DNS Single Point of Failure Detection using Transitive Availability Dependency Analysis

Florian Maury
florian.maury@gmail.com

Abstract. The Domain Name System (DNS) is one of the cornerstones of modern Internet, allowing users to access data from a distributed database, using domain names as reference keys. Data includes IP addresses of servers. DNS servers are no exception, and their names must be resolved into IP addresses, as well. The crucial difference between the name of a DNS server and, say, the name of a web server, is that one must resolve the name of a DNS server in order to query it and proceed with a user request such as "what's the address of that web server?". DNS experts advocate for various naming strategies for DNS servers, each having their own set of distinctive advantages and drawbacks. During this study, we analyzed over four million domain names of websites from the `.fr` country-code top Level Domain (ccTLD) and from Alexa top 1 million domain names, to detect single points of failure (SPOF) from DNS servers and DNS alias naming strategies, and IP address dispersion. We discovered that 83 % of the studied domain names delegated from the `.fr` ccTLD present SPOFs that could easily be avoided. We also discovered that over one domain out of 20 from Alexa top 1 Million web server domain names depend on a single IP address to work properly. In this paper, we detail our measurement methodology, break down the generating causes for SPOFs into classes of misconfigurations and provide guidance to improve the resiliency of the DNS.

1 Introduction

The DNS is a hierarchical database that is distributed on different parties using a mechanism known as delegation. DNS delegations refer queriers to DNS servers more knowledgeable about a subdomain of the domain that the queried server is responsible for. They consist of data known as NS records, which contains the names of DNS servers responsible for a branch of the DNS tree. The DNS query process is illustrated in figure 1. The resolver performs the resolution of "`www.broken-by-design.fr.` AAAA?" by iteratively querying the DNS, following delegations, starting from the root servers, and then down to `d.nic.fr.`, which is authoritative for the `.fr` zone. The process repeats itself until it finds the answer to the user query.

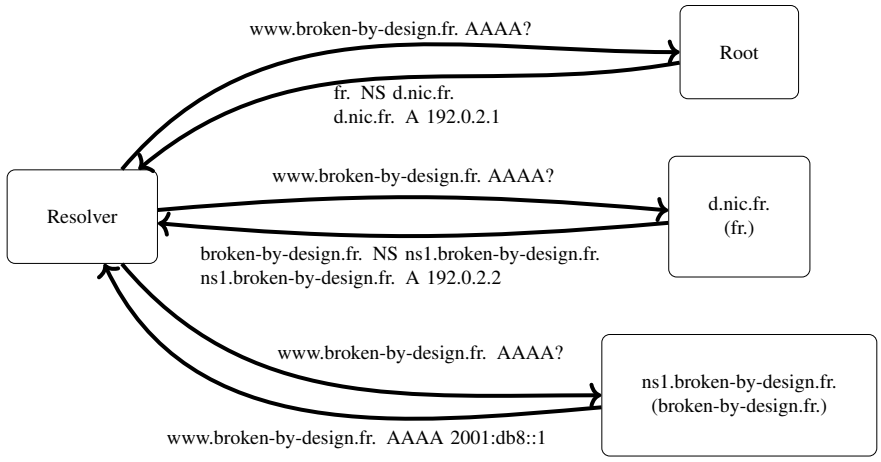

Fig. 1. Simple DNS query resolution. Delegations use glue records.

The names contained into NS records can either be names under the delegator responsibility, with respect to the tree organization of the DNS, or outside of it. In the former case, DNS experts speak of in-bailiwick domain names, while in the latter case, the domain names are said to be out-of-bailiwick. In the case of in-bailiwick domain names, special DNS records, known as glue records, are used to specify the IP addresses associated with these domain names. If those were not present, there would exist circumstances where a subdomain name would need to be resolved before their parent domain could be resolved. These so-called glue records resolve this chicken-and-egg problem. Figure 2 provides examples of both in-bailiwick and out-of-bailiwick domain names and glue records.

```
; in-bailiwick domain name
broken-by-design.fr. IN NS ns1.broken-by-design.fr.
; glue record
ns1.broken-by-design.fr. IN A 192.0.2.1

; in-bailiwick domain name from the fr. bailiwick
broken-by-design.fr. IN NS ns.example.fr.
; optional "glue record"
ns.example.fr. IN A 192.0.2.2

; out-of-bailiwick domain name
broken-by-design.fr. IN NS ns1.x-cli.eu.
```

Fig. 2. Example of NS and glue records from the fr. bailiwick.

In the case of out-of-bailiwick domain names, a DNS resolver trying to answer a user request cannot proceed without putting the user request on-hold, resolving the out-of-bailiwick domain name into IP addresses, and then resuming the previous resolution, using the obtained IP address. Figure 3 represents a resolution of a domain name involving NS records containing out-of-bailiwick domain names. The `.net` servers indicate that to resolve "`www.example.net.` A?", one must query the server named `ns1.example.com`. Thus, the resolver first resolve `ns1.example.com` into IP addresses, and then query one of them for "`www.example.net.` A?". This query procedure is to be compared with the much simpler one from figure 1, where the resolver followed delegations with glue records.

