Zone Scanning at a ccTLD: Detection and Analysis

Pascal Huppert*†, Moritz Müller*‡, Thymen Wabeke‡, Cristian Hesselman*§,
Pieter-Tjerk de Boer*, Ralph Holz*†
*University of Twente
{pascal.huppert,m.c.muller,c.e.w.hesselman,r.holz,p.t.deboer}@utwente.nl
†University of Münster
{pascal.huppert,ralph.holz}@uni-muenster.de
‡SIDN Labs
{moritz.muller,thymen.wabeke}@sidn.nl
§SURF
cristian.hesselman@surf.nl

***Abstract*—The Domain Name System (DNS) contains data that is valuable for research, and domain names themselves can be of substantial commercial value. For this reason, actors from both academia and industry systematically query the DNS. However, it is unclear how frequently these DNS scans occur and how they affect the DNS infrastructure. Identifying scans in DNS traffic allows operators and researchers to make informed choices about provisioning of infrastructure and implementation of countermeasures, as well as enabling applications such as DNS-based popularity lists to ignore artificial traffic.**

We provide a first look at DNS scanning from the perspective of the operator of .nl, a popular country code top-level domain (ccTLD). We extrapolate that about 33% of all traffic can be attributed to scanning. We identified scans manually and through clustering, and we categorized the different kinds of scanning. Most scanning is for lists of second-level domains (2LDs) or subdomains (domain names with more than 2 labels), often involving tens of resolvers, such as machines from the networks of hosting providers or public resolvers.

I. Introduction

The DNS is responsible for translating human-readable domain names into IP addresses. The information it contains is relevant for commercial [1], academic [2] and malicious purposes alike [3]. The DNS is also a frequent target of scanning, i.e. large-scale, systematic querying for gathering data in bulk, which supports these activities by detecting new domain names or observing trends in the DNS.

DNS operators, eager to optimize their setup for eyeball customers, may want to know which traffic is caused by scans and can therefore be ignored in optimization efforts. Scans can also artificially inflate the popularity of domain names in DNS-based popularity lists such as Tranco [4] and others. Detecting scans could help to improve the accuracy of these lists and quantify the load that scans put on DNS infrastructure.

Our work takes a first step in this direction and assesses the prevalence of *DNS scanning* and its impact on .nl name servers. We perform classification of scanning traffic and describe the types and volume of scanning we detect.

We use the perspective of .nl, one of the oldest and top 10 most popular ccTLDs at over 6 million registered domain names [5]. We use passive measurement data and examine it for signs of scanning, relying on manual classification and k-Means clustering. .nl has a large market share in the Netherlands, giving it an important domain name economy and making it particularly interesting for scanning. The zone file of .nl is not publicly available. Our findings may also apply to similar top-level domains (TLDs) with unpublished zone files, including all top-10 ccTLDs.

We extrapolate from our results that about 33% of traffic stems from scans on a typical day. Some scans are performed from single machines, others from multiple resolvers. A single such scan caused 4.6% of traffic on one unremarkable day we analyzed (April 3rd, 2024) using public resolvers, whereas exceptionally large and rare scans can cause daily traffic to double. In addition to scanning for 2LDs, scans for deeper-level domains (subdomains) are commonplace. Finally, we find evidence that scans can inflate the popularity ranking of domain names: Multiple names repeated by large scanning campaigns appeared in the Tranco popularity list [4] despite running neither a web server nor any other service.

The rest of his paper is structured as follows: After providing background and related work (section II, section III), we lay out the considerations for our methodology in section IV and describe our analysis in section V. We present the results in section VI, interpret them in section VII and give our conclusions in section VIII. This work builds on a thesis by the first author [6].

II. Background

A. The Domain Name System (DNS)

The DNS is the distributed system of name servers that translates human-readable domain names like `example.nl` into IP addresses such as `94.198.159.35`.

The DNS name space is managed by different entities. SIDN runs the authoritative name servers (AuthNSes) for .nl. These harbor the zone files, containing name server data for all

2LDs such as `example.nl`. When a client device needs to resolve a domain name, it contacts a recursive resolver, which performs name resolution on its behalf. Unless the results are present in cache, for any domain name ending in .nl, SIDN's AuthNSes are contacted.

Some registries choose to make their zone files public, thereby providing a complete list of 2LDs. .nl's zone file is not public, but is available for academic research under an NDA.

B. Motivation for DNS scanning

The DNS is a frequent target of scanning, both for academic [2], [7]–[12] and commercial purposes, with tools for fast scanning available [13], [14]. Various motivations exist.

1) Domaining: Businesses may buy an unclaimed domain name, profiting from its future resale. This practice is commonly called *domaining*. Domainers benefit from knowing which popular domain names are available or registered. They can find out by scanning DNS zones and WHOIS services. From conversations with domain resellers, we learned of one group using both databases for finding names. Domaining also includes the practice of drop catching[1], i.e., registering a domain of interest immediately after expiry, using high-frequency automated queries to monitor the status. Other purposes of DNS scanning include vulnerability scanning as part of penetration tests, and trademark protection.

2) Malicious purposes: Malicious actors may register expired or look-alike domain names for impersonation or phishing. Other motivations include finding vulnerable software installations, and malware trying to discover command and control (C&C) servers (domain generation algorithms [3]).

