ANYway: Measuring the Amplification DDoS Potential of Domains

Olivier van der Toorn*, Johannes Krupp†, Mattijs Jonker*, Roland van Rijswijk-Deij*,
Christian Rossow†, Anna Sperotto*
*University of Twente
{o.i.vandertoorn, m.jonker, r.m.vanrijswijk, a.sperotto}@utwente.nl
†CISPA Helmholtz Center for Information Security
{johannes.krupp, rossow}@cispa.de

***Abstract*—DDoS attacks threaten Internet security and stability, with attacks reaching the Tbps range. A popular approach involves DNS-based reflection and amplification, a type of attack in which a domain name, known to return a large answer, is queried using spoofed requests. Do the chosen names offer the largest amplification, however, or have we yet to see the full amplification potential? And while operational countermeasures are proposed, chiefly limiting responses to 'ANY' queries, up to what point will these countermeasures be effective?**

In this paper we make three main contributions. First, we propose and validate a scalable method to estimate the amplification potential of a domain name, based on the expected ANY response size. Second, we create estimates for hundreds of millions of domain names and rank them by their amplification potential. By comparing the overall ranking to the set of domains observed in actual attacks in honeypot data, we show whether attackers are using the most-potent domains for their attacks, or if we may expect larger attacks in the future. Finally, we evaluate the effectiveness of blocking ANY queries, as proposed by the IETF, to limit DNS-based DDoS attacks, by estimating the decrease in attack volume when switching from ANY to other query types.

Our results show that by blocking ANY, the response size of domains observed in attacks can be reduced by 57%, and the size of most-potent domains decreases by 69%. However, we also show that dropping ANY is not an absolute solution to DNS-based DDoS, as a small but potent portion of domains remain leading to an expected response size of over 2,048 bytes to queries other than ANY.

I. Introduction

Distributed Denial of Service (DDoS) attacks have become an everyday threat, leading to mere frustration to severe loss of business revenue. Amplification DDoS remains *the* most powerful tool to ramp up the attack volumes. To this end, attackers abuse UDP-based Internet protocols and spoof request such that public services unwittingly flood an attacker-chosen target with inadvertent and large responses. The sheer traffic volume achieved by amplification DDoS is staggering: with a mixture of protocols, the largest attack recorded to date reached 2.54 Tbps [1], taking offline core Internet services for billions of users worldwide.

There are over a dozen protocols that attackers can abuse for amplification [2]. Whereas most protocols have "fixable" amplification vectors that ultimately will disappear, this is not trivial for the Domain Name System (DNS). By design, DNS replies are typically larger than requests, fueled by additional records sent along such as glue records or cryptographic signatures. Not surprisingly, the DNS protocol alone has thus resulted in amplification attacks with peaks of 363 Gbps [3]. By using 'ANY' queries – a pseudo query type, for which DNS resolvers combine all available record types together in the answer – attackers can typically achieve a much larger response with a relatively small query, thus achieving a large amplification factor. The amplification factor can further increase in the case of DNSSEC queries, as DNSSEC adds signature data to DNS responses, making it attractive to abuse in DDoS attacks [4]. However, a scalable approach to evaluate the overall potential of DNS-based DDoS attacks is still missing. Consequently, it remains an open question if network operators, among others, have yet to be faced with the full potential of DNS-based DDoS attacks.

The DNS community has proposed several countermeasures to mitigate the amplification potential of DNS. A countermeasure that has received quite some attention is the proposal to minimize responses to 'ANY' queries, as detailed in RFC 8482 [5]. The prominent provider Cloudflare explains that this will help in multiple areas, including reducing traffic, but chiefly against DDoS attacks [6]. This, however, raises the question: up to what point does limiting 'ANY' solve the DNS-based DDoS problem.

The main contributions of this paper are that we:

- propose and validate a method to estimate the response size to 'ANY' queries, based on large-scale active DNS measurement data covering over 65% of the global namespace;
- compare and rank the expected response sizes of domains observed in attacks to domains from our dataset, to investigate if DNS-based DDoS attacks can become worse in the future;
- perform a quantitative analysis of the consequence of limiting 'ANY' queries as proposed in RFC 8482 [5].

