The Forgotten Side of DNS: Orphan and Abandoned Records

Raffaele Sommese*, Mattijs Jonker*, Roland van Rijswijk-Deij*, Alberto Dainotti†, K.C. Claffy†, Anna Sperotto*

*Design and Analysis of Communication Systems (DACS)
University of Twente
Enschede, the Netherlands
{r.sommese, m.jonker, r.m.vanrijswijk, a.sperotto}@utwente.nl

†CAIDA
UC San Diego
La Jolla, CA, USA
{alberto, kc}@caida.org

***Abstract*—DNS zone administration is a complex task involving manual work and several entities and can therefore result in misconfigurations. Orphan records are one of these misconfigurations, in which a glue record for a delegation that does not exist anymore is forgotten in the zone file. Orphan records are a security hazard to third-party domains that have these records in their delegation, as an attacker may easily hijack such domains by registering the domain associated with the orphan. The goal of this paper is to quantify this misconfiguration, extending previous work by Kalafut *et al.*, by identifying a new type of glue record misconfiguration – which we refer to as *abandoned records* – and by performing a broader characterization. Our results highlight how the situation has changed, not always for the better, compared to a decade-old study.**

***Index Terms*—DNS, orphan records, abandoned records, misconfiguration**

1. Introduction

The Domain Name System (DNS) [1] is part of the core Internet infrastructure and also one of the most complex parts. The DNS is organised as a hierarchical, distributed database with built-in redundancy. The responsibility for domains is arranged through a process of delegation, in which an entity at a higher level in the hierarchy diverts responsibility for a subset of the name space to another party. The hierarchy starts at the root of the DNS, which delegates top-level domains (TLDs) such as `.com`, `.net`, `.nl`, etc. These TLDs in turn delegate to second-level domains, which in turn may further delegate parts of the name space. Administration of these delegations, especially at the TLD level, can be a complex task involving many entities. As a matter of fact, within the context of the DNS, we typically identify three types of stakeholders: *registries*, *registrars* and *registrants*. The registry is the entity responsible for the administration of a TLD. The registrar provides an interface between registrant and registry. It manages the administrative parts related to the selling of the domain (e.g. manage payment, renewal, billing and collecting owner information) and usually provides customers an interface to interact and modify the records related to their domains. Finally, the registrant is the customer that registers a domain.

Given the complexity of managing DNS information, misconfiguration and errors can occur, with an impact on the overal security and reachability of the DNS.

The goal of this paper is to analyze a specific misconfiguration, defined as the *orphan record* [2] in a TLD. Such orphan records are a leftover of a domain that has expired, and should have been removed, by the registry or by the registrar, together with the expired domain they belong to. Orphan records form a security risk as unwitting third party domains may still point to these orphan records in their delegation. An attacker can easily hijack domains referring to orphan records by re-registering the domain they belong to.

This work reproduces and extends the analysis performed by Kalafut *et al.* [2] in 2010 and is motivated by the main question: *A decade after the original analysis, what does the orphan records phenomenon look like?* Compared to [2], we characterize the *orphan records* phenomenon through a significantly larger dataset of ∼2K TLDs and over a wider time window of 25 months. We also introduce a related type of misconfiguration, which has not been considered before, that we refer to as *abandoned records*, which could be predecessors to new orphan records.

2. Background

Zone files and glue records. A *zone* is a portion of the DNS managed by a single entity. A *zone file* describes all the *Resource Records* (RRs) related to a zone, and its format is defined in two IETF standards [3], [4]. Resource Records map a host name to a specific type, e.g. an `A` record maps a host name to an IPv4 address. A top-level domain (TLD) is a special type of zone that typically only has one task: to delegate second-level domains. This delegation uses `NS` records that identify the name server for a domain. If the `NS` record for a domain points to a record that is inside the domain (called *in-bailiwick*), that name is included in the zone as a *glue record* to enable the resolution process to continue. Consider for example: `example.com NS ns1.example.com`. To resolve `example.com`, we need to resolve `ns1.example.com`, but this implies resolving the `example.com` delegation. Defining the `A/AAAA` glue record for `ns1.example.com` in the

parent zone file breaks this circular dependency and allows the domain to be resolved. *Glue records* are usually the only A/AAAA records admitted in TLD *zone files*. A notable exception to this condition represented by the `.de` *zone* is explained in § 5.3.

