Never say Never: Authoritative TLD nameserver-powered DNS amplification

Marios Anagnostopoulos
Singapore University of Technology and Design
Singapore
marios_a@sutd.edu.sg

Georgios Kambourakis
Dept. of Information and Communication Systems Engineering
University of the Aegean, Greece
gkamb@aegean.gr

Stefanos Gritzalis
Dept. of Information and Communication Systems Engineering
University of the Aegean, Greece
sgritz@aegean.gr

David K. Y. Yau
Singapore University of Technology and Design
Singapore
david_yau@sutd.edu.sg

***Abstract*—DNS amplification attack is a significant and persistent threat to the Internet. Authoritative name servers (ANSes) of popular domains, especially the DNSSEC-enabled ones, give attractive leverage for attackers in distributed denial-of-service (DDoS) attacks. Particularly, the ANS list of top-level domains (TLD) is publicly accessible, including by would-be attackers, in the form of a root.zone file. In this work, we examine the potential of TLD ANSes to be exploited as unknowing agents in DNS amplification attacks. Specifically, over a period of 12 months that covers two different versions of the root.zone file, we assess the amplification factor (AF) that these servers may provide to attackers when replying to both individual and multiple queries. Also, we measure the degree of actual adoption of the recommended response rate limiting (RRL) countermeasure for the ANSes. Our major findings are that (i) 70% of the distinct ANSes and 47% of the possible DNS queries for the TLDs produce a large AF that exceeds 60, (ii) 10% of the distinct ANSes reflect inbound network traffic and magnify it by a factor that exceeds 50, (iii) the number of most useful ANSes for the attacker, in terms of their role as amplifiers, appears increasing during the monitoring period, and (iv) there still exists a significant number of ANSes that do not implement the RRL or leave it inactive.**

***Index Terms*—DNS Amplification, DNSSEC, Top Level Domains, Response Rate Limiting**

I. Introduction

The Domain Name System (DNS) can be considered a cornerstone of the Internet; its operation precedes virtually any other online accesses. It comes as no surprise that the DNS is often exploited by attackers as a platform for launching powerful DNS amplification attacks [1], [2] to cause outage of specific victim servers or domains. This attack can be a particularly damaging form of distributed denial-of-service (DDoS), since it can undermine the normal operations of basic nameservers (NSes) in the DNS hierarchy. Indeed, many DNS amplification attacks have been reported to date against important services or critical infastructures. A report by Symantec [3] highlights the prevalence of DNS amplification among diverse types of DDoS attacks. A real-world example of the attack was directed against Spamhaus [4]. During a one-week period, the Spamhaus infrastructure experienced a flood of unsolicited DNS replies that reached 300 Gbps at its peak.

978-1-5386-3416-5/18/$31.00 © 2018 IEEE

While the role of DNSSEC-enabled zones in DNS amplification has received recent research attention, to the best of our knowledge, no prior work has quantified the potential of exploiting top-level domain (TLD) ANSes as either (or both) amplifiers and reflectors. From an attacker's viewpoint, the TLD servers can be especially relevant. Their identities are public knowledge, downloadable as the root.zone file [5]. Their resource records (RR) for different lookup types can also be readily parsed and interpreted along with the corresponding glue records. The root.zone file does not give a static list, but rather a dynamic one that changes on an almost daily basis. For example, the *New gTLD Program* managed by the Internet Corporation for Assigned Names and Numbers (ICANN) admits the registration of new TLDs. Since October 2013, more than 1,200 new gTLDs have been delegated and added to the root.zone file [6]. In this dynamic environment, there is always the possibility of an NS with low security standards coming on board the root.zone file.

To date, a great majority (90%) of the TLD zones have already adopted DNSSEC [7]. Paradoxically, this adoption could allow attackers to maximize the amplification effects of their actions, since the DNSSEC-related RRs are large [8]. In contrast, only about 4% of the second-level domains (SLD/2LD) are signed [7]. In general, if the attacker relies solely on SLDs, they will need to perform a zone walking to locate and extract the most profitable DNSSEC-related RRs for their purposes. The attacker may discover that at most 4% of the domains are applicable in this zone walking process. On the other hand, by targeting TLDs alone, potential attackers need not crawl the DNS hierarchy extensively to locate the DNSSEC enabled zones, since most entries in the root.zone file are already suitable. Furthermore, since the TLD zones form the pillars of the Internet, they typically host sizable computational and network bandwidth resources, supported by state-of-the-art techniques such as clustering and anycast routing, for accommodating high loads. They thus

have the promise of being powerful agents for potential DDoS perpetrators.

Practically, the fully qualified domain name (FQDN) of an TLD is often short; many of the domain names have two or three characters only. Since the queried domain names appear in the DNS requests used by attackers to initiate their attacks, these attackers can thus achieve smaller query sizes and therefore larger amplification factors (AFs), which make the attacks more powerful. More fundamentally, because the TLDs are close to the root of the DNS hierarchy, in principle they are involved in every DNS resolution no matter the lookup name. Hence, it is generally infeasible to blacklist traffic originating from a TLD ANS exploited as reflector, because we rely on data from the same server to answer many legitimate users also in benign operations. In comparison, if only open DNS resolvers/forwarders are utilized as reflectors, such as those collected by crawling the Internet [9], [10], it is relatively easier for defenders to temporarily ban traffic originating from those servers identified to be involved in an ongoing attack.