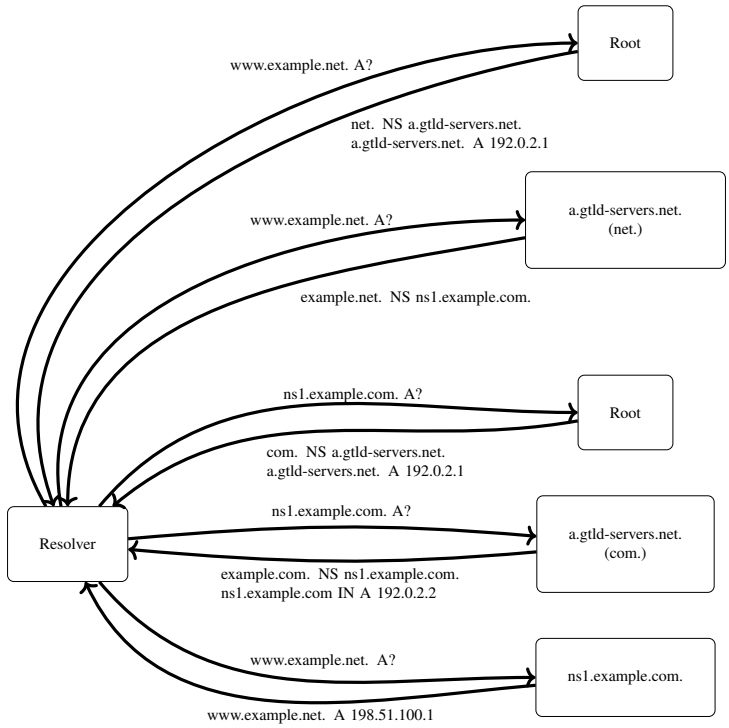

Fig. 3. DNS query resolution with delegations using out-of-bailiwick domain names.

If the DNS servers responsible for the out-of-bailiwick domain names are unavailable or compromised, the situation may result in the incapacity to answer the user request or, even worse, in providing the user with a response of the attacker's choice. In the previous example, this would occur

if `.com`, `example.com` or `ns.example.com` could not be resolved. This dependency of one domain to the proper operation of an out-of-bailiwick domain name is referred to, in the literature, as *transitive dependency*, because these dependencies can be chained (e.g. a domain A may depend on a domain B, itself depending on a domain C, and thus making C a dependency of A). Transitive dependency risks are very real; for instance, in 2015, the domain name tools.ietf.org became unavailable because all of its DNS server names were (and still are, at the time of writing) subdomains of the domain `levkowetz.com`, which remained down for several hours. Unfortunately, this transitive dependency issue is not always as easy to spot as in the tools.ietf.org example. Sometimes, the SPOF is several links down the chain of transitive dependency. Moreover, the risk may evolve over time, when administrators of names further down the chain modify their own delegations, without even realizing that their change might increase risks of down time on some remote relying parties they never heard about. The simplest example of these chained transitive dependencies is illustrated in figure 4. In that figure, a domain name A is dependent on either the domain names B or C, and both B and C are dependent on a domain name D. In that case, even if A believes that its configuration is resilient because if C breaks, B is still available (and vice versa), the domain name D is a SPOF for A, because if it becomes unavailable, it can bring down B and C simultaneously. An instance of this example can be easily imagined if D is a popular CDN platform.

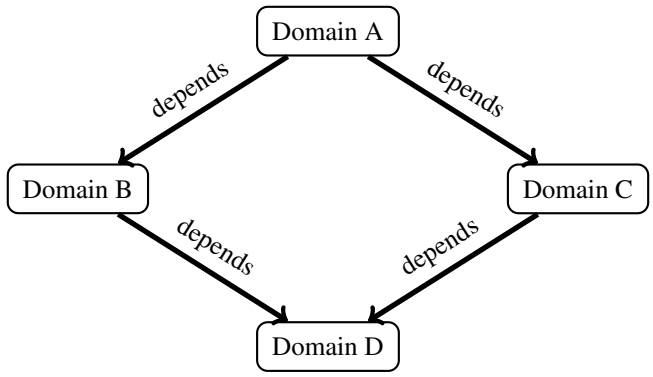

Fig. 4. Dependency graph presenting an indirect single point of failure.

Our contribution consists in assessing the risks of transitive dependency from the availability perspective. For this, we discover and build

a graph from DNS queries. Then we apply an algorithm to detect which nodes in the graph are critical and would render the target domain name unavailable if that node was unavailable. In our graph, nodes can be domain names, IP addresses, Autonomous System (AS) numbers and network prefixes covering the IP addresses of the DNS servers. To perform these measurements and analysis, we developed a specific-purpose DNS resolver, that we published as open source software under BSD license.