The remainder of this paper is organized as follows. In Section II, we discuss background information and related work. In Section III, we discuss the datasets we used in this paper. In Section IV, we discuss our methodology for estimating 'ANY' queries and validate our method. Section V discusses the ranking we made using our estimations of the

978-3-903176-36-2 ©2021 IFIP

'ANY' response size. In Section VI, we study the effect of dropping 'ANY' queries. Finally, we discuss operational considerations in Section VII and conclude in Section VIII.

II. Background & Related work

In this section, we provide a brief introduction to DDoS attacks and DNS, particularly focusing on how both come together in DNS-based amplification DDoS attacks.

A. DNS

The DNS is a core Internet component. It can be seen as the Internet's *phonebook* as it allows, among others, domain names to be resolved to IP addresses. Domain name operators can publish various types of resource records (RR) in the DNS zone of their domain, for example A records (IP addresses), MX records (mail exchanger for inbound e-mail), and TXT records, which are 'freetext' of variable length and are used for a variety of purposes [7]. Records are held by the *name server* that is *authoritative* for a given domain name.

Typically, when a DNS client wishes to retrieve information from the DNS, it sends a query for a specific RR type (e.g., A) to a *recursive resolver*, either directly or through a *stub resolver*. When a client wishes to learn all RRs in a zone, or as many as possible, it may be able to use an ANY query. However, the response to an ANY query is dependent on the resolver software and the state of its cache. A resolver may decide to return a partial answer (i.e., RR subset) from cache rather than retrieve a full answer from the authoritative name server. It is important to note that when a specific (and existing) RR is queried for, the DNS response is typically relatively larger than the request, because the answer carries record data. This applies even more to ANY queries, because multiple record types may be included in the response. As we will explain later, this makes the DNS attractive for attackers.

The original specification of the DNS [8] indicated a maximum response size of 512 bytes. The introduction of DNSSEC [9] – adding integrity DNS answer through cryptographic signing – required the DNS to support larger answers, as DNSSEC signed answers contain signatures to validate the answer. EDNS [10] enabled responses larger than 512 bytes, while theoretically limited to the maximum UDP payload size, 4,096 bytes is typically used as an initial value for the maximum size of the answer. Support for EDNS is signalled through an pseudo record ('OPT') in the query.

B. DDoS Attacks

With Denial of Service (DoS) attacks, attackers aim to 'deny service' by overwhelming a victim with requests or network traffic. These attacks continue to be one of the most perilous threats to Internet security and stability. While an attacker can send requests to the victim directly from a single host, large volumetric attacks commonly distribute attack traffic generation through a botnet or by abusing innocent but vulnerable third-party services as so-called reflectors. This makes DoS attacks *distributed* (i.e., DDoS).

Although botnet-based attacks [11] have seen a resurgence in recent years, largely attributed to a rising number of insecure Internet of things (IoT) devices [12], they require attackers to first exploit a sufficiently large number of devices before attacks can be launched. In contrast, reflection attacks are less taxing: most of the preparatory work can be done with an Internet-wide scan for reflectors, which can be performed in less than one hour [13].

In a reflection attack, the attacker claims the network identity of the target, by sending a request to some third-party service (i.e., the reflector) with a spoofed source IP address. Services on top of UDP are particularly susceptible to spoofing, because of its stateless nature and lack of authentication on header data. As the third-party service cannot easily distinguish legitimate requests from forged requests, it will send a response to the purported requester, i.e., victim. This is referred to as *reflection*. By carefully selecting services that generate relatively large responses to small requests, so-called amplification is brought about, which allows attackers to reflect large volumes of network traffic to the victim. To avoid the bandwidth of an individual reflector becoming the bottleneck for attack traffic volume, attackers typically use hundreds to thousands of reflectors in a given attack.

