097 were excluded because no valid creation date could be extracted from WHOIS.

Query: Results

Adoption rates by year, for the first couple of years up until roughly 1997, are all over the place. As shown in 4.2.4, the overall count of domains from the early years is very low, therefore making for a smaller sample size. From 1998 until 2019, a slight downward trend is visible, ending in a short-lived peak around 2021 and 2022. Domains registered in 2024 have the lowest adoption rate across all years of reasonable sample size of just 2.39%.

5 Discussion

5.1 Findings

When looking at the various breakdowns of the dataset, the following insights can be extracted:

1. Overall adoption, even 20 years later, still remains very low,
2. however, across various categories (TLDs, registrars, nameservers), outliers showing high adoption rates can be identified.

5.2 Contributions Helping Adoption

Looking at the results from 4.3.3, some of the TLDs with the highest adoption rates (`.dk`, `.se`, and `.nu`) support automatic configuration of DNSSEC at the registry [Clo23] [The24]. This allows nameserver providers to configure DNSSEC on the domain owners' behalf, thereby circumventing the lack of incentive or education. Besides, these efforts demonstrate greater interest in pushing DNSSEC and its adoption compared to other registries.

Registries for `.nl` and `.se` offered a small discount on the registration fee to registrars for every domain that adopted DNSSEC [Le+18], which led to many registrars automatically configuring DNSSEC for their customers.

Evidently, these efforts have resulted in good, if not amazing, adoption rates, and these registries have proven that adoption rates of >50% *are* achievable with the right structure in place.

5.3 Adoption Hurdles

Some TLDs do not support registering `DS` records at the registry, therefore preventing all subordinate domains from adopting DNSSEC. The United Arab Emirates' TLD `.ae` is, with 95,788 occurrences, the most popular TLD in the dataset that does not support DNSSEC. While this is hindering overall adoption, the market share of TLDs without DNSSEC support is insignificant compared to other TLDs, especially `.com`.

In private communication, Bert Hubert, the original author of PowerDNS[14], suggests that DNSSEC poses a risk to availability and is therefore avoided by uptime-critical domains, especially by infrastructure operators like CDNs [Hub24]. Instead, these companies rely on TLS and monitoring CT logs to ensure that users are routed to the correct destination [Com23]. This explains why adoption rates within the *Cloudflare Top 100* and beyond are below average (see 4.3.2), despite those companies having a lot of in-house expertise.

Unlike the cases mentioned in 5.2, enabling DNSSEC often still requires manual intervention by domain owners, even if it is just pressing a single button, like for domains registered with Namecheap [Nam24]. When using third-party nameservers, this becomes more complicated, as `DS` records need to be manually configured at the registrar.

GoDaddy sees some of the worst adoption rates, at just 0.23% (see 4.3.4). This is not surprising given that the registrar is selling DNSSEC as a paid add-on for more than the price of a domain registration [GoD24]. Judging by their marketing material, GoDaddy caters to an audience that wants to minimize effort spent on anything digital. Given that there is no widely visible indicator for DNSSEC as there is for HTTPS, most domain owners won't be aware of DNSSEC, let alone be willing to pay for it.

[14] `https://www.powerdns.com/`

6 Conclusion

While we have seen some small progress towards greater DNSSEC adoption, it would certainly be naive to believe that the standard will be widely adopted in the near future, or possibly ever. Especially given that since its standardization in 2005, over 19 years of rather little engagement have passed, there are no signs of this changing and, with only a few exceptions, no direct incentive exists for site owners or registrars to adopt DNSSEC.

Initiatives from some registries and registrars have proven that high adoption rates *are* possible if DNSSEC is configured without manual intervention. Sadly, this is not due to a lack of initiative [IS18] but rather a lack of collaboration and acceptance of these initiatives. Instead, companies like GoDaddy are actively pushing against DNSSEC adoption through intentional anti-consumer behavior, locking DNSSEC and other basic security features behind a costly upgrade.

The survey treated DNSSEC adoption as boolean, as determined by whether Cloudflare's resolver was able to validate the response it got, using the set of algorithms it supports. A more complete picture could be achieved by also considering broken and incomplete configurations. `DS` and `DNSKEY` records were collected but were not used in the analysis. These data points might be used to find domains where the necessary records for DNSSEC are present but where validation fails.

Additionally, inaccuracies were introduced into the survey's results by (1) the dataset only covering around half of all domains, gathered from a source that implies certain security awareness, (2) the fact that domains were surveyed over the span of a couple of weeks instead of all at once, and (3) a sizeable number of records lack information about registrar and creation date, either because the utilized WHOIS library wasn't set up to parse the response format or because aggressive rate limits were applied on WHOIS queries by the registry, which could not be circumvented with reasonable efforts.

The question of why the majority of the largest sites do not use DNSSEC, despite having appropriate in-house knowledge and resources, remains unanswered. This thesis presented the argument of DNSSEC posing a risk to availability, a theory that needs further investigation and validation. Such efforts would most likely also reveal additional weaknesses and problems of DNSSEC. Conversely, there are plenty of supporters of DNSSEC who are willing to put up with the potential risks involved. These outliers, especially in TLDs and registrars, should be investigated further to understand their reasoning beyond monetary incentives.