In this paper, we assess the potential of exploiting the TLD ANSes as both amplifiers and reflectors. To do so, we estimate the maximum size of a DNS response packet to a single DNS request. We perform measurements with the ANY and DNSKEY query types, since they are expected to elicit large responses. Moreover, we assess empirically the degree to which the response rate limiting (RRL) mechanism is adopted at the TLD. This mechanism is a recommended defensive measure for ANSes against exploitation attempts. We perform the assessment by dispatching a stream of consecutive DNS requests during a limited time window and observing the number of complete, truncated, and missed responses that result. We take into consideration both positive and negative answers, i.e., we issue queries for both existent and non-existent domain names (NXDOMAIN).

The contributions of our work are summarized as follows:

- We show that the existing TLD ANSes can be exploited effectively as unknowing agents in DNS amplification DDoS attacks. We present measurement results for the volume of the maliciously reflected responses to DNSSEC-related queries and calculate the corresponding AFs.
- We argue that the aforementioned effective exploitation potential can be attributed to two main reasons: (a) RFC 7766 [11] guidelines are not respected by a large majority of the ANSes, and (b) the majority of them also respond to ANY requests although they should not.
- We demonstrate that still exists a significant fraction of TLD ANSes that do not implement the RRL mechanism properly or leave it inactive; this omission allows the servers to be exploited as reflectors for attacking a third party.
- We base our analysis on large-scale data using two different versions of the root.zone file; these results cover a period from February 2016 to January 2017.

The remainder of the paper is organized as follows. The next section provides essential background for DNS amplification attacks and related countermeasures relevant to our work. Section III presents our methodology and experimental results for assessing the feasibility of using TLD ANSes as amplifiers or reflectors in attacks. Section IV summarizes our main findings and discuss their implications. Section V discusses related work, before the paper concludes.

II. Background

DNS amplification combines the characteristics of amplification and reflection in a DDoS internet attack. It aims to induce aggressive DNS reply traffic from normal DNS servers, which is then reflected to hit victim servers as targets [12]. The amplification allows the attacker to invest a small amount of query traffic but obtain a significantly larger stream of attack traffic, while the reflection obfuscates the origins of the attack and related forensic efforts. The attack's efficiency is measured by its amplification factor (AF). The greater the AF, the more powerful the attack. Two equations are commonly used to express the AF, as follows.

$$AF_1 = \frac{\text{size(response)}}{\text{size(request)}} \tag{1}$$

$$AF_2 = \frac{\text{sum of size(responses)}}{\text{sum of size(requests)}} \tag{2}$$

Equation (1) [13] gives the ratio of the size of the response to that of the corresponding request. This formula is applicable when a single request is sent to the leveraged NS in question; it is utilized in the first phase of our experiments presented in Section III-B. On the other hand, Equation (2) [12] measures the cumulative size of the responses divided by that of a stream of corresponding requests. It is applicable when a batch of requests sent to the leveraged NS, as the sizes of the corresponding responses may vary, due to say some of them being truncated or missing. Equation (2) is used in the second phase of our experiments presented in Sections III-C and III-D. Following common practice in the existing literature [1], [13] we focus on the application-layer messages only and the sizes of lower layer (UDP) packet headers are ignored in the calculations. Hence, our AF numbers are directly comparable with those in closely related work.

For accomplishing reflection, the attacker spoofs the IP source address field of request packets, making them appear to come from the victim. The spoofing is trivial for DNS that runs on UDP, since it does not require a connection back to the client. Moreover, many ISPs do not check for falsified source addresses in outbound packets; i.e., they do not comply with the guidelines of RFC 2827 (BCP 38) [14]. It is estimated that only about half (49.8%) of the measured Autonomous Systems (AS) are completely unspoofable [15]. There are thus ample "low hanging fruits" for potential DDoS attackers to choose from to best serve their goals.

Figure 1 shows how an attacker is able to exploit the TLD ANSes as reflectors (left side) or amplifiers (right side). It

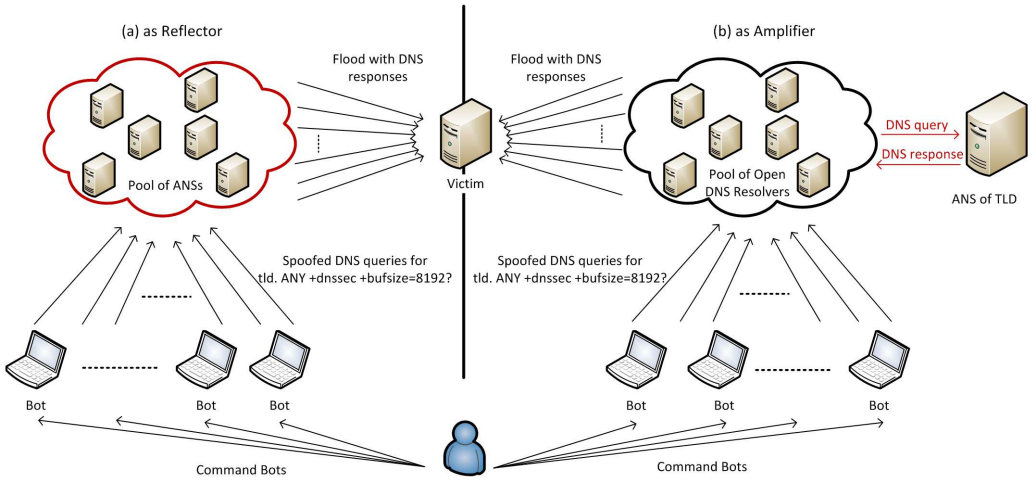

Figure 1: Exploiting ANSes administrating TLD zones

depicts two different mechanisms that share the common goal of flooding the victim with large DNS responses. Interactions with the ANSes are depicted with red lines. The attacker issues spoofed DNS requests to a pool of open DNS recursive resolvers. The queried domain name of the request belongs to the example TLD. The DNS resolvers contact with the TLD ANS, cache the response from it, and forward the response to the victim.

