Unresolved Issues: Prevalence, Persistence, and Perils of Lame Delegations

Gautam Akiwate
UC San Diego
gakiwate@cs.ucsd.edu

Mattijs Jonker
University of Twente
m.jonker@utwente.nl

Raffaele Sommese
University of Twente
r.sommese@utwente.nl

Ian Foster
DNS Coffee
ian@dns.coffee

Geoffrey M. Voelker
UC San Diego
voelker@cs.ucsd.edu

Stefan Savage
UC San Diego
savage@cs.ucsd.edu

KC Claffy
CAIDA/UC San Diego
kc@caida.org

ABSTRACT

The modern Internet relies on the Domain Name System (DNS) to convert between human-readable domain names and IP addresses. However, the correct and efficient implementation of this function is jeopardized when the configuration data binding domains, nameservers and glue records is faulty. In particular *lame delegations*, which occur when a nameserver responsible for a domain is unable to provide authoritative information about it, introduce both performance and security risks. We perform a broad-based measurement study of lame delegations, using both longitudinal zone data and active querying. We show that lame delegations of various kinds are common (affecting roughly 14% of domains we queried), that they can significantly degrade lookup latency (when they do not lead to outright failure), and that they expose hundreds of thousands of domains to adversarial takeover. We also explore circumstances that give rise to this surprising prevalence of lame delegations, including unforeseen interactions between the operational procedures of registrars and registries.

CCS CONCEPTS

• **Networks** → **Naming and addressing**; *Public Internet*.

ACM Reference Format:
Gautam Akiwate, Mattijs Jonker, Raffaele Sommese, Ian Foster, Geoffrey M. Voelker, Stefan Savage, and KC Claffy. 2020. Unresolved Issues: Prevalence, Persistence, and Perils of Lame Delegations. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3419394.3423623

1 INTRODUCTION

The Domain Name System (DNS) plays a critical role in the functioning of the Internet by resolving human-readable domain names into routable IP addresses (among other tasks). Because this function is distributed, its operation implicitly depends on the nature of the delegations configured across the DNS namespace. In particular, the ability of a domain to be efficiently resolved is predicated on all of its nameservers being resolvable and that those nameservers, in turn, are able to provide authoritative answers. In the common case, all of these requirements are satisfied, but there are a significant minority where they are not.

When a nameserver is delegated authority over a domain, but is unable to provide authoritative answers about that domain, a *lame delegation* is created. In the best case, lame delegations can result in increased resolution latency, as queries must timeout and be redirected to other hopefully correctly configured nameservers. However, in other situations, lame delegations can provide sufficient purchase for attackers to monitor or hijack DNS resolution.

In this paper, we explore the prevalence and causes of such lame delegations in the DNS name hierarchy. We explore this issue both longitudinally, using nine years of zone snapshot data comprising over 499 million domains in both legacy and new generic TLD (gTLD) namespaces (respectively, e.g., `.com` and `.xyz`) as well as in the current DNS namespace using active measurements covering over 49 million domains. We find that lame delegations are relatively common, roughly 14% of registered domains actively queried have at least one lame delegation and the clear majority of those have no working authoritative nameservers. We identify reasons why lame delegations persist, including: cross-zone delegations, which current protocols are unable to validate; and non-working IP addresses in glue records, which similarly cannot be validated statically using registry zone data. Moreover, we identify an unforeseen interaction between existing registrar practice and the constraints of registry provisioning systems that has inadvertently created hundreds of thousands of lame delegations.

Our measurements show that lame delegations can have significant impacts even when there are alternative working authoritative nameservers for a domain. Lame delegations can result in a significant increase in average resolution latency (3.7×), unnecessary load on existing nameservers (roughly 12% of requests to GoDaddy's nameservers are for domains for which they are not authoritative [24]) and, most importantly, the potential for malicious parties to monitor or hijack DNS lookups. We have identified many tens

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10.
https://doi.org/10.1145/3419394.3423623

of thousands of domains vulnerable to such hijacking and, in several instances, we have identified single domains that, if registered by an attacker, would have allowed the hijacking of thousands of domain names. Finally, we describe our efforts working with the registrar and registry communities to understand the source of these problems and establish efforts to address them going forward.

2 BACKGROUND

The Domain Name System (DNS) provides a distributed lookup service mapping a hierarchical namespace to a variety of associated resource records (RRs). In its most familiar usage, a DNS client (such as a web browser) will request the address records (either `A` for IPv4 or `AAAA` for IPv6) corresponding to the fully-qualified domain name (FQDN) found in a URL. However, the basic request-response protocol used by clients to make requests belies the considerable complexity in how resolution works, how namespaces are configured and delegated, and in how protocols and operational practices provision this state. This section sketches the basics of this process to provide the context necessary to describe the range of problems we identified in our measurement study.

2.1 DNS Protocol

DNS is fundamentally a lookup service. Clients make requests, following the protocol first specified in RFC 1035 [20], to resolve individual RR's (such as `A` records) for a given fully-qualified domain name. Thus, a client seeking to reach `www.cs.cmu.edu` might request its `A` record and obtain the IP address `128.2.42.95` in return. In typical use, a client's request is directed to a configured recursive resolver, either a local DNS server usually provisioned to the operating system via DHCP, or a public resolver such as Google's `8.8.8.8`. Recursive resolvers, if they do not have an appropriate and fresh answer in their cache, take responsibility for performing the series of distributed requests needed to complete the resolution, or to identify that the resolution cannot be satisfied (e.g., resulting in an `NXDOMAIN` response).

Recursive resolution. Recursive resolvers use the same protocol as clients, but parse the domain from left to right, dropping a domain name's prefixes until they encounter a portion of the name space for which they know of an authoritative server to query.

Absent any previously cached information, all recursive resolvers at least include the hard-coded IP addresses of the global DNS root servers. These servers will not be able to provide authoritative information about the FQDN being queried, but will return authoritative information about the nameserver (`NS`) records for the associated top-level domain (TLD).[1] We say that these `NS` records represent a *delegation* of the namespace. For example, nameservers for `.edu` are delegated responsibility for the namespace below `.edu`. Then, using an appropriate TLD nameserver, the recursive resolver can issue its query again, each time obtaining answers about nameservers responsible for a more narrowly delegated portion of the namespace until a nameserver is reached that can provide an authoritative `A` record, identifying the IP address for the original query received from the client.

[1]These records include legacy gTLDs such as `.com` and `.edu`, country-code TLDs (ccTLDs) such as `.uk` and `.ru`, and 1000+ new generic TLDs such as `.xyz`.

As a concrete example, a query for `www.cs.cmu.edu` to a newly started recursive resolver might produce a request to a root server who, in turn, would reply with `NS` records for the `.edu` nameservers (i.e., `[a-m].edu-servers.net`). Sending the same request to these servers would produce a reply pointing to the `cmu.edu` nameservers (i.e., `nsauth1.net.cmu.edu`, among others) who, upon being queried themselves, would point to the `cs.cmu.edu` nameservers (i.e., `nsauth-ib1.net.cmu.edu`, among others).[2] Finally, the authoritative nameservers for `cs.cmu.edu` would provide the resulting `A` record for `www.cs.cmu.edu`.[3]

Glue records. It is important to note that `NS` records are names themselves (e.g., `nsauth1.net.cmu.edu`) and a recursive resolver must obtain `A` records for those names to properly contact them. This resolution can be problematic, however. For instance, if the domain `example.com` is delegated to `ns1.example.com` (a common idiom), there is no way to query `ns1.example.com` to obtain its IP address. For this reason, the DNS protocol allows nameservers to provide additional records, called *glue records*, which are `A` or `AAAA` records for the identified nameservers (`ns1.example.com` in this example). To improve latency, nameservers may also provide *sibling glue records*, which are glue records for sibling domains in the zone file. Thus, it is common for nameservers to provide corresponding `A` or corresponding `AAAA` records (i.e., IP addresses) for any `NS` records they return authoritative answers for. Critically, a requester will only accept additional records that are *in-bailiwick*, i.e., portions of the namespace for which the server provides authoritative answers. `NS` records that are *out of bailiwick* for a domain will typically not include glue, since resolvers will not accept them. For example, delegating `example.com` to nameserver `ns1.example.org` would be glue-less; the `.com` TLD nameservers would not provide glue for `ns1.example.org`.

2.2 Zone Provisioning and Management

Equally important is the procedure by which domains and nameserver records are provisioned and managed. Each TLD is operated by a single registry organization (e.g., Verisign is the registry for `.com`, PIR for `.org`, etc.) who is responsible for the TLD namespace and for ensuring the availability and consistency of its authoritative nameservers. Registries typically contract with registrars (e.g., GoDaddy or Network Solutions) to register domains under the registry's namespace on behalf of the registrar's customers.