In addition to the figures presented in this thesis, further insights such as adoption rates across combinations of registrars and nameservers could be extracted from the same dataset.

The full dataset is available for download under a CC BY 4.0 license. The source code for scraping the CT logs as well as for surveying the domains is also publicly available.

A SQL Queries

A.1 Distribution of TLDs; entire dataset

```
SELECT
  count(*) AS count,
  tld
FROM
  domains
GROUP BY
  tld
ORDER BY
  count DESC,
  tld
```

A.2 Distribution of TLDs; Cloudflare Top 1000

```
SELECT
  count(*) AS count,
  tld
FROM
  domains
WHERE
  domain IN (SELECT domain FROM top_1000)
GROUP BY
  tld
ORDER BY
  count DESC,
  tld
```

A.3 Distribution of Registrars; entire dataset - Discovery

```
SELECT
  count(*) AS count,
  registrar
FROM
  domains
GROUP BY
  registrar
ORDER BY
  count DESC
```

A.4 Distribution of Registrars; entire dataset - Results

```
SELECT
  count(*) AS count,
  multiIf(
    lower(registrar) LIKE '%daddy%', 'GoDaddy',
    lower(registrar) LIKE '%namecheap%', 'Namecheap',
    lower(registrar) LIKE '%tucows%', 'Tucows',
    lower(registrar) LIKE '%squarespace%', 'Squarespace, fka. Google Domains',
    lower(registrar) LIKE '%wix%', 'Wix',
    lower(registrar) LIKE '%dynadot%', 'Dynadot',
    lower(registrar) LIKE '%pdr%', 'Public Domain Registry / PDR',
    lower(registrar) LIKE '%namesilo%', 'Namesilo',
    lower(registrar) LIKE '%hostinger%', 'Hostinger',
    lower(registrar) LIKE '%network solutions%', 'Network Solutions',
    lower(registrar) LIKE '%gmo internet%', 'GMO Internet',
    lower(registrar) LIKE '%cloudflare%', 'Cloudflare',
    lower(registrar) LIKE '%enom%', 'Enom / Tucows',
    registrar = 'unknown', 'Unknown',
    'Others'
  ) AS provider
FROM
  domains
GROUP BY
  provider
ORDER BY
  count DESC
```

A.5 Distribution of Registrars; Cloudflare Top 1000 - Discovery

```
SELECT
  count(*) AS count,
  registrar
FROM
  domains
WHERE
  domain IN (SELECT domain FROM top_1000)
GROUP BY
  registrar
ORDER BY
  count DESC
```

A.6 Distribution of Registrars; Cloudflare Top 1000 - Results

```
SELECT
  count(*) AS count,
  multiIf(
    lower(registrar) LIKE '%markmonitor%', 'MarkMonitor',
    lower(registrar) LIKE '%daddy%', 'GoDaddy',
    lower(registrar) LIKE '%csc corp%', 'CSC Corporate Domains',
    lower(registrar) LIKE '%gandi%', 'Gandi',
    lower(registrar) LIKE '%alibaba%', 'Alibaba',
    lower(registrar) LIKE '%amazon%', 'Amazon',
    lower(registrar) LIKE '%nom-iq%', 'Com Laude / Nom-IQ',
    lower(registrar) LIKE '%namecheap%', 'Namecheap',
    lower(registrar) LIKE '%network solutions%', 'Network Solutions',
    lower(registrar) LIKE '%cloudflare%', 'Cloudflare',
    registrar = 'unknown', 'Unknown',
    'Others'
  ) AS provider
FROM
  domains
WHERE
  domain IN (SELECT domain FROM top_1000)
GROUP BY
  provider
ORDER BY
  count DESC
```

A.7 Distribution of Primary Authoritative Nameservers; entire dataset - Discovery

```
SELECT
  domain,
  count(*) AS count
FROM (
  SELECT
    arrayStringConcat(arraySlice(splitByChar('.', ns), 2), '.') AS domain
  FROM (
    SELECT
      arrayElement(records_ns, 1) AS ns
    FROM
      domains
    WHERE
      length(records_ns) > 0
  )
  WHERE ns != ''
)
GROUP BY
  domain
ORDER BY
  count DESC
```

A.8 Distribution of Primary Authoritative Nameservers; entire dataset - Results

```
SELECT
  count(*) AS count,
  multiIf(
    like(lower(records_ns[1]), '%.domaincontrol.com.'), 'domaincontrol.com / GoDaddy',
    like(lower(records_ns[1]), '%.ns.cloudflare.com.'), 'ns.cloudflare.com / Cloudflare',
    like(lower(records_ns[1]), '%.wixdns.net.'), 'wixdns.net / Wix',
    like(lower(records_ns[1]), '%.googledomains.com.'), 'googledomains.com / Google Cloud',
    like(lower(records_ns[1]), '%.dns-parking.com.'), 'dns-parking.com / Hostinger',
    like(lower(records_ns[1]), '%.namefind.com.'), 'namefind.com / GoDaddy',
    like(lower(records_ns[1]), '%.registrar-servers.com.'), 'registrar-servers.com / Namecheap',
    like(lower(records_ns[1]), '%.bodis.com.'), 'bodis.com / BODIS',
    like(lower(records_ns[1]), '%.dan.com.'), 'dan.com / GoDaddy',
    like(lower(records_ns[1]), '%.ovh.net.'), 'ovh.net / OVHcloud',
    like(lower(records_ns[1]), '%.siteground.net.'), 'siteground.net / SiteGround',
    like(lower(records_ns[1]), '%.abovedomains.com.'), 'abovedomains.com / Above.com',
    like(lower(records_ns[1]), '%.wordpress.com.'), 'wordpress.com / Automattic',
    like(lower(records_ns[1]), '%.bluehost.com.'), 'bluehost.com / Bluehost',
    like(lower(records_ns[1]), '%.afternic.com.'), 'afternic.com / GoDaddy',
    like(lower(records_ns[1]), '%.sedoparking.com.'), 'sedoparking.com / Sedo',
    like(lower(records_ns[1]), '%.one.com.'), 'one.com / one.com',
    like(lower(records_ns[1]), '%.parkingcrew.net.'), 'parkingcrew.net / ParkingCrew',
    'Others'
  ) AS provider
FROM
  domains
GROUP BY
  provider
ORDER BY
  count DESC
```