3) Academic research: Interest of academics in scanning the DNS stems from various types of attacks on the DNS, and attacks using it, due to its relevance to network operations [7], [9]. SIDN itself runs a DNS-scanning tool to validate and measure the implementation of DNS Security Extensions (DNSSEC).

C. Definitions

We regard a *DNS scan* as a large number of DNS queries performed in a coherent and systematic fashion. Scans gather information about domain names in bulk, not serving immediate user needs such as surfing the web, sending email or using other software only peripherally concerned with the DNS. This broad range of activities includes domaining, monitoring (uptime or registration status), and even collecting data for bulk email sending or web scraping/crawling. We intentionally choose such a broad and open definition in order to not exclude types of scans that are still unseen. We use the term *resolver* (or *source*) to indicate a single IPv4 or IPv6 address, unless specified otherwise.

III. Related Work

Our work provides a first look at DNS scanning, and its classification, types, and quantity. To the best of our knowledge, we are the first to provide an overview of the amount and types of scanning found in any part of the DNS.

[1]e.g. https://www.pronkwebsites.nl/sidn-dropcatch-script/

a) Active DNS measurement: Various researchers actively queried DNS servers, e.g. in order to measure prevalence of standards [8], [9], [11], gather datasets for other measurements [12], find rogue or unintentionally open servers [10], [15] or gather historic data [2], [7].

b) Targets and Tools: DNS zone scans require a source of target names because the potential value range for domain names is too vast to probe exhaustively. A majority of names can be mined from Certificate Transparency (CT) logs and Common Crawl data. Not all domain names can be found this way, however, and new names are usually found with some delay [16], [17].

DNS scanning software is freely available, such as ZDNS [13], [18], MassDNS [14], and variations. MassDNS also comes with lists of open resolvers and common subdomains as well as scripts for generating subdomains, extracting names from CT logs and reverse-lookup of the IPv4 address space [14], making it ready-to-use for subdomain enumeration and other scans.

OpenINTEL [2] is a large-scale measurement project with proprietary software, running active scans since 2015. The authors claim that "[I]f lots of researchers were to set up similar infrastructures, this would have a significant and possibly disruptive impact on the Internet", which has not been assessed yet.

c) Characterizing name server traffic: Several studies tried to characterize traffic observed at authoritative name servers. Various ones used traffic from various authoritative name servers to understand resolver characteristics, including the deployment of new protocol extensions [11], issues with the DNS standard itself [19], and more.

Little research studied *classification* of individual resolvers: [20] used machine learning to classify monitoring traffic, e.g. to exclude it from DNS-based popularity rankings, achieving an accuracy of over 99%. Similar to this, the authors of [21] use an approach of feature engineering and machine learning on traffic from .nl to classify resolver types, e.g. resolvers used by eyeball networks. Both studies had ground-truth data available for model training. Another case study [22] used learned Word2Vec embeddings of resolvers and domain names, resulting in embeddings that contain information about a resolver's organization, country and number of queries sent.

Our goal is related: detecting and examining *scanning* resolvers. We perform clustering for classification and for data exploration. We thereby hope to help operators gain a clear picture of what and how much scanning activity exists. Our feature set differs significantly from those used by [20] and [21], and we have very few ground-truth examples available. To the best of our knowledge, no previous study attempted to detect DNS scanning.

IV. Considerations about DNS Scanning Patterns

As there is no exhaustive list of scan types and their behavior, we first inventorize activities that we consider DNS scans and hypothesize how traffic created from these scans might look like from the perspective of a ccTLD operator

(Figure 1). We compare these patterns against traffic patterns expected from users, enabling classification.

For example, domaining could lead to a large number of queries within a short time frame asking for non-repeating 2LDs (Figure 2, Resolver 2), with the goal of validating the registration status of a large number of names. Other traits include a high total query count, very few repetitions, a consistent delay in-between queries, and more.

We hypothesize that scan traffic is more homogeneous (less diverse and mixed) than user traffic and follows certain patterns. The main reason for this is that DNS scans are generated algorithmically, whereas ordinary traffic is generated by a multitude of actions and users: Following popularity, using different query types, sometimes containing spelling mistakes and so on. Even if a scan uses diverse lists of domain names, we still expect the other query parameters to show patterns. We describe some patterns here in detail, while Figure 1 provides an exhaustive overview for reference.

Time Because scans are run by a algorithms, we expect their queries to show little variance in traffic volume in time, creating unusually even traffic volumes or short bursts (Figure 2 Resolver 2, Figure 3). User traffic changes organically with usage patterns and the time of day.

Name order Name lists may be ordered, meaning scans may go through a list in alphabetical order (domaining, Figure 2 Resolvers 2 and 3), or target some specific names/2LDs repeatedly (subdomain enumeration, monitoring). User traffic should have no particular order, but be influenced by popularity.

Query options Parameters such as query type may be chosen according to the specific purpose of the scan and show less variance than user traffic (Figure 2 Resolver 2 and 3), e.g. domaining may use a single query type such as `AAAA` for all queries. User applications, on the other hand, use various different query types.

Properties of names Depending on scan type and motivation, scanned names could be almost completely nonexistent (e.g. when using a name list from another TLD), or existent (checking a list of known existing names, web crawling, monitoring, drop catching). Queries may ask for exactly 2 labels (expected from domaining, but also due to query name (qname) minimization [23]), or always for more than 2 labels (subdomain enumeration). Names may never be repeated (when scanning for registration status), or constantly (2nd label in subdomain enumeration). In user traffic, longer names generally occur less often, and popularity and language determine a characteristic distribution of starting letters. Depending on the type, this may not be the case for scans.