Currently, there are two foremost and simple countermeasures for preventing the exploitation of an NS as amplifier and/or reflector: truncated response (TC) and DNS RRL [16]. Regarding the TC countermeasure, RFC 7766 [11] recommends that if a UDP-based DNS response is larger than 512 bytes, then the responsive NS should truncate the response with a TC flag indicating that the requestor should follow up on the missing information in TCP mode. The use of TCP makes it significantly more challenging to spoof the source IP address due to the need of completing the TCP handshake. In Section III-B, we determine empirically which ANSes in the root.zone file enforce truncation. Each measurement is for an individual (unique) query, not a stream of them. This is because a stream of the queries could trigger the RRL mechanism, which mandates the truncation of some replies. On the other hand, the DNS RRL limits the number of identical responses that can be returned by an ANS to the same requestor within a given time interval. The RRL is applicable to ANSes only, since a normal flow of DNS requests to an ANS is expected to contain limited duplicate queries from the same recursive resolver. This is due to the caching facility of recursive resolvers, which allows it to store RRs locally for fulfilling subsequent requests. The DNS RRL is already implemented in BIND 9.

III. Methodology and Results

Initially, we measure the sizes of DNS replies to identify those TLD ANSes that can give sizeable answers or large AFs. Next, we investigate the adoption degree of the RRL mechanism for both cases of requests returning positive and negative answers. Our experiments span a period of 12 months and cover two different snapshots of the root.zone file [5]. The older one corresponds to serial number 2016021300 of the Start of Authority (SOA) record type, while the newer one corresponds to serial number 2017011700. In the following, the older version will be referred to as Ver. 1 and the newer as Ver. 2. These two versions are almost one year apart in time. We execute each run of our measurements twice, to get an idea of whether the investigated ANSes exhibit changing performance due to say dynamic loads or other factors. The machines executing the probes were located inside a network of our university campus.

Ver. 1 of the root.zone file contains 5,972 unique records of the NS type, accompanied by 3,150 glue records of the A type. Altogether, Ver. 1 contains 1,227 unique zones (including the root zone ".") served by 2,755 servers with distinct IPv4 addresses. Ver. 2 of the root file consists of 7,255 NS RRs and 4,311 glue A RRs. The total number of discrete zones is 1,529, served by 3,784 servers (of distinct IPv4 addresses). On average, each TLD zone is served by 4–5 ANSes. If these ANSes were all distinct, they would number 5,000 or more. Instead, however, each version has less than 4,000 servers. Hence, it is evident that some of the ANSes are authoritative for multiple zones. Another immediate observation is that the majority of the newly introduced TLDs are served by different ANSes and their total number has increased substantially, adding almost 1,000 new IPv4 addresses. That many of the NSes act as authoritative for multiple zones does not affect our results, as the RRL mechanism examines the similarity of the responses, which in this case differ by the queried domain name. In total, we extracted 5,938 and 7,231 unique tuples of <domain name, IPv4 address> respectively for the two root.zone file versions. Note that both versions contain two ANSes that only support IPv6 addressing; we excluded them from further investigation. These ANSes are found in 34 NS RRs.

A. Types of queries

To augment the impact of their actions, attackers aim to use a query that resolves to a response containing multiple sizeable RRs. From this point of view, the most attractive types of DNS queries to the attacker are ANY and DNSKEY. The first one essentially returns all the available RRs about the queried domain name. We cannot predict what types of and how many records the response will contain. The latter type (DNSKEY) retrieves the public keys of the inquired zone. As expected, a public key infrastructure is necessary for the validation of DNSSEC-related RRs. These keys add to the message sizes. Indeed, RFC 6781 [17] suggests the usage of RSA public keys with a key length of at least 1024 bits, and each zone usually possesses two keys, i.e., the Key Signing Key (KSK) and the Zone Signing Key (ZSK) [17]. An aspiring attacker may also take advantage of responses for non-existent domain names. In this case, an ANS returns a negative answer in the form of an NXDOMAIN status, or in the case of DNSSEC, it includes an NSEC/NSEC3 RR type in its answer. Typically, such a response contains up to two NSEC or up to three NSEC3 records [18], together with the corresponding RRSIG records. Nevertheless, some zones tend to resolve non-existent (random) domains with wildcard RRs. In such a case, their reply will contain a regular RR, as it is considered a positive response.

B. Response size for a single query

For each extracted tuple of <domain name, IPv4 address>, we issue a DNS query, with the DO flag enabled, to the ANS's IPv4 address directly. The query is about the matching domain name for both types of records. With the DO flag, we force the ANS to include DNSSEC-related RRs in its response. We repeat the execution three times, each of which advertising a different buffer size for the EDNS0 mechanism [19], i.e., 1,024, 8,192 or 65,535 bytes. As summarized in Table I, a great majority of the ANSes in each root.zone file replied with no error status. However, a tiny portion of them are not interesting for the purposes of our experiments; the attacker will be indifferent to them. Specifically, some ANSes returned a FormErr, ServFail, or Refused response status, meaning that they were unable to process our DNSSEC-related requests or do not admit the ANY query type. Also, it appears that nearly 0.3% of them still did not support the EDNS0 extension, so they were unable to provide responses larger than 512 bytes. Additionally, there are a number of ANSes that timed out in their response. Essentially, many of these ANSes did not respond to any of our requests during either phase of our experiments. An aspiring attacker will avoid those ANSes in execution, as they have limited amplification capabilities. However, for the sake of completeness, we include them in the subsequent analysis. Note that if an ANS fails to respond to even one of our queries, we flag it as "timeout." This means that if an ANS is not always reachable, it is deemed unsuitable for the attacker's objectives.