A.9 Distribution of Primary Authoritative Nameservers; Cloudflare Top 1000 - Discovery

```
SELECT
  domain,
  count(*) AS count
FROM (
  SELECT
    arrayStringConcat(arraySlice(splitByChar('.', ns), 2), '.') AS domain
  FROM (
    SELECT
      arrayElement(records_ns, 1) AS ns
    FROM
      domains
    WHERE
      domain IN (SELECT domain FROM top_1000) AND
      length(records_ns) > 0
  )
  WHERE ns != ''
)
GROUP BY
  domain
ORDER BY
  count DESC
```

A.10 Distribution of Primary Authoritative Nameservers; Cloudflare Top 1000 - Results

```
SELECT
  count(*) AS count,
  multiIf(
    like(lower(records_ns[1]), '%.awsdns-%.%.'), 'awsdns-*.* / AWS',
    like(lower(records_ns[1]), '%.akam.net.'), 'akam.net / Akamai',
    like(lower(records_ns[1]), '%.ns.cloudflare.com.'), 'ns.cloudflare.com / Cloudflare',
    like(lower(records_ns[1]), '%.nsone.net.'), 'nsone.net / IBM NS1',
    like(lower(records_ns[1]), '%.azure-dns.com.'), 'azure-dns.com / Microsoft Azure',
    like(lower(records_ns[1]), '%.google.com.'), 'google.com / Google',
    like(lower(records_ns[1]), '%.dnsv5.com.'), 'dnsv5.com',
    like(lower(records_ns[1]), '%.googledomains.com.'), 'googledomains.com / Google Cloud',
    like(lower(records_ns[1]), '%.alidns.com.'), 'alidns.com / Alibaba Cloud',
    'Others'
  ) AS provider
FROM
  domains
WHERE
  domain IN (SELECT domain FROM top_1000)
GROUP BY
  provider
ORDER BY
  count DESC
```

A.11 Distribution of Original Domain Creation Years

```
SELECT
  toYear(created_at) as year,
  count(*) as count
FROM
  domains
GROUP BY
  year
ORDER BY
  year
```

A.12 Overall DNSSEC Adoption

```
SELECT
  dnssec,
  count(*)
FROM
  domains
GROUP BY
  dnssec
```

A.13 DNSSEC Adoption by Domain Popularity

```
SELECT
  countIf(domain IN (SELECT domain FROM top_100) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_100)) AS top_100,
  countIf(domain IN (SELECT domain FROM top_500) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_500)) AS top_500,
  countIf(domain IN (SELECT domain FROM top_1000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_1000)) AS top_1000,
  countIf(domain IN (SELECT domain FROM top_5000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_5000)) AS top_5000,
  countIf(domain IN (SELECT domain FROM top_10000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_10000)) AS top_10000,
  countIf(domain IN (SELECT domain FROM top_50000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_50000)) AS top_50000,
  countIf(domain IN (SELECT domain FROM top_100000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_100000)) AS top_100000,
  countIf(domain IN (SELECT domain FROM top_500000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_500000)) AS top_500000,
  countIf(domain IN (SELECT domain FROM top_1000000) AND dnssec = 't')
    / countIf(domain IN (SELECT domain FROM top_1000000)) AS top_1000000,
  countIf(dnssec = 't') / count(*) AS entire_dataset
FROM
  domains
```

A.14 TLDs by Adoption Rate; entire dataset

```
  SELECT
  row_number() OVER (ORDER BY adoption_percentage DESC, tld) AS position,
  tld,
  round(100.0 * SUM(CASE WHEN dnssec = 't' THEN 1 ELSE 0 END) / count(*), 2) AS adoption_percentage,
  count(*) AS total_domains
FROM
  domains
GROUP BY
  tld
```

A.15 TLDs by Adoption Rate; min. 1,000 domains

```
SELECT
  row_number() OVER (ORDER BY adoption_percentage DESC, tld) AS position,
  tld,
  round(100.0 * SUM(CASE WHEN dnssec = 't' THEN 1 ELSE 0 END) / count(*), 2) AS adoption_percentage,
  count(*) AS total_domains
FROM
  domains
GROUP BY
  tld
HAVING
  count(*) > 1000
```