Technical implementation Instead of well-established DNS resolver software (e.g., BIND and Unbound), scanners may use simpler ad-hoc implementations with incorrect or unusual technical implementation, e.g. not randomizing transaction IDs, not using TCP when necessary or even wrongfully setting the recursion desired flag, which is invalid for AuthNS.

TABLE I
FIELDS OF THE MEASUREMENT DATA

From	Field
DNS Query	Transaction ID, Question type, Recursion desired, Query class, DNSSEC checking disabled, EDNS UDP packet size indication, DNSSEC ok, EDNS client subnet, Query name, Request length, Question count, Zero bit, Operation code
DNS Response	Authoritative answer, Additional record count, Recursion available, Name server count, Truncation, Response code, Authenticated data, Result length, Answer count
UDP/TCP	Source port, Destination port
Derived Resolver Details	Country, AS organization, AS number, Public resolver indication
IP	Protocol, IP version, Source address, TTL, Destination address
Server-side	Timestamp, Server location, Processing Time

V. METHODOLOGY

First, we explore the dataset by manually analyzing TLD traffic (i.e., queries) from individual resolvers. Then, we form a set of numerical features and apply a clustering algorithm to the resolvers. This results in clusters that contain resolvers with "similar" behavior, according to the chosen features. By inspecting each cluster and the feature space, we find more scans as well as further types of campaigns, and we can draw conclusions about traffic in total. The limitations of this approach are that the outcome of the clustering is dependent on the choice of features, and that the manual inspection of each cluster in costly to perform on a large scale.

Figure 1 lists the visualizations we use in manual classification in the rightmost column and visualizes the reason for including each of them, for lookup from right to left.

A. Dataset

We collected DNS traffic from all anycast instances of AuthNSes using a tool called ENTRADA [24], [25]. The data fields are listed in Table I. Due to the large volume of traffic, we use data from only two individual days: Dataset 1 (D1) from April 3rd and D2 from July 3rd, 2024. D1 consists of 4.9 billion queries from 1.6 million resolvers and D2 of 3.6 billion queries from 1.5 million resolvers.

We perform our data exploration, analysis and model building on dataset D1. Because our clustering uses over 100 different features and 41 different weights (Table IV) as parameters, we take care not to over-adapt (overfit) our model to D1 and use D2 to validate our method. This also allows us to calculate statistics about D2.

B. Manual Classification

For each IP address, we lay out all visualizations and data that are described in the rightmost column of Figure 1, to be judged manually. We chose these in such a way that all columns of the data Table I are represented and related ones (e.g. query name and response code) are visualized together, allowing scans to be spotted. Figure 2 and Figure 3 show

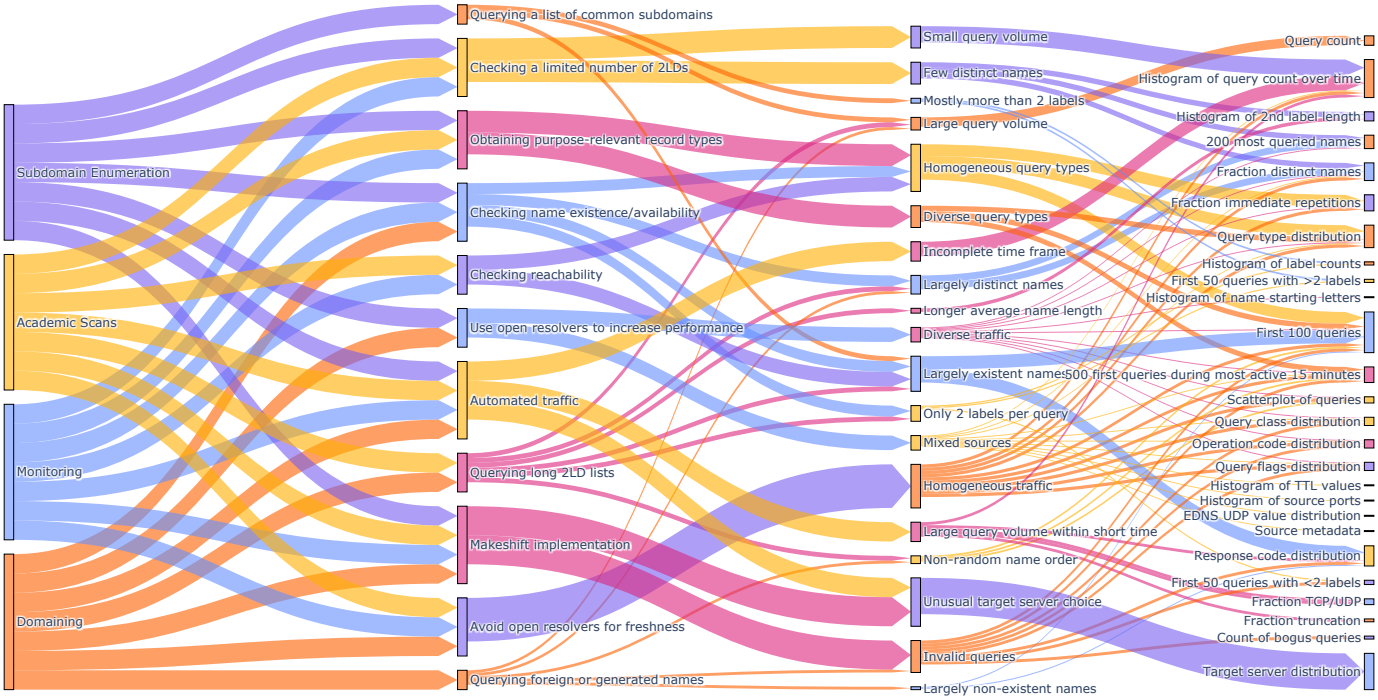

Fig. 1. Our building of hypotheses from left to right: Scan types that we expected (left), what approaches they may use, attributes of their respective traffic and in which visualization to spot them during classification (right). An arrow from A to B means "A is likely to show up in/as B".