EPP - the Extensible Provisioning Protocol. EPP is an XML text protocol defined in RFC5730 [5]. The primary goal of EPP is to permit multiple service providers to manipulate objects in a shared centralized object directory. EPP was introduced by *Hollenbeck* to provide a standard Internet domain registration protocol between registrars and registries. The protocol defines and describes the interaction between these two parties via a standard set of atomic and idempotent commands. EPP defines three main object types: domains, contacts and hosts. EPP also defines *actions*, such as: check, info, create, update, delete, transfer and renew. *Domain* objects represent the domain itself, *contacts* are the WHOIS contact information, and finally *hosts* represent the glue records.

In EPP, creation of host and domain resources are two independent operations. A registrar typically creates glue records in case of the aforementioned in-bailiwick `NS` record case. The specification [5] does not define who is responsibile to clean up glue records if they are no longer required (*e.g.,* in case of expired domains). A registrar could periodically check this. Alternatevely, a registry could check on if glue records in its database are actually required by an *in-bailiwick* domain before publishing a zone on its name servers. When a domain expires, after a grace period the registrar return the responsibility of managing the domain to the registry, which should ensure that all the related resource records are deleted by the registrar.

Orphan and abandoned records. Glue records are supposed to be removed after a delegation is removed or changed. Earlier work indicates, however, that this does not always happen in practice [2]. In this paper, we define an **orphan record** as a former glue record for which the related domain no longer exists in the zone (the delegation has been removed). We also define an **abandoned record** as a former glue record for which the related domain still exists in the zone but the delegation no longer requires that glue record. Under normal operation, abandoned records do not show up in the DNS resolution, as there is no longer a relation with the domain the record served. Abandoned records show up in the additional section only when they are referred by a delegation of other domains of the zone. However, it is still questionable if they should remain at all in a TLD zone file. Finally, we define **junk records**, as the union of orphan and abandoned records.

Related work. Kalafut *et al.* [2] characterized the problem of orphan records in terms of their spread, usage, lifetime and hosted resources, for a 31-day timeframe. The authors considered zone files for 6 TLDs as well as malware URL feeds. We reproduce part of their results, but for a significantly longer, 25-month timeframe, enabling long-term characterization of the junk records phenomenon. Moreover, as 10 years have passed and we focus on a recent period, we can analyze how this problem has evolved over the past decade. Where possible, we run our analysis on the same zones as in [2], but also extend the analysis to other zones available to us. Liang *et al.* [6] proposed a method for keeping DNS records locked in the cache of open DNS resolvers after the domain expires. The authors defined these records as *ghosts* and prove that by performing queries against open resolvers and by crafting an ad-hoc response in the controlled authoritative nameserver, it was possible to refresh the *TTL* value for the record in the cache of the open resolver even if the domain no longer exists in the parent zone. While their work does not specifically focus on junk records, it indirectly refers to generic expired domains in the parent zone.

example.com	86400	IN	NS	ns.external.org
ns1.example.com	3600	IN	A	1.2.3.4
ns1.expired1.com	3600	IN	A	3.2.5.4
ns1.expired2.com	3600	IN	A	8.4.5.6
active.com	86400	IN	NS	ns1.expired2.com
good.com	86400	IN	NS	ns1.good.com
ns1.good.com	3600	IN	A	1.2.3.5

TABLE 1: Example `.com` Zone File
■ Algorithm 1 (O) ■ Algorithm 2 (A)

3. Methodology and Dataset

Methodology. We developed two algorithms to respectively identify orphan and abandoned records inside zone files. These algorithms rely on the principle that in the zone file the only `A` records available are glue records[1].

Algorithm 1 identifies orphan records and it is similar to the one described in [2]. The algorithm first collects all the domains in `A` records available inside the zone file. Then it trims the domains to the second level domain (SLD). Finally, it looks for SLDs that do not have any associated `NS` record.