A.16 Top TLDs by Adoption Rate; min. 10,000 domains

```
SELECT
  tld,
  round(100.0 * SUM(CASE WHEN dnssec = 't' THEN 1 ELSE 0 END) / count(*), 2) AS adoption_percentage
FROM
  domains
GROUP BY
  tld
HAVING
  count(*) > 10000
ORDER BY
  adoption_percentage DESC
LIMIT
  20
```

A.17 DNSSEC Adoption by TLD; most popular TLDs

```
SELECT
  countIf(tld = 'ca' AND dnssec = 't') / countIf(tld = 'ca') AS "ca",
  countIf(tld = 'ch' AND dnssec = 't') / countIf(tld = 'ch') AS "ch",
  countIf(tld = 'co' AND dnssec = 't') / countIf(tld = 'co') AS "co",
  countIf(tld = 'com' AND dnssec = 't') / countIf(tld = 'com') AS "com",
  countIf(tld = 'com.au' AND dnssec = 't') / countIf(tld = 'com.au') AS "com.au",
  countIf(tld = 'com.br' AND dnssec = 't') / countIf(tld = 'com.br') AS "com.br",
  countIf(tld = 'de' AND dnssec = 't') / countIf(tld = 'de') AS "de",
  countIf(tld = 'eu' AND dnssec = 't') / countIf(tld = 'eu') AS "eu",
  countIf(tld = 'fr' AND dnssec = 't') / countIf(tld = 'fr') AS "fr",
  countIf(tld = 'in' AND dnssec = 't') / countIf(tld = 'in') AS "in",
  countIf(tld = 'info' AND dnssec = 't') / countIf(tld = 'info') AS "info",
  countIf(tld = 'io' AND dnssec = 't') / countIf(tld = 'io') AS "io",
  countIf(tld = 'it' AND dnssec = 't') / countIf(tld = 'it') AS "it",
  countIf(tld = 'net' AND dnssec = 't') / countIf(tld = 'net') AS "net",
  countIf(tld = 'nl' AND dnssec = 't') / countIf(tld = 'nl') AS "nl",
  countIf(tld = 'online' AND dnssec = 't') / countIf(tld = 'online') AS "online",
  countIf(tld = 'org' AND dnssec = 't') / countIf(tld = 'org') AS "org",
  countIf(tld = 'pl' AND dnssec = 't') / countIf(tld = 'pl') AS "pl",
  countIf(tld = 'ru' AND dnssec = 't') / countIf(tld = 'ru') AS "ru",
  countIf(tld = 'shop' AND dnssec = 't') / countIf(tld = 'shop') AS "shop",
  countIf(tld = 'top' AND dnssec = 't') / countIf(tld = 'top') AS "top",
  countIf(tld = 'tv' AND dnssec = 't') / countIf(tld = 'tv') AS "tv",
  countIf(tld = 'us' AND dnssec = 't') / countIf(tld = 'us') AS "us",
  countIf(tld = 'xyz' AND dnssec = 't') / countIf(tld = 'xyz') AS "xyz",
  countIf(tld NOT IN (
    'ca', 'ch', 'co', 'com.au', 'com.br', 'com', 'de', 'eu', 'fr', 'in', 'info', 'io',
    'it', 'net', 'nl', 'online', 'org', 'pl', 'ru', 'shop', 'top', 'tv', 'us', 'xyz'
  ) AND dnssec = 't') / countIf(tld NOT IN (
    'ca', 'ch', 'co', 'com.au', 'com.br', 'com', 'de', 'eu', 'fr', 'in', 'info', 'io',
    'it', 'net', 'nl', 'online', 'org', 'pl', 'ru', 'shop', 'top', 'tv', 'us', 'xyz'
  )) AS Others,
  countIf(dnssec = 't') / count(*) AS "Entire Dataset"
FROM
  domains
```