The technical mechanism for interfacing between registrars and registries is the Extensible Provisioning Protocol (EPP) principally documented in RFC 5731 and RFC 5732 [10, 11]. Registries use EPP to provide a degree of administrative access to the registry database and to allow registrars the ability to install newly registered domains into the database and manage the records for those domains. EPP provides a degree of isolation between registrars and ensures

[2]Note that there is no requirement that each "." in the domain name represent a delegated portion of the namespace. Indeed, while it so happens that `cs.cmu.edu` operates in a separately delegated "zone" from `cmu.edu`, that delegation is an administrative choice. In an alternate implementation, `nsauth1.net.cmu.edu` could have provided an authoritative `A` record for `www.cs.cmu.edu` directly.

[3]Note that this complete set of queries is rarely performed in practice because answers, at each level of the namespace, are cached for the period of time designated in the time-to-live (TTL) field in each nameserver answer.

Records	Type
foo.com NS ns1.example.com	Well Configured
foo.com NS ns2.exmple.com	Misconfigured
ns1.bar.com A 132.239.1.1	Well Configured
ns2.bar.com A 13.239.1.1	Misconfigured

Table 1: Example lame delegation due to typos.

the consistency of the overall database. EPP's consistency constraints can have unintuitive consequences. For example, one registrar can create a host object entry in EPP to delegate a nameserver (`ns1.example.com`) for a domain that they have registered. If a different registrar registers a domain that uses `ns1.example.com` as *its* nameserver, then the first registrar will no longer be able to delete the host object `ns1.example.com` nor its associated domain object `example.com`, so long as the other domain registered by the second registrar continues to use `ns1.example.com`. In addition to the baseline constraints of EPP, registries and registrars can impose their own restrictions on names registered through them.

Finally, many names in the DNS rely on multiple registries. For instance, `example.com` might have two nameservers spanning two TLDs: `ns1.example.com` and `ns1.example.org`. While the registry for `.com` (Verisign) is in a position to validate and enforce policies about `ns1.example.com`, it is unable to do the same for the `NS` records (`ns1.example.org` in this case) outside its authority.

2.3 Lame delegations

Absent issues like network or server outages, every fully-qualified domain name should be resolvable by any nameserver delegated to provide authoritative answers for that portion of the namespace. However, as this paper documents, there are a significant number of cases where this is not so. In particular, a range of configuration errors produce *lame delegations* — a situation where an `NS` record for a given domain does not lead to authoritative answers for that domain. Lame delegations result in wasted DNS queries, sometimes to hosts that do not even exist [8, 25].

In some cases all of a registered domain's delegations are lame. It is also possible for a domain to be *partly lame*, i.e., at least one nameserver is deficient, but not all of them. The former case is likely to be fixed quickly because the namespace is unusable. Partly lame domains are more insidious because name resolution continues to operate, but with increased latency and potential security risks. The increased latency arises because if a recursive resolver uses the lame nameserver first, it will need to timeout before it will try a correctly configured nameserver.

The potential for security risk is more nuanced. Consider the case in which the misconfiguration is a result of a typo such as shown in Table 1. Whoever controls `exmple.com` can control the resolution for the fraction of requests for `foo.com` that are resolved through the `ns2.exmple.com` nameserver. Similarly, whoever has control of the mistyped IP address can control the resolution of the domain names that use `ns2.bar.com`. Lame delegations create an attack surface for would-be hijackers of the delegating domains.

3 RELATED WORK

The complexity of DNS configuration, and associated prevalence of misconfigurations, was recognized decades ago [8, 25]. In 2004, Pappas et al. used active measurements to study ∼52 k domain names and found that on average about 15% of registered domains under several TLDs (i.e., `.com`, `.net`, `.org`, `.edu` and various ccTLDs) had lame delegations [21].

A TLD may contain glue records for a nameserver, even when the registered domain name of the nameserver has expired. Such a nameserver is considered *orphaned*. Kalafut et al. [16] passively analyzed six TLDs over a 31-day period in April 2009, and identified 16 k orphan nameservers per day on average. The TLDs under consideration accounted for about 60% of all domains on the Internet at the time. Kalafut et al. also found that certain TLDs accounted for a disproportionate number of orphan records, and that some orphans were evidently used for malicious purposes. In 2019, Sommese et al. [26] revisited this behavior. They found that some TLDs had fewer orphan nameservers than 10 years earlier, but other TLD operators had more orphan records than before, and they were prevalent among new gTLDs. Notably `.com` and `.net` no longer had any, implying those TLD operators are now automatically preventing them.

Liu et al. investigated the presence of *pointers* to invalid resources in the DNS, a type of *dangling DNS record* [18]. They used active measurement to highlight dangling records created by use of ephemeral IP addresses on cloud services and via expiring domains.

Lame delegations can also occur with reverse delegations. Some Regional Internet Registries (RIRs) automatically detect lame reverse delegations, such as APNIC [6] and LACNIC [17]. ARIN previously had a similar policy, but retired it in 2014 [7]. In 2016, Phokeer et al. showed that reverse delegations are frequently lame in AFRINIC's `41.in-addr.arpa` zone [22]. At the time AFRINIC did not have automated detection, but later instituted it [2] and substantially reduced the prevalence of lame reverse delegations [3]. Our study focuses on forward delegations, which determine control and availability of mappings.

In 2020, Sommese et al. [27] found that ∼8% of registered domains under the largest gTLDs (i.e., `.com`, `.net` and `.org`) have inconsistent parent (delegation) and child zones. They investigated the risk that such inconsistencies pose to the availability of misconfigured domain names.

These previous studies used only active measurements to study delegation-related security risks in the DNS namespace. Ours is the first to use comprehensive collections of both active and passive DNS measurements to explore and quantify these risks, allowing us to not only identify long-term trends in lame delegations, but also analyze root causes of their surprising prevalence in some cases.

4 DATA SETS

We use two data sets for analysis: a passive collection of TLD zone files, and a data set of active DNS resolutions.

4.1 DNS Coffee: TLD Zone Data

Our primary data set is a large collection of zone files from the `dns.coffee`[4] service [9]. This data set contains daily snapshots of

[4] CAIDA now offers the same collection through `CAIDA-DZDB` at https://dzdb.caida.org

Year	TLD Zone Files	Year	TLD Zone Files
2011	12	2016	1221
2012	12	2017	1237
2013	49	2018	1241
2014	462	2019	1235
2015	828	2020	1206

Table 2: TLD zone files per year in DNS Coffee data set.

Domains	Nameservers (NS)	IPv4 (A)	IPv6 (AAAA)
499.3 M	19.9 M	5.1 M	91.9 k

Table 3: Records in the DNS Coffee data set.

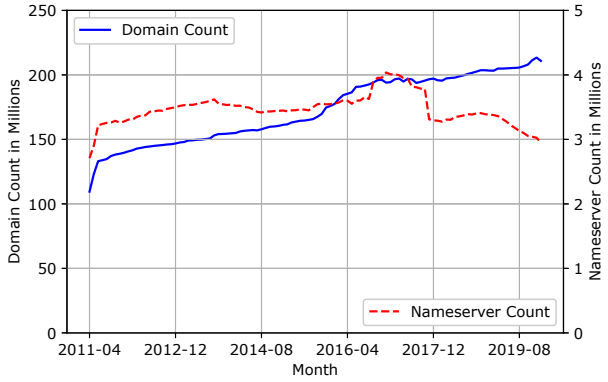

Figure 1: Number of distinct domains and nameservers in DNS Coffee zones over the years.

zone files from April 2011 through January 2020, covering nearly nine years. Over time, as zone files for new TLDs became available, `dns.coffee` added them to its collection. Table 2 shows the number of unique TLDs collected over time, and Figure 1 shows the number of distinct domains and nameservers across the zone files every year. As of September 2020, the service collects zone files for over 1250 different zones on an ongoing basis. The snapshots include the zone files of legacy generic TLDs (gTLDs), the `.us`, `.se` and `.nu` country-code TLDs (ccTLDs), and new generic TLDs (ngTLDs) made available through the ICANN Centralized Zone Data Service [14].

One issue when analyzing zone files is that records can refer to TLDs outside of that zone. As we aggregate records from the zone files together, cross references across zones are automatically consolidated in the data set. However, for records that refer to TLDs for which we do not collect zone files, we have to make assumptions, e.g., that the resolution is valid.

4.2 Active DNS measurement

Certain characteristics of real-world DNS behavior cannot be learned from zone files. Zone files may list `NS` records for nameservers that do not have authoritative data, are not reachable, or do not even exist. Active measurement data can reveal these additional insights into lame delegations, although capturing comprehensive data would require exhaustively querying all nameservers listed in the zone files for a given domain, and all IP addresses for each nameserver. Open data sets like the OpenINTEL [28] project do not exhaustively query all nameservers in the zone file; instead they perform resolutions as a typical nameserver would, and stop when they receive an authoritative response for a domain. This approach will not capture comprehensive data on availability or authoritativeness of nameservers listed in the zone file, a particular problem for partly lame delegations. Thus, to gain a more comprehensive picture, we perform our own active DNS measurements. We describe our methodology for doing so later, in Section 6.1. Given the intrusive nature and overhead of exhaustive probing, we limit the number of domains we actively probe. Further, we supplement these measurements with OpenINTEL data to ascertain the potential "real-world" impact of lame delegations.