The remaining of this paper is organized as follows: in section 2, we compare our approach to those of previous works. Section 3 contains the details of our measurement methodology, and presents the tool we developed. Then, section 4 presents the results of our analysis of the dependency graphs of the web servers of the domains delegated from the `.fr` TLD and Alexa top 1 million web server domain names. Finally, we discuss the situation and some recommendations in section 5.

2 Previous Work

Transitive trust in the DNS is a concept that was introduced, back in 2005, by Venugopalan Ramasubramanian et al. [8]. Transitive trust dependency is the study of transitive dependency from the integrity perspective. Each out-of-bailiwick domain and each additional DNS server implicated in the resolution of a target domain name increase the attack surface of that domain. In their paper, Venugopalan Ramasubramanian et al. reported that the classic dependency graph of a domain name is generally very large, implying over 46 DNS servers on average. Any of these servers, if exploited correctly by a skilled attacker, could lead to the hijack of the target domain name.

This threat was and still is a significant concern for all domain names that are not protected with DNSSEC. Indeed, DNSSEC, a DNS extension that uses cryptographic signatures covering DNS records, is a valid countermeasure to the threat of response forgery. However, it is worth noting that DNSSEC operation is intricate and that it requires expertise, or at the very least understanding, in its inner working. This is especially true during migrations, domain transfers, key and algorithm rollovers or similarly complex procedures. As a result, failures to correctly implement and maintain DNSSEC has been observed in the wild at regular intervals [1]. Also, DNSSEC operational failures are indistinguishable from attacks: to protect users, DNS experts made the reasonable choice of failing safe, meaning that in case of operational failure or attack, the domain name being resolved is simply marked as "unavailable due to server failure". While

this makes perfect sense security-wise, the unavoidable consequence is that DNSSEC may, in some circumstances, cause DNS zones to be broken, thus affecting domain name availability and service resiliency. So DNSSEC is simultaneously a boon for security from the integrity perspective and a scourge from the availability perspective when not properly operated.

In 2010, Casey Deccio et al. explored transitive dependency from the availability perspective by developing a new model for server dependencies [5]. In their model, domain name dependency is represented as a boolean expression of prerequisites for a domain name to be available: availability of an IP address, of an AS number or of a network prefix. Their measurements used domain names drawn from traffic captures during a conference and the domain names listed in the Open Directory. They found that 6.7 % of the analyzed domain names required querying a suboptimal number of IP addresses, thus raising the risk of down time for the target domains if one of these IP addresses/servers were unavailable. The present paper presents results on transitive availability dependency, building on top of Deccio's model and methodology. The main differences are as follows: firstly, we assume DNSSEC deployment and consider the additional threat to availability that DNSSEC deployment causes; secondly, we resolve the boolean expression, setting to false leaf nodes of that expression represented as a tree, to detect possible single points of failure. Finally, one of our data sources is the complete `.fr` ccTLD. This allows us to detect and study phenomenons that are country-specific, such as those generated by popular DNS registrars in France.

The concept of graph dependency in the DNS was further studied by Eric Osterweil et al. in 2011 [6]. In their paper, they presented and discussed the trade-off between large dependency graphs to improve resiliency and the performance hit of such a practice. Our findings demonstrate that there generally exists little benefit from having a large, or even medium size, dependency graph.

The author of the present paper also published some previous results regarding transitive availability dependency in the 2015 and 2016 reports from the Observatory of the Internet Resiliency in France, an entity acting under the aegis of the ANSSI, the French network and information security agency [7]. These results were partial in that the analysis only covered the risks implied by the direct naming strategies of the NS records of subdomain names under the `.fr` TLD. An issue located further down the chain of transitive dependency was ignored. The present paper goes further, by analyzing the whole transitive dependency chain and by also searching SPOFs based on IP addresses, network prefixes, and AS numbers.

3 Measurement Methodology

3.1 Toolset Description

To discover the dependency graph of domain names and detect single points of failure, we developed our own toolset, from scratch, and published it as open source software (`https://github.com/ANSSI-FR/transdep`) under BSD license. Using Go's most popular DNS library (`https://github.com/miekg/dns`), we implemented a sort of DNS resolver. In parallel of the DNS name resolution, this resolver builds a dependency graph of the queried domain names. This graph is tree-shaped, representing a boolean expression, whose operands are DNS components or actors that may cause unavailability. These operands include DNS zones that may break in case of DNSSEC operational failure or other kind of zone-wide operational failures and IP addresses of servers that may be down, compromised or otherwise unable to answer DNS queries. From IP addresses, we extrapolate other components that may break and that we consider during our SPOF detection: network prefixes that may be hijacked using BGP or down due to operational failures, IP network versions for resolvers that are not dual-stacked (i.e. having connectivity over IPv4 and IPv6 at the same time), and Autonomous Systems (AS), which can, sometimes, though rarely, be unavailable in case of a major outage.