Algorithm 2 identifies abandoned records. The algorithm collects the list of domains in the `A` records available in the zone file. Alike Algorithm 1, it trims the domain to the SLD and looks for the SLDs for which the `NS` records do not point to the extracted `A` records.

Table 1 provides an example of records retrieved by the two algorithms. Algorithm 1 identifies `ns1.expired1.com` and `ns1.expired2.com` as orphans since no `NS` records exist for `expired1.com` or `expired2.com`. Algorithm 2 marks `ns1.example.com` as abandoned (a delegation for `example.com` exists, but points elsewhere).

Dataset. The TLDs we consider for this study are `.aero`, `.asia`, `.biz`, `.ca`, `.com`, `.fi`, `.info`, `.mobi`, `.name`, `.net`, `.nu`, `.org`, `.ru`, `.se` and `.us`. We also include 1184 new gTLDs introduced by ICANN, which we collectively refer to as "CZDS"[2]. We make use of OpenINTEL, a large-scale DNS measurement platform [7]. OpenINTEL collects zone files on a daily basis. The combined zones cover a period of 25 months, from April 2017 to May 2019 (760 days). This set of zone files contains per day on average 3,283,404 unique `A` records, 199,249,769 unique domains, and 1,317,987 unique in-bailiwick domains. A zone file might occasionally not be collected (*e.g.,* due to contract renewal processes). This

1. So as to reproduce the work in [2], we did not consider `AAAA` RRs.
2. The ICANN system that regulates access to the zones for these domains is called the Centralized Zone Data Service (CZDS).

TLD	Coverage	Start-Date	End-Date	TLD	Coverage	Start-Date	End-Date
info	98.3%	2017-04-01	2019-04-30	ca	94.3%	2017-04-01	2019-04-30
mobi	96.2%	2017-04-01	2019-04-30	fi	97.5%	2017-04-01	2019-04-30
asia	94.4%	2018-11-20	2019-04-30	aero	94.4%	2018-11-20	2019-04-30
org	99.9%	2017-04-01	2019-04-30	biz	93.8%	2018-11-20	2019-04-30
com	96.0%	2017-04-01	2019-04-30	name	94.4%	2018-11-20	2019-04-30
net	98.6%	2017-04-01	2019-04-30	nu	99.3%	2017-04-01	2019-04-30
us	99.4%	2018-11-19	2019-04-30	se	99.3%	2017-04-01	2019-04-30
ru	99.1%	2017-06-17	2019-04-30	CZDS	99.9%	2017-04-01	2019-04-30

TABLE 2: Overview of datasets used in this work

TLD	#Glue records/day	#Orphan/day	Min	Max	Mean	Prev Orphan	Prev #Glue	#Abandoned/day	Min	Max	Mean	Sum
.info	169946	43687	17.3%	36.0%	24.9%	18.8%	139126	70180	36.5%	49.0%	41.8%	66.7%
.mobi	6855	1602	5.5%	37.5%	22.7%	10.7%	4062	2972	36.0%	54.2%	44.0%	66.7%
.asia	6122	1140	17.8%	19.9%	18.6%	7.5%	1313	3294	52.2%	55.5%	53.8%	72.4%
.org	364568	21929	4.4%	7.5%	6.0%	3.7%	206513	234256	62.8%	65.5%	64.2%	70.2%
.com	1873668	0	0.0%	0.0%	0.0%	0.4%	1566392	602641	31.4%	33.0%	32.2%	32.2%
.net	303387	0	0.0%	0.0%	0.0%	0.2%	331896	55327	9.5%	19.6%	18.2%	18.2%
.us	26042	1869	6.8%	8.0%	7.2%	3931*	N/A	1577	5.1%	8.5%	6.1%	13.3%
.ru	79492	54	0.0%	0.1%	0.1%	1801*	N/A	2998	2.0%	4.1%	3.8%	3.8%
.ca	23537	980	3.9%	4.5%	4.2%	1368*	N/A	3467	13.9%	15.7%	14.7%	18.9%
.fi	3908	0	0%	0%	0%	N/A	N/A	0	0.0%	0.0%	0.0%	0.0%
.aero	626	22	3.0%	4.3%	3.6%	N/A	N/A	342	53.5%	55.7%	54.7%	58.3%
.biz	22958	6	0.0%	0.0%	0.0%	N/A	N/A	2113	7.5%	12.5%	9.3%	9.3%
.name	1820	50	2.2%	3.0%	2.7%	N/A	N/A	42	1.8%	2.8%	2.3%	5.0%
.nu	2184	0	0.0%	0.0%	0.0%	N/A	N/A	0	0.0%	0.0%	0.0%	0.0%
.se	20053	48	0.2%	0.3%	0.2%	N/A	N/A	0	0.0%	0.0%	0.0%	0.2%
CZDS	387897	17144	0.5%	23.9%	5.0%	N/A	N/A	84805	3.5%	31.5%	22.2%	27.2%