A.18 DNSSEC Adoption by Registrar

```
SELECT
  countIf(lower(registrar) LIKE '%alibaba%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%alibaba%') AS "Alibaba",
  countIf(lower(registrar) LIKE '%amazon%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%amazon%') AS "Amazon",
  countIf(lower(registrar) LIKE '%cloudflare%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%cloudflare%') AS "Cloudflare",
  countIf(lower(registrar) LIKE '%nom-iq%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%nom-iq%') AS "Com Laude / Nom-IQ",
  countIf(lower(registrar) LIKE '%csc corp%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%csc corp%') AS "CSC Corporate Domains",
  countIf(lower(registrar) LIKE '%dynadot%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%dynadot%') AS "Dynadot",
  countIf(lower(registrar) LIKE '%gandi%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%gandi%') AS "Gandi",
  countIf(lower(registrar) LIKE '%gmo internet%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%gmo internet%') AS "GMO Internet",
  countIf(lower(registrar) LIKE '%daddy%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%daddy%') AS "GoDaddy",
  countIf(lower(registrar) LIKE '%hostinger%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%hostinger%') AS "Hostinger",
  countIf(lower(registrar) LIKE '%markmonitor%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%markmonitor%') AS "MarkMonitor",
  countIf(lower(registrar) LIKE '%namecheap%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%namecheap%') AS "Namecheap",
  countIf(lower(registrar) LIKE '%namesilo%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%namesilo%') AS "Namesilo",
  countIf(lower(registrar) LIKE '%network solutions%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%network solutions%') AS "Network Solutions",
  countIf(lower(registrar) LIKE '%pdr%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%pdr%') AS "Public Domain Registry / PDR",
  countIf(lower(registrar) LIKE '%squarespace%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%squarespace%') AS "Squarespace",
  countIf(lower(registrar) LIKE '%tucows%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%tucows%') AS "Tucows",
  countIf(lower(registrar) LIKE '%wix%' AND dnssec = 't')
    / countIf(lower(registrar) LIKE '%wix%') AS "Wix",
  countIf(registrar = 'unknown' AND dnssec = 't')
    / countIf(registrar = 'unknown') AS "Unknown",
  countIf(NOT has(arrayMap(x -> lower(registrar) LIKE x, [
    '%alibaba%', '%amazon%', '%cloudflare%', '%nom-iq%', '%csc corp%', '%dynadot%',
    '%gandi%', '%gmo internet%', '%daddy%', '%hostinger%', '%markmonitor%', '%namecheap%',
    '%namesilo%', '%network solutions%', '%pdr%', '%squarespace%', '%tucows%', '%wix%'
  ]), 1) AND registrar != 'unknown' AND dnssec = 't') / countIf(NOT has(arrayMap(x -> lower(registrar) LIKE x, [
    '%alibaba%', '%amazon%', '%cloudflare%', '%nom-iq%', '%csc corp%', '%dynadot%',
    '%gandi%', '%gmo internet%', '%daddy%', '%hostinger%', '%markmonitor%', '%namecheap%',
    '%namesilo%', '%network solutions%', '%pdr%', '%squarespace%', '%tucows%', '%wix%'
  ]), 1) AND registrar != 'unknown') AS "Others",
  countIf(dnssec = 't') / count(*) AS "Entire Dataset"
FROM
  domains
```

A.19 DNSSEC Adoption by Primary Authoritative Nameserver

```
SELECT
  countIf(arrayExists(x -> like(lower(x), '%.abovedomains.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.abovedomains.com.'), records_ns)) AS "abovedomains.com / Above.com",
  countIf(arrayExists(x -> like(lower(x), '%.afternic.com.'), records_ns) AND dnssec = 't')
```

```
    / countIf(arrayExists(x -> like(lower(x), '%.afternic.com.'), records_ns)) AS "afternic.com / GoDaddy Afternic",
  countIf(arrayExists(x -> like(lower(x), '%.akam.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.akam.net.'), records_ns)) AS "akam.net / Akamai",
  countIf(arrayExists(x -> like(lower(x), '%.alidns.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.alidns.com.'), records_ns)) AS "alidns.com / Alibaba Cloud",
  countIf(arrayExists(x -> like(lower(x), '%.awsdns-%.%.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.awsdns-%.%.'), records_ns)) AS "awsdns-*.* / AWS",
  countIf(arrayExists(x -> like(lower(x), '%.azure-dns.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.azure-dns.com.'), records_ns)) AS "azure-dns.com / Microsoft Azure",
  countIf(arrayExists(x -> like(lower(x), '%.bluehost.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.bluehost.com.'), records_ns)) AS "bluehost.com / Bluehost",
  countIf(arrayExists(x -> like(lower(x), '%.bodis.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.bodis.com.'), records_ns)) AS "bodis.com / BODIS",
  countIf(arrayExists(x -> like(lower(x), '%.dan.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.dan.com.'), records_ns)) AS "dan.com / GoDaddy Dan.com",
  countIf(arrayExists(x -> like(lower(x), '%.dns-parking.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.dns-parking.com.'), records_ns)) AS "dns-parking.com / Hostinger",
  countIf(arrayExists(x -> like(lower(x), '%.dnsv5.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.dnsv5.com.'), records_ns)) AS "dnsv5.com",
  countIf(arrayExists(x -> like(lower(x), '%.domaincontrol.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.domaincontrol.com.'), records_ns)) AS "domaincontrol.com / GoDaddy",
  countIf(arrayExists(x -> like(lower(x), '%.google.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.google.com.'), records_ns)) AS "google.com / Google",
  countIf(arrayExists(x -> like(lower(x), '%.googledomains.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.googledomains.com.'), records_ns)) AS "googledomains.com / Google Cloud",
  countIf(arrayExists(x -> like(lower(x), '%.namefind.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.namefind.com.'), records_ns)) AS "namefind.com / GoDaddy",
  countIf(arrayExists(x -> like(lower(x), '%.ns.cloudflare.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.ns.cloudflare.com.'), records_ns)) AS "ns.cloudflare.com / Cloudflare",
  countIf(arrayExists(x -> like(lower(x), '%.nsone.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.nsone.net.'), records_ns)) AS "nsone.net / IBM NS1",
  countIf(arrayExists(x -> like(lower(x), '%.one.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.one.com.'), records_ns)) AS "one.com / one.com",
  countIf(arrayExists(x -> like(lower(x), '%.ovh.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.ovh.net.'), records_ns)) AS "ovh.net / OVHcloud",
  countIf(arrayExists(x -> like(lower(x), '%.parkingcrew.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.parkingcrew.net.'), records_ns)) AS "parkingcrew.net / ParkingCrew",
  countIf(arrayExists(x -> like(lower(x), '%.registrar-servers.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.registrar-servers.com.'), records_ns)) AS "registrar-servers.com / Namecheap",
  countIf(arrayExists(x -> like(lower(x), '%.sedoparking.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.sedoparking.com.'), records_ns)) AS "sedoparking.com / Sedo",
  countIf(arrayExists(x -> like(lower(x), '%.siteground.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.siteground.net.'), records_ns)) AS "siteground.net / SiteGround",
  countIf(arrayExists(x -> like(lower(x), '%.wixdns.net.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.wixdns.net.'), records_ns)) AS "wixdns.net / Wix",
  countIf(arrayExists(x -> like(lower(x), '%.wordpress.com.'), records_ns) AND dnssec = 't')
    / countIf(arrayExists(x -> like(lower(x), '%.wordpress.com.'), records_ns)) AS "wordpress.com / Automattic",
  countIf(NOT arrayExists(ns -> arrayExists(pattern -> like(lower(ns), pattern), [
    '%.abovedomains.com.', '%.afternic.com.', '%.akam.net.', '%.alidns.com.', '%.awsdns-%.%.', '%.azure-dns.com.',
    '%.bluehost.com.', '%.bodis.com.', '%.dan.com.', '%.dns-parking.com.', '%.dnsv5.com.', '%.domaincontrol.com.',
    '%.google.com.', '%.googledomains.com.', '%.namefind.com.', '%.ns.cloudflare.com.', '%.nsone.net.', '%.one.com.',
    '%.ovh.net.', '%.parkingcrew.net.', '%.registrar-servers.com.', '%.sedoparking.com.', '%.siteground.net.',
    '%.wixdns.net.', '%.wordpress.com.'
  ]), records_ns) AND dnssec = 't') / countIf(NOT arrayExists(ns -> arrayExists(pattern -> like(lower(ns), pattern), [
    '%.abovedomains.com.', '%.afternic.com.', '%.akam.net.', '%.alidns.com.', '%.awsdns-%.%.', '%.azure-dns.com.',
    '%.bluehost.com.', '%.bodis.com.', '%.dan.com.', '%.dns-parking.com.', '%.dnsv5.com.', '%.domaincontrol.com.',
    '%.google.com.', '%.googledomains.com.', '%.namefind.com.', '%.ns.cloudflare.com.', '%.nsone.net.', '%.one.com.',
    '%.ovh.net.', '%.parkingcrew.net.', '%.registrar-servers.com.', '%.sedoparking.com.', '%.siteground.net.',
    '%.wixdns.net.', '%.wordpress.com.'
  ]), records_ns)) AS "Others",
  countIf(dnssec = 't') / count(*) AS "Entire Dataset"
FROM
  domains
```