An example of a DNS delegation and how it is translated in our model is provided in figure 5. In this example, the transitive availability dependency of `www.example.com`, a subdomain of `example.com` residing in the `example.com` zone, is considered. It is self-evident that if the root zone, the `.com` zone or the `example.com` zone are unavailable, `www.example.com` cannot be resolved, as these zones are direct ancestors of `www.example.com` in the DNS tree. As a consequence, the full dependency graphs of the `example.com` zone, of the `.com` TLD and of the root zone are unavoidable dependencies for the correct operation `www.example.com`. The delegation information from `.com` to `example.com` is composed of four NS records, three of which are in-bailiwick, and thus have glue records. Of these glue records, two point to the same IP address, and the third one is contained within the same /24 IPv4 prefix[1] as the other glue records. As a consequence of all glues being in the same maximum length prefix, these glue records are all under the responsibility of the same AS, from the Internet connectivity standpoint. The out-of-bailiwick NS record indicates that

[1] The choice of /24 for IPv4 prefixes, and of /48 for IPv6 prefixes is based on the commonly accepted maximum length of network prefixes that can be advertised as part of prefix advertisements over BGP across independent AS.

the full dependency graph of this name must also be considered when analyzing the dependency graph of `www.example.com`.

```
example.com. IN NS ns1.example.com.
example.com. IN NS ns2.example.com.
example.com. IN NS ns3.example.com.
example.com. IN NS ns.example.net.
ns1.example.com. IN A 192.0.2.1
ns2.example.com. IN A 192.0.2.2
ns3.example.com. IN A 192.0.2.2; deliberate typo
```

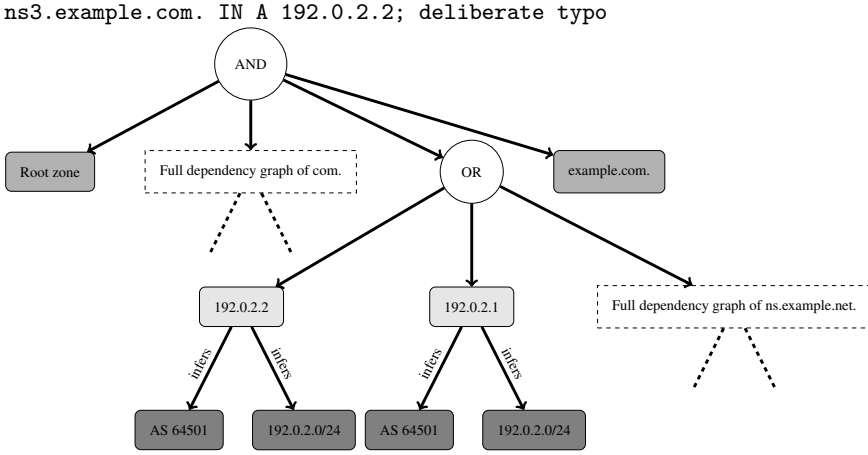

Fig. 5. Example of translation of a domain name delegation into a boolean expression.

Once the complete dependency graph of a target domain name is built, including the subgraphs built recursively for all dependencies of the dependencies of the currently considered domain name, we perform a simplification of the boolean expression that the graph represents. For this, we use the commutative and associative properties of the boolean operators to merge nested AND or OR operations, so as to limit the graph depth. We do not try to reduce the expression to its disjonctive normal form (DNF), as were doing Deccio et al. in [5]. Indeed, the DNF memory cost may be of exponential complexity, while the benefits are not clear, considering the next steps of our algorithm.

Once the dependency graph is simplified, a set of functions is generated to convert leaf nodes/operands (DNS zones and IP addresses) into boolean values. Each function is built to return false for a specific set of leaf nodes or else true. When a leaf node is converted to the false value, it means that this node is in a simulated outage state.

Here follows the list of generated functions:

- one function per unique DNS zone contained in the dependency graph.
- one function per unique IP address in the dependency graph;
- one function per unique IPv6 address in the dependency graph. These functions return false when the evaluated leaf node is an IPv4 address or when the evaluated leaf node is the IPv6 that was considered when generating that function. These functions allow the detection of SPOF when a resolver is IPv6-only.
- one function per unique IPv4 address in the dependency graph. These functions serve a purpose similar to the function set for IPv6, except the goal of this function set is to detect SPOF for resolvers that are IPv4-only.
- one function per unique maximum length prefix covering an IP address in the dependency graph.
- one function per unique AS announcing a network prefix covering an IP address in the dependency graph.

Once the function set is generated, we pick and remove a function from the set until the set is empty. Each picked function is applied on all leaf nodes, and each resulting boolean expression is evaluated. If an evaluation result is false, it means that at least one of the leaf nodes that were in a simulated outage state by the picked function was critical to the availability of the domain name whose dependency graph is being analyzed. For instance, for `www.exemple.com`, if the `.com` zone is in a simulated outage, we know that the expression will evaluate to false because it is one of the unavoidable dependencies of `www.example.com`, as previously discussed. On the other hand, the simulated outage of `.net` bears no consequence on the availability of `www.example.com`, because `.net` is part of an OR expression and both 192.0.2.1 and 192.0.2.2 are evaluated to true when the function that simulates the outage of `.net` is applied on all leaf nodes.