*Kalafut *et al.* report the number of orphans instead of the percentage for these TLDs due to lack of access to the zone files [2].

TABLE 3: Orphan and Abandoned records for each TLD over *2017-04-01 – 2019-04-30*. The absolute numbers shown are daily averages.

happens at maximum for 5.6% of the measurement period (42 days for `.asia`). This means that we can consider our results a lower-bound for the orphan and abandoned records problem. Table 2 lists the effective start and end dates for each TLD in our dataset, and the percentage of days covered in the range. We used Apache Spark [8], an open source cluster computing framework, to perform our analysis.

4. Characterizing Orphan and Abandoned

4.1. Orphan record distribution

Kalafut *et al.* [2] identified `.info` as the TLD with the highest percentage of orphan records in the period between 2009-04-01 and 2009-05-1. Our analysis shows that 10 years later, the number of orphan records in this TLD is still rising (Table 3). Of an average of 169,946 `A` records per day, in the studied period, an average 24.9% of these are orphan records, with a maximum of 36% and a minimum of 17.3%. Comparing these results to [2] we find an increase of 6.1% on average and of 17.2% as a maximum.

The `.mobi` TLD shows a similar trend. With an average of 6,855 `A` records per day. The mean percentage of orphan records is 22.7%, with a peak of 37.5%. Compared to [2], the number of orphan records tripled, with an increase of 12% in the total number of records, whereas the total number of records become 1.71x. We note that the number of orphan records for `.mobi` has decreased over the last year, but found no evidence that this is due to a targeted cleanup action. A remarkable difference with [2] is that `.com` and `.net` no longer contain any orphan records. We discuss the case in § 5.3.

For `.asia` and `.org`, we find more records compared to [2] with an almost constant trend. For `.us`, `.ca` and `.ru` instead, we identify fewer records compared to [2], which already underestimated the number of records for these TLDs, as they did not have access to the respective zone files. This means that these TLDs improved their management of glue records.

4.2. Abandoned record distribution

The TLD with the highest percentage of abandoned records is `.org`, with a mean of 64.2% abandoned records. The `.info` TLD exhibit a lower percentage (41.8%). Considering the percentage of orphan records and abandoned records together, `.mobi`, `.info`, `.asia` and `.org` show a percentage of junk records around 65%, casting doubts on the management of these zones. Also `.com` and `.net` show a relevant number of abandoned records. For `.fi`, `.nu`, and `.se`, we do not find any abandoned records.

4.3. IP address and domain distribution

We now investigate how many domains and how many IP addresses are related to junk records. The distribution of the IP addresses related to the orphan and abandoned records shows 38% of records that point to a single IP address for orphans and 67% for abandoned records. Moreover, 88% of orphans and 92% of abandoned records refer to a single or to two IP addresses, with an average of 2.32 and 1.83 orphan and abandoned records, respectively, per IP. Compared to [2], the number of orphans per IP decreased (from an average of 3.2 orphans per IP in [2]). Since `.com` and `.net` dominated the number of orphans in [2], we assume that the cleaning of these zones is reflected in this decrease. There are also some peculiar cases. For example, in `.info`, 1,754 orphan records point to Cloudflare's public resolver `1.1.1.1`. This is