As mentioned in Section II, in this phase we apply the formula AF_1 to calculate the AF provided by each ANS, based on the query and response sizes. Typical DNS queries have a size of around 20–60 bytes [13]. Usually, a domain name consists of at least two labels, where the second-level label is typically longer. In our case, as we deal with TLD zones only, the inquired domain names contain one label only, which has two or three characters in most cases. Hence, the size of the query packet is smaller, and the AF is correspondingly larger. Nevertheless, the usage of EDNS0, which is mandatory in our experiments as we target DNSSEC-related RRs, adds an overhead of 11 bytes. The smallest query packet size, i.e., 28 bytes, is when we are looking up the root zone (i.e., empty label). Conversely, the largest query size of 53 bytes corresponds to a TLD name composed of 24 characters. The average query packet sizes, over distinct domain names, are 33.97 and 34.65 bytes respectively for the two versions of the root.zone file.

As expected, all the ANSes obey the limit of 1,024 bytes when the EDNS0's buffer size is set to 1,024. The majority of ANSes responded similarly (i.e., with approximately the same response size) in both runs for the ANY query, where only 6 tuples out of 7,231 exhibited a size difference larger than 500 bytes when the maximum buffer size was advertised. The corresponding size variation for the DNSKEY query type is similarly small. The largest response packet for the Ver. 2 file has a size of 7,137 bytes, containing 16 answers and 29 additional RRs in the Answer section. Note that this answer consists of nine fragmented IP packets. We can expect this ANS to provide a large AF of up to 230, as its triggering query packet has a size of 31 bytes only. This largest response was produced by an ANY type query, independent of the defined buffer size, i.e., 8,192 or 65,535 bytes. For the Ver. 1 file, we saw a response of 7,728 bytes, containing in its Answer section 32 answers and 37 additional RRs. This response was carried in six fragmented IP packets, and produces an AF of up to 249. However, the corresponding server ceased to operate as a TLD ANS since Ver. 1, and it is no longer present in the Ver. 2 file.

The AF calculated from the responses of the TLD ANSes are given in Table II. We choose to present the results for the queries with a buffer size of 8,192 only, since the difference between the two runs (8,192 vs. 65,535) is insignificant and a potential attacker will probably prefer to advertise a small buffer size to avoid suspicions. In examining the AF in the case of negative responses, we select to query for a domain name with a 5 character random string as a second label. Doing so results in a random domain name that is nonexistent. In total, the size of the query packet increases by 6 bytes, i.e., 5 bytes for the random string and 1 byte for the extra label. The AFs of the ANY query type for negative responses (i.e., ANY NSEC) for both versions of the root.zone file are shown in Table II as well.

C. Examining RRL mechanism for positive responses

In the context of this work, we are mostly concerned about the following parameters of the RRL mechanism as it is implemanted on BIND: *window*, *responses-per-second*,

	Ver. 1	Ver. 2
NoError	2,656 (96.41%)	3,654 (96.56%)
NoEDNS0	10 (0.36%)	11 (0.29%)
FormError, ServFail or Refused	14 (0.51%)	11 (0.29%)
TimeOut	75 (2.72%)	108 (2.85%)
Total	2,755 (100%)	3,784 (100%)

Table I: Demographics of the TLD ANSes.

Amplif. Factor	Ver. 1			Ver. 2		
	ANY	DNSKEY	ANY NSEC	ANY	DNSKEY	ANY NSEC
<=20 or TimeOut	1,173 (19.75%)	1,044 (17.58%)	2,976 (50.12%)	1,132 (15.65%)	1,079 (14.92%)	4,359 (60.28%)
20−40	367 (6.18%)	4,254 (71.64%)	2,797 (47.10%)	550 (7.61%)	3,774 (52.19%)	2,679 (37.05%)
40−60	2,489 (41.92%)	618 (10.41%)	137 (2.31%)	2,118 (29.29%)	2,251 (31.13%)	177 (2.45%)
60−80	1,265 (21.30%)	18 (0.30%)	28 (0.47%)	1,880 (26.00%)	125 (1.73%)	8 (0.11%)
80−100	395 (6.65%)	2 (0.03%)	0 (0.00%)	1,114 (15.41%)	0 (0.00%)	8 (0.11%)
100−150	241 (4.06%)	2 (0.03%)	0 (0.00%)	428 (5.92%)	2 (0.03%)	0 (0.00%)
>150	8 (0.13%)	0 (0.00%)	0 (0.00%)	9 (0.12%)	0 (0.00%)	0 (0.00%)
Total	5,938 (100%)			7,231 (100%)		

Table II: Amplification factor for a single query with EDNS0 buffer size 8,192 bytes.

nxdomain-per-second, and *slip* [16], [20]. The *window* parameter with a default value of 15 seconds designates the time period over which the rate limiting is calculated and during which any excesses are stored. The *responses-per-second* parameter determines the maximum number of times that the same response can be returned to the same requestor. The *nxdomain-per-second* defines the maximum number of times that the same negative response will be returned to the same requestor independently of the queried domain name. Finally, *slip* specifies the rate at which successive identical requests will be answered with a truncated response. The default value for this parameter is two, meaning that every other identical response gets truncated.