3.2 Data Sources

The lists of analyzed domain names were downloaded, in January 2018, from two sources: Alexa top 1 million domain names and Afnic's Open data [2], which contains the list of all domains delegated under the `.fr` ccTLD. We then prefixed all domain names from these lists with the `www` DNS label, to query for the website associated with these domains. We did so because we observed, during our preliminary tests, that many

websites use CNAMEs, a form of DNS aliasing, that must be resolved before one can get the IP addresses associated with a website. These CNAMEs generally induce dependencies on third-party domain names, such as those of CDN providers.

During the measurements, which were conducted in January 2018, we collected the NS records from the parent zones instead of the authoritative NS records from the child zone (i.e. we fetched the NS records regarding `example.com` from the `.com` zone instead of the ones in the `example.com` zones). There is three reasons for this choice, which might be otherwise regarded as doubtful from a DNS purist standpoint. The first one is that this is the standard behavior for a DNS resolver having a cold/empty cache[2]. The second one is that we observed many domains being served by so-called DNS servers that answer with "server failure" (rcode 2) or ignore the query altogether, when queried for any DNS query types other than those for IP addresses (A and AAAA). While these servers won't answer to our queries while we are searching for delegation points and similar information, the parent zone that delegated to these servers must provide NS record or else the delegation would not exist. Finally, circular dependencies may exist in some domain configurations. For instance, the authoritative NS records for the `.com` and `gtld-servers.net` NS record sets present a circular dependency: the authoritative NS records for the `gtld-servers.net` domain make use of subdomains of `nstld.com` while the `.com` authoritative NS records make use of subdomains of `gtld-servers.net`.

3.3 DNS Errors and Standard Violations

Of the one million domain names from Alexa, 120,000 domains were classified by our toolset as impossible to analyze. For the `.fr` TLD, our tool failed to analyze 476,000 domain names.

Impossibility to analyze a domain name is a verdict that is returned when a domain name dependency graph cannot be completed. In that case, the partial graph is discarded and the domain name is ignored in our statistics. This situation occurs if a DNS error is obtained in response to a query. Among these errors, we identified some patterns:

- rcode 3 (NXDOMAIN) or rcode 5 (REFUSED): the queried domain name does not exist or it is not hosted by this server. This error code

[2] Some implementations do confirm the delegation information using the authoritative answer before proceeding with them, though.

may occur for domains listed by Alexa or Afnic, because the DNS zones might have evolved between the moment these lists were generated and the time we sent our queries. This also happens in case of dangling NS records and dangling CNAME records. Indeed, Matthew Bryant discovered that many domain names, including the .io TLD, specify in various DNS records domain names that no longer exist or that are unassigned, thus leaving an opportunity for domain name hijacking [4]. Another reason to receive an NXDOMAIN error code during our dependency graph discovery is non-compliance with RFC8020, which interferes with our delegation point chasing (i.e. NS record retrieval) algorithm. RFC8020 states that the DNS is tree-shaped and that a domain name having subdomains must exist. Unfortunately, several CDN providers (including Akamai, Edgesuite, Cedexis, etc.) among other entities were found to return NXDOMAIN on empty non-terminal (ENT) DNS names, i.e. domain names having no data on their own, but having existing subdomain names. To improve the completeness of our study, we thus defined a command-line flag for our toolset that can be set to work around this RFC violation.

- rcode 2 (SERVFAIL): all DNS servers responsible for a domain name reported a server failure.
- rcode 1 (FORMERR): all DNS servers responsible for a domain name answered that they did not understand our query format. This might occur because our toolset only supports EDNS0-enabled DNS servers. If a server is not compatible with this 19 year-old standard (RFC6891), we therefore give up on querying it, and if a domain name is only served by servers that are not compatible with this DNS extension, then it is arbitrarily excluded from this study. Being compatible with DNS servers not supporting EDNS0 could be an improvement for our toolset.
- non-authoritative answers: we observed that some DNS implementations violate the DNS standard by returning non-authoritative DNS answers for CNAME records. This should not happen and answers with CNAMEs should always have the authoritative flag set. We discussed this situation with other DNS resolver implementers and were told they put up with this situation by accepting them, even though this is a clear violation of the standard. We arbitrarily decided to exclude these domains from our study.
- truncated flag set and no support for TCP: some servers could not be queried over TCP, while they answered a truncated answer. This

situation is probably caused by a misconfigured firewall or some kind of middle box such as a load balancer accepting only UDP traffic.

DNS violations can also be found into our own toolset.

First, we are in violations of RFC6672, because we do not support DNAME at the time of writing.

We also did not implement time-to-live (TTL) support within our cache. Once a DNS record is cached, we never invalidate it during a single execution of our toolset. To mitigate this issue, we ran out toolset over batches of 100,000 domain names and then cleared the cache, before the next batch. A 100,000 domain name batch took about three hours to query, so this standard violation is equivalent to rewriting DNS record TTL to three hours in the worst case scenario. We look forward to implementing TTL support in future version of this toolset.