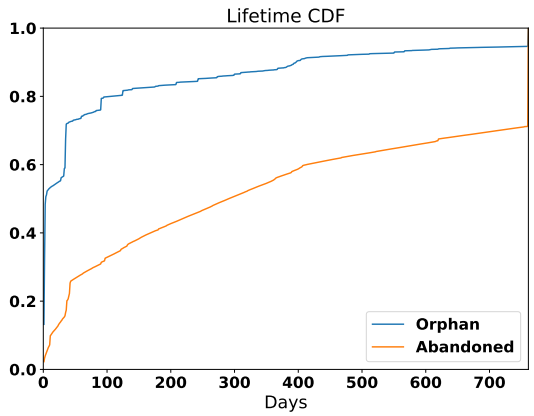

Figure 1: Lifetime of orphan and abandoned records

Orphan records											
Ref by	ru	org	asia	CZDS	se	name	mobi	us	ca	info	Total
aero	0	0	0	0	0	0	0	36	4	0	40
asia	0	0	26	23	0	0	0	0	0	11	60
biz	0	17	12	93	4	0	2	62	101	96	387
ca	0	16	0	5	0	0	0	2	10320	24	10367
com	0	1337	41	276	34	2	19	3923	6223	1111	12966
CZDS	3	194	11	404	0	0	1	44	208	292	1157
fi	0	0	0	0	0	0	0	0	0	38	38
info	0	101	5	126	4	1	7	291	219	453	1207
mobi	0	8	4	99	0	0	30	7	149	5	302
name	0	0	0	28	2	68	0	0	9	0	107
net	0	277	10	139	19	0	3	1566	874	136	3024
nu	0	14	0	0	18	0	0	0	1	0	33
org	0	288	6	133	14	2	3	4695	1038	160	6339
ru	43	26	0	3	0	0	24	0	0	14	110
se	0	0	0	0	110	0	0	0	0	2	112
us	0	33	5	22	0	0	0	3287	67	20	3434
Total	46	2311	120	1351	205	73	89	13913	19213	2362	39683

TABLE 4: NS records pointing to orphan records

the result of a misconfiguration of NS resolvers (circular dependency), which causes unreachability of the domain.

We also analyze domain distributions. For orphan records, in 94% of the cases, we find two orphan records for a single SLD. This result is consistent with the common configuration practice of DNS, in which administrators set up two authoritative nameservers, thus two A glue records for a domain. In 7% of the cases, we find one orphan record for a single SLD. For abandoned records, in 85% of the cases, we find two abandoned records for a single SLD, and in 21% of the cases we find one.

4.4. Lifetime of records

Fig. 1 shows the lifetime CDF of orphan and abandoned records. Lifetimes are the uninterrupted time segments during which we consider glue records to be orphaned or abandoned in our analysis. The plot contains only data for .info, .mobi, .org, .ca, .se, and CZDS. We do not include other TLDs since their zone collection started later in time in OpenINTEL. However, the shape of the CDF is similar across the different TLDs, with some exceptions that we explain later. The number of orphan records that lived at most 1 day is 19%, which is higher compared to the 12% found in [2]. Also, [2] indicates that only 2% of the orphan records last their entire measurement duration (31 days). In our case, 4% of orphan records survived for more than 760 days (the time frame of our analysis). The results for abandoned records are higher than the orphan ones: only 8% of records lived one day or less and 28% of records lived more than 760 days. Interestingly, when we look at individual TLDs, we find this difference between orphan and abandoned lifetime mainly present in .org and the new gTLDs, where abandoned records lived longer than orphan records.

The CDF in Fig. 1 shows that ∼4% (21,640) of all the orphan records we find (541,002), persisted for more than 760 days (our observation period). These records were orphans during all the period of our analysis and represent a significant fraction of the orphan records we observed daily (fourth column in Table 3). In a similar way, for abandoned records, we discover 496384 persistent records that survived more than 760 days. These results confirm that junk records are *a long-term misconfiguration*, which persistently affects the TLD zones.