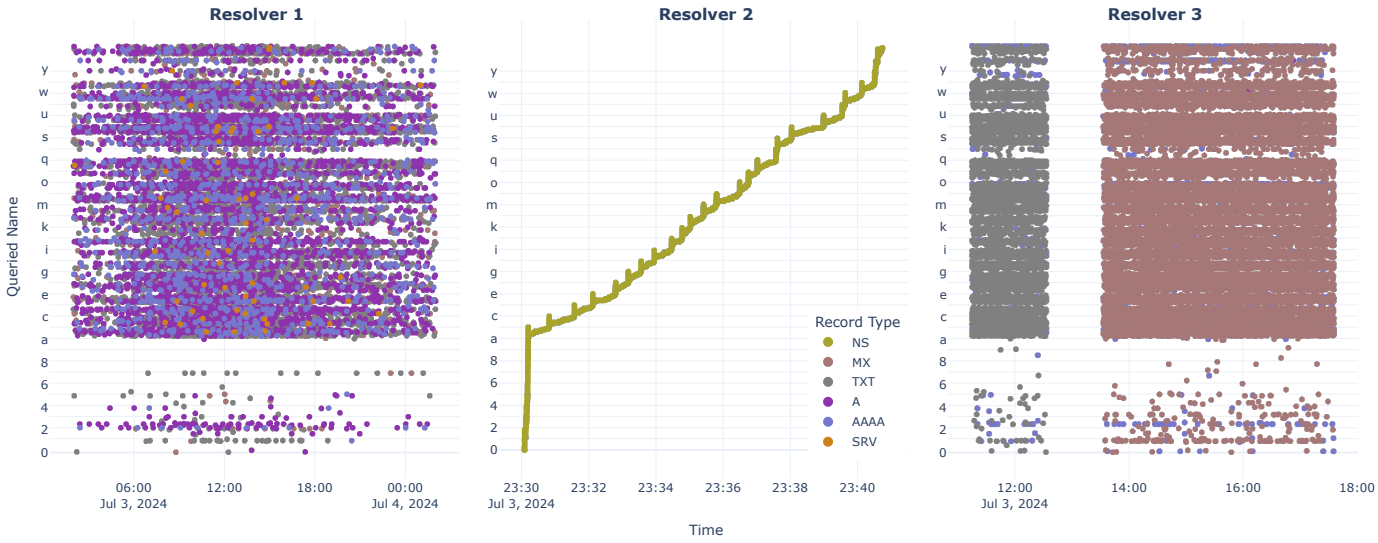

Fig. 2. Examples of resolver behavior (from D2) as visualized in scatterplots for manual classification. Each dot is a query, with time on the x-axis and the domain name's second label on the y-axis. Colors indicate query types. Resolver 1: *Non-scan traffic* from a Microsoft resolver (no apparent patterns). Resolver 2: *Scan* (campaign #15 in Table III) working in alphabetical order. Resolver 3: *Multi-phase scan* with varied query types, not represented in Table III.

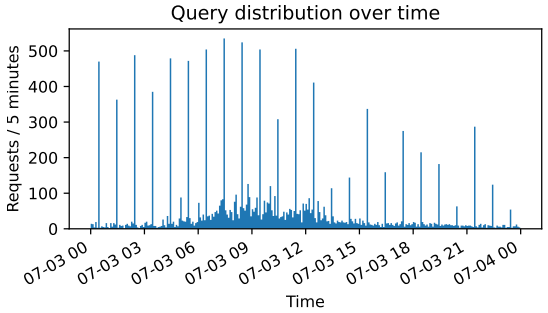

Fig. 3. Histogram of a resolver's (mixed) traffic, showing periodic scanning.

examples of the visualizations, while the full list of outputs used can be seen in the rightmost column in Figure 1.

While most scans are clearly distinguishable as such, and other resolvers showed no suspicious features (Figure 1, third column), others showed a mixed features that could not with confidence be labeled either scan or inconspicuous. For example, if only the time distribution is abnormal (as Figure 3), then this alone is insufficient for us to classify the traffic as scanning. Additionally, traffic from multiple sources may be handled by a single recursive resolver and therefore be mixed

TABLE II
TRAFFIC FROM DIFFERENTLY SIZED RESOLVERS IN D1

Filtering by	Resolvers remaining		Queries from these	
All	1 627 978	100.0%	4 855 654 326	100.0%
#Queries $\geq$ 1 000	197 530	12.1%	4 740 800 824	97.6%
#Queries $\geq$ 10 000	52 776	3.2%	4 228 242 991	87.1%

from our vantage point. We therefore mark such resolvers as "unclear", introducing a three-class classification. Resolvers are classified "scan" if the vast majority (i.e. more than 80%) of their traffic is judged to be scanning, and "non-scan" when very little ($< 5\%$) is judged to be scanning. We also validated our hypotheses and annotations through discussion with DNS experts other than the authors, including the DNS operators of .nl. In total, we classified 695 resolvers from D1 manually, among them the most active scans in D1.