5 LAME DELEGATIONS INFERRED FROM ZONE FILES

Our first analysis uses the nine years of zone file data to identify unresolvable nameservers that cause lame delegations. We delineate three periods of a nameserver's lifetime during which lame delegations occur, each period associated with different causes and implications. In this context, we characterize the prevalence of unresolvable nameservers and affected domains overall, how long domains are lame delegated, and how an unusual concentration in the `.biz` TLD reveals an undocumented registrar operational practice. We then examine unresolvable nameservers and lame delegations longitudinally over the nine years, identifying trends, prominent events that indicate causes of large-scale lame delegations, and their associated risks.

5.1 Methodology for static analysis

Our analysis of longitudinal zone file data performs "static resolution" of domains and nameservers to identify unresolvable nameservers that lead to lame delegations. Specifically, we infer lame delegations by following chains of records in zone files to establish that a nameserver has a valid resolution path. We use the zone file snapshots over time to derive the date ranges for when each nameserver has a valid resolution path. We then identify the registered domains that depend upon the nameservers during their valid time periods. Any (*domain*, *nameserver*) pair where the domain relies on a nameserver outside of that nameserver's periods of valid resolution is a lame delegation. We refer to registered domains in the zone files simply as domains, and specifically mention in context if a domain is a fully qualified domain name.

To explain this static resolution process, we use mock `NS` and `A` records (Table 4) to show how we use four criteria to derive the "valid resolution" date ranges for the nameservers (Table 5). Each record has a start and an end date. For each TLD we record

TLD	Records	Start Date	End Date
.com	foo.com NS ns1.bar.in	2011-04-11	2013-10-31
.com	foo.com NS ns1.baz.org	2011-06-18	2013-10-31
.com	foo.com NS ns1.qux.org	2011-04-11	2013-10-31
.com	foo.com NS ns.1qux.org	2011-06-11	2013-10-31
.com	foo.com NS ns1.thud.org	2011-06-11	2013-10-31
.org	thud.org NS ns1.baz.org	2011-06-06	2013-10-31
.org	ns1.baz.org A 93.14.2.34	2011-06-06	2013-10-31
.org	ns1.qux.org A 93.14.2.36	2011-06-06	2013-09-01

Table 4: Mock `NS` and `A` records to illustrate static resolution.

Nameserver	Start Date	End Date	Reason
ns1.bar.in	2011-04-11	2013-10-31	Other TLD
ns1.baz.org	2011-06-06	2013-10-31	Glue Record
ns1.qux.org	2011-04-11	2011-06-05	Late Access
ns1.qux.org	2011-06-06	2013-09-01	Glue Record
ns1.thud.org	2011-06-06	2013-10-31	Parent Resolution
ns.1qux.org	-	-	No Records

Table 5: Resolvability at the end of static resolution.

the first time we imported the zone file for that TLD. The earliest information we have for `foo.com`, and generally any domain in `.com`, is `2011-04-11`. We derive resolution validity if any of these four criteria hold:

(1) **Other TLDs:** Domains in our set of zone files can have `NS` records with nameservers in TLDs for which we do not have a zone file. In Table 4, `foo.com` has a nameserver `ns1.bar.in`. Since we do not have the zone file for the `.in` TLD we conservatively assume that `ns1.bar.in` can be resolved from `2011-04-11` to `2013-10-31` (Table 5). Of the ∼20 M nameservers in our zone file data set, 1.4 M (7%) of them belong to such TLDs and we assume that they are resolvable.

(2) **Late Access TLDs:** We do not always have the earliest zone file for a given TLD, e.g., our earliest copy of the `.org` TLD zone file is from `2011-06-06`. If we see earlier references to nameservers in the `.org` TLD in other zone files, we conservatively mark them as resolvable for the duration before we have visibility into the TLD.

(3) **Glue Records:** If a nameserver has a glue record in the zone files, then we assume that the nameserver is resolvable for the duration of the glue record. In Table 4, `ns1.baz.org` and `ns1.qux.org` have glue records that make them resolvable for the durations shown in Table 5.

(4) **Parent Resolution:** Domains using a nameserver that does not have a glue record can still resolve via the resolution on the nameserver's parent domain. In Table 4 consider `ns1.thud.org`. While `ns1.thud.org` does not have a glue record, the nameserver parent `thud.org` can be resolved by `ns1.baz.org` since it has a valid resolution path via its glue records. Thus, in Table 5 we consider `ns1.thud.org` resolvable from `2011-06-06` to `2013-10-31` as a result of parent resolution. Determining parent resolution may involve multiple layers of redirection before reaching a nameserver with a valid resolution path. Otherwise, a nameserver without a glue record is unresolvable.

We illustrate this static analysis process by working through the mock examples in Tables 4 and 5. Table 5 presents the durations for which a nameserver is conservatively resolvable. Nameservers `ns1.bar.in`, `ns1.baz.org`, and `ns1.thud.org` have a valid resolution path for the entire period during which they are the nameservers of `foo.com`. However, consider `ns1.qux.org` whose glue record is valid only until `2013-09-01`. Thus, we infer `ns1.qux.org` was unresolvable for the period `2013-09-02` to `2013-10-31`. Additionally `ns.1qux.org`, an example of a typo of the actual nameserver, never has any records associated with it. This typo results in a security risk since someone can register `1qux.org`, set the glue record for `ns.1qux.org` to a private nameserver, and control the resolution of `foo.com` for the fraction of requests that come its way.

Applying the static analysis across all nameservers for the full time period of our data set, we delineate a nameserver's "unresolvability", i.e., when it is unresolvable, across three periods:

(1) **Pre-Life**: The nameserver is referenced by a domain before the nameserver is first resolvable, typically due to delayed glue or delayed registration of the nameserver domain.
(2) **In-Life**: The nameserver is temporarily unresolvable after previously being resolvable. The most common type of lame delegation, it is frequently the result of a nameserver domain expiring and then being renewed, or its glue records being misconfigured.
(3) **Post-Life**: The nameserver is no longer resolvable or was never resolvable. Typically, it is a result of an expired nameserver domain not being renewed, or a typo when entering the nameserver domain.

We found these categories useful for identifying causes and implications of lame delegations.

Our static resolution assumes that a nameserver with a glue record is routable, reachable, and operates an authoritative DNS server for the domain. Consequently, the static analysis results are lower bounds on unresolvable nameservers and lame delegations. Even so, static analysis uncovers a wide variety of DNS behavior that leads to lame delegations. Complementing this analysis, Section 6 describes our active measurements that derive a snapshot of lame delegations via operational execution of the DNS protocol.

5.2 Prevalence of lame delegations

We start by characterizing the overall prevalence of unresolvable nameservers across the zone files in our data set. Table 6 shows the total number of unresolvable nameservers and the total number of domains affected. The table also includes two breakdowns of the overall numbers: by time period (columns), and by TLD (rows).

Unresolvable nameservers may be a small percentage of nameservers (4%), but they result in more than 4.11 M lame delegated domains. Most unresolvable nameservers are unresolvable in-life, which is not surprising since they correspond to issues at any point during a nameserver's lifetime. The smallest category of unresolvable nameservers are those that are unresolvable pre-life; these cases are typically delayed registration of the nameserver domain. EPP constraints do not allow unregistered nameserver domains in the same TLD, so this situation arises only when the nameserver domain is in a different TLD from the domain itself.

	Unresolvable Nameservers by TLD				Across All TLDs
NS TLD	**Unresolvable NS**	**Pre Unr. NS**	**In Life Unr. NS**	**Post Unr. NS**	**Lame Domains**
.com	367,054 (4.25%)	17,660 (0.20%)	277,379 (3.22%)	85,899 (1.00%)	2,122,825
.net	85,039 (4.91%)	2,531 (0.14%)	61,372 (3.55%)	24,997 (1.45%)	899,082
.org	34,669 (3.51%)	828 (0.08%)	17,540 (1.78%)	17,438 (1.77%)	246,486
.info	39,184 (3.28%)	831 (0.07%)	29,092 (2.44%)	10,207 (0.86%)	67,796
ccTLDs	9,480 (1.41%)	333 (0.05%)	4,947 (0.74%)	4,920 (0.73%)	28,193
ngTLDs	28,472 (0.57%)	2,830 (0.06%)	12,351 (0.25%)	14,474 (0.29%)	446,906
.biz	191,211 (50.8%)	7,968 (2.12%)	8,454 (2.25%)	181,211 (48.1%)	551,201
All TLDs	755,109 (4.07%)	32,981 (0.18%)	411,117 (2.22%)	339,146 (1.83%)	4,114,750

Table 6: Summary of unresolvable nameservers in our zone file data set, broken down by nameserver TLDs. Includes unresolvable nameservers as percentage of all nameservers in same TLD. We categorize by TLD of nameserver and not the domain. Recall that nameservers and domains can be lame in more than one time period, so the sum of the time period columns is generally larger than the overall total.

