

DNS Magnitude - A Popularity Figure for Domain Names, and its Application to L-root Traffic

Alexander Mayrhofer, Michael Braunöder, Aaron Kaplan - nic.at GmbH*

August 5, 2020

1 Abstract

When estimating the popularity of services on the internet, researchers often take simple surrogate measurements on individual services - most notable number of requests, or number of “visitors”. While that approach is simple, and usually works well for a select service (for example a web site), it fails when it comes to estimating the popularity of a domain name as a whole, including all services that are offered under that domain. To fill this gap, this report defines a DNS-based metric for estimating the popularity of a domain name - the “*DNS Magnitude*”. As almost all services on today’s internet depend on the DNS, usage of services under a domain instigates DNS traffic to the domain. That DNS traffic hence reflects the overall popularity of the domain’s services.

We start the report with a brief introduction to the Domain Name System (DNS), discuss weaknesses of domain popularity measures based on a simplistic number of requests measures (which usually ignore the TTL-effect) and look at previous work in this field. Next, we explain the design principles of DNS Magnitude and lay open our methodology. We then arrive at a formal definition of DNS Magnitude and discuss the contexts in which DNS Magnitude may be used, and its limits. We also give an informal reasoning why DNS Magnitude appears more resilient to TTL variances compared to counting the absolute number of DNS queries. Manipulation attempts (for example via changes to the TTL values) or via spoofed source IP addresses are discussed, as are other aspects of DNS that might influence resulting magnitude values.

After these theoretical aspects, we apply the DNS Magnitude metric to a L-Root Traffic data set, explore the resulting distributions and time series, and dig deeper into some specific aspects of those results, most notably the very prevalent non-existing Top Level Domains. We conclude that DNS Magnitude based statistics can provide a valuable additional measurement in many cases where large volumes of DNS data are to be aggregated.

In this paper we reason that DNS Magnitude is a useful addition to the DNS researcher’s toolbox, allowing to estimate the overall popularity of the set of services addressed by a domain, and express it in a meaningful, simple, human friendly single number.

*alexander.mayrhofer@nic.at, kaplan@nic.at, braunoeder@nic.at

Contents

1	Abstract	1
2	Terminology	4
3	Introduction	4
3.1	The Domain Name System	4
3.2	Domain Name Popularity	4
3.3	Query traffic and the Time-to-Live	5
3.4	Previous Work & Development Timeline	6
4	The “DNS Magnitude”	6
4.1	Design Principles and Reasoning	6
4.1.1	Counting Unique Client Addresses instead of Queries	6
4.1.2	Logarithmic Scale	6
4.1.3	Normalization	8
4.2	Formal Definition	8
4.3	DNS Magnitude based Rank	8
4.4	Contexts	9
5	Resilience against TTL variance	9
6	Manipulation potential and cost	10
6.1	IPv6 Allocation Ranges	11
6.2	Spoofed Source IP Addresses	11
7	Applying DNS Magnitude to L-Root Traffic	11
7.1	L-Root Dataset Description	12
7.2	Data Completeness	12
7.3	Supplementary Datasets	13
7.4	Description of Data Processing	13
8	L-Root Dataset Results	13
8.1	Top 20 TLDs by DNS Magnitude	14
8.2	Distribution of DNS Magnitude of delegated TLDs	14
8.3	Temporal Fluctuation of DNS Magnitude	15
8.4	Prevalence of Non-Existing TLDs by DNS Magnitude Ranking	16
8.5	A Deeper Dive into nxTLDs	18
8.5.1	nxTLDs with strong Fluctuation of DNS Magnitude Values	19
8.5.2	Name Collision Management Framework ‘High-Risk’ Strings	23
8.5.3	ISO3166-1 User-Assigned Code Elements	25

8.6	Effects of IP Address Aggregation	26
8.7	From the Cradle to the Grave	29
9	Limitations and further research	31
10	Conclusions	31
11	Acknowledgements	32

2 Terminology

In addition to the common DNS terminology as defined in [18], we define the following terms:

Domain Within the scope of this paper, we define *Domain* (with an uppercase “D”) as the *Domain name* and all of its *Subdomains* (both defined in [18]). Example: *Domain* “example.com” contains “example.com” itself, “www.example.com”, “_sip._tcp.example.com” and all other *Subdomains* beneath the “example.com” label sequence.

Unique clients By this we mean the number of unique client IP addresses (or client IP networks, in case when aggregation is applied) a (recursive or authoritative) set of name servers receives queries from.

Domain Name Popularity The DNS query volume of domain d that reflects the popularity of services instigating queries (directly or indirectly) to d . See section 3.2.

DNS Magnitude A measurement for *Domain Name Popularity* of a *Domain*, as defined in Section 4.2 of this document.

Rank An ordinal number assigned to a *Domain* within a *Context*, based on its *DNS Magnitude*, as described in Section 4.3.

Context The environment where measurements for calculation of *DNS Magnitude* are obtained. See section 4.4.

nxTLD Non-existing Top Level Domain: A label that is not delegated in the root zone.

3 Introduction

3.1 The Domain Name System

The Domain Name System (DNS) [23][24] is today’s predominant naming infrastructure on the Internet. It is a distributed hierarchical scheme that is globally available. Since almost any transaction on today’s internet starts with resolving a host name (which, in turn, triggers a query to the DNS), trillions of DNS transactions are performed each day - some larger operators alone report figures in the range of several hundred billions [27].

The DNS hierarchy is structured into a single “root” node from which (as of November 04 2019) 1 583 Top-Level-Domains (“TLDs”) [7] are administratively and technically delegated to organizations all around the world. These delegations form the “Root Zone”. Delegated TLDs below the root zone then contain further levels of hierarchy, their structure depending on the local policy of the respective TLD. As of August 2019, about 355 millions of “Domain Names” were registered across all TLDs [30].

The root zone is hosted on 13 “Root Servers”, named alphabetically *a.root-servers.net* to *m.root-servers.net*, and operated by 12 organizations as of Nov 2019 [6] (there’s only 12 operators because Verisign, Inc. hosts two instances). By means of IP Anycast [1] the 13 root servers are hosted on a globally distributed set of 1 031 instances [29] (as of Dec 2019) to ensure ubiquitous and uninterrupted availability of the root DNS service. Similar infrastructure, albeit typically on a smaller scale, is in place for most TLDs [12].

3.2 Domain Name Popularity

Estimating the popularity of a Domain has a variety of use cases. Popularity is competitive, so Domain owners often want to compare the popularity of their Domains with those of their competitors, or gauge it against well known, very popular names which occupy the top spots within a TLD. Popularity also attracts attention, both of the good (more business) and bad

kind (more attacks). Registrars are interested in grouping their Domain portfolio into popularity classes, for example to identify potential upselling candidates to counteract both the good and bad type of attention (more resilient infrastructure, more extensive security features).

Understanding the popularity of a Domain also gives TLD operators additional insight into their customer base - for example, preliminary research appears to confirm the hypothesis that Domains with a higher popularity are less likely to be deleted. And finally, significant shifts in popularity of a Domain may be used to discover anomalies (see Section 8.3 for a practical example).

Different names in the DNS hierarchy attract widely varying amounts and patterns of query volume. The DNS is rarely the user desired end service itself, but consulting the DNS is a prerequisite for accessing any service that uses the domain name itself or any of its Subdomains. This leads to the first observation: the more often services under a Domain are accessed, the more often name resolution for the respective Domain is required. In other words: the popularity of services offered via names under a Domain is therefore related to the amount of DNS query traffic that can be observed for the domain name itself, or any Subdomains of that domain name:

$$Pop(d) \sim \text{traffic}(d)$$

where $Pop(d)$ is the Domain Name Popularity function of domain name d and $\text{traffic}(d)$ is the set of DNS resolutions requests for d or any Subdomain of d .

We therefore define for the scope of this paper the *Domain Name Popularity* as a function of the DNS query traffic observed for a domain name d or any Subdomain of d :

$$Pop(d) = f(\text{traffic}(d))$$

where f is some to be discussed function.

3.3 Query traffic and the Time-to-Live

The most obvious method to measure popularity based on DNS query traffic would be to count the number of queries observed over a certain period of time for a specific Domain, and declare that figure the popularity of that Domain. However, the DNS contains a caching mechanism that constitutes an integral part of its efficient functioning:

When a DNS client receives a response, that response is accompanied by “Time-to-live” (TTL) information, which, as described in RFC1035 [24] “*specifies the time interval that the resource record may be cached before the source of the information should again be consulted*”. That means when the same client (such as a recursive resolver) receives a new downstream query for a resource for which an older response is still in its cache, it will answer with that cached response, and not send another query to an upstream authoritative server for the domain. Shorter TTLs therefore incur earlier expiring of cached responses, earlier re-querying, and higher query rates on upstream name servers.

TTLs are configurable by the administrator of a zone, and typical values span a wide range from a few seconds up to several weeks. In practice, this means that two domain names with identical popularity of their underlying services, but different TTL values will generate different DNS traffic volumes. This is undesirable for any query-based measurement of domain name popularity, as it introduces bias that a) reflects internal workings of the DNS protocol, rather than an actual difference in service popularity and b) is subject to manipulation by modifying TTL values.

Note that there is also an important operational aspect: Assuming that there is a benefit associated with achieving higher (perceived) popularity, lowering TTL values to induce higher query rates could become attractive for domain name administrators. However, lower TTLs also increase the load on recursive and authoritative DNS servers, and effectively reduce the resilience of the DNS [26] - a very undesirable consequence!

For the reasons outlayed above, it is desirable for a DNS based popularity measurement that changes to the TTL values have little impact on the resulting measurement values.

3.4 Previous Work & Development Timeline

Calculating/estimating popularity of domain names not novel. The internet community has well known estimates such as the Alexa top-1M list, but these don't extend to arbitrary domain names. Holmes et. al [19] discuss (in a Google patent) a domain popularity metric, which goes far beyond looking at DNS traffic only. In addition, the domain investment industry also performs estimations and measurements of how popular (in terms of reselling value) a domain is. None of these approaches try to estimate a domain names' popularity based on its DNS traffic only.

To the best of our knowledge, the first such approach was presented by Sebastian Castro[11] in May 2015 (DNS-OARC Workshop, Amsterdam, The Netherlands), using term frequency-inverse document frequency (TF-IDF) as the general algorithm. As there was concern that TF-IDF will heavily punish very popular domains, they developed a revised popularity ranking, and presented that "compound probability algorithm" to the CENTR R&D group in May 2016. During that presentation, Castro first mentioned the idea that counting unique hosts rather than packets could be a strategy that offsets for variance of the DNS' caching TTL.

Inspired by that discussion, DNS Magnitude was developed by Alexander Mayrhofer over the course of the following months, and first presented to the CENTR R&D group[22] in Nov 2016.

In dec 2016, Cisco's Umbrella 1 Million list [20] was made public. A blog post posted by OpenDNS before their acquisition by Cisco showed that the algorithm used was the compound probability algorithm developed by Internet NZ (as described above).

The DNS based malware detection system *Kopis* [4], presented in 2011, proposed a related concept of "Requestor Diversity", where IP addresses are first mapped to BGP prefixes, AS numbers, country codes, and the distribution of those three features is then analyzed.

This very paper, developed in 2019/2020, to the best of our knowledge, represents the first formal definition of *DNS Magnitude*.

4 The "DNS Magnitude"

4.1 Design Principles and Reasoning

4.1.1 Counting Unique Client Addresses instead of Queries

Because of the forementioned impact of TTL values on query volume (see section 3.3), DNS Magnitude relies on counting unique client addresses observed for a certain Domain, rather than the number of queries. This should significantly reduce the impact of different TTL values for Domains in identical contexts. An empiric observation that confirms this hypothesis is included in section 5.

We propose that DNS Magnitude uses the number of unique client IP addresses (client cardinality), rather than the number of queries, as the basis for calculation.

4.1.2 Logarithmic Scale

As described in section 3, many Contexts of DNS traffic have a high disparity between few extremely busy Domains at one end, and many Domains with very low traffic levels at the other end of the scale. Even when considering the number of Unique Clients (rather than the number

of queries), there is a high disparity amongst, for example, the set of second level domains below a TLD:

We investigate the disparity of unique hosts per second-level domain of the Austrian country code TLD (ccTLD) *.at*¹, based on traffic observed on authoritative nameservers. At the time of the investigation, the *.at* TLD is delegated to 8 different name server names. DNS traffic from the following subset of these nameservers is available for the investigation: *r.ns.at* (all 38 instances), *d.ns.at* (“Frankfurt” node only), *n.ns.at* (“Amsterdam” node only), *ns1.univie.ac.at* (“Vienna” node only) and *ns9.univie.ac.at* (“interxion Vienna” node only). We gather the traffic for the full month of August 2019 from those servers, which contains data for 5,45 billions of DNS queries. Of these queries, 4,21 billions were answered with response code (Rcode) 0 (NOERROR), indicating that a delegation for the requested name existed at the time of the query. 989 million queries triggered Rcode 3 (NXDOMAIN), indicating that the requested name does not exist.