4.5. NS references to orphans

As we discussed in the introduction, NS records for other domains may refer to orphan records. This creates a serious vulnerability: an attacker can register the domain of the orphan record, thus redirecting all queries to a malicious authoritative name server under their control. By controlling the orphan resource records in a malicious authoritative name server, an attacker can divert traffic to any malicious destination (NS hijacking). This hijacking affects all domains that define as an authoritative name server the hijacked orphan record.

We find 39,683 NS records that refer to orphan records, either in the same zone, or in other zones. In Table 4, we show the reference matrix for each TLD. For typographical reasons we exclude empty and non relevant columns. The most referenced orphan records are in .ca, and are referenced ∼10k times in the .ca and ∼6k times in .com. This matrix also helps us understand that the removal of an orphan record could have an impact on other domains in other TLDs, and for this reason removal should be analyzed carefully.

4.6. Orphan DNSSEC signed records

Another issue with orphan records concerns DNSSEC-signed zones (featured by most TLDs [9]). Normally, glue records are not signed, since the TLD is not authoritative for the domain (the name servers to which the domain is delegated are). If the domain is deregistered, however, the glue records are implicitly (and unintentionally) promoted by the registry to records that are part of the TLD zone, and will be DNSSEC-signed. This behaviour results into signing and providing warranty about the authenticity of junk records, increasing the zone file size and raising doubts about the legitimacy of these signatures.

5. Origin of Orphan and Abandoned Records

5.1. Relationship between orphan and abandoned records

Orphan and abandoned records are both useless glue records left in the zone file of TLDs. This raises the

TLD	Dist(O)	Dist(O)*	Dist(A)	$A \implies O$	$O \implies A$
info	127014	103650	169702	44938	3769
mobi	4395	4037	6575	2047	81
asia	1679	1679	3857	475	27
org	82231	66814	369330	52559	5763
CZDS	320158	297219	357123	31346	1880
Total	535477	473399	906587	131365	11520

* Orphan records with with *birth date* $\geq$ *2017-04-01*

TABLE 5: Relationship between (O)rphan and (A)bandoned

question if an abandoned record can become an orphan record or vice-versa. An abandoned record that becomes an orphan could be a sign of poor management practices at the registry or registrant, who do not clean the zone file. Moreover, it could help us to infer records that will likely be orphan records after the expiration of the related domain. An orphan record that becomes an abandoned record indicates that someone registered the domain related to the orphan records. This could happen for legitimate reasons (e.g. people ot being aware of the orphaned status of the domain), or for malicious purposes (e.g. to take control of the orphan record). Table 5 shows the results of our analysis.

Abandoned $\implies$ *Orphan*. Our dataset shows a total of 535,477 distinct orphan records, of which 473,399 come into being in our window of analysis, *i.e.,* records with *birth date* $\geq$ *2017-04-01*. We focus on this category to investigate if there is any relation with abandoned records. We find indeed that roughly 27.7% of orphan records were previously abandoned records. This strong correlation confirms that abandoned records are likely to become orphans at a later point in time.

Orphan $\implies$ *Abandoned*. We do not find many records morphing from orphan to abandoned. In fact only 11,520 of 535,477 orphan records in our dataset became abandoned (2.1%). A likely explanation is that orphan records get registered again. This can occur without the registrant being aware of the orphan status, meaning that unnecessary (junk) records related to a domain might be present in the zone without the registrant noticing it. However, if the registrant is aware of the orphan status of a record, then we might have witnessed a hijack (see § 4.5).

5.2. WHOIS orphan and abandoned

We performed a WHOIS information dump of orphan and abandoned records on *2019-12-03* using SpiderWho [10] for parallel lookup. We analyzed the WHOIS information to understand: (i) if the domains are in the WHOIS database (i.e. domain registered), (ii) if these misconfigurations belong to a specific registrar, or (iii) to a special administrative status of the domains (e.g. locked, expired, etc.).