To investigate the adoption of the RRL mechanism [16], we dispatched a stream of 10,000 DNS queries from the same source IP address within a time window of 65–75 seconds. Then, we calculated the success ratio of this effort as a percentage of the returned answers versus the total number of submitted queries (in our case 10,000). We conducted this experiment twice to see possible variations due to dynamic workload and congestion at the ANS side. Special care was taken to minimize the burden imposed by our experiments on the examined ANSes. For example, in order to minimize the window of the request bursts, we do not query the same ANS consecutively for a different domain name under its administration. For the same reason, we examine the root ANSes only once. In any case, as the query volume received by the TLD ANSes is huge, e.g., the A-root NS receives on average more than 3.5 billion IPv4 UDP queries per day [21], we believe that the magnitude of our requests (10,000 per experiment) is small compared with that baseline. Very similar is the traffic volume for the remaining root servers, which also publicize their usage statistics. It should be noted that some of the responses bear erroneous response codes (REFUSED or SERVFAIL, or even ICMP-related conditions indicating that the host/port is unreachable). However, as we are solely interested in the size of the response, rather than its status per se, we include them in the analysis. These erroneous responses have the effect of diminishing the overall AF, since their packet size usually does not exceed 50–60 bytes.

During the execution of this phase, for Ver. 1, we observe that nearly 62% of the examined tuples (3,679 out of 5,938) exhibit a steady performance giving a difference not exceeding 1% in terms of the success ratio. Moreover, most of them (90.1% or 5,347 out of 5,938) show a variation of no more than 10%, but there exists a small portion (2.2% or 128 out of 5,938) that exhibits a larger variation between the two runs (50%–99%). Regarding Ver. 2, we see that more than 45.8% of the tuples (3,313 out of 7,231) display a negligible difference (no more than 1%) in the success ratio of positive responses. Furthermore, the great majority (86.1% or 6,224 out of 7,231) demonstrate a variation of no more than 10%. The small variations evidence that the corresponding ANSes perform as reliable reflectors. Nevertheless, a small percentage (6.3% or 454 out of 7,231) has a variation higher than 50% between the two runs. We infer that the performance of the specific ANSes can be more highly influenced by changing workload or implemented countermeasures.

It should be noted that the most useful reflector to the attacker exhibits a success ratio that exceeds 96% (97.26% and 96.55% in the two runs respectively). Precisely, its response has a size of 4,011 bytes while the triggering query is 32 bytes long. Hence, the specific ANS reflects the ingress malicious traffic after multiplying its volume by a factor of 121. It is worrying from the security point of view that the particular server is authoritative for 37 distinct TLDs, as their corresponding glue records contain its unique IP address, but with different hostnames. In most cases, this server behaved beneficially for the attacker, as it achieved a success ratio of over 90%, with large responses of around 4,000 bytes. The

aforementioned observations correspond to Ver. 2, whereas in Ver. 1 the same server administered only 5 domains, exhibiting however similar "attacker friendly" behavior.

Tables III and IV summarize the results for the ANSes that show a performance variation of less than 10%. In total, the tables include 90.1% and 86.1% of unique <domain name, IPv4 address> tuples extracted respectively from Ver. 1 and Ver. 2. A conservative attacker might consider only ANSes that show consistent performance. Each table displays the success ratio of the ANSes, meaning the number of successfully received responses to the total number of queries, along with the corresponding AF calculated over the successful queries only. This way, the reader can see the sizes of the actually received responses. To facilitate interpretation of the tables, we circle the actual cumulative AF, after considering the volume of all the submitted queries, for the most profitable ANSes to the attacker.

D. Examining RRL mechanism for negative responses

An insidious attacker might reason that it could evade the RRL mechanism by continually looking up random domain names. Since the corresponding responses will probably contain different records of the NSEC/NSEC3 type, the threshold of the RRL mechanism would not be triggered as often as in the case of positive responses, whose content will be identical for all the queries. To assess the validity of this thinking, similarly to the previous phase, we executed a burst of 10,000 queries, each time with a five character random string as the second label of the queried domain name. Then, we recalculated the success ratio for the negative responses.

For Ver. 1, we observe that about 76% of the ANSes (4,498 out of 5,938) display a negligible difference, whereas 96.3% (5,718 out of 5,938) have a rather expected variation of 10%. Merely 0.9% or 57 out of 5,938 show a significant difference that exceeds 50%. Regarding Ver. 2, we notice that nearly 38% of the ANSes demonstrate a difference of no more than 1% (2,723 out of 7,231) in the success ratio of negative responses. Additionally, the majority of them (91.3% or 6,605 out of 7,231) exhibit a variation that does not exceed 10%, whereas only 4.7% (337 out of 7,231) show a high disparity between the two runs. Tables V and VI summarize the results for the ANSes that have a variation of less than 10% between the two runs of this phase. Similar to the case of positive responses, we include 96.3% and 91.3% of unique <domain name, IPv4 address> tuples respectively for each version of the root file.