The dataset contains 2 237 236 observed unique client IP addresses, and queries for 522 890 793 different second level domain names (Names under *ac.at*, *gv.at* and *priv.at* were excluded, because as they are delegated to a different set of servers, and hence 3rd-level names under those Subdomains always trigger a response code 0 (NOERROR) on the authoritative nameservers for *.at*. As the *.at* TLD contained only about 1.3 millions of delegated names, the vast majority of names queried trigger NXDOMAIN.

We filter the data for *NOERROR* transactions, and aggregate the data to a set of tuples containing the *domain name* the number of unique client IP addresses observed (*hostcount*) for each domain. Query names are aggregated to their 2nd level domain, or the 3rd level domain for names under *co.at* and *or.at*.

We then plot (see Figure 1) the distribution of *hostcount* by domain in that aggregated data set, once with the *hostcount* unmodified (linear scale, Figure 1a), and once with the natural logarithm applied to *hostcount* (Figure 1b). We notice that the linear density exposes the extreme disparity between few “busy” and many “quiet” domains (as discussed above), even on the aggregation level of unique hosts. In comparison, we notice that the logarithmic scale exposes a much more natural distribution, and can be fitted quite well to a normal distribution:

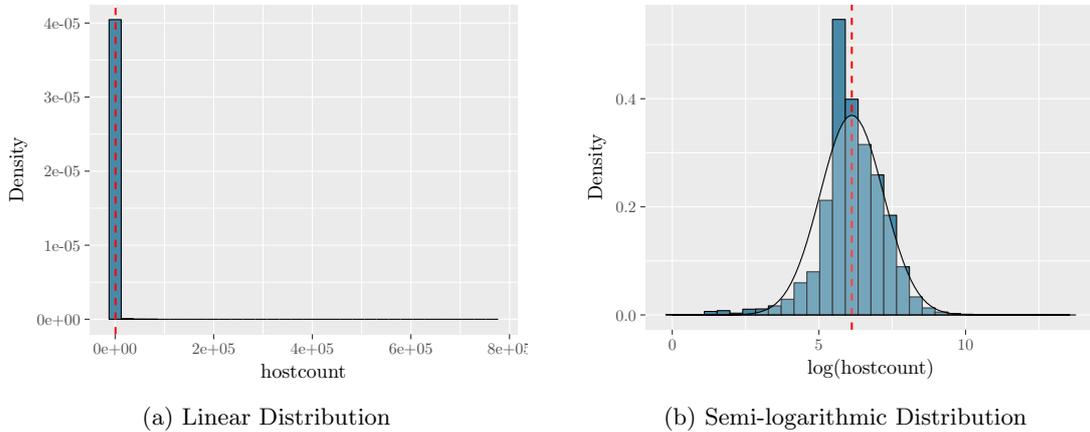


Figure 1: Linear and semi-logarithmic distributions of unique host count per domain for existing *.at* domain names

This is also reflected in the general “scale-free” architecture of the Internet (see [8]). It is therefore no surprise that a logarithmic distribution fits the distribution of DNS traffic to Domains, as the popularity distribution of nodes on the Internet is reflected in the DNS traffic of their names.

¹operated by nic.at, the employer of the authors

For the reasons outlined in this section, we propose to use *the natural logarithm of Unique clients per Domain*, rather than a linear scale, to describe DNS Magnitude values.

4.1.3 Normalization

The number of queries and unique client addresses observed will vary greatly between different environments. A home or small enterprise network will typically see much fewer queries than the recursive resolver of their upstream internet service provider. However, in any environment, the maximum number of unique client addresses accessing a certain domain cannot exceed the observed total number of unique client addresses (the extent of the context). Given the goal of DNS Magnitude is to measure a relative popularity within a given context, it makes sense to normalize the resulting figure to the extent of that context.

When normalization is applied by means of dividing the observed unique clients per domain by the total unique number of observed client addresses, the resulting figure is in the range of 0 to 1.

As DNS Magnitude also has the goal to be human-friendly and -understandable (and humans are used to work in the decimal system, with digits ranging from 0 to 10), we propose to multiply that resulting figure by 10, in short:

We propose to normalize resulting DNS Magnitude values to a scale of 0 to 10.

Because DNS Magnitude is - (as described above) - normalized to a specific Context, this means that values - even for identical Domains - cannot be directly compared cross-context.

4.2 Formal Definition

DNS Magnitude is defined as follows:

Where A_d is the set of client addresses observed querying a specific Domain d (and therefore either the Domain name of d itself, or any of its Subdomains) in a given context (see 4.4) during a specific time interval, and A_{tot} is the total set of client addresses observed querying in the same Context during the identical time interval, DNS Magnitude $mag(d)$ for a Domain d in that Context is defined as the natural logarithm of the cardinality of A_d , divided by the natural logarithm of the cardinality of A_{tot} , normalized to the range 0 – 10:

$$mag(d) = \frac{\ln(|A_d|)}{\ln(|A_{tot}|)} * 10 \quad (1)$$

Note that because $\forall A \forall d : |A_d| \leq |A_{tot}|, \forall d : 0 \leq mag(d) \leq 10$. For the empty set of client addresses $A_d = \{\}$, the respective DNS Magnitude $mag(d)$ is undefined.

For a concise prosa description of DNS Magnitude, we propose the following text:

“DNS Magnitude is a logarithmic measure for the DNS popularity of a Domain, based on counting unique clients, normalized to a range of 0-10.”

4.3 DNS Magnitude based Rank

In some cases, the actual value of a measurement might be less important than comparing the value of different items against each other, as long as measurements are acquired in the same Context (using the same methodology). One such method is to create a ranking, based on the order of the measurement values, and consider the rank of each item as the result, rather than the actual measurement value itself.

DNS Magnitude can be used to create such a ranking of Domains within a specific Context: To assign a rank to each Domain, the set of Domains is sorted by their DNS Magnitude value in descending order, and ordinal numbers (starting from 1) are assigned to each Domain, starting with the Domain with the highest DNS Magnitude value.

A Domain's Rank is always specific to a Context.

Note that ranks in a Context are not affected by the calculation of the logarithm nor the normalization step. Therefore, ranks can also be calculated directly from the number of Unique Clients.

4.4 Contexts

For the purpose of DNS Magnitude calculation, a *Context* is defined as the environment in which the DNS transactions (query/response pairs) have been observed, including any filtering / truncation steps performed on the set of observed transactions and addresses before the calculation defined in 4.2 is applied.

A specific context entails:

- The potential set of DNS clients
- The potential set and function of DNS servers
- Filtering performed on observed queries
- Aggregation performed on client IP addresses
- Aggregation performed on query names (Qnames)

Because DNS Magnitude values are always specific to a Context, cross-context comparisons are potentially misleading and must be done with great caution. Such comparisons are only sensible when two Contexts expose significant correlation of the underlying DNS traffic.

DNS Magnitude information is useless and invalid without the description of the respective Context.

An example of such a Context would be the set of recursive resolvers of an Internet Service Provider (ISP), their IP address ranges used for customers as potential client addresses, the information that IPv6 addresses have been truncated to /56 prefixes (for example because this matches the local allocation policy for end customers), while IPv4 addresses have not been aggregated, that Qnames have been truncated to one additional level beyond the Public Suffix List, while no filtering for QTYPEs was performed.

Another example of a context would be a subset of authoritative servers for a TLD, the global IP address space as potential clients, queries filtered to NXDOMAIN type responses, QNAMEs aggregated to 2nd-level domains, and IP addresses truncated to /24 (IPv4) and /48 (IPv6).

5 Resilience against TTL variance

As described above, one of the main design goals of DNS Magnitude is resilience against query volume differences which are triggered by TTL variations, rather than representing differences in popularity of the underlying services.

During preparation of an earlier study of DNS Magnitude (within the .at TLD), we noticed that a specific 2nd-level Domain exposed an much higher ratio of queries to Unique clients, compared to other Domains with similar client address counts (See Table 1):

(Context / data set: Queries for the .at TLD to the authoritative nameservers *r.ns.at*, *n.ns.at* (Amsterdam node only), and *ns9.univie.ac.at*, observed between Mar 23 2019 and Mar 29 2019.)

Domain	Unique clients	observed queries	queries / Unique clients ratio
<i>univie.ac.at</i>	543 437	41 428 395	76,23
<i>anexia.at</i>	444 859	203 186 012	456,74
<i>telekom.at</i>	336 942	11 178 255	33,17
<i>google.at</i>	295 864	4 940 158	16,70
<i>nessus.at</i>	283 206	12 850 393	45,37

Table 1: Query-to-Client ratio observed before TTL change

Upon closer inspection of the properties of the affected domain *anexia.at*, we discovered that some of the relevant TTL values were configured to unusually low values:

- *60 seconds* for the authoritative *NS* resource record set (RRSet)
- *120 seconds* for the *A/AAAA* RRSet of the glue record nameservers.

The domain name owner was notified of those low TTL settings, and subsequently changed all TTL values mentioned above to *10800 seconds* (a more sensible and common value, see Figure 3 of [25]). Once the new TTLs had settled, we took new measurements (again, for a full week). The results of both measurements are compared in Table 2.

Measurement	Unique clients	Observed queries
(a) short TTLs	444 859	203 186 012
(b) long TTLs	422 278	21 409 529
relation b/a	94,94 %	10,53 %

Table 2: Comparison of client and query volume before and after TTL change

While the number of observed queries is reduced by 89,47% of the original volume, the cardinality of the client IP addresses is only 5,06% lower. This empiric observation indicates that host cardinality seems indeed more resilient against changes in TTL values than query counts, and hence more suitable for popularity measurements.

Note that the domain primarily serves as an “infrastructure” domain, meaning that hosts under the domain serve as authoritative nameservers for a large number of other domains. Such infrastructure domains expose a high popularity, because these names need to be resolved before any of the hosted names can be resolved.

Also note that when the measurement period is shorter than a TTL, only a subset of the clients requesting a Domain will be observed. For example, given a TTL of 2 days, and a measurement period of 12 hours, only about 25

6 Manipulation potential and cost

DNS Magnitude could be used in situations where achieving a certain ranking provides benefits to the owner of a domain name. Because of such benefits, owners might be tempted to explore options to artificially influence the DNS Magnitude for their name. This section explores some potential manipulation options, their associated cost, and prospective countermeasures.

As described in section 4.1.1, DNS Magnitude is based on the cardinality of client IP addresses. Therefore, for manipulation of DNS Magnitude, that number of unique clients observed in the DNS traffic of a certain Context is the only target for manipulation. By creating non-organic traffic from a set of IP addresses for a certain name, DNS Magnitude of that name can be artificially elevated. Artificially lowering the magnitude of a name directly is not possible, unless suppressing DNS queries from existing organic clients is possible. However, Magnitude can be indirectly lowered by creating additional traffic for other names, hence increasing the

base population used in the calculation, and reducing the relative fraction of hosts seen for the target name.

Counting hosts is more resilient against manipulation than counting packets, because much more effort / cost is required for introducing traffic from additional IP addresses rather than simply sending more queries from the same IP address. However, two properties of the current internet infrastructure lower this barrier significantly:

6.1 IPv6 Allocation Ranges

In version 4 of the Internet Protocol (IPv4) address assignments to users and even small enterprises often contain just a single IP address. However, assignments to end users in version 6 of the Internet Protocol (IPv6) typically comprise whole prefixes (See RFC 6177 [28]). The typical allocation prefix size is /56 to /64, and any of those prefix sizes contains more IP addresses than the entire IPv4 space. This means that even a single IPv6 allocation could be used to increase the number of IP addresses observed on a server almost arbitrarily, and hence significantly inflate the total set of client IP addresses used as the basis for DNS Magnitude calculation.

As a countermeasure, IPv6 addresses observed could always be truncated to /64, as this is a very common (and recommended) prefix length for single-host allocations. Subsequently, each observed prefix is then counted as a unique client (rather than each observed individual address). Section 8.6 examines the general impact of such IP address aggregation on DNS Magnitude results.

6.2 Spoofed Source IP Addresses

Today, a lot of DNS traffic is transported using the User Datagram Protocol (UDP). Because UDP is stateless, and deployments do typically not verify whether the source address used in a packet is actually assigned to the device (or routed to the network sourcing that packet, see BCP-38 [13]), sending packets with arbitrary source IP address is possible in large parts of the internet. Such packets with *spoofed* source addresses can then be used to, again, artificially and almost arbitrarily increase the number of unique hosts observed for a Domain.

Spoofing of source IP addresses might not be possible in more controlled environments (for example between the access plane of an ISP and the ISP's own recursive resolver). In such cases, DNS Magnitude is safe from manipulation by spoofed addresses. However, that level of control does not exist on the open internet.

One countermeasure would be to limit measurements to hosts seen via TCP-based DNS protocols. Hence the effectivity of that countermeasure depends on the prevalence of such protocols in the respective context. For the recursor-to-authoritative leg, such protocols are still under standardization at the time of this writing, and hence not available. Alternative options could include statistical methods to detect (and remove) artificially created traffic (“Resolver classification”), but such methods are out of scope of this paper.