A.20 DNSSEC Adoption by Original Domain Creation Year

```sql
SELECT
  toYear(created_at) as year,
  round(100.0 * SUM(CASE WHEN dnssec = 't' THEN 1 ELSE 0 END) / count(*), 2) AS adoption_percentage
FROM
  domains
GROUP BY
  toYear(created_at)
ORDER BY
  toYear(created_at) ASC
```

References

[3K97] 3rd, Donald E. Eastlake and Kaufman, Charles W. *Domain Name System Security Extensions*. RFC 2065. Jan. 1997. doi: 10.17487/RFC2065. url: https://www.rfc-editor.org/info/rfc2065.

[3rd99] 3rd, Donald E. Eastlake. *Domain Name System Security Extensions*. RFC 2535. Mar. 1999. doi: 10.17487/RFC2535. url: https://www.rfc-editor.org/info/rfc2535.

[87] *Domain names - implementation and specification*. RFC 1035. Nov. 1987. doi: 10.17487/RFC1035. url: https://www.rfc-editor.org/info/rfc1035.

[Aas15] Aas, Josh. *Why ninety-day lifetimes for certificates?* Blog post. Let's Encrypt, Nov. 2015. url: https://letsencrypt.org/2015/11/09/why-90-days/ (visited on 10/17/2024).

[Abl+19] Abley, Joe et al. *Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY*. RFC 8482. Jan. 2019. doi: 10.17487/RFC8482. url: https://www.rfc-editor.org/info/rfc8482.

[APN24] APNIC Labs. *DNSSEC Validation Rate*. APNIC Labs, 2024. url: https://stats.labs.apnic.net/dnssec (visited on 11/20/2024).

[Are+08] Arends, Roy et al. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*. RFC 5155. Mar. 2008. doi: 10.17487/RFC5155. url: https://www.rfc-editor.org/info/rfc5155.

[Ble16] Bleich, Holger. *US-Provider GoDaddy übernimmt die Host Europe Group*. German. Dec. 2016. url: https://www.heise.de/news/US-Provider-GoDaddy-uebernimmt-die-Host-Europe-Group-3563541.html (visited on 10/21/2024).

[Clo23] Cloudflare. *Changing Internet Standards to Build A Secure Internet*. Cloudflare, Inc., June 2023. url: https://blog.cloudflare.com/dk-dnssec/ (visited on 11/20/2024).

[Clo24a] Cloudflare. *Cloudflare Radar Glossary: Domain Rankings*. Cloudflare Developers Documentation. Cloudflare, 2024. url: https://developers.cloudflare.com/radar/glossary/#domain-rankings (visited on 10/21/2024).