Characterizing the prevalence of unresolvable nameservers by TLD, we found that unresolvable nameservers appear more often and with roughly similar prevalence in the old generic TLDs in the first four rows: between 3–4%. Country-code TLDs have comparatively fewer unresolvable nameservers, and the many new gTLDs grouped under ngTLDs have the fewest unresolvable nameservers.[5] While domain management practices could be better in the newer TLDs, a common practice in the new gTLDs is to have the `NS` records for domains point to nameservers in another TLD, often `.com`. Our method attributes any resulting unresolvable nameserver to `.com` and not the newer gTLD.

5.3 DROPTHISHOST anomaly

We placed `.biz` at the end of the Table 6 since it stands out in sharp contrast to other TLDs. The `.biz` TLD has had 381,475 nameservers across nine years of zone files. Of these, nearly half had no valid resolution path ever, yet domains still pointed to them. These results uncovered a long-standing undocumented practice among some registrars when dealing with expired nameserver domains.

Nearly 66% of these unresolvable `.biz` nameservers (118,905) have the substring "DROPTHISHOST" followed by a random unique string (indicating a generated GUID) in their FQDN. Very few of these nameserver domains have ever been registered, placing the domains served by the nameservers at risk of hijacking. Examining the history of such domains revealed a pattern: the change in their `NS` records to a unique "DROPTHISHOST" nameserver happens after the previous nameservers in the `NS` records expire.[6]

The naming, scale, and longevity of this pattern suggested systematic behavior. We reached out to the `.biz` registry and a large registrar to understand our findings. The registry was unaware of the extent of the issue because they had no visibility into it—these nameservers are not actually registered in `.biz`, and hence `.biz` does not have any records for them in its registry database. They just appear as names in `NS` records in the databases of other TLDs.

The registrar solved the mystery. For decades registrars have used an undocumented practice to clean up expired nameserver domains, a practice developed in response to a situation created by requirements of the EPP specification. A registrar cannot delete the record for a nameserver domain that expires if there are other records (e.g., domains) in the same TLD that refer to a host object for that domain (Section 2.2). However, by crafting a nameserver hostname in another TLD, and updating the host object record to use this "sacrificial" nameserver hostname instead — in effect updating the `NS` record of all domains referring to the original nameserver to use the new sacrificial nameserver host in a different superordinate domain — the registrar can then garbage collect the original expired nameserver object (RFC 5731 Section 3.2.2 [10]). Domains pointing to the sacrificial nameserver become lame delegated, but domain owners can always change the `NS` records to use a valid nameserver again if they choose. Anecdotally, it appears registrars chose `.biz` because it was a new gTLD at the time.

There are a few potential options to solve the problem going forward. The first option is to create sacrificial nameserver domains under a "sink" domain that the registrar controls. Some registrars already use this option. However, this option leaves the registrar responsible for answering queries for lame delegated domains, and for operating the "sink" domain. Another option would rely on the AS112 project `empty.as112.arpa` [1], which established a distributed anycast service that DNS operators could use to sink DNS traffic relating to parts of the global namespace under their control. Doing so would not require coordination among zones, and would ensure that such nameserver domains would never be registered by another party. To minimize query latency, responses could return `NXDOMAIN` with a long TTL. But this project relies on volunteers willing to donate resources to operate an AS112 anycast server. More concerning, a malicious actor could set up their own AS112 server and hijack queries intended for the AS112 server. More recently, however, ICANN's Security and Stability Advisory Committee (SSAC) has recommended that ICANN reserve a private-use TLD that might offer a useful path forward [15] to resolving this issue.

[5] We examined unresolvable nameservers among the individual gTLDs in the ngTLDs group and no particular gTLD stood out.

[6] As an example, see the current and past nameservers of a test domain at https://dns.coffee/domains/ORPHAN-FINDER.COM.

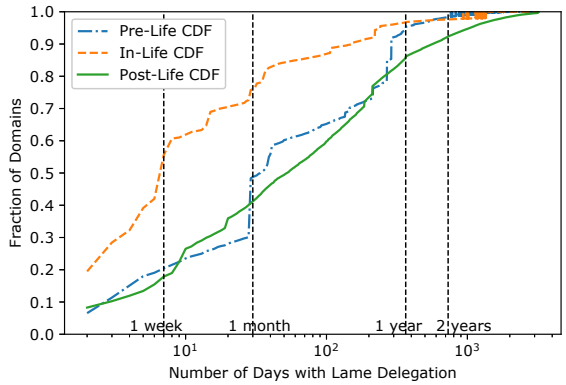

Figure 2: Fraction of domains with lame delegations for at most X days.

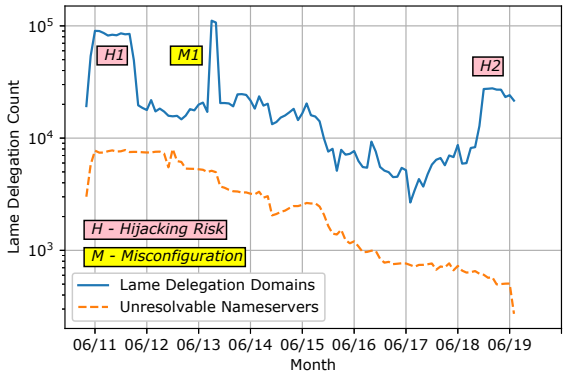

Figure 3: Pre-life lame delegations (blue) due to dependency on nameservers that are not yet unresolvable (red), because the nameserver domain or associated glue is not yet active.

The registrar we talked with was also surprised at the extent of the current situation; indeed, our findings motivated a change in their operational practice. However, cleaning up the existing DROPTHISHOST and similar sacrificial nameservers is more challenging. Given the restrictions in EPP that prevent external records from being modified (Section 3.2.5 of RFC 5732 [11]), purging these records will require coordination among registrars whose domains point to such nameservers and the registrars who created them. We plan to continue working with the registrar and registry communities to find a viable alternative approach to renaming expired nameservers as well as cleaning up the existing records.

5.4 Duration of lame delegations

How long do lame delegations persist? Figure 2 shows the fraction of domains with lame delegations to pre/in/post-life unresolvable nameservers for at most X days. Domains that are lame as a result of in-life lame nameservers are lame for the shortest time: nearly 50% of the affected domains are lame delegated for less than a week. These lame delegations suggest intermittent causes such as misconfigurations that are discovered relatively quickly. The mode at five days reflects an event in November 2011 where `cwgsh.com` and all of the nameservers under it became unresolvable after the domain `cwgsh.com` expired, causing nearly 60 thousand domains to have lame delegations.

Both pre-life and post-life unresolvable periods of nameservers have durations substantially longer than in-life unresolvable periods. For pre-life periods, the inflection at 29 days is due to a misconfiguration of `nic.tel`, and the last inflection corresponds to an issue with `cwgsh.org`, which had domains pointing to it for 289 days before it was registered (Section 5.5).

The distribution of post-life unresolvable periods has the longest tail, reflecting intentional use of lame delegations to park domains. Some domains are lame for up to 3,000 days, nearly the timeframe of our data set. Parking domains for long durations is a risk since the nameserver domain can mistakenly be allowed to expire, exposing them to hijacks (Section 5.5.1).

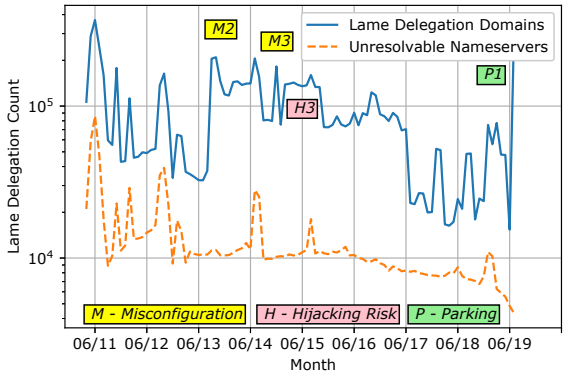

Figure 4: In-life lame delegations due to nameservers that *become* unresolvable (red), often due to temporary expiration of nameserver domain or misconfiguration of glue.