7 Applying DNS Magnitude to L-Root Traffic

In the second part of this paper, the definition of DNS Magnitude is applied to traffic from the ICANN Managed Root Server System (IMRS), also known as *L-Root* or *l.root-servers.org*. That practical part of the study was performed in two stages:

During an initial (experimental) phase, we had access to traffic snapshots of 10 minutes. In that phase, we focused on exploring the structure and properties of the data set, and developed DNS Magnitude calculation software curtailed to the given input format. The software is available via a GitHub repository [9].

For the subsequent research phase, we had access to full traffic data of several months, as described in the following section.

7.1 L-Root Dataset Description

The dataset available for the study is based on pcap (“*packet capture*”) data gathered by ICANN from the L-Root servers. For each 10-minute interval of each server instance, a separate pcap file is produced, collected, and pre-processed into a text-format which stored using gzip compression.

Those files contain one line of text per DNS response sent to a client. From the available fields for each response, the destination of the response (client IP address) and the Query Name (QNAME) are relevant for the calculation of DNS Magnitude.

The data of an average day comprises about 52 000 files, with a total (compressed) volume of about 120 GB. The average day sees traffic from a total of 3,5 million unique client IP addresses.

Based on the available curated data, we decided for the 5 complete months of April 2019 to August 2019 as the observation window. This resulted in a total count of 8 024 722 files with a total compressed volume of 17,95 TB.

7.2 Data Completeness

During a late stage of the research, we noticed that the data of some days exposed a significantly lower number of unique client IP addresses per day than average days. We performed a size based validation by normalizing the daily figures of (a) unique client IP addresses, (b) number of data files per day, and (c) total file size per day to their respective maxima, and comparing them against each other (See figure 2a). The observed irregularities fall into two different categories:

1. **Incomplete calculations:** Days with normal file counts and / or normal file sizes, but lower than expected unique client counts indicate complete data, but a problem with the calculation of DNS Magnitude for that day. Specifically: 2019-05-10, 2019-05-11, 2019-07-28, 2019-08-09 and 2019-08-31. We could trace down some of these incomplete results to reboots of the machine during the time-intensive calculation.
2. **Incomplete source data:** Days with lower than expected number of unique clients, but also fewer than expected data files (or lower than expected total source data size) indicate missing source data (but correct DNS Magnitude calculations). For example, first few days of August 2019.

Based on these findings, we were able to correct some of the incomplete calculations (namely 2019-05-10, 2019-05-11, 2019-08-09), but none of the “incomplete source data” cases, as the original data was unavailable by the time the issue was discovered. We do, however, believe that this allows to examine the resilience of DNS Magnitude against incomplete measurement data, and therefore constitutes an interesting research aspect. Figure 2b illustrates data completeness post corrections - this data was used for the subsequent findings. When focusing on a single day, we used 2019-08-30 (last complete working day of the observation period).

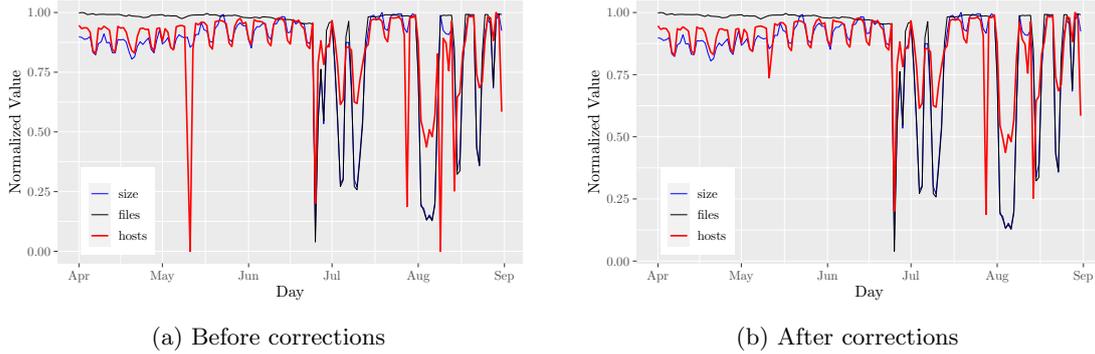


Figure 2: Size based Validation of Dataset

7.3 Supplementary Datasets

Besides the L-root traffic data described in section 7.1, we used the following datasets to augment the calculated DNS Magnitude results:

- **Historic root zones:** To identify whether or not a label existed as a delegation in the root zone on a given day, we use a private root zone archive managed by Arsen Stasic from the University of Vienna. This archive was created by performing zone transfers from *b.root-servers.org* on a daily basis. For the dates of 2019-05-21, 2019-07-29, and 2019-08-06 no root zone could be transferred. For those dates, we used the root zone file of the previous day.
- **ICANN new gTLD Application List:** To identify whether or not a TLD is part of ICANN’s new gTLD Program, we use a CSV file of new gTLD applications[15], published by ICANN itself.

7.4 Description of Data Processing

Based on the data set described in section 7.1, we created daily lists of DNS Magnitude per TLD. Calculations for one day required about 5 hours of script run time.

As a compromise between result file size and extent of the analysis, we truncate the DNS Magnitude lists to TLDs for which at least 100 unique client IP addresses were observed during the day. Depending on the specific day, this censors the data to TLDs with a minimum DNS Magnitude of about 3,2, and contains about 160k TLDs for each day. The minimum number of 100 unique clients was chosen so that (on most days), all delegated TLDs are contained in the output files.

8 L-Root Dataset Results

We initially use the data from a single day, August 30 2019², calculate the DNS Magnitude for each TLD, and sort the results by DNS Magnitude in descending order to create a ranking of TLDs³ by the observed DNS Magnitude. We augment the results with information from the supplementary datasets (See section 7.3).

The result file contains 169 507 TLDs for which at least 100 unique clients have been observed. Given that 1 528 TLDs existed in the root zone on that day, 99,1 % of rows represent non-existing TLDs (nxTLDs).

²Last non-weekend day of the observation period for which complete data is available

³Note that we include the root zone itself as well, even strictly speaking it does not constitute a “TLD”

8.1 Top 20 TLDs by DNS Magnitude

We first look at the TLDs with highest DNS Magnitude values. The total number of client addresses observed on this day was 3996847. For brevity, we truncate the data to the top 20 (see Table 3).

Rank	TLD	DNS Mag.	Unique clients	TLD exists	new gTLD	% Total Clients
1	com	9.50	1861869	TRUE	FALSE	46.58%
2	.	9.48	1825446	TRUE	FALSE	45.67%
3	net	9.45	1735329	TRUE	FALSE	43.42%
4	org	9.03	917320	TRUE	FALSE	22.95%
5	uk	8.82	666371	TRUE	FALSE	16.67%
6	info	8.75	595804	TRUE	FALSE	14.91%
7	au	8.68	538117	TRUE	FALSE	13.46%
8	de	8.64	502903	TRUE	FALSE	12.58%
9	arpa	8.59	471062	TRUE	FALSE	11.79%
10	eu	8.56	444635	TRUE	FALSE	11.12%
11	biz	8.55	443072	TRUE	FALSE	11.09%
12	local	8.45	376781	FALSE	FALSE	9.43%
13	br	8.43	368388	TRUE	FALSE	9.22%
14	cn	8.43	365896	TRUE	FALSE	9.15%
15	io	8.35	326590	TRUE	FALSE	8.17%
16	it	8.32	310880	TRUE	FALSE	7.78%
17	jp	8.31	305900	TRUE	FALSE	7.65%
18	ru	8.31	305562	TRUE	FALSE	7.65%
19	co	8.26	282390	TRUE	FALSE	7.07%
20	pl	8.24	273454	TRUE	FALSE	6.84%

Table 3: Top 20 TLDs by DNS Magnitude for 2019-08-30

We see that the set of top 20 TLDs contains widely known, long-established TLDs with a high number of registrations on subsequent levels, so those TLDs correlate with the empiric observation that these are indeed “busy” TLDs. The *.local* TLD is the only nxTLD in the top 20, with a popularity similar to larger country code TLDs. This is not surprising, however, as the popularity of that invalid TLD is well known, and has been described before [10]. We further examine the prevalence of non-existent (invalid) TLDs in section 8.4.

8.2 Distribution of DNS Magnitude of delegated TLDs

We filter the results described in section 8.1 so that it contains only data for the 1528 TLDs which existed that day⁴. Plotting a histogram of the resulting data, we observe that the distribution exposes bimodal characteristics (see Figure 3a).

Because a manual review of the data suggests a high concentration of new gTLDs from the 2012 round in the bottom 50% of the scale, we initially suspect that bimodality to originate from the introduction those TLDs (as the Top 20 shown above contains exclusively “old” TLDs). We split the histogram into two groups: TLDs introduced by ICANN’s new gTLD program [16] in the 2012 round (“*ngtld*”), and others (TLDs pre-dating the new gTLD program plus IDN ccTLDs) (“*old*”). The resulting distributions (see Figure 3b) refutes that hypothesis:

The *old* group exposes a relatively unimodal distribution - the slight peak of low-magnitude TLDs contains (presumably unused) IDN ccTLDs. The *ngtld* group itself shows a significant bimodality, indicating that this group itself contains two groups with disjoint magnitude values.

We conclude that neither group of domains is uniform, and both the *old* and *ngtld* groups contain sub-groups of “busy” and “idle” TLDs. We can show, however, that the *old* group

⁴The whole file contains data for 169507 root labels

exposes a higher median popularity than the *ngtld* group.

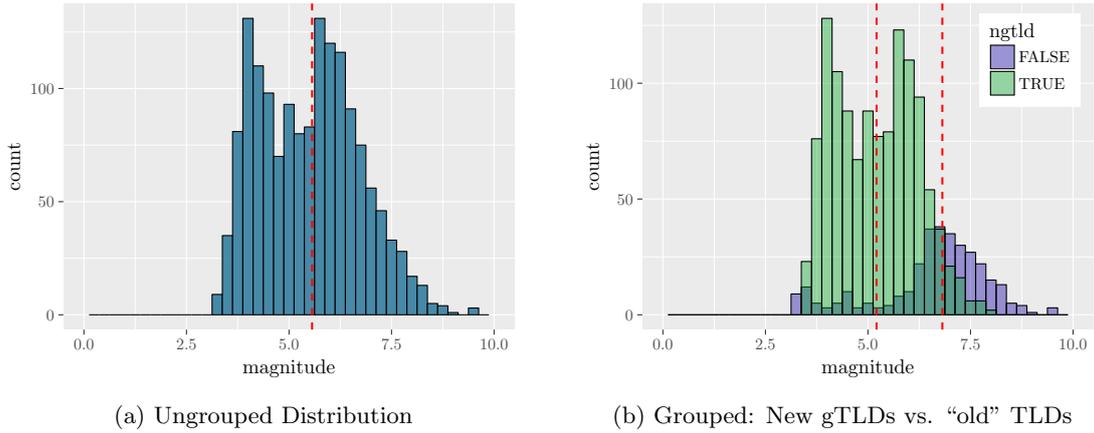


Figure 3: Distribution of DNS Magnitude from August 30 2019 for delegated TLDs. Dashed red line reflects the median DNS Magnitude

8.3 Temporal Fluctuation of DNS Magnitude

As we calculate DNS Magnitude values for each TLD on a per-day basis, and the observation period spans several months, we can also investigate the fluctuation of the measurement values of each TLD over time. This investigation can be done for the actual DNS Magnitude values, but also for the Ranks (see Section 4.3) derived from those values. The observation period contains 153 days, so splitting the data by TLD yields a time series with 153 data points each for Magnitude and Rank, for each TLD.

We calculate the median DNS Magnitude for each TLD, and select the Top 20 TLDs to plot their daily DNS Magnitude values over time (See Figure 4).

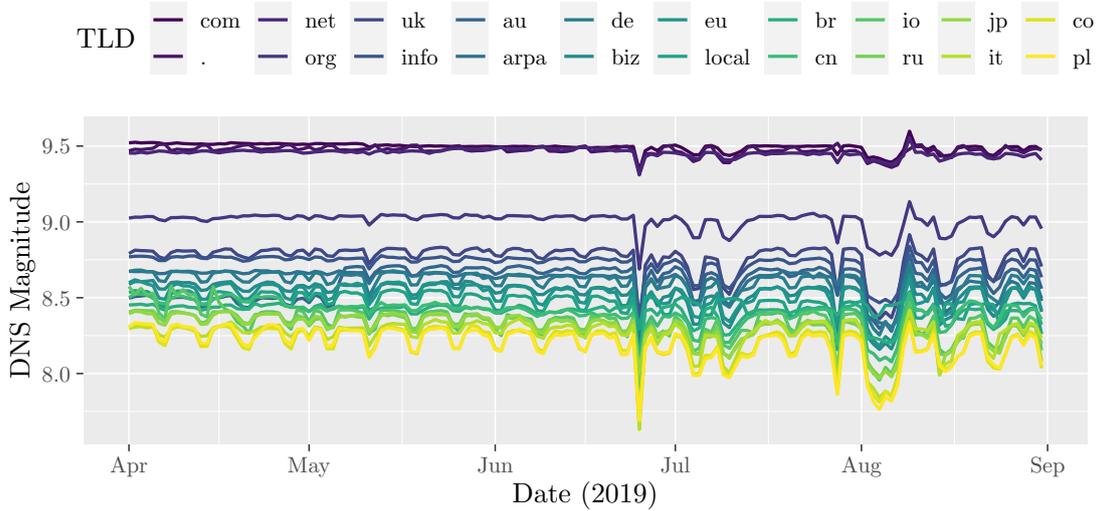


Figure 4: DNS Magnitude values of Top 20 TLDs over time

We observe that:

- DNS Magnitude values for those very busy TLDs are generally very stable over time, at least up to the last week of June.