[Clo24b] Cloudflare. *DNS amplification attack*. Cloudflare Learning Center. Cloudflare, Inc., 2024. url: https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/ (visited on 11/24/2024).

[Clo24c] Cloudflare. *Domain Rankings*. Cloudflare Radar Domain Rankings. Cloudflare, 2024. url: https://radar.cloudflare.com/domains (visited on 10/21/2024).

[Com23] Community, Let's Encrypt. *How come most websites do not support DNSSEC?* Let's Encrypt Community Forum. Internet Security Research Group, Oct. 2023. url: https://community.letsencrypt.org/t/how-come-most-websites-do-not-support-dnssec/214728/5 (visited on 11/20/2024).

[DNI24] DNIB Staff. *The DNIB Quarterly Report Q3 2024*. Quarterly Report. Data and Analysis of the Domain Name Industry. Domain Name Industry Brief, 2024. url: https://dnib.com/articles/the-domain-name-industry-brief-q3-2024 (visited on 11/24/2024).

[DNS09] DNSCurve Project. *Introduction to DNSCurve*. DNSCurve Project, June 2009. url: https://dnscurve.org/ (visited on 11/21/2024).

[DNS24a] DNS Institute. *Proof of Non-Existence (NSEC and NSEC3)*. DNSSEC Guide, Chapter 6. Advanced Discussions. DNS Institute, 2024. url: https://dnsinstitute.com/documentation/dnssec-guide/ch06s02.html#advanced-discussions-nsec3-salt (visited on 11/26/2024).

[DNS24b] DNSCrypt Project. *DNSCrypt: A protocol to improve DNS security*. DNSCrypt protocol and implementation information. DNSCrypt Project, 2024. url: https://www.dnscrypt.org (visited on 11/21/2024).

[GH14] Gilad, Yossi and Herzberg, Amir. “Off-Path TCP Injection Attacks”. In: *ACM Trans. Inf. Syst. Secur.* 16.4 (Apr. 2014). issn: 1094-9224. doi: 10.1145/2597173. url: https://doi.org/10.1145/2597173.

[GoD24] GoDaddy. *Premium DNS Hosting*. DNS hosting service. GoDaddy Inc., 2024. url: https://www.godaddy.com/hosting/premium-dns (visited on 11/21/2024).

[Gol+16] Goldberg, Sharon et al. *NSEC5 from Elliptic Curves: Provably Preventing DNSSEC Zone Enumeration with Shorter Responses*. Cryptology ePrint Archive, Paper 2016/083. 2016. url: https://eprint.iacr.org/2016/083.

[Goo24a] Google. *How CT works*. Certificate Transparency Project - How CT Works. 2024. url: https://certificate.transparency.dev/howctworks/ (visited on 11/17/2024).

[Goo24b] Google. *HTTPS encryption on the web*. Transparency Report. Google, 2024. url: https://transparencyreport.google.com/https/overview?hl=en (visited on 10/17/2024).

[GW03] Guðmundsson, Ólafur and Wellington, Brian. *Redefinition of DNS Authenticated Data (AD) bit*. RFC 3655. Nov. 2003. doi: 10.17487/RFC3655. url: https://www.rfc-editor.org/info/rfc3655.

[HM18] Hoffman, Paul E. and McManus, Patrick. *DNS Queries over HTTPS (DoH)*. RFC 8484. Oct. 2018. doi: 10.17487/RFC8484. url: https://www.rfc-editor.org/info/rfc8484.

[Hof23] Hoffman, Paul E. *DNS Security Extensions (DNSSEC)*. RFC 9364. Feb. 2023. doi: 10.17487/RFC9364. url: https://www.rfc-editor.org/info/rfc9364.

[HS13] Herzberg, Amir and Shulman, Haya. *Towards Adoption of DNSSEC: Availability and Security Challenges*. Cryptology ePrint Archive, Paper 2013/254. 2013. url: https://eprint.iacr.org/2013/254.

[HS14] Herzberg, Amir and Shulman, Haya. “Retrofitting Security into Network Protocols: The Case of DNSSEC”. In: *IEEE Internet Computing* 18.1 (2014), pp. 66–71. doi: 10.1109/MIC.2014.14.

[Hu+16] Hu, Zi et al. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. May 2016. doi: 10.17487/RFC7858. url: https://www.rfc-editor.org/info/rfc7858.

[Hub24] Hubert, Bert. *Private Communication*. Email to Felix Wotschofsky, November 18, 2024. Nov. 2024.

[HW12] Hoffman, Paul E. and Wijngaards, Wouter. *Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC*. RFC 6605. Apr. 2012. doi: 10.17487/RFC6605. url: https://www.rfc-editor.org/info/rfc6605.

[Int24] Internet Assigned Numbers Authority. *Delegation Record for .XYZ*. IANA Root Zone Database. IANA, Feb. 2024. url: https://www.iana.org/domains/root/db/xyz.html (visited on 10/21/2024).

[IS18] Isasi, Sergi and ShresthaVicky. *Expanding DNSSEC Adoption*. Cloudflare Blog. Cloudflare, Inc., Sept. 2018. url: https://blog.cloudflare.com/automatically-provision-and-maintain-dnssec/ (visited on 11/25/2024).