Out of 54,421 domains belonging to orphan records, 29,418 (54%) have no associated WHOIS information at all in the WHOIS server of the relative TLD, meaning they were potentially available for registration. Of the remaining 25,003 domains, 19,491 were registered through Namecheap, 2,201 domains were in clientHold state, 294 are inactive, and 290 in pending delete state. Interestingly, when we tried to recreate an orphan through the Namecheap web interface – in order to understand if the registrar behaves in a bad way (*i.e.,* not deleting orphan records) – we had no success.

For abandoned records, by definition, all related domains were registered. We found only few records not available in the WHOIS database (related to failure in querying or parsing). Of 165,492 domains, 34,334 are registered with GoDaddy and 20,512 with Namecheap. We tried to recreate an abandoned record through Namecheap and GoDaddy without success. We suspect that is possible to create these records through API calls for managing the domains, which we did not verify as a premium account is required.

5.3. The Case of `.com` and `.net`

A difference in our results compared to Kalafut *et al.*, is that `.com` and `.net` no longer contain any orphan records. Our dataset does not allow us to pinpoint the point in time when these records disappeared as this occurred prior to *2017-04-01*. For this reason we use the archives of `.com` and `.net` zone data provided by DNS-OARC [11] in order to detect the date of occurrence. Figure 2 shows the temporal evolution of orphan records

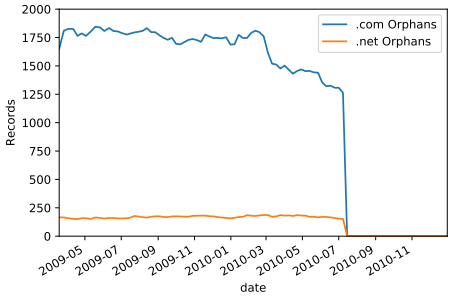

Figure 2: Orphans in `.com` and `.net` 2009 – 2010

between May 2009 and December 2010. Differently from our data collections, DNS-OARC does not collect and publish zone files daily. Based on these data we can pinpoint the disappearance of orphan records between July 7 and July 15, 2010. Figure 2 also shows a net decrease of orphans already in March 2010. We can trace this back to [12] in which Verisign explicitly announced that from March 1, 2010, glue records would no longer be promoted to authoritative status. However, they also stated that: "These records will not actually be removed: although they will not be returned when queried for directly, they will appear in the additional section of referrals that reference them". After July 2010, Verisign also started removing orphan records from the zone files that are made available through the zone file access program. Since these records are still returned in DNS referrals, this means that the zone file made available through the zone access program no longer exactly reflects the state of the registry database.

5.4. The Case of `.se`

In the case of .se, we find 48 orphan records. These records are all subdomains of: `org.se`, `pp.se`, `fhsk.se`, `d.se`, `g.se`, `ns.se`, `ac.se`, and `fh.se`.

These domains reflects the former structure of the *.se* namespace (pre 2003), in which any registered domain was a subdomain of a registry managed second level domain [13]. The *.se* domain name structure was then liberalized. However, *.se* operators confirmed to us that some old records are maintained in the zone file for legacy purposes and are banned for registration. Therefore, their presence does not pose any issues.

5.5. The Case of `.de`

Even though we do not analyze `.de` domains, their administration policy represents a special case that was not considered in [2]. DENIC permits users to publish and manage domains directly in the `.de` zone with the following three restrictions: (i) maximum 5 RRs; (ii) only `A`, `AAAA` and `MX` RRs; (iii) records are checked (for legitimacy) by the DENIC staff. This opportunity to directly manage subdelegations breaks the common operating model of TLDs and impacts the discovery of orphan and abandoned records. In particular, these A records can be false positives for the analysis conducted by Kalafut in [2]. However, given that `.de` is not considered in our analysis (due to lack of access to `.de` zone files) and `.info`, `.mobi` and `.org` (which are the most affected by the orphan and abandoned misconfiguration) do not allow this operational model of directly publishing and managing records in the zone, we are confident that this does not impact the conclusions and the main results of our analysis.