- Values for some TLDs fluctuate weekly, exposing a visible difference between working days and weekends. This effect appears to be more pronounced in ccTLDs. Note that a similar effect is also visible for May 1st, which is a holiday in many countries.
- The period with lower stability of DNS Magnitude value (starting in last week of June) correlates with our observation of incomplete data (See section 7.2). We therefore assume that those fluctuations of DNS Magnitude are an effect of the incomplete data, rather than actual changes of popularity of the respective TLDs.

We also observe that in periods with missing data, absolute DNS Magnitude values are visibly lower. However, all TLDs in the chart appear to be affected to a similar degree, so differences between Magnitude values (and therefore Ranks) seem to be more stable than actual values.

To verify that assumption, we create a bump chart of Ranks of those same 20 TLDs (See figure 5). We do indeed observe that the periods of missing data are not immediately visible in the chart, and fluctuation of Ranks in such periods is much less affected than the fluctuation of actual values.

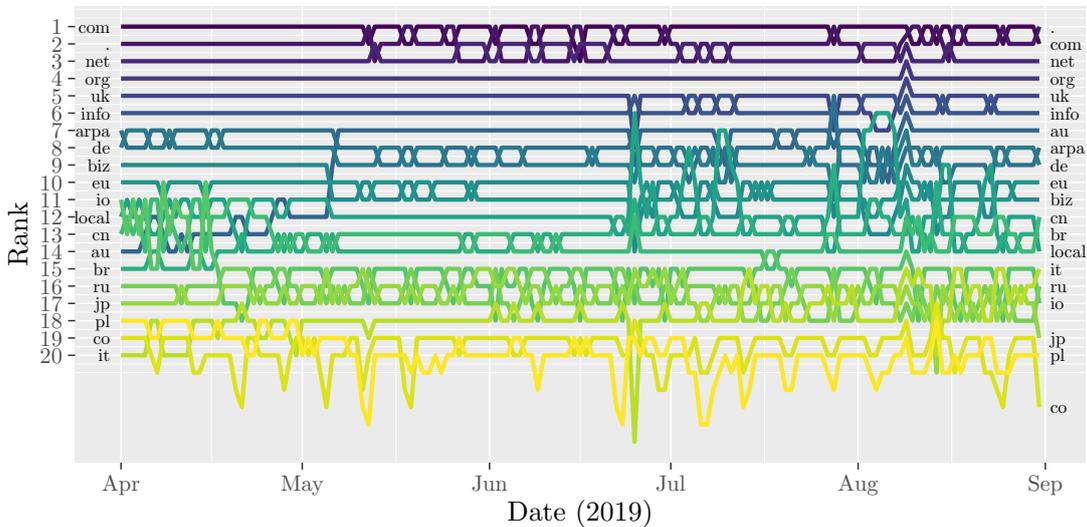


Figure 5: DNS Magnitude based Ranks of Top 20 TLDs over time

We therefore conclude that when there's risk of incomplete data, comparisons should be performed on a Rank basis, rather than actual DNS Magnitude values of individual Domains.

8.4 Prevalence of Non-Existing TLDs by DNS Magnitude Ranking

As described in section 8.1, even the list of top 20 TLDs (ranked by DNS Magnitude) contains a non-existent (invalid) TLD. This poses the question how those TLDs are distributed amongst a more extensive ranking list. More specifically, we are interested whether *.local* represents a notable exception amongst the busiest delegated TLDs, or whether nxTLDs are more prevalent amongst the top-ranking delegated TLDs.

We take the existing result file for 2019-08-30, and create bins of 200 TLDs, starting from the top ranked TLD. The first bin therefore contains TLDs with ranks #1 to #200, while the second bin contains TLDs ranked #201 to #400, etc. We count the number of delegated TLDs in each of the bins, and plot the results on a bar chart (see Figure 6). We initially truncate the chart to the top 5000 TLD, in the naive belief that this would contain almost all of the 1528 delegated TLDs.

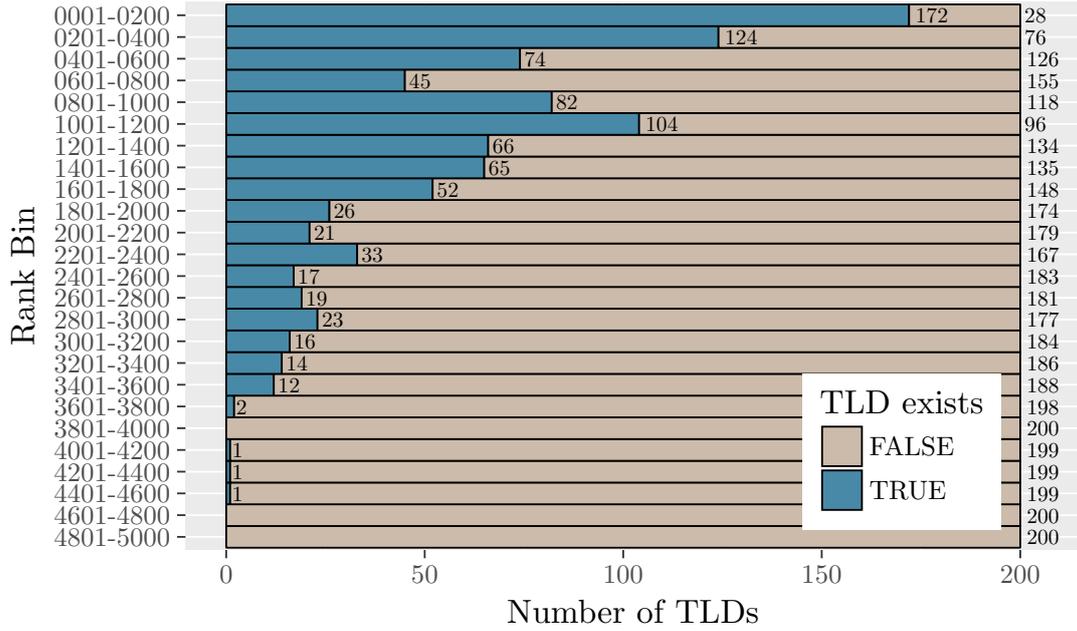


Figure 6: Number of delegated and non-existent TLDs by binned rankings

However, we find that only 970 out of the total 1 528 delegated TLDs are contained in the top 5 000 TLDs. Even the top 2000 TLDs are literally infested with 1 190 nxTLDs (59,5 %).

We therefore create another chart with a bin size of 5 000, representing the entire dataset. All 970 delegated TLDs represented in figure 7 are hence contained in the first bin. The chart exposes a surprisingly long tail of delegated TLDs with low-ranking DNS Magnitude values.

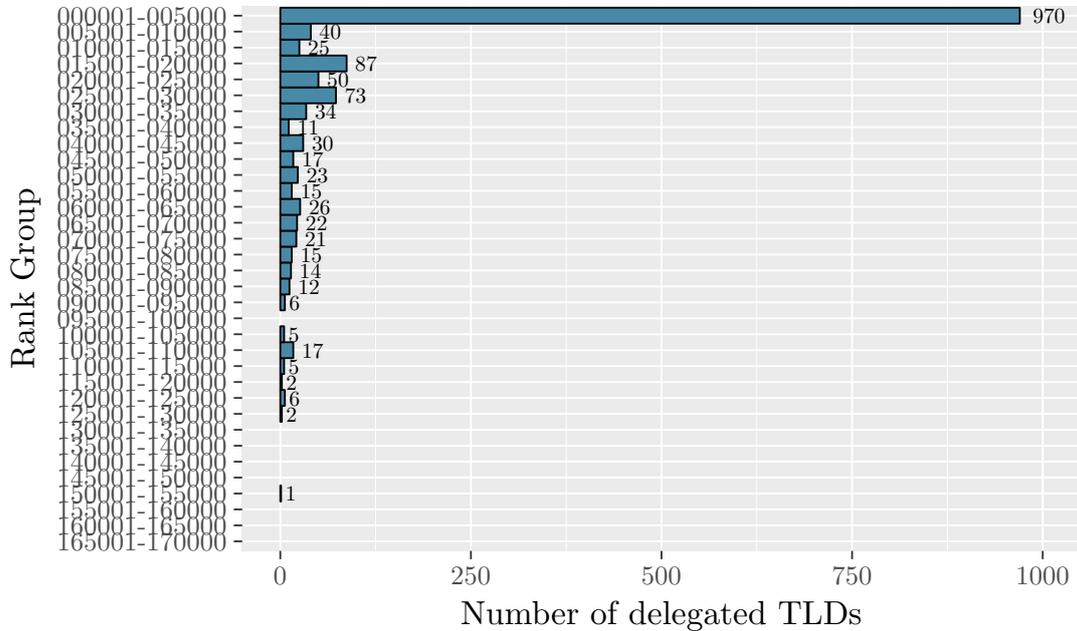


Figure 7: Number of delegated TLDs by binned rankings

The IDN ccTLD *xn-mix891f* had the lowest DNS Magnitude value of all delegated TLDs on August 30 2019. It ranked 151 243 out of 169 507 labels with at least 100 unique clients

observed during that day. This means that on that day, there were 149 714 nxTLDs with a higher popularity than this specific delegated TLD. The position of that TLD was fairly constant during the observation period, Its rank fluctuated between 140 000 and 170 000.

8.5 A Deeper Dive into nxTLDs

In Section 8.4 we discussed the prevalence of nxTLDs amongst top-ranking labels. Because of this prevalence, we take a closer look at the properties of DNS Magnitude values of nxTLDs.

To identify the top-ranking nxTLDs, we filter the daily DNS Magnitude result files for non-existing labels, and calculate the median of the DNS Magnitude values for each nxTLD across the observation period. Results truncated to the Top 20 ranking rows are included in Table 4. Besides the Rank amongst nxTLDs, the table also includes the ranking of each nxTLD in the full, unfiltered list of observed TLDs.

Rank	nxTLD	DNS Magnitude	Rank across all TLDs
1	local	8.48	12
2	localdomain	7.86	45
3	_ta-4f66	7.71	64
4	home	7.70	65
5	lan	7.65	71
6	tcs	7.56	81
7	gif	7.47	91
8	internal	7.41	105
9	invalid	7.41	106
10	com/	7.38	107
11	wpad	7.37	112
12	1	7.34	116
13	corp	7.29	125
14	null	7.28	130
15	1]	7.20	145
16	_tcp	7.15	155
17	com_1	7.07	164
18	2	7.06	177
19	loc	7.05	183
20	_msdcs	7.03	186

Table 4: Top-ranking 20 nxTLDs by DNS Magnitude

To understand whether popularity of nxTLDs fluctuates in different ways than that of delegated TLDs, we create a chart of daily DNS Magnitude values across the observation period for those 20 nxTLDs (See figure 8).

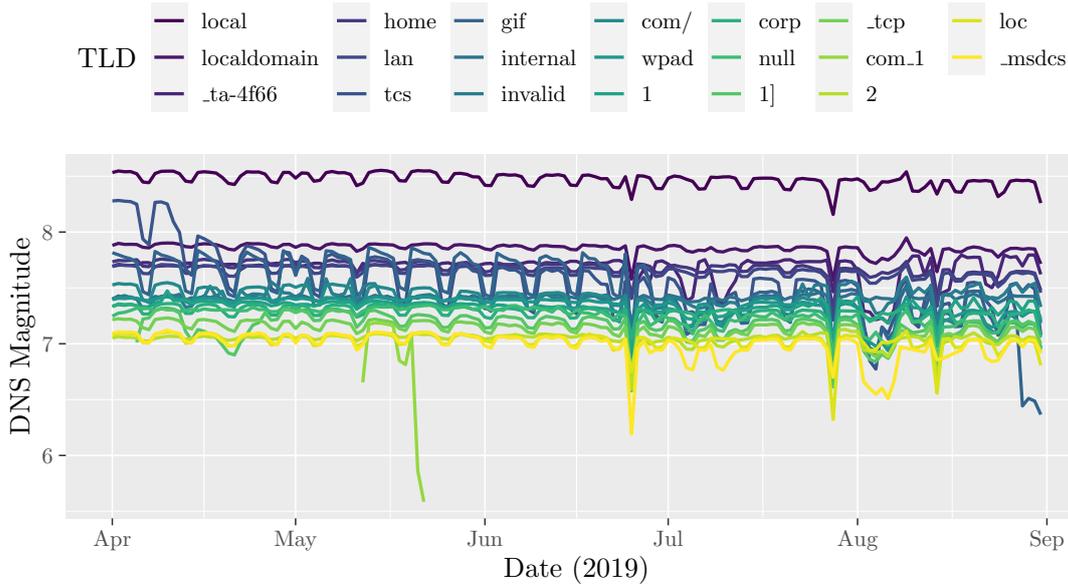


Figure 8: DNS Magnitude values of Top 20 nxTLDs over time

Comparing Figure 8 with the similar Figure 4 (containing the Top 20 TLDs, regardless of whether they exist or not) exposes a similar basic structure. However, some TLDs in the nxTLD chart appear to expose different or stronger fluctuation of DNS Magnitude values

8.5.1 nxTLDs with strong Fluctuation of DNS Magnitude Values

To identify the nxTLDs with stronger popularity fluctuation, we calculate the standard deviation of the DNS Magnitude value per nxTLD across the observation period (see Table 5). We also extend the list of nxTLDs to the Top 70, as this (a) includes more Domains with “interesting” fluctuation shape and (b) contains the nxTLD *.mail*.