C. Feature Engineering & Clustering

We develop numerical features based on our successful scan identifications, similar to the previously used visualizations. Table IV in the appendix lists these features. We apply the feature calculation on D1 and D2 to all resolvers sending at least 10 000 queries. This improves the meaningfulness of the features and makes individual users less likely to be identified. Only little traffic is excluded (Table II).

Some features are just standardized (normalized), while we apply preprocessing to others (e.g., using a logarithmic scale for counts). Because not all features are similarly important, they are then weighted (Table IV). We use distances in the resulting feature space to double-check manual classification results.

We then apply k-Means, a linear clustering method, with the goal of partitioning resolvers into homogeneous clusters, meaning no cluster should contain both scanning and non-scanning resolvers. We ignore resolvers marked as "unclear" and *obtain a binary classification* from the clustering as follows: Each cluster is given a label according to the majority of its labeled samples. Every cluster in our result for D1 contained at least one labeled resolver (scan/non-scan).

We optimize for accuracy in this classification, penalizing clusters that contain both scanning and non-scanning resolvers. More precisely, we adapt the cluster count k, preprocessing and feature weights to obtain best accuracy on the manually labeled resolvers in D1 while retaining sensible values and explainability, i.e., a sufficiently small k, straightforward preprocessing, and just 5 groups of feature relevance (Table IV). We keep this simplicity to reduce overfitting, which can generally occur when adding more parameters (in this case, the weights).

Finally, we tested the clustering on D2 by classifying 10 randomly chosen resolvers from each cluster manually using the same process as described before, but furthermore estimating the fraction of scan traffic from each resolver, i.e., labeling by a percentage, while keeping the "unclear" class. We do so through further careful examination of the scatterplots and histograms. As these samples are representative of their clusters, we extrapolate from these to estimate the total traffic composition of D2.

VI. RESULTS

A. Clustering

We reached an accuracy of 98% and an F_1 score of 99% on the small ("training") set of 695 manually classified resolvers from D1, using $k = 17$ clusters. This means that 2% of resolvers would have been misclassified. Increasing k improves this only slowly.

The 17 clusters from D2 contain 116–6 600 resolvers and 5–653 million queries each (we address them as clusters 0 to 16). By manual classification of 10 samples per cluster, we obtained the following results: Three clusters contained purely scanning traffic (100%); for three others, more than 70% of their traffic is scanning; seven clusters contained 30% or less scan traffic. The other four clusters were mixed (30–70%), and some contained resolvers with quite different classification. Assuming the same binary classification of resolvers as for D1, accuracy of the clustering on D2 is 93.3%, and the F_1 score is 95.5%, significantly lower than the clustering on D1, likely due to the non-random choice of samples in D1. Many resolvers contain mixed traffic, adding to the limitations of the simplified classification used.

B. Total scanning traffic

We estimate the fraction of total scan traffic from the samples from D2. This yields that approximately 33% of queries in the clustered data are caused by scanning (between 21% and 54% with 95% confidence). The details of this calculation are given in section XI. The small resolvers excluded from clustering increase uncertainty by a maximum of $[-4\%, +7\%]$. In D1, 9.7% of traffic (471 million queries) was manually identified by us as scanning (lower bound).

C. Campaigns

We identified 60 different campaigns originating from 438 IP addresses in D1, with each campaign using from 1–207 resolvers. The smallest operation only sends 33k queries from one address, while the largest one sends 233 million through hundreds of public resolvers (cf. Table III). This subdomain scanning campaign causes millions of excess and repeated queries and thereby 4.6% of the traffic. ASes of hosting providers account for 9 out of the largest 15 scans.

D. Scan Types

2LD scanning and subdomain enumeration are the two most common kinds of scans we encountered.

1) Academic Scans: A campaign by the Georgia Institute of Technology and SIDN's own DNSSEC validation tool sent enough traffic to be among the largest campaigns. We also observed scans performed by OpenINTEL and the Technical University of Munich (TUM), which we could confirm with the respective researchers. All of these organizations are involved in DNS research (TUM: [23], GA Tech: [13]).

TABLE III
OVERVIEW OF SCAN CAMPAIGNS FOUND IN D1

#	#Queries	#IPs	Querying	Origin
#1	233.2M	83[1]	Subdomains	Open resolvers, scans 23 2LDs
#2	65.3M	4	2LDs	German hosting provider
#3	39.8M	207	2LDs	Multiple Swedish hosting providers
#4	21.7M	16	2LDs	TU Munich
#5	21.2M	67	2LDs	Georgia Institute of Technology
#6	17.3M	1	2LDs	South Korean research institute
#7	14.3M	2	2LDs	Dutch hosting provider
#8	11.5M	1	2LDs	US/international hosting
#9	9.3M	1	2LDs	German hosting provider
#10	7.1M	4	2LDs	OpenINTEL [2]
#11	4.1M	2	2LDs	own DNSSEC validation tool
#12	3.7M	2	2LDs	US hosting provider
#13	3.4M	1	2LDs	Dutch hosting provider
#14	3.0M	1	2LDs	US/international hosting provider
#15	2.7M	1	2LDs	Austrian hosting provider
Rest	13.7M	45	*	45 others, each <2M queries
Total	471.1M	438	*	9.7% of traffic in D1

[1] This number is an underestimate, because many resolvers are affected partially. The campaign involves many open resolvers and a fixed list of 2LDs, allowing us to determine only the number of queries sensibly. Some leak the actual origin of the campaign in the EDNS client subnet field: The AS of a Turkish hosting provider.