IV. Lessons learned

Looking at Table II, we can assert that the ANY type gives the most profitable kind of RRs for a potential attacker amongst the examined query types. The most significant observation is that in Ver. 2 nearly 70% (i.e., 2,634 out of 3,784) of the distinct ANSes yield an AF of 60 or more for at least one of the examined query types given the advertised buffer size. Moreover, there exist 7% of the ANSes (i.e., 280 out of 3,784) that provide an AF of at least 100, which can be considered severe from an amplification point of view. The corresponding percentages for Ver. 1 are respectively 48% and 7% (i.e., 1,327 and 203 out of 2,755). Furthermore, almost half of the possible tuples (47% of 3,431 out of 7,231) in Ver. 2 generate an AF greater than 60 and nearly 6% greater than 100. The corresponding percentages are 32% and 4% respectively for Ver. 1. Hence, the number of attractive ANSes in terms of their role as amplifiers has increased during the monitoring period. This observation is also corroborated by the information in an older root.zone file of serial number 2015090700, which we examined tentatively at the beginning of our experiments. These results suggest that an aspiring attacker could choose practically at random which domain names to query, and then almost half of them would be beneficial to their purposes.

Another interesting discovery is that only five out of the 13 permanent root ANSes, as well as 53 out of the 2,742 (from Ver. 1) and 58 out of the 3,771 (from Ver. 2) remaining ANSes replied with TC responses. That is, considering both versions of the root.zone file, the TC bit was enabled in only 464 of the 5,938 and 434 of the 7,231 responses respectively, when we issued a single request. As expected, the sizes of the responses that had the TC bit off were significantly high. Indeed, on average, these responses were 1,870 and 2,129 bytes long, respectively, for the case of the ANY query type and 8,192 buffer size. Interestingly, one can see that the more recent version (Ver. 2) exhibited worse behaviour from a defender's viewpoint, in that the newly introduced ANSes in that version gave a sizeable response. The results given in Tables III and IV, in comparison with those in Tables V and VI, verify the claim that the accomplished AF is higher in the case of positive responses. Nevertheless, the success ratios due to the RRL mechanism are similar in both cases, which in turn implies that the relevant parameters (i.e., *responses-per-second* and *nxdomain-per-second*) have similar values.

In a nutshell, our observations support strongly the view that the TLD ANSes give attractive leverage to attackers for launching DNS amplification attacks. Depending on specific intention of the perpetrators and the resources available to them, they could readily identify specific existing ANSes to best suit their purposes. In the case that the attacker already possesses a pool of reflectors, such as open resolvers or open forwarders [22], they might employ only those ANSes that provide a high AF. Since typically the resolvers and the forwarders provide DNS caching, it will be hard for the involved ANSes to realize that they are taking part in such an attack. To illustrate this point, the 5th column of Table II shows that a considerable portion (more than 47%) of DNS queries for TLD would be fruitful in such an attack scenario, contributing an AF that exceeds 60.

On the other hand, if the attacker lacks reflectors, they may enlist those ANSes that exhibit a high success ratio. There is a substantial fraction of the ANSes that responded to nearly all the queries and induced an AF of significant magnitude. Specifically, we notice that 362 ANSes out of the total 3,784 (9.6%) serving the TLDs for Ver. 2 reflected the inbound query traffic after magnifying its volume by a factor that exceeds 50, independent of whether the query was positive or negative. The

		Amplification Factor				
		<=20	20−40	40−60	>60	**TOTAL**
Success Ratio	<=70%	676 (12.64%)	11 (0.21%)	91 (1.70%)	151 (2.82%)	929(17.37%)
	70%−80%	3 (0.06%)	1 (0.02%)	(41) 14 (0.26%)	(53) 43 (0.80%)	61(1.14%)
	80%−90%	3 (0.06%)	0 (0.00%)	(52) 1 (0.02%)	(58) 6 (0.11%)	10(0.19%)
	90%−100%	898 (16.79%)	(38) 348 (6.51%)	(50) 2,074 (38.79%)	(75) 1,027(19.21%)	4,374(81.30%)
	TOTAL	1,580 (29.55%)	360 (6.73%)	2,180 (40.77%)	1,227(22.95%)	5,347 (100.00%)

Table III: Percentage of ANSes for positive responses (Ver. 1).

		Amplification Factor				
		<=20	20−40	40−60	>60	**TOTAL**
Success Ratio	<=70%	2,652 (42.61%)	100 (1.61%)	335 (5.38%)	1,006 (16.16%)	4,093 (65.76%)
	70%−80%	3 (0.05%)	4 (0.06%)	5 (0.08%)	(59) 23 (0.37%)	35 (0.56%)
	80%−90%	12 (0.19%)	3 (0.05%)	(44) 10 (0.16%)	(66) 22 (0.35%)	47 (0.76%)
	90%−100%	863 (13.87%)	(36) 381(6.12%)	(43) 461 (7.41%)	(81) 344 (5.53%)	2,049 (32.92%)
	TOTAL	3,530 (56.72%)	488 (7.84%)	811 (13.03%)	1,395 (22.41%)	6,224 (100.00%)

Table IV: Percentage of ANSes for positive responses (Ver. 2).

		Amplification Factor				
		<=20	20−40	40−60	>60	**TOTAL**
Success Ratio	<=70%	924 (16.16%)	277 (4.84%)	3 (0.05%)	0 (0.00%)	1,204 (21.06%)
	70%−80%	30 (0.52%)	3 (0.05%)	0 (0.00%)	0 (0.00%)	33 (0.58%)
	80%−90%	495 (8.66%)	235 (4.11%)	(35) 100 (1.75%)	0 (0.00%)	830 (14.52%)
	90%−100%	1,486 (25.99%)	(27) 2,106 (36.83%)	(46) 37 (0.65%)	(69) 22 (0.38%)	3,651 (63.85%)
	TOTAL	2,935 (51.33%)	2,621 (45.84%)	140 (2.45%)	22 (0.38%)	5,717 (100.00%)

Table V: Percentage of ANSes of negative responses (Ver. 1).