TLD	Standard deviation of DNS Magnitude value
com_1	0.56
gif	0.30
tcs	0.30
zervdns	0.28
***	0.23
adsl	0.18
com/	0.15
.ta-4f66	0.13

Table 5: nxTLDs with largest standard deviation in DNS Magnitude values

To inspect the fluctuation characteristics of those, we highlight the respective nxTLD in separate plots, set against the whole array of 70 time series. Interestingly, even that small set of nxTLDs exposes very different shapes:

nxTLD 'com_1' (see figure 9) suddenly jumps into the Top 70 nxTLDs around mid May, retains a fairly constant DNS Magnitude for about a week, and then drops back as quickly as it appeared. Because it appeared and disappeared so abruptly, we believe that this was a misconfiguration in a single popular service or software (or a small set thereof).

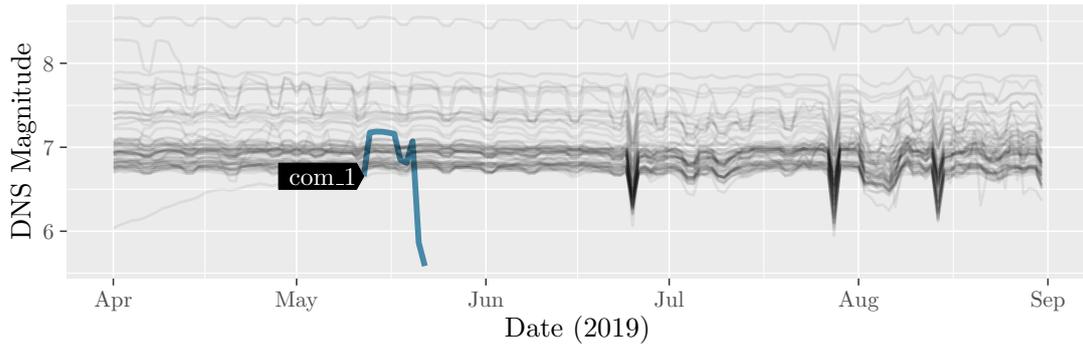


Figure 9: DNS Magnitude time series of 'com_1' nxTLD

nxTLD 'gif' (See Figure 10) has significant, constant popularity with a strong weekly periodicity. It appears to lose some of its popularity from the beginning of July, and then again in the last week of August. As the change in popularity roughly coincides with the start of the “incomplete data” period, we’re cautious on any conclusions regarding the decrease of popularity. As “GIF” (Graphics Interchange Format) is a popular image format on the Web, our (unconfirmed) hypothesis is that the source of the queries for this nxTLD is incorrectly authored Uniform Resource Locators, creating host parts ending in “.gif”.

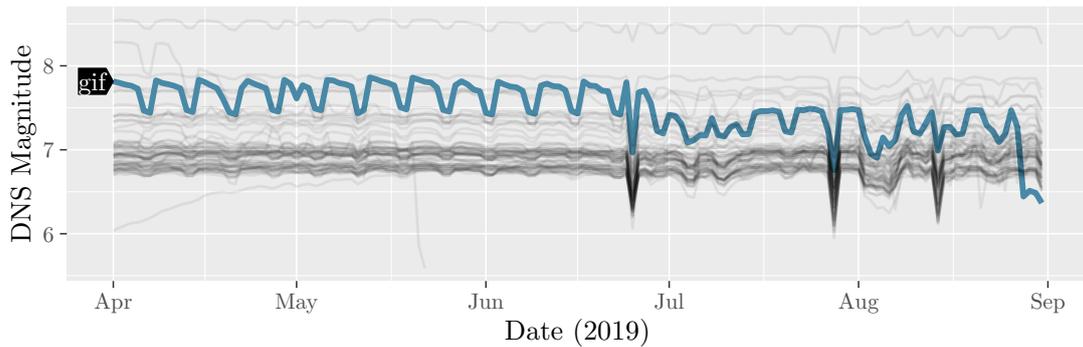


Figure 10: DNS Magnitude time series of 'gif' nxTLD

nxTLD 'tcs' (See Figure 11) exposes weekly periodicity that is similar to that of 'gif', but starts with a much higher initial DNS Magnitude. It does consistently lose some of its popularity over time, and there’s no noticeable step at the beginning of the period of incomplete data. There are reports that Windows SmartScreen attempts to resolve *canonicalizer.ucsuri.tcs* [2]. Because the original data was unavailable at the time of this finding, we could not inspect the original query data whether the popularity of the “tcs” label was indeed due to that single host name. The string 'tcs' was not applied for under ICANN’s new gTLD program in the 2012 round.

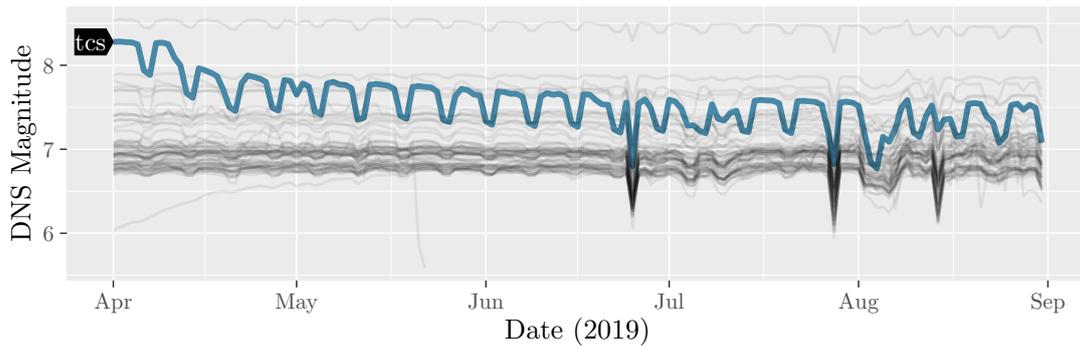


Figure 11: DNS Magnitude time series of 'tcs' nxTLD

nxTLD 'zervdns' (See Figure 12) suddenly appears with relatively high DNS Magnitude in the last week of July. Our hypothesis here is that the reason for these queries lies in incorrectly configured Pointer (PTR) records in some networks⁵. Subsequently, when servers receive traffic from those networks, and attempt to verify the reverse DNS by performing a correlating forward lookup, they trigger a query for said nxTLD.

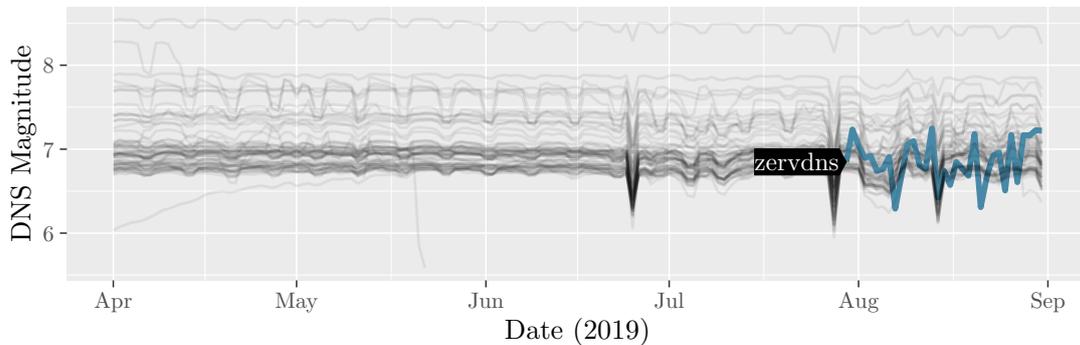


Figure 12: DNS Magnitude time series of 'zervdns' nxTLD

nxTLD '***' (See Figure 13) steadily rises in DNS Magnitude value in the first month of the observation period, with a slower increase during the remaining months. We fail to come up with an hypothesis for the source of those queries, but can observe that the chart follows the weekend / workday pattern observed with other nxTLDs of similar Magnitude.

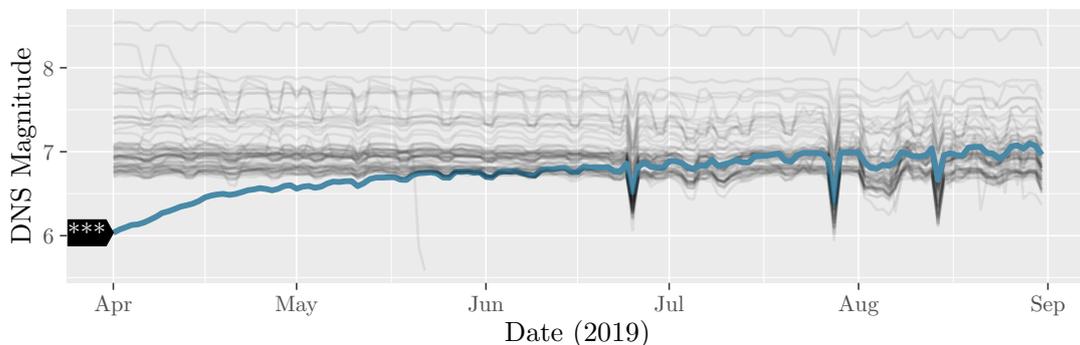


Figure 13: DNS Magnitude time series of '***' nxTLD

⁵For example, as of 2020-05-25, the PTR record for *22.38.118.92.in-addr.arpa.* resolves to *ip-38-22.ZervDNS.*

nxTLD 'adsl' (See Figure 14) does not expose a visible trend in DNS Magnitude, but varies for individual days in a pattern that looks more random than the common workday / weekend pattern. The source of those queries is unclear, though the name hints to some relation to end subscriber lines / customer premises equipment.

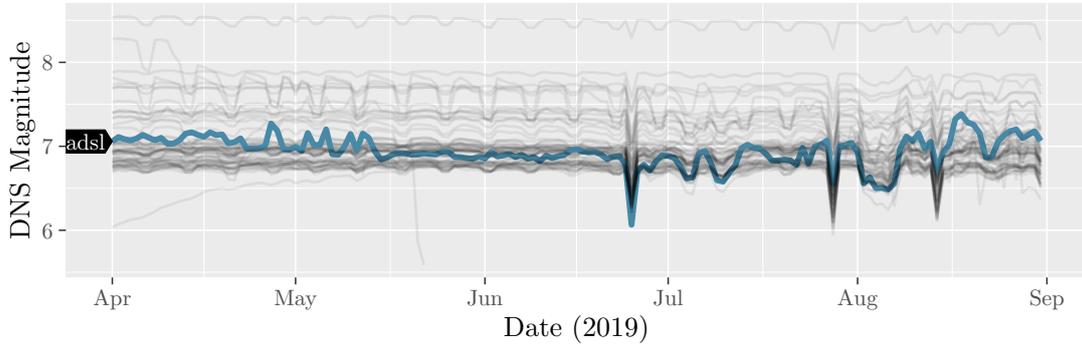


Figure 14: DNS Magnitude time series of 'adsl' nxTLD

nxTLD 'com/' (See Figure 15) exposes both a clearly visible workday / weekend pattern as well as a slight downward trend of Magnitude. Due to its similarity with the largest TLD 'com', the use of the character '/' in URLs, and its slight downward trend, our (unconfirmed) “best guess” for the reason for these queries is a malfunctioning web client software that gets gradually replaced during the measurement period.