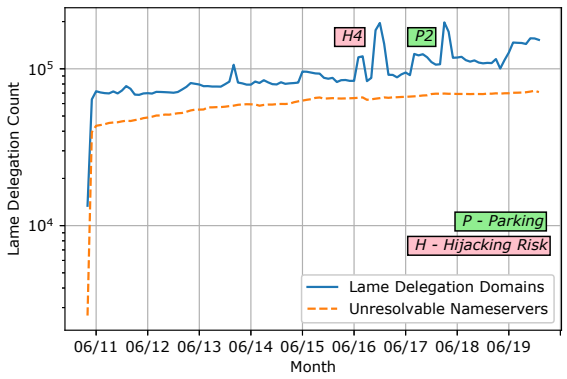

Figure 5: Post-life lame delegations (blue) due to nameservers that are no longer or were never resolvable (red), typically due to permanent expiration of a nameserver domain or typo of a nameserver.

5.5 Lame delegations over time

The duration of our zone data set allowed us to analyze long-term trends in lame delegations caused by unresolvable nameservers. We observed trends, discovered prominent events, and considered associated risks. For pre-life, in-life, and post-life, Figures 3—5 show the number of unresolvable nameservers causing lame delegations, and the number of domains affected by them, over time.

The pre-life timeseries (Figure 3) shows a downward trend in this kind of unresolved nameserver. Over the last few years, significantly fewer nameservers are named in NS records before those nameservers are resolvable. Yet the number of domains affected has increased substantially. The contrast indicates that the practice of adding nameservers in NS records before they are resolvable is on the rise, but concentrated on fewer nameservers. The sudden increase in concentration (H2) is a result of a single typo causing roughly 20,000 domains to be lame delegated.

The in-life timeseries (Figure 4) shows a generally stable baseline number of unresolvable in-life nameservers through 2014, and a slight decreasing trend since then. The most common cause of in-life periods is mismanagement, e.g., failure to renew, deleting required glue records.

The post-life timeseries (Figure 5) shows increasing trends in the number of post-life unresolvable nameservers and the number of domains affected by them. The steady increase could reflect the increasing use of unresolvable nameservers for parked domains, or for domains that have expired but have yet to be released.

These timeseries also show spikes in the number of unresolvable nameservers and and their associated lame delegated domains. These spikes correspond to significant events that caused many domains to become lame delegated. In the rest of this section, we study these events to highlight major causes of lame delegation and associated risks.

5.5.1 Hijacking Risk. Lame delegations can pose a risk to domain owners since attackers can take advantage of expired nameserver domains or typos to hijack domain resolutions. Consider the events labeled "Hijacking Risk" in Figures 3—5. In May 2011 (H1 in Figure 3) roughly 29,000 domains pointed to three unresolvable nameservers. These lame delegations were a result of three nameservers created by the Conficker Working Group (CWG) to use for sinkholed and preemptively registered domains used by Conficker [23]. However, these nameserver domains expired and someone else acquired them, thus controlling resolution of the domains using those nameservers [5]. Further, in May 2015 (H3 in Figure 4) the `cwgsh` nameserver registrations expired again.

In December 2016 (H4 in Figure 5) nearly 100,000 domains suddenly become lame when their nameserver's domain expired. Specifically, the domains using nameservers `ns[1,2].oigjaeiug.xyz` become unresolvable when the registered domain `oigjaeiug.xyz` expired. Surprisingly, domains continued to point to these unresolvable nameservers for five more months, until May 2017. Further, the domain `oigjaeiug.xyz` was available for registration at the end of this period, posing a hijacking risk: an attacker registering that domain name could immediately have become authoritative for domains that pointed to it in this period.

Finally, in December 2018 (H2 in Figure 3) the appearance of roughly 20,000 lame delegated domains was due to the use of the unregistered nameserver `ns5.dsndun.net`, which is a typo on the intended `ns5.dnsdun.com`. The domain `dsndun.net` was registered six months later, but the historical zone files reveal that `ns5.dsndun.net` did not resolve to the same addresses as `ns5.dnsdun.net`. In this case, whoever registered `dsndun.net` hijacked resolutions for nearly 20,000 domains for six months before the original domain owner removed the typoed nameserver from its list of authoritative nameservers.

Quantifying the Hijacking Risk: To make this risk concrete, we quantified the hijacking opportunity, i.e., the potential to gain some degree of DNS resolution control over currently lame delegated domains. Our zone file data showed that as of January 2020, there were 70,605 nameservers under 48,185 unique registered domains used by 151,422 lame delegated domains. Of these nameserver domains, 42,579 (88%) were available for purchase, placing nearly 75,000 domains at risk. For instance, by purchasing just 10 of these domains (each under $10 per domain), anyone could have potentially become the authoritative nameserver for around 4,000 domains.

While these domains may not have much intrinsic value, they could be a source of cheap domains. For the cost of registering a nameserver domain, an actor effectively gains use of all domains that name it in their NS record. Even though a purchaser does not own the delegated domains, they have control over how they are resolved and can even get SSL certificates signed for them.

This risk is not hypothetical. We see evidence of actors purchasing nameserver domains to take advantage of lame delegations. For instance, the owner of `phonesear.ch` has been registering nameserver domains that are authoritative for many lame delegated domains,[7] apparently for search engine optimization. Section 5.6 describes a set of lame delegations that left a county government in the U.S. at risk of hijacking for over a year.

5.5.2 Misconfiguration. A common cause of lame delegation is misconfiguration. We describe the three examples (M1-M3) annotated in Figures 3 and 4.

In September 2013, new nameservers were added to the `nic.tel` zone without glue records (M1), followed by existing nameserver glue records being dropped (M2). These configuration issues are consistent with reports of ongoing troubles the registry operator had with their delegations [12]. In May 2017 `.tel` transferred ownership [13], after which issues with the `nic.tel` nameservers disappeared.

The nameservers `conficker-sinkhole.{com,net}` were registered as a fix for letting the `cwgsh` domains expire, and efforts were made to move some domains over to these new nameservers from the `cwgsh` nameservers (which were no longer under the Conficker Working Group Control). Unfortunately, in December 2014 (M3), these domains expired and for five days were unresolvable while the registrar held them for the grace period. Fortunately, based on `whois` information, the domains were renewed in the grace period avoiding a repeat of the hijacking seen with the `cwgsh` domains (Section 5.5.1).

5.5.3 Parking. Registrars often try to monetize traffic to parked or expired domains. Typically, this monetization takes the form of

[7] https://dns.coffee/nameservers/A.NS.PHONESEAR.CH

many domains serviced by a single nameserver that directs visitors to advertisements. When such nameservers become unresolvable, the number of lame delegations jumps. We highlight two examples.

In July 2019 (P1 in Figure 4) roughly 285 k domains became lame, caused by the domain `domainparkingserver.net`, along with the glue records for its nameservers in the zone, disappearing for seven days from the zone files.

Similarly, the spike (P2) in March 2017 was due to a nameserver used for parked domains expiring. Since domains still pointed to the expired nameserver, the registrar could not delete the nameserver. The registrar followed industry practice and changed the NS record to `ns1.pendingrenewaldeletion.com.lamedelegation.org`, making the original nameserver domain available for registration again. In this case, the registrar used a domain it owns to act as a "sacrificial nameserver", and therefore created no hijacking risk.

5.6 Discussion

The lame delegation issues highlighted by our longitudinal passive analysis may involve only a small fraction of nameservers and domains in the DNS, and relatively unpopular ones at that. However, we argue that these issues are still important for a variety of reasons.

First, misconfigurations due to expired nameservers, nameserver records with typos, etc., represent a gap between expected and actual operation. When all nameservers for a domain are lame (fully lame), the domain is entirely unresolvable. When a subset of nameservers for a domain are lame (partly lame), the domain may still resolve but persistent unresolvable nameservers reduce the resiliency of DNS resolution for those domains. Section 6 discusses the operational impact of these issues.

Second, lame domains have sufficient value in practice to motivate some actors to capture their traffic by strategically registering dangling nameservers, as illustrated by the `phonesear.ch` example in Section 5.5.1.

Finally, even "unpopular" domains may identify critical infrastructure. As a concrete example, consider `whitecounty.net`, the official domain for White County, Georgia. This domain had the same two authoritative nameservers `ns2.internetemc.com` and `ns1.hemc.net` from our first import of the `.net` zone file until June 30, 2019 when the domain `internetemc.com` expired. To work around the EPP constraint of freeing a domain (`internetemc.com` in this case) when host objects associated with the domain have live references, the registrar renamed the host object associated with the domain `ns2.internetemc.com` to a sacrificial nameserver `ns2.internetemc1aj2tkdy.biz` in a different TLD.[8] This renaming followed a similar practice to that described in Section 5.3, just using a different pattern for the sacrificial nameserver.

As a result, starting on July 1, 2019, one of its nameservers was unresolvable and `whitecounty.net` was partly lame delegated. By registering the domain name `internetemc1aj2tkdy.biz`, an attacker could have received a fraction of the resolution requests for an official county government domain. Note that redundancy in DNS worked as intended since the other nameserver still worked and resolved everything correctly, albeit with a delay at times if the resolver chose to query the lame nameserver first. Ironically, though, because redundancy masked the long-term unresolvable nameserver, this issue went undiscovered by the domain owner. Given the sensitive nature of White County's domains, we reached out to the registry who notified the domain registrant. The domain configurations were fixed soon after.