6. Ethical Considerations

We perform our study of orphan and abandoned records through the analysis of zone files provided from registry operators to the OpenINTEL project. The results related to orphan records could lead to potential NS hijacking. For this reason, and because of the contractual restrictions under which OpenINTEL gets access to most zone files, we publish only aggregated results and we do not refer to specific cases. Furthermore, we performed the WHOIS scan using a conservative approach and on a single day in order to not overload the WHOIS servers.

7. Conclusions

Our work was prompted by the 2010 work by Kalafut *et al.* [2] and aimed at evaluating the state of orphan misconfiguration a decade later. We discovered that for the `.com` and `.net` TLDs, the number of orphan records has fallen to zero, which means that operators have introduced mechanisms for cleaning their zone files. Unfortunately, these best practices are not adopted by all TLD registry operators. For some TLDs, the number of orphan records have increased over 10 years. Also, in the new gTLDs introduced after [2], this misconfiguration is widely spread among TLDs. We also discover and analyze another misconfiguration: the abandoned record. Our analysis shows that this misconfiguration is broader than the orphan one. Even if these records are not resolved, common sense would suggest they should be removed by registers or registrars, as they potentially represent the initial stage of orphan creation. Our study also shows that the removal of these records from the zone file may not be a simple operation, since it can incur the risk of breaking other domains. Future work, in collaboration with registry operators, will address the nature of the resources related to orphan records (*i.e.,* hosted websites or domains) and of their related traffic by actively registering these domains and intercepting it. Finally, we suggest all registry operators address this misconfiguration by at least making domains related to orphan records not available for registration or by considering to clean up their *zone* removing orphans.

Acknowledgments

We thank Duane Wessels, Ulrich Wisser and Joe Abley for their valuable feedback on our research. This work is partially funded by the NWO-DHS MADDVIPR project (628.001.031/FA8750-19-2-0004), the PANDA project (NSF OAC-1724853) and the EU H2020 CONCORDIA project (830927). We thank DNS-OARC for providing us access to OARC archives copies of the TLD zone files. This material is based on research sponsored by the Air Force Research Laboratory under agreement number FA8750-18-2-0049. The views and conclusions in this paper do not necessarily reflect the opinions of a sponsor, Air Force Research Laboratory or the U.S. Government.

References

[1] J. Klensin, "Role of the Domain Name System (DNS)," Internet Requests for Comments, RFC Editor, RFC 3467, February 2003.

[2] A. J. Kalafut, M. Gupta, C. A. Cole, L. Chen, and N. E. Myers, "An empirical study of orphan DNS servers in the Internet," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 308–314.

[3] "Domain names - concepts and facilities," RFC 1034, Nov. 1987. [Online]. Available: https://rfc-editor.org/rfc/rfc1034.txt

[4] P. Mockapetris, "Domain names - implementation and specification," Internet Requests for Comments, November 1987, http://www.rfc-editor.org/rfc/rfc1035.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1035.txt

[5] S. Hollenbeck, "Extensible Provisioning Protocol (EPP)," RFC 5730 (Standard), Internet Engineering Task Force, August 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5730.txt

[6] J. Jiang, J. Liang, K. Li, J. Li, H. Duan, and J. Wu, "Ghost domain names: Revoked yet still resolvable," 2012.

[7] 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, June 2016.

[8] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 10–10.

[9] "TLD DNSSEC Report," http://stats.research.icann.org/dns/tld_report/.

[10] I. Foster, "Spiderwho," Dec. 2019, original-date: 2013-01-10T23:01:02Z. [Online]. Available: https://github.com/lanrat/SpiderWho

[11] DNS-OARC, "Zone File Repository," https://www.dns-oarc.net/oarc/data/zfr.

[12] M. Larson, "[dns-operations] Upcoming DNS behavior changes to .com/.net/.edu name servers," Jan. 2010. [Online]. Available: https://lists.dns-oarc.net/pipermail/dns-operations/2010-January/004841.html

[13] A. Nyman, "Stoppa domnskojarna!" [Online]. Available: https://web.archive.org/web/20170925230114/https://www.iis.se/lardig-mer/guider/stoppa-domanskojarna/domanskojarnas-metoder/3/