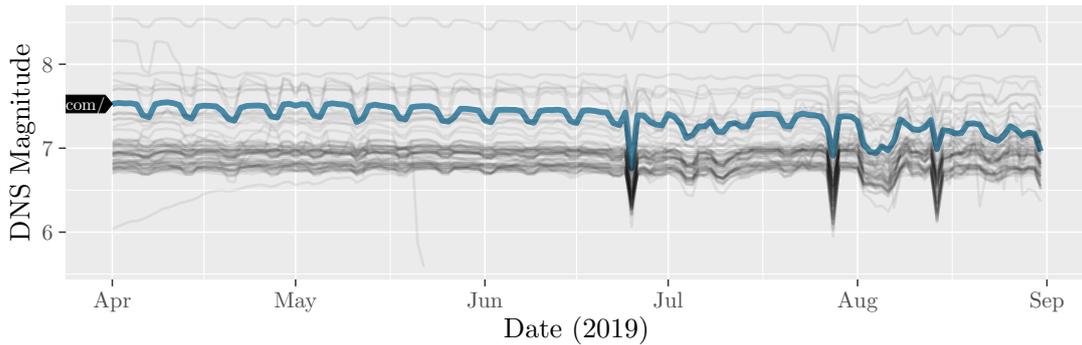


Figure 15: DNS Magnitude time series of 'com/' nxTLD

nxTLD '_ta-4f66' (See Figure 16) is a very special case amongst the labels discussed. The label is an effect of “Signaling Trust Anchor Knowledge in DNS Security Extensions” specified in RFC 8145 [31]. By means of that protocol, resolvers indicate the list of DNSSEC trust anchors they support to authoritative servers. For the measurement period with complete data (up to last week of June), the nxTLD has an almost constant DNS Magnitude value. We believe that this can be explained by the fact that queries for this label are created by resolvers automatically (rather than by user interaction with a service), and DNS Magnitude hence reflects the number of resolvers supporting this protocol, rather than the popularity of services. The fact that DNS Magnitude exposes very slight but regular upward weekend bumps indicates that more such resolvers are active on weekends than on working days.

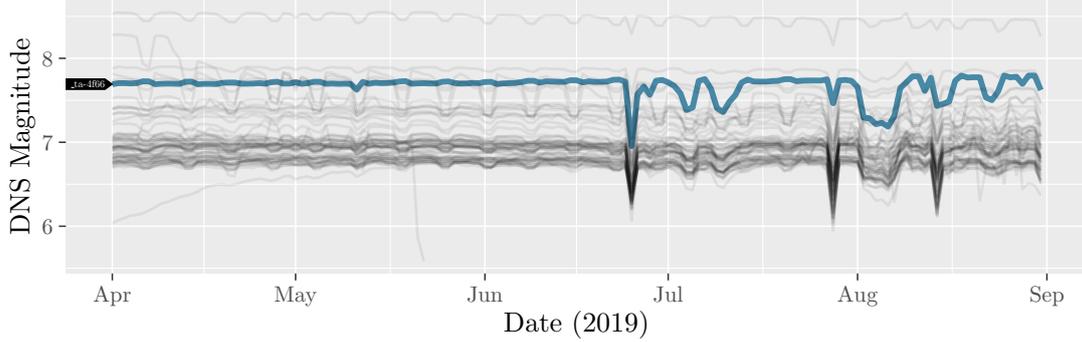


Figure 16: DNS Magnitude time series of ‘_ta-4f66’ nxTLD

8.5.2 Name Collision Management Framework ‘High-Risk’ Strings

During the 2012 round of applications for new gTLDs [15], there was concern about “Name Collision” between applied-for gTLD labels and existing use of those DNS names, for example by configuration of those names in local networks. Several studies were conducted (see “Name Collision in the DNS” [17], also known as the *Interisle Report*, and “Mitigating the Risk of DNS Namespace Collisions” [3], known as the *JAS Report*). The ICANN Board subsequently approved the “Name Collision Management Framework” [14], declaring *home*, *corp* and *mail* as “high-risk strings” whose delegation should be deferred indefinitely.

These “high-risk” strings were primarily identified by their volume of DNS queries to the root. Even though about 6 years have passed between the reports and the data collection for this paper, it is still interesting to explore whether those strings - even today - expose significant DNS Magnitude values.

Using the data created for Table 4, we extract the rows for the three strings. Table 6 lists the Rank of those three nxTLDs across all nxTLDs as well as their rank across all names (excluding the root itself). For illustration, the table furthermore includes name and DNS Magnitude of the delegated (existing) TLD with the popularity closest to the respective “high-risk” string. We see that there is still significant query traffic observed for these strings, comparable to that of medium-sized ccTLDs.

nxTLDs	Rank (nxTLDs)	DNS Magnitude	Rank (all strings)	comparable to
home	4	7.70	65	pt (7.70)
corp	13	7.29	125	ml (7.29)
mail	54	6.79	284	tm (6.79)

Table 6: Rankings of “High-Risk” gTLD application strings

When we plot the time series of DNS Magnitude values highlighted against the remaining Top 70 nxTLDs (See Figure 17), we observe that the values of these TLDs are very stable across the observation period, expose no obvious general trend, and follow the working day / weekend pattern.

The *Interisle report* describes that, in 2013, the two strings *home* and *corp* “occur with at least order-of-magnitude greater frequency than any others” (See Page 4 of [17]). As many of the gTLD strings have been delegated and put into service in those 6 years, we set the “high-risk” strings against all applied-for gTLD strings (see Table 7).

While the data is not directly comparable (Interisle primarily used queries rather than unique clients in their study), we see that quite a few new gTLDs now exceed the popularity of the “high-risk” strings. As we have shown above, both *corp* and *home* expose stable popularity, so we assume that the reason that those strings have been “overtaken” meanwhile is mainly due to the rising popularity of the new gTLDs since their delegation, rather than the sinking popularity

of the “high-risk” strings.

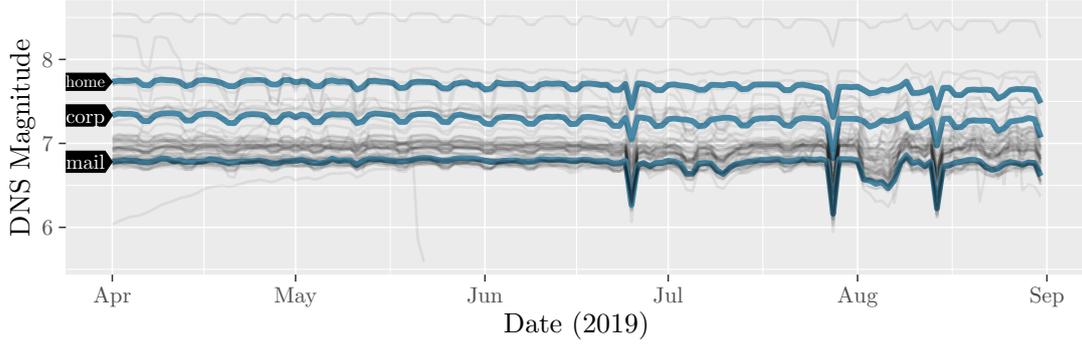


Figure 17: DNS Magnitude time series of home, corp, mail nxTLDs

ngTLD Rank	TLD	DNS Magnitude	Rank (all strings)
1	cloud	7.99	36
2	google	7.81	46
3	xyz	7.77	51
4	works	7.74	54
5	home	7.70	65
6	goog	7.68	67
7	tech	7.63	71
8	network	7.59	77
9	link	7.59	78
10	online	7.57	80
11	site	7.55	83
12	club	7.53	89
13	top	7.49	95
14	host	7.43	101
15	business	7.34	114
16	technology	7.33	117
17	media	7.31	123
18	corp	7.29	125
19	space	7.28	127
20	live	7.28	128
...
61	tools	6.80	278
62	mail	6.79	284
63	press	6.76	294

Table 7: Top-ranking new gTLD strings by DNS Magnitude

We conclude that DNS Magnitude based exploration shows that the “high-risk” strings identified in 2013 still carry significant (and stable) popularity, similar to a medium-sized ccTLDs or top-ranking new gTLD. We therefore believe that DNS Magnitude could be a useful input metric for future similar assessments.

The *JAS Report* includes in RECOMMENDATION 11 that some form of “medium-latency, aggregated summary feed describing queries reachings the DNS root” should be explored. We believe that for the reasons layed out in this document, per-TLD DNS Magnitude values could be a valuable component of such a summary feed.

8.5.3 ISO3166-1 User-Assigned Code Elements

The list of ISO3166-1 2-letter codes [21] forms the basis for the assignment of country code TLDs (ccTLDs). Out of the 676 two-letter combinations, 42 are “User-assigned code elements” (‘aa’, ‘qm’ to ‘qz’, ‘xa’ to ‘xz’, and ‘zz’). These are not delegated in the root zone, and hence constitute nxTLDs.

In November 2019, Arends and Lewis published an Internet Draft [5] that proposes the use of those as strings for private internets (“Private TLDs”). A previous version of the draft recommended designating ‘zz’ as a single private-use TLD, but the version current as of May 2020 proposes that any of them can be used by a network or application for private use.

While there is no risk that name collision may happen (due to the fact that those TLDs cannot be delegated under current policies), it is interesting to investigate the DNS Magnitude characteristics of these 42 strings at the root level.

We filter the Magnitude data across the observation period for the 42 user-assigned code elements, and calculate DNS Magnitude median and standard deviation for each of the proposed Private TLDs. Using the data for all 676 ISO3166-1 code elements, we also assign a Rank for the each of the 42 user-assigned values (see Table 9).

We observe that the 4 most popular user-assigned values (in terms of DNS Magnitude) are those where first and second letter are identical. Our hypothesis is that those are more attractive to humans (or faster to type), and therefore configured more frequently than other combinations. The least popular code is ‘qv’, which ranks 673 out of 676 ISO3166-1 elements (only ‘qj’, ‘vq’, ‘zv’ rank lower).

Figure 18 contains a plot of the time series of DNS Magnitude values of all 42 user-assigned code elements, with the TLDs ‘aa’ (as the most popular), ‘qv’ (as the least popular), and ‘zz’ (as the TLD poposed in the first version of the Arends & Lewis draft) highlighted. Many user-assigned strings (except the top-ranked) experience a notable upwards trend in Magnitude over the first three months of the observation period, though the reason for that is unclear. This appears to affect all proposed Private TLDs equally, as a plot of the Rank time series shows no visually obvious trends (See Figure 19).

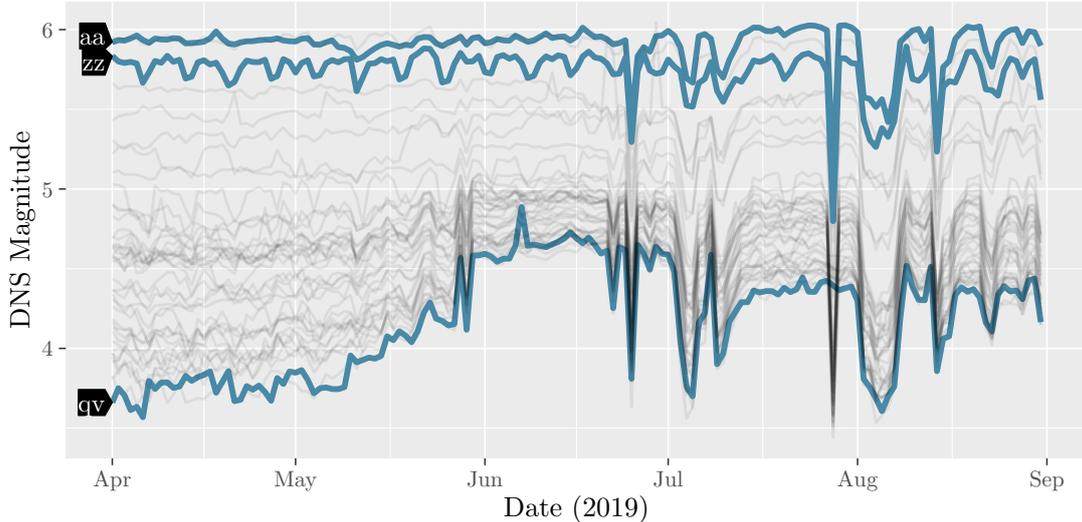


Figure 18: DNS Magnitude time series of ISO3166-1 User-assigned code elements

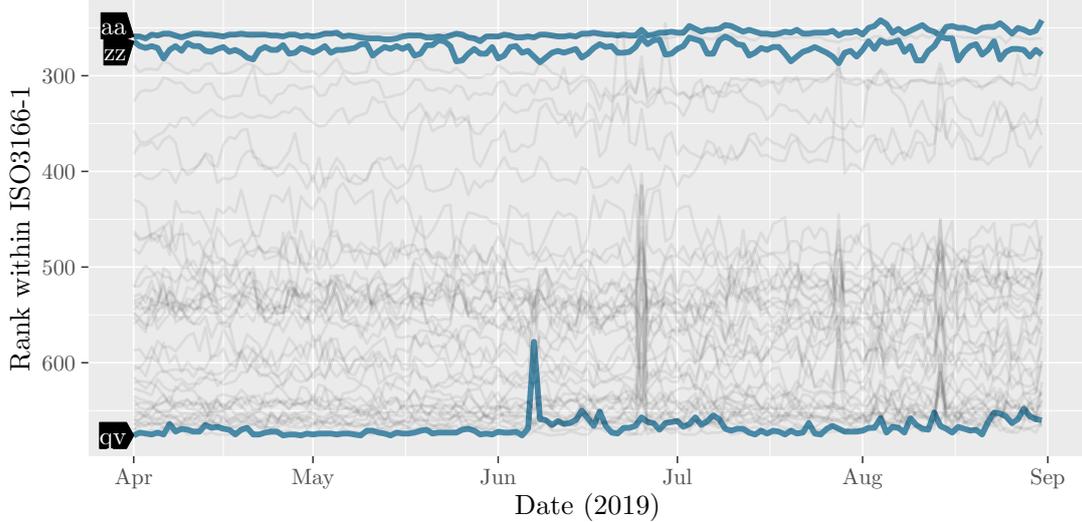


Figure 19: “Private Use” TLDs ranked amongst all ISO3166-1 strings

Further, we note that the user-assigned TLDs are almost perfectly segregated from the delegated ccTLDs. Across all 676 strings, Only 13 existing ccTLDs rank lower in terms of DNS Magnitude than the highest-ranking user-assigned ISO3166-1 string ‘aa’ (See Table 8).