		Amplification Factor				
		<=20	20−40	40−60	>60	**TOTAL**
Success Ratio	<=70%	2,615 (39.59%)	276 (4.18%)	4 (0.06%)	1 (0.01%)	2,896(43.85%)
	70%−80%	31 (0.47%)	15 (0.23%)	0 (0.00%)	0 (0.00%)	46 (0.70%)
	80%−90%	147 (2.23%)	47 (0.71%)	(36) 2 (0.03%)	0 (0.00%)	196 (2.97%)
	90%−100%	2,437 (36.90%)	(26) 984 (14.90%)	(42) 39 (0.59%)	(61) 7 (0.11%)	3,467 (52.49%)
	TOTAL	5,230 (79.18%)	1,322 (20.02%)	45 (0.68%)	8 (0.12%)	6,605 (100.00%)

Table VI: Percentage of ANSes of negative responses (Ver. 2).

corresponding extent for Ver. 1 is 1,025 out of the total 2,755 ANSes (37.2%). We observe that the percentage of reflecting ANSes has decreased from the older to the newer zone file, indicating that the number of ANSes adopting RRL or similar countermeasures is increasing. The aforementioned ANSes are only those that showed steady performance between the two runs. There are other ANSes that did not produce high success ratios in both the runs.

Deployment of the TC countermeasure [11] will significantly reduce the overall AF. We can observe that the average message size decreases inversely with the percentage of truncated responses. Essentially, this countermeasure diminishes the AF to nearly 10, when almost every response is truncated. Hence, it limits the damage of potential attacks, even if a high success response ratio is preserved. Therefore, it is preferable to enforce TC rather than drop incoming queries that look suspicious, because the latter countermeasure may hamper legitimate clients from receiving their required answers, which may in turn drive them to cope by increasing their DNS query rates. A complementary section that elaborates on possible countermeasures is accessible at [23].

V. Related Work

In the first documented study of the DNS amplification attack, Vaughn and Evron [24] conclude that perpetrators tend to exploit large TXT RRs for amplifying their attacking network traffic and open resolvers to reflect the traffic towards the target. Specifically, an attacker might place a sizeable TXT RR of about 4 KB in the DNS hierarchy, and repeatedly send spoofed requests for the specific RR to numerous open resolvers. The authors report an achieved AF of 60. From the initial announcement of DNSSEC-bis [25]–[27], there have been concerns that its deployment may facilitate aspiring aggressors in mounting forceful DNS amplification attacks due to the increased record size. For example, Ariyapperuma and Mitchell [28] express skepticism about the size of a DNSSEC response. The first comprehensive study of DNS amplification attack involving DNSSEC-related RRs is given in [22]. The authors take advantage of the increased size of DNSSEC-related RRs for amplifying DDoS ramifications, along with a vast number of open DNS forwarders existing in the wild as reflectors. Specifically, they repeatedly dispatched a stream of DNSSEC-related requests with spoofed source IP address towards a pool of known (precompiled) open forwarders. During the conducted experiments, the authors observed a maximum AF of 44. They also highlight the observation that the aspiring aggressors may utilize existing available resources, namely DNSSEC-signed zones and open DNS forwarder/recursive NS for accomplishing the DDoS, without the need to register or install an NS of their own. This way, forensic evidence of the attack is significantly obscured.

More recently, Rijswijk-Deij et al. [13] affirm that DNSSEC-related RRs can be exploited to augment the AF of a DNS amplification attack. Specifically, they calculate the AF produced by almost 2.5 million DNSSEC-signed zones under six major TLDs. They conclude that the ANY query may generate an AF of 47 on average, while the worst case has an AF of almost 179. Also, Rossow [1] has evaluated the potential of exploiting 14 UDP-based network protocols including DNSSEC for amplification attacks. From a sample of 1,404 ANSes crawled from the IP space, they calculate that the top 10% of them provide an AF of nearly 98 for the ANY query.

For reflecting their attacking traffic, aggressors mainly take advantage of open DNS recursive resolvers. The work of Dagon et al. [29] is the first that calls attention to the issue of open resolvers. The authors found that almost 10.5M devices functioned globally as open resolvers at each run of their probes. In 2010, the initiative of Open Resolver Project began its operation with the purpose of locating and shutting down open recursive resolvers. Initially, it was evident that nearly 30M devices functioned as open resolvers. After their efforts to eliminate open recursion, the Open Resolver Project recently recorded about 17.5M open resolvers at each probe [9]. These numbers have been further verified by the DNS Factory Measurement [30].

Our work in this paper is most closely related to [1], [13]. However, our methodology does not deploy zone walking nor require collaboration with a zone's administrator for extracting domain zones and their matching ANSes. Instead, we take advantage of publicly available data, namely the root.zone file, and focus our research on TLDs instead of SLD/2LD, thereby complementing the earlier work. Furthermore, we investigate the reflection capabilities of the examined ANSes, besides their role as amplifier. In comparison with the findings in [13] quantitatively, our results based on all the possible 7,231 DNS tuples (cf. Section III-B) demonstrate a cumulative AF of 57, with the worst case exceeding 230, whereas they find an AF of 47, based on a larger set of ANSes (about 2.5M). Also, by our experiments, the 10% most profitable tuples generate an AF of 109, whereas Rossow [1] observes an AF of nearly 98 from the top 10% of the 1,404 samples of ANSes. Note that both [1] and [13] calculate the AF based on the DNS packet size only, and we adopt the same approach to get directly comparable results.