[8] See the timeline illustrated at https://dns.coffee/domains/WHITECOUNTY.NET

6 LAME DELEGATIONS MEASURED WITH ACTIVE QUERIES

Static analysis revealed many aspects of lame delegations, particularly over time, but it is a lower bound. Active measurement shows that the prevalence of lame delegations is significantly higher in operational practice. We can detect lame delegations operationally by performing active domain resolutions, much as clients do when resolving domains. We characterize the prevalence of lame delegations across the major gTLDs, explore nameserver consistency issues, and quantify the impact of lame delegations on domain resolution time.

6.1 Methodology

We targeted NS queries at all nameservers listed in the zone file for a domain, from a single, well-provisioned vantage point connected to the Netherlands NREN. We supplemented our measurements with active resolution data provided by OpenINTEL for additional context about lame delegated domains within the recent past.

We started with a snapshot of the ngTLD zone files and `.com`, `.net` and `.org` to learn all the registered domain names under these zones. Next, we extracted the nameservers specified in their NS records. Finally, we extracted IP addresses in any existing glue records for nameservers.

We performed the following measurement steps:

(1) Actively resolve all NS names and record the IPv4 addresses[9] learned per name.[10]
(2) For each registered domain name, and for every NS name of each particular domain, we targeted up to *five* actively resolved IP addresses for the NS name in question with an explicit NS query for the registered domain name. We instantiate a local DNS resolver to contact the nameserver, so caching mechanisms will not affect our measurements.
    - We recorded the set of NS records returned by the NS query, including response flags set by the nameserver.
    - In case of an error (e.g., a connection timeout or a DNS-specific error), we record the error type.

Between March and May 2020 we queried over 49 million domains: 13 million randomly sampled domains from `.com`, 13 million randomly sampled domains from the combined set of all ngTLDs, and all domains from `.net` and `.org`. This selection balances coverage against the overhead of an exhaustive crawl of the entire DNS with the exponential fan-out from multiple nameservers per domain, and then multiple IP addresses per nameserver.

When resolvers cannot use a provided NS record (i.e., delegation) to obtain authoritative answers for a registered domain, we infer

[9] We contacted nameservers over IPv4 only. Our rationale is that a nameserver that is unresponsive over IPv4 and reachable only over IPv6 is still lame to resolvers (e.g., clients) with no IPv6 connectivity.

[10] Successful resolution requires any part of the delegation chain for the NS name to work. We do not exhaustively check every step of the chain as our perspective does not require it and doing so would exponentially increase measurement overhead.

	.com	ngTLDs	.net	.org	Total
Domains	13,000,000	13,000,000	13,174,611	10,015,702	49,190,313
Fully Lame	8.7%	9.6%	10.5%	9.2%	9.5%
Partly Lame	11.8%	19.8%	13.5%	11.7%	14.3%
Nameservers	620,561	278,657	724,518	552,665	1,325,856
IPs	299,319	143,095	347,413	273,906	534,214
Fully Lame	14.5%	17.1%	16.2%	16.4%	15.7%
Partly Lame	41.9%	44.0%	43.3%	44.1%	45.3%
~AA	0.3%	0.2%	0.5%	0.4%	0.5%

Table 7: Active DNS Resolution Lame Delegation Results: Breakdown by TLD.

#NS	#Lame Domains (%)	#NS	#Lame Domains (%)
1	11,926 (54.5%)	9	675 (40.7%)
2	5,732,799 (14.7%)	10	304 (12.6%)
3	499,652 (14.1%)	11	80 (61.1%)
4	551,592 (10.7%)	12	295 (3.4%)
5	132,428 (13.1%)	13	71 (28.9%)
6	97,472 (30.6%)	14	2 (100%)
7	17,817 (26.3%)	15	2 (100%)
8	9,176 (5.7%)	16	1 (100%)

Table 8: Partly lame domains by number of delegated `NS`.

NS TLD	Total NS	Fully Lame NS(%)
.com	176,897	57,137 (32.3%)
.net	97,160	30,896 (31.8%)
.org	38,825	14,792 (38.1%)
.info	2,690	731 (27.2%)
ccTLDs	65,041	16,585 (25.5%)
ngTLDs	40,792	19,213 (47.1%)
.biz	14,311	10,533 (73.6%)
Total	435,716	149,887 (34.4%)

Table 9: Fully lame nameservers relative to all nameservers in the same TLD.

the delegation is *lame* (Section 2.3). Our measurement reveals cases in which `NS` hosts do not exist, do not run a nameserver, or are not able to provide authoritative responses. It is not always possible to distinguish non-operational servers from network outages. Nameservers that we cannot reach after repeated attempts, we infer to be lame.

6.2 Domain Perspective

Table 7 summarizes the results of our active measurements, including the number of domains resolved, the total number of nameservers used by those domains, and the total number of IP addresses associated with the nameservers. The table classifies domains into two categories, fully and partly lame. A fully lame domain means that we did not obtain an authoritative answer from *any* nameserver or IP enumerations for that domain. A partly lame domain means that we did not obtain an authoritative answer from *at least one* nameserver and IP enumeration for that domain. Note that the partly lame metric also includes the fully lame cases.

At the time of our measurements roughly 10% of domains were fully lame (not resolvable) consistently across the TLDs.[11] This number increased to 14% of domains when considering partly lame domains (has at least one lame delegation, but not all). There are various reasons why actively resolving a domain can fail, from typos in names to placing recursive (non-authoritative) resolvers in NS records. For the 10% fully lame domains, the most prevalent issues that we encountered are nameservers that do not (or cannot) provide an authoritative answer, or nameservers that cannot be reached (i.e., query timeouts). Only a small percentage of cases resulted from typos in `NS` records.

Partly lame delegations were only 3–5% more common than fully lame. Since Table 7 counts the fully lame cases as also partly lame, it shows that more often than not, if a domain has any lame nameserver path, all of its paths do not resolve. The exceptions are the new gTLDs grouped under ngTLDs. Nearly 20% of domains we queried in ngTLDs had at least one nameserver path that did not resolve. This behavior could derive from ngTLDs domains being concentrated on many fewer nameservers than other TLDs. ngTLDs have roughly half the number of nameservers and corresponding IPs when compared to legacy gTLDs with similar number of domains.

Table 7 also breaks down the nameserver IP addresses into fully and partly lame. A fully lame IP address means that, when querying that IP to resolve a domain, that IP does not return an authoritative answer for *all* domains for which we queried it. A partly lame IP address means that the IP does not return an authoritative answer for *at least one* domain for which we queried it.

The fact that partly lame domains still resolve underscores the benefits of redundancy in the DNS. Table 8 classifies partly lame domains by the number of delegated nameservers. The first row corresponds to domains with just one nameserver, which by definition are misconfigured since RFC 1034 requires a domain have two nameservers at least [19]. With one lame nameserver, these domains are all unresolvable. As the number of nameservers increases, the percentage of partly lame domains naturally increases. The more delegated nameservers, the higher the probability that at

[11] Note that this percentage is similar to the results from Pappas et al. [21] in 2004. As the timeseries from Section 5 highlights, lame delegations have long been a persistent issue in the DNS.

NS IP	Country	#Lame Domains
52.20.26.87	US	144,327
60.12.122.226	CN	117,462
103.26.77.114	CN	117,462
218.98.111.162	CN	80,142
183.2.194.161	CN	80,142

NS Domain	#Lame Domains
0088dns.com	117,462
sinkhole.shadowserver.org	45,401
verification-hold.suspended-domain.com	41,804
sav.com	35,431
icmregistry.net	32,377
expirenotification.com	32,369

Table 10: Top fully lame delegated NS IPs and domains.

least one of them at any time is lame. These results highlight the fact that DNS redundancy may obscure configuration issues, since the domain is often still resolvable even when misconfigured.

6.3 Nameserver Perspective

We next look at the nameservers responsible for lame delegations. Table 9 shows the number and percentage of actively discovered fully lame nameservers across TLDs. Consistent with the results of our static analysis, `.biz` stands out with an unusually high percentage of fully lame nameservers (Section 5.3). There were 14,311 nameservers in `.biz` in our active measurement set, and 73.6% of them were fully lame. Domains using the unresolvable `.biz` nameservers predominantly come from the legacy TLDs `.com` and `.net`, again consistent with the long-standing practice of handling expired nameservers within those TLDs.

Looking at nameserver domains and their IPs more closely, lame delegation is concentrated and the most prevalent nameservers and IPs suggest that they, at least, are lame delegated by design and not due to misconfiguration. Table 10 reports the top fully lame delegated `NS` records and IPs. The top nameserver domain is associated with suspicious bulk domain registrations.[12] Manual inspection shows that the others are primarily sinkholes for security, abuse, and expired domains where delegated domains have been made lame intentionally. The top IP serves parked and for-sale domains, the next two IPs are used by `0088dns.com`, and the last two IPs are used in glue records for nameservers in `maff.com` with a large number of apparently abusive domains.