Finally, and for completeness, we did not care for DNS answers containing enough glue records to overflow our EDNS0 buffer size of 4096 bytes. If this rare situation occurs, we only consider the returned glue records and ignore all IP addresses that were left out. While this situation may lead to false positive since some redundancy might be ignored, it is difficult to detect missing glue records accurately. Indeed, glue records are additional information, and when some are missing, the DNS truncated flag, whose purpose is to signal that the whole DNS answer could not fit the buffer, is not set.

4 Results

4.1 DNS Zone Availability

For the `.fr` ccTLD, DNS zone availability is a major concern, with 82.7 % of the studied web server domain names having at least one avoidable dependency to a DNS zone. This number is to put into perspective with previous results reported in [7], where over 99 % of DNS zones delegated from the `.fr` ccTLD were having only glueless delegations. This means that only 17 % of the studied domain names adopted a naming strategy for their glueless delegations and aliases that is resilient in case of a third-party domain name failure.

The repartition of the number of dependencies per web server domain name is provided in figure 6. For 68 % of the studied domain names, two avoidable DNS zone dependencies are detected. In general, these two names are the domain names of the DNS servers of the registrar that hosts the zone (e.g. `ovh.net`) and the TLD from which the registrar domain

name is delegated (e.g. `net`). A summary of the most frequent avoidable dependencies in naming strategies is provided in figure 7.

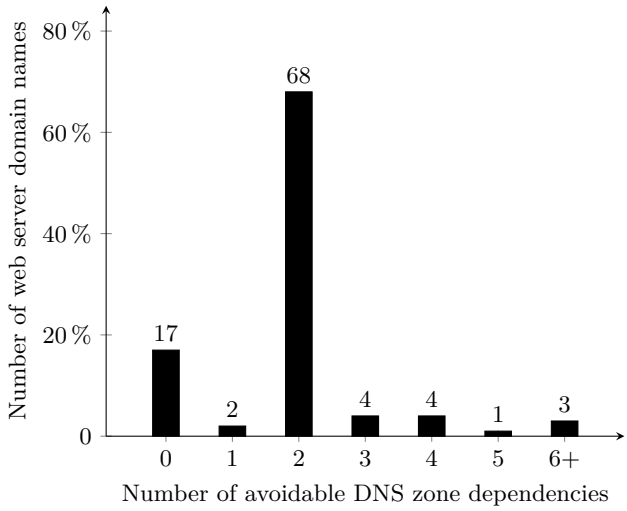

Fig. 6. Avoidable DNS zone dependency count per studied domain names under `.fr`.

Rank	.fr ccTLD	#domains	Alexa	#domains
1	net.	1,574,938	com.	192,003
2	ovh.net.	851,505	net.	150,681
3	com.	547,328	cloudflare.com.	59,767
4	gandi.net.	347,512	jp.	16,155
5	me.	119,528	domaincontrol.com.	15,716
6	anycast.me.	119,101	google.com.	14,062
7	it.	47,902	dynect.net.	11,085
8	register.it.	46,318	amazonaws.com.	10,351
9	nordnet.fr.	46,176	ovh.net.	10,109
10	amazonaws.com.	45,082	dnsv2.com.	9,420

Fig. 7. Most frequent avoidable domain name dependencies.

For Alexa top 1 million web server domain names, the naming strategies chosen by 51.5 % of the analyzed domain name introduce at least one avoidable DNS zone dependency. The repartition of the number of dependencies per domain name is represented in figure 8 and the list of domain names being dependencies are listed in figure 7.

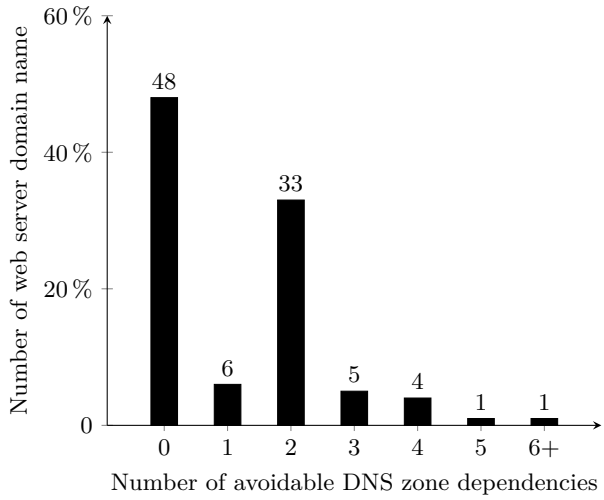

Fig. 8. Avoidable DNS zone dependency count per studied domain name from Alexa.