These academic scans were not distinct from other 2LD scanning in their characteristics, and we could only judge them as academic through knowledge of their origin networks and DNS reverse pointers. This shows the importance of setting up reverse DNS pointers when performing scans.

The same scans are present in D2. Cluster 2 (67% scanning) contains 60 IP addresses from GA Tech. Cluster 7 (24%) contains 10 addresses from OpenINTEL. Cluster 9 (83%) contains 4 addresses from OpenINTEL plus one from a secondary prefix, and SIDN's DNSSEC validation tool (2 addresses). Cluster 12 (16%) contains 18 addresses from TUM. While similarly behaving resolvers from the same operation were never split between two clusters, 28 addresses from OpenINTEL and TUM are misclassified by the clustering. OpenINTEL's main scanning addresses are correctly classified.

While OpenINTEL scans the full zone file (100% coverage of 2LDs due to sharing of the zone file), no other scan achieves similar coverage, with a maximum of approximately 75% for campaign #2 even with a query count that is a multitude of the number of 2LDs ($\approx$ 6 million).

2) Domaining: We speculate that 2LD scanning from the networks of hosting providers most likely has commercial purposes (domaining). This would make domaining the most common motivation for scanning, with most of the largest scans falling into this category (9 of the top 15, Table III).

3) Subdomain Enumeration: 2LD scanning is most often carried out from one or more IP addresses at hosting providers, but we see subdomain scanning performed only indirectly, making up a majority of the traffic of at least 83 open resolvers in D1 (Table III).

Recursive resolvers should be able to serve the 2LDs involved from their cache instead of sending redundant queries to our servers. Still, we see these queries from Cloudflare, Google and other resolvers. From Cloudflare we learned that their resolver's cache had run full during D1.

4) Monitoring: Cluster 0 in D2 contains only monitoring resolvers (116 resolvers, 5M queries). In D1, we saw monitoring for 2LDs as well as monitoring of the zone itself, e.g. Start of Authority records, presumably to check for zone updates.

VII. DISCUSSION

DNS scans increase load on AuthNS and recursive resolvers, skew DNS-based popularity lists such as Tranco [4], and are quite likely also used for malicious purposes.

A. Impact on name server operations

Our name servers see no measurable decrease in performance from scanning and are capable of handling orders of magnitude larger traffic volumes than is caused by scans. Large AuthNS operators with similarly overprovisioned infrastructure will likely not suffer from scanning of this extent. The .nl zone recently saw a scan on the order of 2.6 billion queries (54% of the day's traffic), received at only 3 server sites. Even this caused no increase in processing times. Simple rate limiting has proven an effective means of limiting scans. Scanners should nevertheless follow best practices, as any scan causes large overhead not only to AuthNS.

Zone file sharing could decrease the 2LD scanning we see, but may not meet the freshness demands of some scanners.

B. Limitations

Due to the large number of resolvers, it is impossible for us to classify each one, and the clustering quality (silhouette score) indicates that the resolvers are not easily grouped into categories using a feature set like ours. Future work could profit from focusing on a narrower definition of DNS scanning. While we believe that manual classification works well, it is more time-consuming and possibly error-prone.

Scans may have escaped our attention through distributing traffic across a large number of IP addresses, e.g. a whole IPv6 prefix. As most traffic is sent by large resolvers (Table II), this is likely not common practice. In addition, a significant share of traffic could not conclusively be classified manually.

Lastly, our data covers only two non-consecutive days of traffic for .nl. The days we analyzed are both average in traffic volume and scans were similar between the two. However, applicability to other datasets is yet unclear.

VIII. SUMMARY AND FUTURE WORK

In this paper, we presented a first look at DNS scanning at a ccTLD, giving statistics on the volume, composition and notable examples of scans in the DNS zone of .nl. These insights help to understand the potential danger that scanning might pose to name servers, and their potential motivation. Our analysis was limited to 2 days of data from .nl.

Further research is necessary on the topic, particularly the usage of IP prefixes and sources of domain names used by scanners in practice.

ACKNOWLEDGMENTS

This work was funded by the Dutch Research Council (NWO) projects INTERSECT (NWA.1160.18.301) and CATRIN (NWA.1215.18.003). We thank SIDN Labs' other members for their input and support.

IX. Appendix

References

[1] P. Salvador and A. Nogueira, "Analysis of the internet domain names re-registration market," *Procedia Computer Science*, vol. 3, pp. 325–335, 2011, World Conference on Information Technology, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2010.12.056.

[2] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A high-performance, scalable infrastructure for large-scale active DNS measurements," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1877–1888, Jun. 2016, ISSN: 1558-0008. DOI: 10.1109/JSAC.2016.2558918.

[3] M. Antonakakis, R. Perdisci, Y. Nadji, *et al.*, "From Throw-Away traffic to bots: Detecting the rise of DGA-Based malware," in *21st USENIX Security Symposium (USENIX Security 12)*, Bellevue, WA: USENIX Association, Aug. 2012, pp. 491–506. [Online]. Available: https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/antonakakis.