[Le+18] Le, Tho et al. “Economic incentives on DNSSEC deployment: Time to move from quantity to quality”. In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 2018, pp. 1–9. doi: 10.1109/NOMS.2018.8406223.

[LGS22] Lyu, Minzhao, Gharakheili, Hassan Habibi, and Sivaraman, Vijay. “A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques”. In: *CoRR* abs/2201.00900 (2022). arXiv: 2201.00900. url: https://arxiv.org/abs/2201.00900.

[Lu+19] Lu, Chaoyi et al. “An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?” In: *Proceedings of the Internet Measurement Conference*. IMC ’19. Amsterdam, Netherlands:

Association for Computing Machinery, 2019, pp. 22–35. isbn: 9781450369480. doi: 10.1145/3355369.3355580. url: https://doi.org/10.1145/3355369.3355580.

[Lüc12] Lück, Folker. *Host Europe krallt sich Mesh Digital*. German. Internationale Expansion. May 2012. url: https://www.connect-professional.de/hardware/host-europe-krallt-sich-mesh-digital.274686.html (visited on 10/21/2024).

[Maj24] Majestic. *The Majestic Million*. Licensed under Creative Commons Attribution 3.0 Unported License. Majestic, 2024. url: https://majestic.com/reports/majestic-million (visited on 10/21/2024).

[MC17] McQuinn, Alan and Castro, Daniel. *Benchmarking U.S. Government Websites*. Mar. 2017. url: https://itif.org/publications/2017/03/08/benchmarking-us-government-websites/.

[Nam24] Namecheap. *Managing DNSSEC for domains pointed to Premium or BasicDNS*. Namecheap Support Knowledgebase. Namecheap, Inc., 2024. url: https://www.namecheap.com/support/knowledgebase/article.aspx/9723/2232/managing-dnssec-for-domains-pointed-to-premium-or-basicdns/ (visited on 11/28/2024).

[NIC24] NIC.GOV.NU. *Story of .NU*. Timeline of events related to the .NU domain. Top-Level Domain Management AB, 2024. url: https://nic.gov.nu/story/ (visited on 11/27/2024).

[Pet22] PeterDaveHello. *Alexa Top 1 Million Domains Backup*. GitHub repository. GitHub, Sept. 2022. url: https://github.com/PeterDaveHello/top-1m-domains/blob/master/backup/alexa.zip (visited on 10/21/2024).

[Rob17] Robinson, Robert. *Prevalence of DNSSEC for hospital websites in Illinois*. 2017. arXiv: 1712.05376 [cs.CY]. url: https://arxiv.org/abs/1712.05376.

[Ros+05a] Rose, Scott et al. *DNS Security Introduction and Requirements*. RFC 4033. Mar. 2005. doi: 10.17487/RFC4033. url: https://www.rfc-editor.org/info/rfc4033.

[Ros+05b] Rose, Scott et al. *Protocol Modifications for the DNS Security Extensions*. RFC 4035. Mar. 2005. doi: 10.17487/RFC4035. url: https://www.rfc-editor.org/info/rfc4035.

[Ros+05c] Rose, Scott et al. *Resource Records for the DNS Security Extensions*. RFC 4034. Mar. 2005. doi: 10.17487/RFC4034. url: https://www.rfc-editor.org/info/rfc4034.

[RSP14] Rijswijk-Deij, Roland van, Sperotto, Anna, and Pras, Aiko. "DNSSEC and its potential for DDoS attacks: a comprehensive measurement study". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, pp. 449–460. isbn: 9781450332132. doi: 10.1145/2663716.2663731. url: https://doi.org/10.1145/2663716.2663731.

[SID24] SIDN Labs. *DNSSEC Statistics*. DNSSEC deployment statistics for .nl domain names. SIDN Labs, 2024. url: https://stats.sidnlabs.nl/en/dnssec.html (visited on 11/20/2024).

[The24] The Internet Foundation in Sweden. *Automated DNSSEC provisioning*. The Internet Foundation in Sweden (IIS), 2024. url: https://internetstiftelsen.se/en/domains/tech-tools/automated-dnssec-provisioning/ (visited on 11/20/2024).

[Tim23] Tim Soulo Joshua Hardwick, Patrick Stox. *SEO Book for Beginners*. Ahrefs Pte. Ltd, 2023.

[Too+21] Toorn, Olivier van der et al. "ANYway: Measuring the Amplification DDoS Potential of Domains". In: *2021 17th International Conference on Network and Service Management (CNSM)*. 2021, pp. 500–508. doi: 10.23919/CNSM52442.2021.9615596.

[Ver24a] Vercel Inc. *Domain Registration Addendum*. Legal Agreement. Vercel Inc., 2024. url: https://vercel.com/legal/domain-registration-addendum (visited on 11/20/2024).

[Ver24b] Verisign. *DNSSEC Scoreboard*. Internet Security Tools. Verisign Inc., 2024. url: https://www.verisign.com/en_US/company-information/verisign-labs/internet-security-tools/dnssec-scoreboard/index.xhtml (visited on 11/21/2024).

[Wan17] Wander, Matthäus. “Measurement survey of server-side DNSSEC adoption”. In: *2017 Network Traffic Measurement and Analysis Conference (TMA)*. 2017, pp. 1–9. doi: 10.23919/TMA.2017.8002913.