4.2 IP Address Availability

IP Addresses For the `.fr` ccTLD, the number of studied domain names for which an IP address is a SPOF is a bit less than 0.3 %. Several factors come in to explain this rather good result, compared to the DNS zone dependency situation. For a long time, delegation information for zones delegated from the `.fr` ccTLD were submitted to automated checks, using the ZoneCheck utility, now rewritten and known as ZoneMaster. Since the `.fr` zone content is mostly stable, as reported in [7], this means that most domains delegated from `.fr` have sane values, such as at least two NS records. Also, most of the domain names delegated from `.fr` have a very typical DNS hosting infrastructure maintained by the selling registrar and have an audience limited enough that it does not require the usage of external services, such as CDNs, and other traffic optimization engines.

The situation regarding Alexa top 1 million web server domain names is far worse, with over 5 % of these domains depending on the availability of a single IP address. This situation has no simple explanation that we can think of, as the operator hosting IP addresses with the highest count of domain names with an IP address SPOF is only responsible for serving about 500 of these domains. The exact reasons for this situation need to be investigated in future work.

IP versions For the `.fr` ccTLD, when a resolver only has IPv6 connectivity, the number of domain names that can still be resolved drops to 66.3 % of the total number of domains delegated from the `.fr` ccTLD. Of the domain names that can still be resolved, 2.8 % (or 50.500 domain names) presents an IP address SPOF, which is much higher than the number of domain names with an IP address SPOF when the resolver has connectivity to IPv4 and IPv6 simultaneously.

When the resolver only has IPv4 connectivity, the numbers are very close to those when the resolver also has connectivity to IPv6. Indeed, 99.9 % of the studied domain names can still be resolved and only 0.4 % of these domains presents an IP address SPOF.

Resolvers with only IPv6 connectivity can only resolve 41.5 % of Alexa top 1 million domain names. Of these resolvable domain names, 6.9 % (or 45,300 domain names) presents an IP address SPOF. When the resolver only has IPv4 connectivity, the ratio of resolvable domain names from Alexa is the same as for the domain names delegated from the `.fr` TLD. However, 5.3 % of these resolvable domain names presents an IP address SPOF.

4.3 Network Prefix Availability

Network prefix availability is a notion related to the risks of BGP prefix hijacking and routing advertisement pollution zone. BGP prefix hijack principle is that an attacker advertise over BGP a network prefix owned by their victim. By doing so, the attacker entices Internet routers to reroute targeted traffic that should go to the victim's network toward the attacker's. The routing advertisement pollution zone is composed of all the routers affected by the attacker's BGP advertisement and that will reroute the traffic toward the attacker's network. In case of a BGP hijack, a mitigation strategy is for the victim to make BGP advertisements that use a prefix length that is longer than the one used in the attacker's fraudulent advertisement. The victim can do so because routers generally route traffic toward the router that advertised the longest prefix length that covers a destination IP address. There is, however, a maximum prefix length that is generally enforced by Internet routers, to prevent the routing table from containing too many entries. If both the attacker and the victim make identical advertisements with the maximum prefix length, then the network traffic is generally split between the attacker and the victim. Thus, if all DNS servers reside in a single maximum length prefix, an attacker can hijack in one advertisement all traffic for all DNS servers responsible for a target domain name. While nothing prevents a deliberate hijacker

from advertising all network prefixes of a victim until all traffic is rerouted toward them, an accidental hijacker, such as a network operator making a honest mistake or typo while configuring a router may hijack one prefix, but probably not all maximum length prefixes of their "victim".

For the `.fr` ccTLD, 8.4 % of the studied domain names are dependent on the availability of a single maximum length prefix (/24 in IPv4 and /48 in IPv6) and there is a relatively high concentration on few prefixes. For Alexa web server domain name list, the results are worse, with 14.6 % of the domain names having this kind of dependency.

These results show that many DNS operators are not cognizant of the risks associated with BGP hijacks and routing advertisement pollution zone. While these notions might be considered advanced routing concerns by some, it is surprising to discover that this practice is sometimes adopted by some platform providers and prominent registrars. The list of the network prefixes that are a dependency to some of the studied domain names is provided in figure 9.

	Network Prefix	Operator Name	#domains
.fr ccTLD	81.88.63.0/24	RegisterIT	65,381
	194.206.126.0/24	Nordnet	46,138
	194.2.0.0/24	Oleane	13,900
	193.252.243.0/24	Pages Jaunes	11,590
	93.88.255.0/24	Infomaniak	5,742
Alexa	162.251.82.0/24	Public Domain Registry	4,602
	46.242.149.0/24	Loopia	2,125
	93.188.0.0/24	Loopia	716
	129.232.248.0/24	Hetzner	639
	192.185.5.0/24	Hostgator	621

Fig. 9. Top 5 of the maximum length prefixes that are a dependency.

4.4 AS Availability

Dependency to an AS is really a subject of open debate. While there are instances where a whole Autonomous System fell, generally due to internal routing incidents or major DDoS causing the infrastructure to collapse, only a scrutiny of an AS infrastructure and a network security evaluation can tell if an AS susceptible to such AS-wide incidents.