VI. Conclusion

ANSes responsible for resolving top-level domain names are essential for nearly every Internet transaction. From a security point of view, these servers prove also to be a tempting target for aggressors determined to exploit them as amplifiers or reflectors in launching (D)DoS attacks. Against this background, the main contribution of this paper is a comprehensive empirical investigation of TLD ANSes acting as unknowing agents in these attacks. By examining key issues like response sizes for the ANY query type and current adoption of the RRL and TC mechanisms, we show that, despite prior corrective efforts, there remains a significant portion of the TLD ANSes that are quite beneficial to potential DDoS attackers. Particularly, we discovered that more than 47% of the queries for distinct <domain name, IPv4 address> tuples provide an alarming AF higher than 60. Furthermore, we found that 10% of the unique TLD ANSes would reflect ingress attack network traffic and multiply its volume by a factor of more than 50.

In this work, we combined the NS and A resource records to map each TLD to the IPv4 addresses of the corresponding ANSes. However, due to anycast, each IP address may correspond to different computer clusters and/or geographical regions. It is therefore interesting to interpret the measurements by clusters, which will require the measurements to be obtained from various vantage points worldwide.

References

[1] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *In Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS)*, 2014.

[2] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "Detecting DNS amplification attacks," in *Critical Information Infrastructures Security*. Springer, 2008, pp. 185–196.

[3] Symantec, "Internet Security Threat Report (ISTR20)," vol. 20, 2015.

[4] R. Lemos. (2013, March) Largest-Ever DDoS Campaign Demonstrates Danger of New Attack Method. eWeek. http://www.eweek.com/security/largest-ever-ddos-campaign-demonstrates-danger-of-new-attack-method.

[5] Internet Assigned Numbers Authority (IANA). Root files. https://www.iana.org/domains/root/files.

[6] New Generic Top-Level Domains. https://newgtlds.icann.org/.

[7] R. Lamb. Dnssec deployment report. [Online]. Available: https://rick.eng.br/dnssecstat/

[8] M. Anagnostopoulos, G. Kambourakis, E. Konstantinou, and S. Gritzalis, *DNSSEC vs. DNSCurve: A Side-by-Side Comparison*. IGI Global, 2012, p. 201.

[9] Open Resolver Project. https://openresolverproject.org.

[10] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 111–125.

[11] J. Dickinson, S. Dickinson, R. Bellis, A. Mankin, and D. Wessels. (2016) RFC 7766: DNS Transport over TCP - Implementation Requirements. https://tools.ietf.org/html/rfc7766.

[12] F. J. Ryba, M. Orlinski, M. Wählisch, C. Rossow, and T. C. Schmidt, "Amplification and DRDoS Attack Defense–A Survey and New Perspectives," *arXiv preprint arXiv:1505.07892*, 2015. [Online]. Available: http://arxiv.org/abs/1505.07892

[13] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks: A Comprehensive Measurement Study," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 449–460.

[14] P. Ferguson and D. Senie, "RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," 2000, http://www.ietf.org/rfc/rfc2827.txt.

[15] Spoofer Project: State of IP Spoofing. https://spoofer.caida.org/summary.php.

[16] P. Vixie and V. Schryver. (2012, June) DNS Response Rate Limiting (DNS RRL). ISC. http://ss.vix.su/ vixie/isc-tn-2012-1.txt.

[17] O. Kolkman, M. Mekking, and M. Gieben. (2012) RFC 6781: DNSSEC Operational Practices, Version 2. https://tools.ietf.org/html/rfc6781.

[18] M. Gieben and M. Mekking. (2014) RFC 7129: Authenticated Denial of Existence in the DNS. https://tools.ietf.org/html/rfc7129.

[19] J. Damas and P. Vixie. (2013) RFC 6891: Extension Mechanisms for DNS (EDNS0). https://tools.ietf.org/html/rfc6891.

[20] Internet System Consortium, *BIND 9 Administrator Reference Manual*, 2014.

[21] VeriSign. (2017) A-root Query Volume (Millions/Day). http://a.root-servers.org/metrics/index.html.

[22] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "DNS Amplification Attack Revisited," *Computers & Security*, vol. 39, Part B, pp. 475 – 485, 2013.

[23] http://goo.gl/BNRWXR.

[24] R. Vaughn and G. Evron. (2006) DNS amplification attacks (Preliminary Release). http://crt.io/DNS-Amplification-Attacks.pdf.

[25] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "RFC 4033: DNS security introduction and requirements," 2005, http://www.ietf.org/rfc/rfc4033.txt.

[26] ——, "RFC 4034: Resource records for the DNS security extensions," 2005, http://www.ietf.org/rfc/rfc4034.txt.

[27] ——, "RFC 4035: Protocol modifications for the DNS security extensions," 2005, http://www.ietf.org/rfc/rfc4035.txt.

[28] S. Ariyapperuma and C. J. Mitchell, "Security vulnerabilities in DNS and DNSSEC," in *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*. IEEE, 2007, pp. 335–342.

[29] D. Dagon, N. Provos, C. P. Lee, and W. Lee, "Corrupted DNS resolution paths: The rise of a malicious resolution authority," in *Proceedings of Network and Distributed Security Symposium (NDSS08)*, 2008.

[30] DNS - The Measurement Factory. http://dns.measurement-factory.com.