[4] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*, ser. NDSS 2019, Internet Society, 2019. DOI: 10.14722/ndss.2019.23386.

[5] SIDN Labs. "Domain names." (2026), [Online]. Available: https://stats.sidnlabs.nl/en/registration.html (visited on 02/14/2026).

[6] P. Huppert, "Identifying DNS scanners from a TLD perspective," Master's Thesis, University of Münster, Oct. 2024. [Online]. Available: https://www.sidnlabs.nl/downloads/2aEkyKM1TuH9U94X5422WK/40f3483a6e953d421d0d237f446ffe78/Master_Thesis_Pascal_Huppert.pdf.

[7] O. Hohlfeld, "Operating a DNS-based active internet observatory," in *Proceedings of the ACM SIGCOMM 2018 Conference Posters and Demos*, ser. SIGCOMM '18, Budapest, Hungary: Association for Computing Machinery, 2018, pp. 60–62, ISBN: 9781450359153. DOI: 10.1145/3234200.3234239.

[8] J. Mao, M. Rabinovich, and K. Schomp, "Assessing support for DNS-over-TCP in the wild," in *Passive and Active Measurement*, O. Hohlfeld, G. Moura, and C. Pelsser, Eds., Cham: Springer International Publishing, 2022, pp. 487–517, ISBN: 978-3-030-98785-5.

[9] S. García, K. Hynek, D. Vekshin, T. Čejka, and A. Wasicek, *Large scale measurement on the adoption of encrypted DNS*, 2021. arXiv: 2107.04436 [cs.CR].

[10] A. J. Kaizer and M. Gupta, "Open resolvers: Understanding the origins of anomalous open DNS resolvers," in *Passive and Active Measurement*, J. Mirkovic and Y. Liu, Eds., Cham: Springer International Publishing, 2015, pp. 3–14, ISBN: 978-3-319-15509-8.

[11] J. Magnusson, M. Müller, A. Brunstrom, and T. Pulls, "A second look at DNS QNAME minimization," in *Passive and Active Measurement*, A. Brunstrom, M. Flores, and M. Fiore, Eds., Cham: Springer Nature Switzerland, 2023, pp. 496–521, ISBN: 978-3-031-28486-1.

[12] S. D. Strowes, *Bootstrapping active IPv6 measurement with IPv4 and public DNS*, 2017. arXiv: 1710.08536 [cs.NI].

[13] L. Izhikevich, G. Akiwate, B. Berger, *et al.*, "ZDNS: A fast DNS toolkit for internet measurement," in *Proceedings of the 22nd ACM Internet Measurement Conference*, ser. IMC '22, Nice, France: Association for Computing Machinery, 2022, pp. 33–43, ISBN: 978-1-45-039259-4. DOI: 10.1145/3517745.3561434.

[14] B. Blechschmidt and Q. Scheitle. "MassDNS - a high-performance DNS stub resolver GitHub." (2024), [Online]. Available: https://github.com/blechschmidt/massdns (visited on 02/14/2026).

[15] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going wild: Large-scale classification of open DNS resolvers," in *Proceedings of the 2015 Internet Measurement Conference*, ser. IMC '15, Tokyo, Japan: Association for Computing Machinery, 2015, pp. 355–368, ISBN: 978-1-450-33848-6. DOI: 10.1145/2815675.2815683.

[16] L. Lehle, P. Sattler, and J. Zirngibl, "Structure and origin of CT based domain lists," 2023. DOI: 10.2313/NET-2023-11-1_15.

[17] R. Sommese, R. van Rijswijk-Deij, and M. Jonker, "This is a local domain: On amassing country-code top-level domains from public data," Sep. 2023. arXiv: 2309.01441.

[18] The ZMap Project. "ZDNS (GitHub)." (2025), [Online]. Available: https://github.com/zmap/zdns (visited on 02/14/2026).

[19] G. C. M. Moura, M. Müller, M. Davids, M. Wullink, and C. Hesselman, "Fragmentation, truncation, and timeouts: Are large DNS messages falling to bits?" In *Passive and Active Measurement*, O. Hohlfeld, A. Lutu, and D. Levin, Eds., Cham: Springer International Publishing, 2021, pp. 460–477, ISBN: 978-3-030-72582-2.

[20] J. Qiao. "Detecting resolvers at .nz." Archived by the Internet Archive. (Nov. 12, 2018), [Online]. Available: https://web.archive.org/web/20220122132201/https://blog.nzrs.net.nz/detecting-resolvers-at-nz/.

[21] M. Açikalin, "Profiling recursive resolvers at authoritative name servers," Thesis, University of Twente, Enschede, Aug. 2019. [Online]. Available: https://purl.utwente.nl/essays/79267.

[22] T. Wabeke, T. van den Hout, and M. Müller, *DNS2Vec: Applying representation learning to DNS data*, Blog Post, Jan. 25, 2024. [Online]. Available: https://www.sidnlabs.nl/en/news-and-blogs/dns2vec-applying-representation-learning-to-dns-data (visited on 02/14/2026).