6.4 Consistency

The DNS ultimately depends upon multiple independent sources of information to operate correctly. However, as a hierarchical, delegation-based distributed system, the DNS does not contain inherent internal mechanisms to ensure consistency across these independent records. Inconsistencies arise for a variety of reasons, two of which we describe: inconsistencies in authoritative responses and glue records.

[12]For example, see current and past delegated domains at https://dns.coffee/nameservers/NS1.0088DNS.COM.

NS IPs	Country	#Domains	Wildcard
91.195.241.7	DE	59,628	Y
91.195.240.7	DE	56,744	Y
185.230.61.173	IL	22,897	Y
185.230.60.173	IL	22,894	Y
31.31.205.59	RU	14,710	Y
31.31.205.62	RU	14,710	Y
209.235.147.130	US	5,751	Y
209.235.147.131	US	5,746	Y
151.236.17.126	DE	5,616	N
149.154.159.77	GB	4,048	N

Table 11: AA false lame delegated IPs.

6.4.1 Authoritative Consistency. By definition, authoritative nameservers should reply with authoritative responses (setting the `AA` flag). In a small percentage of cases (0.1–0.3%) authoritative nameservers do not reply as authoritative, creating lame delegations as a result. Table 11 shows the top 10 nameserver IPs that do not set the `AA` flag, ranked by the number of nameserver domains associated with those IPs. These turn out to be wildcard nameservers, which set the `AA` flag to false for `NS` queries and true for any `A` queries. The top nameservers group in pairs based on country code. In fact, the nameserver pairs are used as the two nameservers for parking domains, which obviates the need to update nameserver zone file records.

6.4.2 Glue Consistency. One can obtain IP addresses of nameservers by examining glue records in zone files, or by actively querying for their `A` records. These two methods should ideally yield the same set of IP addresses, but we find a surprising degree of inconsistency between the glue records in zone files and those returned by active queries. We examined the consistency between the set of IP addresses in the zone glue records ("parent" zone glue P) and the glue records retrieved via DNS queries (the "child" zone glue C).

Table 12 shows two interesting results between the two perspectives. First, similar to the results for domains in Table 7, a significant number of glue records cannot be resolved by querying, particularly for glue record IPs used by nameservers in the ngTLDs.

Second, for the glue records that can be queried, most are consistent ($P = C$). But from 5.5–11.4%, depending on the TLD, have inconsistent glue records. Table 12 further breaks down the relationships between the two sets P and C into four categories: the sets of glue records are completely disjoint ($P \cap C = \emptyset$); the parent zone glue records are a subset of the child zone glue records ($P \subset C$); and the parent zone glue records are a superset of the child zone ($P \supset C$); and sets that otherwise overlap in at least one address ("Rest"). The breakdown shows that the dominant inconsistencies are entirely disjoint: the child zone glue records are completely different from the parent zone glue records. As a result, for nearly 10% of the cases these inconsistencies create two entirely separate resolution paths for the same nameservers: in-bailiwick domains

	.com		ngTLDs		.net		.org	
Unresponsive	80554	16.4%	32410	42.0%	58189	21.9%	28961	29.5%
$P = C$	355055	72.2%	40528	52.5%	184407	69.3%	60739	61.8%
$P \neq C$	56038	11.4%	4223	5.5%	23534	8.8%	8524	8.7%
$P \cap C = \emptyset$	47716	85.1%	3832	90.7%	20541	87.3%	7754	91.0%
$P \cap C \neq \emptyset$	8322	14.9%	391	9.3%	2993	12.7%	770	9.0%
$P \subset C$	5742	69.0%	290	74.2%	2061	68.9%	488	63.4%
$P \supset C$	2317	27.8%	95	24.3%	805	26.9%	269	34.9%
Rest	263	3.2%	6	1.5%	127	4.2%	13	1.7%

Table 12: Parent-Child Glue Record Consistency.

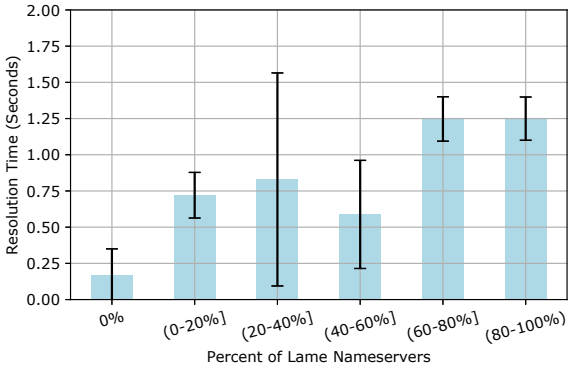

Figure 6: Average times to resolve domains over a month of daily resolutions. Domains are aggregated by the percentage of lame delegated authoritative nameservers they have, e.g., domains with five nameservers where three are lame delegated fall into the "(40,60%]" bucket. The whiskers show standard deviations.

will use the parent zone glue records, whereas out-of-bailiwick domains will use the child zone glue records.

6.5 Impact of Lame Delegation

In addition to their security risks (Section 5.6), lame delegations also degrade DNS resolution performance. In this section we quantify this performance impact and show that it affects even popular domains.

Lame delegations cause useless DNS queries. When resolving a domain that has at least one lame delegated nameserver, a resolver may have to contact multiple nameservers to successfully resolve the domain. As a result, the average resolution time for lame delegated domains will increase. To quantify this impact experimentally, we used data from OpenINTEL [28] to calculate the average resolution time for resolving the roughly 49 million domains in our active measurement set over the month of March 2020. OpenINTEL performs active measurement using a *normal resolver* to resolve domains. The normal resolution method approximates the user experience, and averaging resolution performance measurements over a month minimizes short-term variance.

	Measured	Fully Lame	Partly Lame
Alexa Top 100k	14,483	146	439
Alexa Top 1M	82,420	943	2,867

Table 13: Popular domains with lame delegations: the number of domains in our active measurement set that are on Alexa Top lists, and the number of those that are fully and partly lame.

The average resolution time for domains that are fully resolvable (without any lame delegated nameserver) was 172 ms, whereas domains with lame delegated nameservers had a significantly higher resolution time. For partly lame delegated domains (where a subset of the nameservers are lame), the average resolution time was 720 ms. For fully lame delegated domains, the resolution time was 1743 ms, an order of magnitude higher than fully resolvable domains. Note that these resolution times were bounded by timeout errors and caching since this data came from using a *normal resolver* process. Even entirely lame delegated domains ultimately have a maximum finite resolution time.

Figure 6 breaks down resolution times for the domains in our data set by the percentage of their lame delegated nameservers. For example, for domains with (40,60%] lame delegated nameservers (e.g., domains with two nameservers where one of them is lame, or domains with five nameservers where three are lame), the average resolution time was 0.59 seconds, 3.4× higher than domains with no lame delegated nameservers (the "0%" bucket). Overall the figure shows that a higher percentage of lame delegated nameservers per domain resulted in higher average resolution time.

We also observed that lame delegations occurred even on popular domains. Table 13 shows the number of domains in our active measurement set that are on Alexa Top lists [4], and the number of those that were fully and partly lame. We used the Alexa list for April 13, 2020, which corresponds to the midpoint of our active measurement campaign.

Table 13 shows that lame delegations, while not as ubiquitous, were present even for popular domains. Consider `archive.org`, an Alexa Top 200 site, which has one lame delegation of five possible delegations. As of September 12, 2020, `archive.org` was still partly lame delegated.[13] Surprisingly, we also encountered fully

[13] Note that `archive.org` while misconfigured is not at risk of being hijacked.

lame delegations in popular domains. We found that most domains switched their nameservers soon after, remediating the lame delegation. These observations reinforce our hypothesis that fully lame delegations are likely to be fixed more quickly than partly lame delegations because the domains are unusable when fully lame delegated.

Finally, as yet another perspective indicating that lame delegations are a notable operational issue, GoDaddy estimates that roughly 12% of requests to their nameservers are for domains for which they are not authoritative [24].

7 ETHICAL CONSIDERATIONS

We had to consider ethical aspects of characterization and responsible disclosure of lame delegations. Domains with lame delegations may be at risk of being hijacked. Given the many thousands of at-risk nameserver domains, we cannot defensively register all of them, which would raise its own ethical issues if we could. Without the ability to protect these lame domains, disclosing them increases the risk of harm to their owners and users. We are working on a responsible way to disclose our snapshot of lame delegations.