We believe that the findings could be a valuable input to the decision whether or not the approach described in the Internet Draft shall proceed in standardization. If the draft proceeds, DNS Magnitude could be a tool to monitor adoption in the wild, and also assist in selecting a specific Private Use TLDs for certain applications.

ccTLD	DNS Magnitude	Rank across all ISO3166-1
er	5.87	261
gn	5.82	264
gf	5.82	265
aq	5.73	275
kp	5.70	282
mq	5.66	289
gu	5.63	293
ss	5.51	306
mh	5.48	310
gb	5.40	333
bv	5.24	377
sj	5.19	389

Table 8: ccTLDs with lower DNS Magnitude than the ‘aa’ TLD

8.6 Effects of IP Address Aggregation

DNS Magnitude uses client IP addresses as the basis of calculations. However, using the full address might not be desirable or possible: (a) Section 6.1 describes that aggregating addresses can be a countermeasure against artificially inflated numbers of unique clients by using IPv6 prefixes. Furthermore, (b) full IP addresses are considered Personally Identifiable Information (PII) in many jurisdictions, and policies might require anonymization before DNS query logs can be analyzed. Aggregation of IP addresses to their prefixes is a well-known, accepted, and widespread measure of anonymization.

As aggregation of IP addresses into prefixes influences the number of observed unique clients, it influences DNS Magnitude calculations based on such aggregated data - both the total number of unique clients as well as the per-TLD count.

It is desirable that the effects of aggregation are minimal on the resulting DNS Magnitude values - in other words, that DNS Magnitudes are stable, irrespective of whether or not aggregation was performed. For the purpose of assessing the level of stability, we look at the following two metrics

- The correlation between DNS Magnitude values based on unaggregated and aggregated versions of identical data
- The set stability of lists of TLDs ranked by their DNS Magnitude, again comparing results from unaggregated and aggregated versions of identical data

We investigate the effects using the data set d , containing the L-Root traffic from a single day (2019-08-30). From that set, we derive an aggregated data set $d' = agg(d)$ using the aggregation function $agg()$, which replaces each client IP address with its respective /24 (for IPv4) or /48 (IPv6) prefix. We then perform DNS Magnitude calculations on both data sets d and d' , and compare the resulting sets of values against each other.

We define M_{tld} as the DNS Magnitude of the TLD tld calculated from the full (unaggregated) data d , and M'_{tld} as the DNS Magnitude of the same TLD calculated from the aggregated data set d' .

Value based Correlation:

We create two-dimensional density plots, with the original DNS Magnitude values M_{tld} on the x-axis, and the DNS Magnitude values M'_{tld} from the aggregated data set on the y-axis, and calculate the Pearson correlation coefficient between the two axes. As delegated (existing) TLDs and nxTLDs appear to behave differently, we create separate charts for each group. With an almost-perfect Pearson coefficient of 0,998 existing TLDs (see Figure 20a) expose a near-perfect correlation between M' and M , while correlation for nxTLDs (see Figure 20b) is slightly weaker with a Pearson coefficient of 0,921.

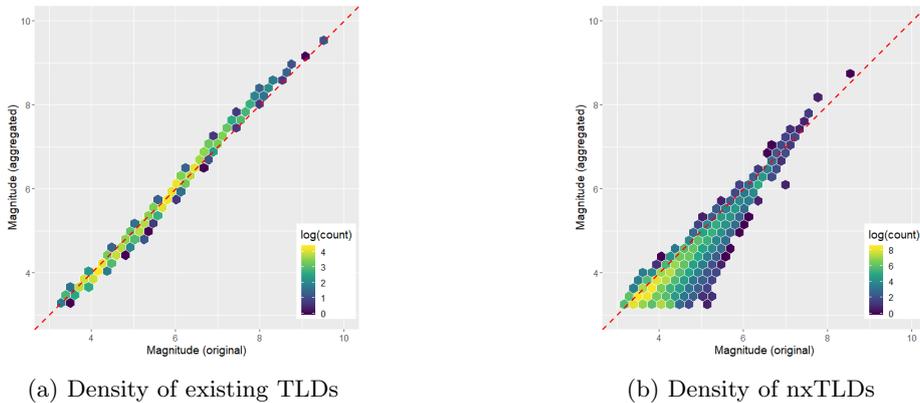


Figure 20: Density chart of TLDs by DNS Magnitude values before and after aggregation

We therefore believe that, for the given data set, (a) the impact of aggregation on actual DNS Magnitude values of existing TLDs is negligible, because the Pearson coefficient of 0,998 is very close to the value 1 (representing perfect correlation); and (b) that while the impact on the DNS Magnitude values of nxTLDs is slightly bigger, aggregated data still provides an excellent approximation for those data points. However, these findings are specific to the given data set, and cannot be translated directly to other Contexts. Repeating the analysis presented above will be required before conclusions can be drawn for other data sets.

Ranking Set Stability

DNS Magnitude values can be used to rank TLDs by their popularity. As aggregation affects the DNS Magnitude values of TLDs, it may also impact their position (rank) on popularity lists. We amend the information presented in Table 3 with DNS Magnitude and ranking information created from the aggregated data, and calculate the Ranking gain/loss as well as the difference in Magnitude (see Table 10).

	TLD	Rank (full)	Rank (agg.)	gain/loss	Mag. (full)	Mag. (agg.)	Mag. Diff.
1	com	1	2	-1	9.50	9.58	0.09
2	.	2	1	1	9.48	9.59	0.10
3	net	3	3	0	9.45	9.52	0.07
4	org	4	4	0	9.03	9.19	0.16
5	uk	5	5	0	8.82	9.02	0.20
6	info	6	6	0	8.75	8.96	0.21
7	au	7	7	0	8.68	8.92	0.24
8	de	8	9	-1	8.64	8.84	0.21
9	arpa	9	8	1	8.59	8.84	0.25
10	eu	10	12	-2	8.56	8.75	0.20
11	biz	11	11	0	8.55	8.78	0.22
12	local	12	10	2	8.45	8.78	0.34
13	br	13	16	-3	8.43	8.59	0.16
14	cn	14	14	0	8.43	8.61	0.18
15	io	15	13	2	8.35	8.67	0.32

Table 10: Unaggregated / aggregated ranking comparion

We see that even in the Top 15 TLDs, aggregation does indeed shift rankings of 8 of the 15 TLDs. However, looking at the set stability, only 1 out of 15 TLDs dropped entirely from the Top 15 list (.br; new rank is 16) - and that entry was near the bottom of the original list. We repeat that analysis for larger Top lists. Table 11 contains:

- **Set size:** The number of TLDs considered for the set, starting with rank 1
- **Retained in set:** The number of TLDs retained in the set after aggregation
- **Percentage:** The percentage of TLDs retained in the set after aggregation
- **Median rank dropped:** The median rank of TLDs which dropped out of the set as an effect of aggregation

	Set Size	Retained in set	Percentage	Median rank dropped
1	15	14	93,33 %	13
2	100	98	98 %	95
3	200	192	96 %	196.5
4	500	448	89,6 %	447
5	1000	932	93,2 %	932
6	2000	1848	92,4 %	1813.5
7	5000	4344	86,8 %	4220
8	10000	8738	87,38 %	7943

Table 11: Ranking set stability impact of IP Address aggregation on various set sizes

While we see that larger sets also have a smaller number of retained (stable) TLDs, the median of dropped TLDs shows that entries dropped are typically near the end of the list, while entries towards the top of the list might switch rankings, but rarely drop off the list entirely.

8.7 From the Cradle to the Grave

Finally, an interesting aspect of the root zone is that set of delegated TLDs is not static - new labels are added, and obsolete labels are removed. Even the relatively short observation period of May 2019 to August 2019 included several such events (five removals and one addition - see Table 12):

TLD	date	event type
honeywell	2019-06-07	<i>removal</i>
bnl	2019-07-29	<i>removal</i>
starhub	2019-08-02	<i>removal</i>
iselect	2019-08-05	<i>removal</i>
gay	2019-08-09	<i>addition</i>
duns	2019-08-30	<i>removal</i>

Table 12: TLDs added and removed from the root zone during the observation period

Note that all TLDs listed above constitute new gTLDs from ICANN’s *New Generic Top-Level Domains Program* [16], and - with the exception of *.gay* TLD - they constitute “Brand” TLDs.

The interesting question here is whether these events - the “birth” and “death” of a TLD in the DNS - have any impact on on their DNS Magnitude values.

We therefore extract daily DNS Magnitude values for each of these TLDs, and inspect the resulting time series for discontinuities around the dates of their addition or removal events.

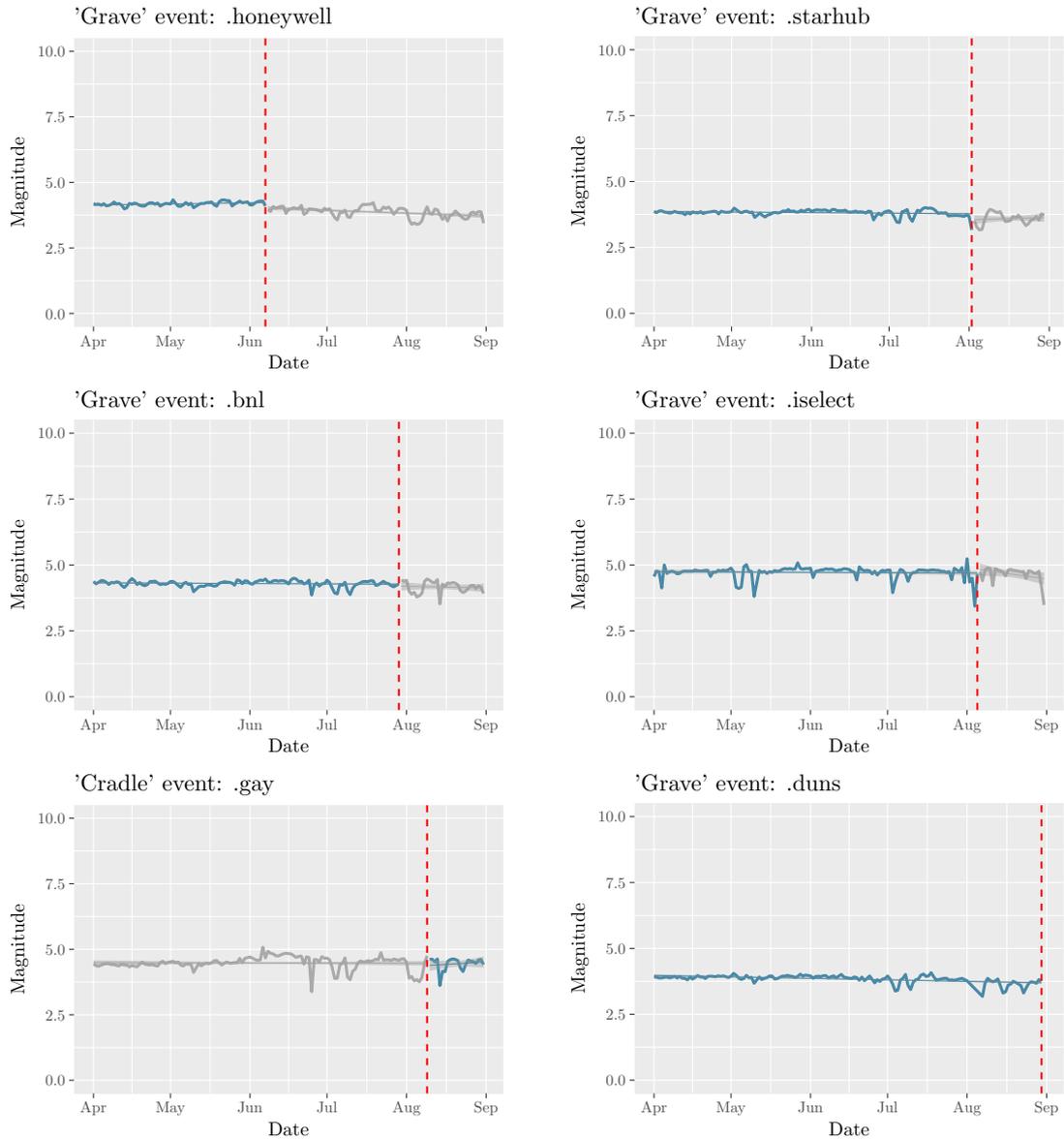


Figure 21: DNS Magnitude for TLDs added or removed from the root zone during observation period

We see that the introduction (or removal) of an (empty) TLD does not appear to significantly influence the DNS Magnitude of the respective TLD immediately. Although the data for the TLD *.honeywell* exposes a slight downward slope post removal, this effect is not apparent in the other TLDs. It might be attributed to the limited observation period.