[23] W. B. de Vries, Q. Scheitle, M. Müller, W. Toorop, R. Dolmans, and R. van Rijswijk-Deij, "A first look

TABLE IV
LIST OF FEATURES USED IN CLUSTERING, AND THEIR WEIGHTS

Weight	Features
16	Distinct Name Fraction
8	Response Code 0 Fraction, No Response Fraction, Repeat Fraction, Domain Name Length Average
4	Query Type x Fraction, Fraction Over Short Time Frame Minimum, Fraction Over Short Time Frame Difference, Name Starts With x Fraction, Per Name Query Count Average, Per Name Query Count Deviation, Recursion Desired Fraction, Response Code 3 Fraction, Domain Name Length Deviation, Fraction Over Short Time Frame Deviation
2	ID Average, TC Fraction, CD Fraction, Source Port Minimum, Punycode Fraction, Time Frame, TCP Fraction, Repeat Time Average, Fraction Over Short Time Frame Maximum, Per Name Query Fraction Average, Count Over Short Time Frame Quotient
1	Repeat Time Deviation, Destination x Fraction, Time Deviation, Time Min, Time Max, Bogus Fraction, ID Minimum, AA Fraction, Source Port Average, Source Port Deviation, Response Code *Other* Fraction, Operation Code x Fraction, Inter Query Time Deviation, Time Average, EDNS UDP Variability

at QNAME minimization in the domain name system," in *Passive and Active Measurement*, D. Choffnes and M. Barcellos, Eds., Cham: Springer International Publishing, 2019, pp. 147–160, ISBN: 978-3-030-15986-3.

[24] M. Wullink, M. Müller, M. Davids, G. C. M. Moura, and C. Hesselman, "ENTRADA: Enabling DNS big data applications," in *APWG Symposium on Electronic Crime Research (eCRIME 2016), Toronto, ON, Canada. June 1, 2, and 3, 2016*, Jun. 2016.

[25] SIDN Labs. "ENTRADA - DNS big data analytics." Archived by the Internet Archive. (2019), [Online]. Available: https://web.archive.org/web/20260104165216/https://entrada.sidnlabs.nl/.

X. ETHICS

IP addresses and domain names can make queries personally identifiable. Furthermore, it is not possible to ask for user consent, as the DNS servers are part of hidden infrastructure. However, personal devices do not perform lookups directly, but through recursive resolvers which handle traffic from numerous users, providing one layer of protection.

We believe that the processing of this dataset is warranted by our goal of understanding AuthNS traffic to assess risks to and avert potential disruptions from our service and that of others. We abstracted the processing of data as much as possible. Potentially personally identifiable data was only stored on local, SIDN-managed servers and access only granted to contract employees of SIDN and only possible from organization-issued devices within the network.

XI. CALCULATION OF THE TOTAL SCAN TRAFFIC MEAN AND CONFIDENCE INTERVAL

From each of the $n = 17$ clusters, 10 random samples were manually labeled either with a percentage $p_{i,j}$ or the label "unclear". Between $m_i = 3$ and $m_i = 10$ resolvers per cluster could be conclusively labeled with a percentage and are therefore used for the estimates.

We define the random variables

$$X_i := \text{number of scanning queries of a resolver in cluster } i$$
$$Y_i := \text{number of total queries of a resolver in cluster } i$$

for each cluster $i = 0, \ldots, 16$ in D2. The samples we take are assumed independent and identically distributed, with X_i and Y_i likely not being independent. From the samples, we know the number of queries $Y_{i,j}$ and the estimated number of scan queries $X_{i,j} = Y_{i,j} \cdot p_{i,j}$ of each resolver. We calculate the estimate

$$\frac{\mathbb{E}(X_i)}{\mathbb{E}(Y_i)} \approx p_i = \frac{\sum_{j=0}^{m_i} X_{i,j}}{\sum_{j=0}^{m_i} Y_{i,j}}$$

Let $n_i :=$ number of queries of all resolvers in cluster i combined. We assume that p_i describes the percentage of scan traffic among all queries of the cluster, interpolating from the labeled samples to the $10 - m_i$ unclear samples and the unlabeled resolvers in the cluster. We calculate the fraction $p = \frac{\mathbb{E}(X)}{\mathbb{E}(Y)}$ of scan traffic among all clusters combined by using the means from our samples and n_i, the number of queries of cluster:

$$p = \frac{\left(\sum_{i=0}^{16} \frac{\mathbb{E}(X_i)}{\mathbb{E}(Y_i)} \cdot n_i\right)}{\sum_{i=0}^{16} n_i} = \frac{\sum_{i=0}^{16} p_i \cdot n_i}{\sum_{i=0}^{16} n_i} \approx 33\%$$

This gives our estimate of the total fraction of scan traffic in the clustered resolvers in D2.

A. Confidence Interval

For calculating the confidence interval, we model the distribution of $\frac{\mathbb{E}(X_i)}{\mathbb{E}(Y_i)}$ for each cluster and then add these.

By using Fieller's theorem, we can calculate the confidence interval of $\frac{\mathbb{E}(X_i)}{\mathbb{E}(Y_i)}$ per cluster, obtain its standard deviation σ_i, and add the distributions. Cluster 1 has a non-symmetric confidence interval, therefore we fall back to an interval of $[0\%, 100\%]$, adding 0 queries to the lower and n_1 queries to the upper bound. We assume the true values of p_i to be normally distributed and independent because all samples were chosen randomly and independently. By multiplying the variances σ^2 by n_i^2, the known traffic (query count) of each cluster, and summing, we obtain the total variance $\sigma^2 = \sum_{i=0, i\neq 1}^{17} (\sigma_i \cdot n_i)^2$. By calculating the lower and upper bound for 95% confidence and adding the queries of cluster 1, we obtain a confidence interval of $21\% \leq p \leq 54\%$.