8 SUMMARY

The Internet, as it is commonly taught, is constructed from simple abstractions implemented via a number of key network protocols. Invariably, however, there is significant daylight between this clean abstract model of how the Internet functions and the frequently messy reality of its concrete operation. Measurement studies such as this one are the mechanisms we use to characterize this gap in understanding. Our work characterizing the presence and risks of lame delegation in the DNS exemplifies the value of this kind of empirical study.

Using comprehensive collections of both active and passive DNS measurements (covering 49 M and 499 M domains respectively), we found that lame delegations are surprisingly common: roughly 14% of registered domains that we actively measured had at least one lame delegation, and most of those had no working authoritative nameservers. However, even for domains with working alternative nameservers, our measurements show that these lame delegations impair DNS performance (average resolution latency increasing by 3.7×) in addition to producing substantial unnecessary load on existing nameservers.

Finally, we found that unregistered or expired domains in lame delegations can create significant security risk. Indeed, over the last nine years, we identified at least three instances in which an attacker could have hijacked thousands of domains by registering a single nameserver domain. Analysis of this phenomenon led us to discover an unforeseen interaction between registrar practice and the constraints of registry provisioning systems that has inadvertently made hundreds of thousands of domains vulnerable to hijacking due to accidental lame delegations. This practice has persisted for over twenty years, but we are now working with registrars to remediate it and its effects.

Going forward, we are exploring ways to combine daily zone data and periodic active measurements to automatically identify and report lame delegations as they are created. An open question remains about the most effective mechanisms for communicating these findings to appropriate stakeholders to incent corrective action. As well, the security issues that arise as unintended byproducts of registrar/registry practices deserve further attention as this aspect of the domain name ecosystem is largely opaque to the research community.

Many domain operators configure redundancy in resolution infrastructure, which can hide underlying systemic issues for long periods of time. Ironically, this engineered robustness poses a security threat, as domain operators rarely take notice of DNS configurations unless their domain stops resolving completely. Thus they are likely to fail to notice partly lame domains that attackers can exploit.

While some systematic issues such as the "DROPTHISHOST anomaly" require registrar-level intervention to fix, domain owners can proactively monitor their own domain configurations. In pursuit of improved monitoring and remediation, we are developing a monitoring tool to allow domain owners to check static zone files for potential delegation-related security risks, and will integrate it into our zone analysis platform. Finally, we have begun an effort to work with the registrar and registry communities to responsibly disclose such risks, establish their underlying causes, and develop improved operational practices to minimize lame delegations going forward.

9 ACKNOWLEDGMENTS

We thank our shepherd Georgios Smaragdakis and the anonymous reviewers for their insightful suggestions. We also thank Cindy Moore, Alistair King, Bradley Huffaker, Daniel Andersen, Paul Biglete, and Vinay Pillai for their support of software and hardware infrastructure necessary for this project. We thank Brian Dickson, Duane Wessels, Joe Abley, Tim April, Patrik Fältström, Steve DeJong, Dave Knight, Casey Deccio, James Galvin, and Roland van Rijswijk-Deij for their valuable time, insights, and feedback.

This work was supported in part by National Science Foundation grants CNS-1629973, CNS-1705050, OAC-1724853, and OIA-1937165, Department of Homeland Security grant AFRL-FA8750-18-2-0087, the Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, the EU H2020 CONCORDIA project (830927), the NWO-DHS MADDVIPR project (628.001.031/FA8750-19-2-0004), and generous support from Facebook and Google. This research was made possible by OpenINTEL, a joint project of the University of Twente, SURFnet, SIDN, and NLnet Labs.

REFERENCES

[1] J. Abley, B. Dickson, W. Kumari, and G. Michaelson. 2015. AS112 Redirection Using DNAME. RFC 7535. https://rfc-editor.org/rfc/rfc7535.txt
[2] AFRINIC. 2019. *AFRINIC ratifies 'Lame Delegations in the AFRINIC reverse DNS' Policy.* African Network Information Centre. https://afrinic.net/lame-delegations-in-afrinic-reverse-dns-policy-ratified
[3] AFRINIC. 2020. *Lame delegations statistics.* African Network Information Centre. https://stats.afrinic.net/lamerdns/
[4] Alexa. 2020. Top 1M sites. https://toplists.net.in.tum.de/archive/alexa/alexa-top1m-2020-04-13_0900_UTC.csv.xz
[5] E. Alowaisheq, P. Wang, S. Alrwais, X. Liao, X. Wang, T. Alowaisheq, X. Mi, S. Tang, and B. Liu. 2019. Cracking the Wall of Confinement: Understanding and Analyzing Malicious Domain Take-downs. In *Proceedings of The Network and Distributed System Security Symposium (NDSS).* Internet Society, San Diego, CA, USA.
[6] APNIC. 2020. *Lame DNS Reverse Delegation.* Asia Pacific Network Information Centre. https://www.apnic.net/manage-ip/manage-resources/reverse-dns/lame-dns-reverse-delegation

[7] ARIN. 2014. *Recommended Draft Policy ARIN-2014-5: Remove 7.2 Lame Delegations.* American Registry for Internet Numbers. https://www.arin.net/vault/policy/proposals/2014_5.html

[8] D. Barr. 1996. Common DNS Operational and Configuration Errors. RFC 1912. https://rfc-editor.org/rfc/rfc1912.txt

[9] DNS Coffee. 2020. *DNS Coffee.* DNS Coffee. https://dns.coffee

[10] S. Hollenbeck. 2009. Extensible Provisioning Protocol (EPP) Domain Name Mapping. RFC 5731. https://rfc-editor.org/rfc/rfc5731.txt

[11] S. Hollenbeck. 2009. Extensible Provisioning Protocol (EPP) Host Mapping. RFC 5732. https://rfc-editor.org/rfc/rfc5732.txt

[12] ICANN. 2007. *IANA Report on the Delegation of the .TEL Top-Level Domain.* ICANN. https://www.iana.org/reports/2007/tel-report-22jan2007.html

[13] ICANN. 2017. *Transfer Report for tel.* ICANN. https://www.iana.org/reports/tld-transfer/20170503-tel

[14] ICANN. 2019. *ICANN CZDS.* ICANN. https://czds.icann.org

[15] ICANN Security and Stability Advisoiry Committee (SSAC). 2020. SSAC Advisory on Private Use TLDs. https://www.icann.org/en/system/files/files/sac-113-en.pdf

[16] A. Kalafut, M. Gupta, C. A. Cole, L. Chen, and N. E. Myers. 2010. An Empirical Study of Orphan DNS Servers in the Internet. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (Melbourne, Australia) *(IMC)*. ACM, New York, NY, USA, 308–314. https://doi.org/10.1145/1879141.1879182

[17] LACNIC. 2020. *Lame Delegation Policy.* Latin America and Caribbean Network Information Centre. https://www.lacnic.net/686/2/lacnic/6-lame-delegation-policy

[18] D. Liu, S. Hao, and H. Wang. 2016. All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS)*. ACM, New York, NY, USA, 1414–1425. https://doi.org/10.1145/2976749.2978387

[19] P. Mockapetris. 1987. Domain Names - Concepts and Facilities. RFC 1034. https://rfc-editor.org/rfc/rfc1034.txt

[20] P. Mockapetris. 1987. Domain Names - Implementation and Specification. RFC 1035. https://rfc-editor.org/rfc/rfc1035.txt

[21] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. 2004. Impact of Configuration Errors on DNS Robustness. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Portland, Oregon, USA) *(SIGCOMM)*. ACM, New York, NY, USA, 319–330. https://doi.org/10.1145/1015467.1015503

[22] A. Phokeer, A. Aina, and D. Johnson. 2016. DNS Lame delegations: A case-study of public reverse DNS records in the African Region. In *Proceedings of the 8th EAI International Conference on e-Infrastructure and e-Services for Developing Countries — AFRICOMM*. ICANN, European Alliance for Innovation, Ouagadougou, Burkina Faso.

[23] D. Piscitello. 2010. *Conficker Summary and Review.* ICANN. https://www.icann.org/en/system/files/files/conficker-summary-review-07may10-en.pdf

[24] GoDaddy Representative. 2020. Personal Communication.

[25] A. Romao. 1994. Tools for DNS debugging. RFC 1713. https://rfc-editor.org/rfc/rfc1713.txt

[26] R. Sommese, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, KC. Claffy, and A. Sperotto. 2020. The Forgotten Side of DNS: Orphan and Abandoned Records. In *Proceedings of the 2020 Workshop on Traffic Measurements for Cybersecurity (WTMC)*. IEEE, Virtual Event.

[27] R. Sommese, G. CM. Moura, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, KC. Claffy, and A. Sperotto. 2020. When parents and children disagree: Diving into DNS delegation inconsistency. In *Proceedings of the International Conference on Passive and Active Network Measurement (PAM)*. Springer, Springer International Publishing, Virtual Event, 175–189.

[28] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras. 2016. A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. *IEEE Journal on Selected Areas in Communications (JSAC)* 34, 6 (2016), 1877–1888.