For completeness' sake, AS dependency was nonetheless studied. For the `.fr` ccTLD, 88.1 % of the studied domain name have a dependency to a single AS. For Alexa list, only 75.9 % do. This would seem to indicate that

most domain name administrators trust a single DNS hosting platform to take care of their domains. A list of AS numbers and the number of domains that are dependent to these AS is provided in figure 10.

	ASN	Operator Name	#domains
`.fr` ccTLD	16276	OVH	1,034,859
	29169	Gandi	351,325
	8560	1&1	349,300
	16509	Amazon	133,085
	39729	RegisterIT	65,849
Alexa	13335	Cloudflare	152,458
	16509	Amazon	63,233
	26496	GoDaddy	59,082
	15169	Google	23,999
	16276	OVH	23,529

Fig. 10. Top 5 of the AS that are a dependency.

5 Discussion

The results presented in the previous section indicate that for many domain names, be it popular ones from Alexa top 1 million domain names or more mundane ones from the `.fr` ccTLD, resilience engineering best current practices are not followed throughly or that indirect dependencies are created, probably without DNS operators ever knowing it.

Indirect dependencies, that is dependencies that are not immediately observable from a domain name delegation information, are of special concern, since some DNS dependency graphs are composed of tens of nodes, and sometimes up to a hundred. With the size of the dependency graph increasing, so does the difficulty of manually analyzing that graph to understand the exact level of risk affecting a domain name. As such, we challenge the tradeoff between resiliency (supposedly brought by using out-of-bailiwick domain names in NS records) and query performance that was presented in [6]. From our perspective, using out-of-bailiwick domain names can only bring both a performance hit (in case of cache miss) and potential resiliency issues, as demonstrated by our results, where only 17 % of web server domain names delegated from the `.fr` ccTLD only have unavoidable dependencies thanks to their naming strategy and their IP address distribution strategy. That is not to say that using out-of-bailiwick systematically implies risking having avoidable dependencies or

SPOFs. For instance, the second most popular registrar in France, 1&1 [2], names its DNS servers using domain names delegated from various TLDs (`1and1.biz`, `1and1.net`, etc.). Doing so does not introduce any SPOFs because of the DNS server naming strategy. However, we argue that this type of deployment is fragile if the out-of-bailiwick domain names are not under the control of a single entity, as several third parties could independently alter their configuration, unwillingly creating SPOFs for one of the domain names that depend of them.

As such, we encourage domain name holders to perform, after each delegation information change, an analysis of the dependency graph of their domain name for SPOF detection. This can be done using the tool we published as open source software, or by any other mean they see fit. We also recommend limiting the complexity of the dependency graph by using mostly in-bailiwick domain names for domain name delegation purposes, as recommended by ANSSI guide "Best Current Practices for Acquiring and Exploiting Domain Names" [3].

6 Conclusion

In this paper, we detailed a methodology for the detection of domain name single points of failure in transitive dependency graphs. We also introduced an implementation of that methodology, published as open source software. Finally, we presented our measurement results over the web server domain names delegated from the `.fr` ccTLD and over the web servers of Alexa top 1 million domain names.

Analysis of these results show that over 83 % of the studied domain names from the `.fr` ccTLD set are configured such that a SPOF exists. These SPOF mostly originate from registrars and DNS hosting platform provider infrastructure choices in their DNS server naming and IP address assignment strategies. Our analysis revealed that popular domain names from Alexa list also present numerous single points of failure, due to similar naming and IP assignment strategy issues, although the root causes are more difficult to track and cluster than with domain names from a ccTLD. Analysis of these root causes is left for future work.

Finally, we discussed some recommendations to improve DNS resiliency and automate the detection of single points of failure.

References

1. https://ianix.com/pub/dnssec-outages.html, 2018.
2. AFNIC. OpenData. https://opendata.afnic.fr/en/.
3. ANSSI. Best Current Practices for Acquiring and Exploiting Domain Names. https://www.ssi.gouv.fr/en/guide/best-current-practices-for-acquiring-and-using-domain-names/, 2014.
4. Matthew Bryant. The .io Error – Taking Control of All .io Domains With a Targeted Registration. https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/index.html, 2017.
5. Casey Deccio, Jeff Sedayao, Krishna Kant, and Prasant Mohapatra. Measuring Availability in the Domain Name System. *INFOCOM, 2010 Proceedings IEEE*, 2010.
6. Eric Osterweil, Danny McPherson, and Lixia Zhang. Operational Implications of the DNS Control Plane. 2011.
7. François Contat, Pierre Lorinquer, Florian Maury, Julie Rossi, Maxence Tury, Guillaume Valadon, and Nicolas Vivet. Internet Resilience in France. https://www.ssi.gouv.fr/uploads/2015/06/internet-resilience-in-france-report_2015_anssi.pdf, 2015.
8. "Venugopalan Ramasubramanian and Emin Gün Sirer. Perils of transitive trust in the domain name system. *IMC '05 Proceedings of the 5th ACM SIGCOMM conference on Internet measurement*, 2005.