The fact that the introduction of the TLD *.gay* into the root does not appear to increase its popularity (DNS Magnitude) immediately is surprising. Whether this effect is consistent with additions of other TLDs is - due to the limited observation period - unclear.

The findings suggest that DNS Magnitude indeed reflects the popularity of the services under a TLD, as the event of delegation of a fresh, “empty” TLD (or the removal of empty, unused TLDs) does not significantly change the amount of services offered by that TLD.

9 Limitations and further research

During the research and creation of the paper, we came up with many ideas and research questions which could not be answered in this paper, primarily due to lack of time, but sometimes also due to lack of data. This section summarizes some of those research questions for eventual further study.

- **Effects of 'Controlled Interruption':** During ICANN's first round of the new gTLD program, a range of TLDs was subjected to 'Controlled Interruption' by adding a special wildcard A and AAAA resource record to the freshly delegated TLD for 90 days. The research question here would be whether that process affected the DNS Magnitude of the TLDs during the 90 day period.
- **DHCP-churn:** The formal definition of DNS magnitude assumed static IP addresses. While it can be argued that most recursors are static and hence DHCP churn does not influence DNS magnitude in the contexts of authoritative DNS servers, the same can not be claimed for the clients of recursive DNS servers. A large ISP's recursive DNS server might see the same host querying the same domain d multiple times (even though it might have changed IP address in the mean time). The research question is whether (and to what degree) such behaviour influences DNS Magnitude values.
- **Calculations with DNS Magnitude values:** As DNS Magnitude relies on set cardinalities, merging of magnitude values (eg. single days to whole weeks) is not directly possible. While we believe that the mean or median of the values (as long as the context is identical) serves as a useful approximation of the precise magnitude for the longer time interval, it would be interesting to empirically calculate the difference.
- **Correlation of TLD age and Popularity]:** While we explored the distribution of DNS Magnitude amongst "old" and "ngtld" groups separately, this could not explain the bimodality of either group. Our hypothesis is that one factor in this bimodality is the age of the TLD. Other factors might include the fact whether a new gTLD is a "brand" TLD, or a TLD constitutes an IDN ccTLD. The research question here is "What is the source of the bimodality in each of the groups?".
- **More "Cradle and Grave" events:** In addition to the events discussed in Section 8.7, it would be interesting to understand the impact of non-gTLD additions to the root. The recently added greek ccTLD for the European Union might be an interesting candidate, but it's delegation data was outside of the available data. Furthermore, a successful General Availability event (in terms of registration volume) of a new gTLD might also be an interesting event. For example, the GA event of the `.app` TLD saw more than 200 000 registrations over a single week - but again, that event was outside of the study's observation period, and hence no data was available.
- **TTL values vs. query volume:** To the best of our knowledge, there's no broad study about the correlation of TTL values and the query volume of a domain seen at recursive and authoritative servers. If such data would be available, the query volume could be normalized based on the findings of the study, and used as a secondary popularity figure along with (or compared with) DNS Magnitude.

10 Conclusions

In this paper, we present and define *DNS magnitude*, a simple, meaningful, human friendly DNS metric, representing an estimation of the popularity of the services offered under a Domain. We describe how popularity of services corresponds to the DNS traffic volume that can be observed for a Domain, explain why counting Unique client IP addresses (rather than individual query/response pairs) is more resilient against changes of the Time-to-live of DNS resource

records, and demonstrate this with an empiric example observed in the *.at* TLD. We also show that a logarithmic scale matches the “long tail” distribution of DNS traffic much better than a linear scale. Finally, we propose to scale the resulting metric to a human-friendly range of 0 to 10, and arrive at the formal definition of DNS Magnitude as well as a prosa description: “*DNS Magnitude is a logarithmic measure for the DNS popularity of a Domain, based on counting unique clients, normalized to a range of 0-10.*”

We explore the manipulation potential of DNS Magnitude, and discuss properties of today’s Internet / DNS infrastructure most suitable for such manipulation (IPv6 allocation range sizes, and spoofing of source IP addresses). As a countermeasure for at least the IPv6 allocation problem, we propose aggregation of IP addresses to prefixes, and later show in an empiric analysis that the effects of such aggregation can indeed be negligible for specific data sets.

We subsequently apply DNS Magnitude calculations to traffic recorded on the L-Root servers, aggregating the data to the level of individual TLDs. For that result data set, we find that even with incomplete traffic data, DNS Magnitude can still provide reasonable results, particularly when comparing rankings, rather than absolute values. The software used to calculate the results is published in open source form.

We explore the fluctuation over time of DNS Magnitude values, and take a look at the distribution of Magnitude across different types of TLDs. We find that a surprising number of TLDs even in the Top 1000 (ranked by DNS Magnitude) are non-existing TLDs, and decide to take a closer look on the properties of these nxTLDs. Based on the standard deviation of the time series of daily DNS Magnitude values, we also demonstrate how simple calculations can be used to locate anomalies in that set of root labels, and discuss notable examples of such anomalies. The section about nxTLDs concludes with the examination of the Name Collision “high-risk” strings *.mail*, *.home* and *.corp* and the discussion of the recently proposed “Private TLDs”. Finally, we examine the impact of addition or removal of TLDs on their DNS Magnitude values.

With the empiric research presented in this paper, we demonstrate that DNS Magnitude can be a useful, simple, human-friendly and efficient addition to the DNS researcher’s toolbox, especially when large amounts of query/response data are to be aggregated to simple per-Domain figures or time series.

11 Acknowledgements

The study of the L-Root traffic and the creation of this paper was supported by ICANN Org’s Office of the CTO. Roy Arends provided access to L-Root data, extensive advice and review during the creation of the paper.

Sebastian Castro initially inspired the work on DNS Magnitude. Michael Braunöder created the software to analyse L-Root traffic, and performed most of the data analysis work. Arsen Stasic from the University of Vienna provided historic root zone files.

Samaneh Tajalizadehkhoob, David Huberman, and Sion Lloyd performed reviews and provided significant suggestions.

References

- [1] J. Abley and K. Lindqvist. Operation of Anycast Services. RFC 4786 (Best Current Practice), December 2006.
- [2] Luka Acha. What is canonicalizer.ucsuri.tcs? <http://security5magics.blogspot.com/2018/05/what-is-canonicalizerucsuritcs.html>, 2018.

- [3] JAS Global Advisors. Mitigating the risk of dns namespace collisions. <https://www.icann.org/en/system/files/files/name-collision-mitigation-study-06jun14-en.pdf>, 2014.
- [4] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, and David Dagon. Detecting malware domains at the upper dns hierarchy. In *USENIX security symposium*, volume 11, pages 1–16, 2011.
- [5] Roy Arends and Ed Lewis. Top-level domains for private internets. Internet-Draft draft-arends-private-use-tld-01, IETF Secretariat, May 2020. <http://www.ietf.org/internet-drafts/draft-arends-private-use-tld-01.txt>.
- [6] Internet Assigned Numbers Authority. Root servers. <https://www.iana.org/domains/root/servers>, 2019.
- [7] Internet Assigned Numbers Authority. Root zone database. <https://www.iana.org/domains/root/db>, 2019.
- [8] Albert-Lszl Barabasi. *Linked: The New Science of Networks*. Perseus Books Group, 2002.
- [9] Michael Braunoeder. Dns magnitude calculation from l-root (imrs) data. <https://github.com/nic-at/dns-magnitude-lroot>, 2020.
- [10] Nevil Brownlee, Kimberly C Claffy, and Evi Nemeth. Dns measurements at a root server. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, volume 3, pages 1672–1676. IEEE, 2001.
- [11] Sebastian Castro. Domain popularity ranking revisited. <https://www.centri.org/members-library/library/centri-event/r-d8-castro-domain-popularity-ranking-revisited-20160517.html>, 2016.
- [12] Xun Fan, John Heidemann, and Ramesh Govindan. Evaluating anycast in the domain name system. In *2013 Proceedings IEEE INFOCOM*, pages 1681–1689. IEEE, 2013.
- [13] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice), May 2000. Updated by RFC 3704.
- [14] Internet Corporation for Assigned Names and Numbers. Name collision occurrence management framework. <https://www.icann.org/en/system/files/files/name-collision-framework-30jul14-en.pdf>, 2014.
- [15] Internet Corporation for Assigned Names and Numbers. List of new gtld applications. <https://newgtlds.icann.org/newgtlds.csv>, 2019.
- [16] Internet Corporation for Assigned Names and Numbers. Icannew gtld program. <https://newgtlds.icann.org/en/about/program>, 2020.
- [17] Interisle Consulting Group. Name collision in the dns. <https://www.icann.org/en/system/files/files/name-collision-02aug13-en.pdf>, 2013.
- [18] P. Hoffman, A. Sullivan, and K. Fujiwara. DNS Terminology. RFC 8499 (Best Current Practice), January 2019.
- [19] Alexander Holmes, Andrew Simpson, Karthik Shyamsunder, Srinivas Sunkara, Eyal Lanxner, Nir Zohar, Leonard Orentas, Matt Larson, Mark Koster, Yona Mankin, et al. Domain popularity scoring, December 9 2014. US Patent 8,909,760.
- [20] Dan hubbard. Cisco umbrella 1 million. <https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/>, 2019.
- [21] ISO. *ISO 3166-1:1997: Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*. 1997.

- [22] Alexander Mayrhofer. Dns magnitude - another approach on domain name popularity. <https://www.centri.org/members-library/library/centri-event/r-d9-mayrhofer-dns-magnitude-20161129.html>, 2016.
- [23] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Internet Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482.
- [24] P.V. Mockapetris. Domain names - implementation and specification. RFC 1035 (Internet Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604, 7766, 8482, 8490.
- [25] Giovane Moura. How to choose ttl values. https://labs.ripe.net/Members/giovane_moura/dns-ttl-values-what-value-is-the-best-one, 2019.
- [26] Giovane C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt, and Marco Davids. When the dike breaks: Dissecting DNS defenses during DDoS. In *Proceedings of the ACM Internet Measurement Conference*, October 2018.
- [27] Sridhar Muppidi. How to use dns analytics to find the compromised domain in a billion dns queries. <https://securityintelligence.com/how-to-use-dns-analytics-to-find-the-compromised-domain-in-a-billion-dns-queries/>, 2018.
- [28] T. Narten, G. Huston, and L. Roberts. IPv6 Address Assignment to End Sites. RFC 6177 (Best Current Practice), March 2011.
- [29] Root Server Operators. root-servers.org. <https://root-servers.org/>, 2019.
- [30] Inc. Verisign. The domain name industry brief. <https://www.verisign.com/assets/domain-name-report-Q22019.pdf>, 2019.
- [31] D. Wessels, W. Kumari, and P. Hoffman. Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC). RFC 8145 (Proposed Standard), April 2017. Updated by RFC 8553.

	TLD	DNS Magnitude	Rank across all ISO3166-1	Standard Deviation
1	aa	5.93	257	0.14
2	xx	5.92	258	0.16
3	zz	5.79	272	0.14
4	qq	5.60	300	0.17
5	xy	5.50	307	0.16
6	xl	5.36	343	0.15
7	xn	5.26	369	0.20
8	qt	5.15	402	0.15
9	xp	5.02	445	0.25
10	xo	4.90	484	0.20
11	xm	4.89	486	0.20
12	xd	4.86	495	0.21
13	xi	4.80	512	0.20
14	xa	4.77	519	0.21
15	qu	4.77	520	0.22
16	xf	4.76	528	0.19
17	xz	4.74	532	0.22
18	xc	4.72	534	0.21
19	xu	4.71	542	0.19
20	qw	4.71	543	0.21
21	xv	4.69	548	0.23
22	xe	4.64	562	0.22
23	qr	4.63	566	0.23
24	xs	4.59	583	0.23
25	xh	4.56	596	0.23
26	xj	4.53	606	0.23
27	qs	4.49	618	0.25
28	xk	4.47	621	0.23
29	qm	4.43	635	0.27
30	xr	4.40	637	0.27
31	xt	4.36	644	0.26
32	qz	4.36	646	0.26
33	xq	4.34	649	0.26
34	xb	4.33	651	0.28
35	xw	4.33	654	0.28
36	xg	4.31	658	0.28
37	qp	4.29	659	0.29
38	qy	4.27	662	0.31
39	qx	4.26	664	0.30
40	qn	4.21	668	0.30
41	qo	4.20	670	0.34
42	qv	4.19	673	0.34

Table 9: DNS Magnitude data for the proposed "Private TLD" strings