A number of protocols have been shown to be prone to amplification by Rossow [2], ranging from legacy protocols such as CharGen and Quote-of-the-Day (QOTD), to widely deployed protocols such as NTP and DNS (more on that below). To measure the attacker's gain from amplification, Rossow introduces the *bandwidth amplification factor (BAF)*:

Definition II.1.

$$BAF = \frac{len\left(UDP\ payload\right)\ amplifier\ to\ victim}{len\left(UDP\ payload\right)\ attacker\ to\ amplifier} \tag{1}$$

C. DNS-based Reflection and Amplification Attacks

As a specific example of a service on top of UDP that can be abused for reflection and amplification, consider the DNS. When the *recursive resolver* (see Section II-A) accepts requests from anywhere on the Internet it is considered *open*. Open resolvers can be abused for reflection. Moreover, because DNS answers are often relatively larger than requests, amplification can be brought about.

MacFarland et al. [14] studied the amplification potential of DNS based DDoS attacks. They determine the amplification risk associated with authoritative servers, and determine the adoption of resource record rate-limiting, a technique to reduce the traffic coming from resolvers. Our paper also studies amplification risks, but from a different perspective. We determine how 'bad' domains used in attacks are compared to domains with high amplification factors. Additionally, we quantitatively analyse the impact of blocking ANY queries. We also contribute a method to estimate ANY response sizes from DNS measurement data, which we built on knowledge of standardized record type lengths [15].

Rijswijk et al. [4] also investigated amplification potential; specifically, that brought about by DNSSEC, which introduces record types with cryptographic keys and signatures. In their study, they used the UDP datagram sizes of requests and responses to infer amplification. Their methodology builds on the same principles, but a notable difference is that we reassemble individual records to infer the size of responses.

III. Data sets

In this paper we mainly use two data sources. The first is comprised of DDoS attack data recorded by the AmpPot project [16]. The second source provides daily snapshots of the content of the DNS, based on the data retrieved by the OpenINTEL active DNS measurement platform [17].

a) Attack data: The AmpPot project operates a set of geographically and logically distributed amplification DDoS honeypots. These honeypots mimic a reflector for popular UDP-based protocols, DNS included. The honeypots lure attackers by sending large responses to scans and next participate in DDoS attacks in a limited fashion (heavily rate limited) to learn details about the attack such as the duration and number of requests. In the case of DNS, the domain name and query type used for amplification are also learned.

To better distinguish domains used for actual DDoS attacks from spurious queries reaching the monitoring points, we focus on domains for which a honeypot recorded at least 10 queries during an attack and which were used against two or more targets. This left us with 100 domains used in 448,156 attacks, 342,274 (76%) of which use only second-level domains and can thus be joined against the OpenINTEL data.

b) OpenINTEL data: OpenINTEL is an active DNS measurement platform currently measuring over 65% of the DNS name space [17]. The platform actively queries around 235M second-level domains on a daily-basis for 12 resource records (for the full list, see Table I). Unfortunately, OpenINTEL does not include the 'ANY' type in its monitoring. ANY responses are, however, vital for measuring the amplification potential of domains. Our honeypot data has shown that the vast majority of DNS-based amplification attacks abuse ANY' requests. We therefore propose a methodology which uses the available OpenINTEL data for estimating the response size of an 'ANY' query. To this end, we used measurement results for the first of every month from January 2019 until December 2020.

Combining the two datasets, i.e., OpenINTEL domain size estimations with the set of abused domains as observed in AmpPot, allows us to make inferences about how attackers optimize for DNS-based amplification DDoS attacks.

IV. ANY response size estimation

In this section, we propose a methodology for estimating the size of a response to an 'ANY' query. We validate our methodology by comparing our estimations to real-world 'ANY' responses.

TABLE I: Estimation of DNS response size

Record type	Equation
header size	= 12 + 4 + len(domain name) + 1 + 11
signature size	= 30 + len(domain name) + 1 + size(rrsig)
A size	= 12 + 4
AAAA size	= 12 + 16
CAA size	= 12 + 2 + len(CAA)
CDNSKEY size	= 12 + 4 + sizeof(CDNSKEY)
CDS size	= 12 + 4 + len(CDS)
DNSKEY size	= 12 + 4 + sizeof(DNSKEY)
DS size	= 12 + 4 + len(DS)
MX size	= 12 + 1 + len(mail exchange) + 1
NS size	= 12 + len(nameserver) + 1
NSEC3PARAM size	= 12 + 4 + sizeof(salt)
SOA size	= 12 + 16 + len(mname) + len(rname)
TXT size	= 12 + len(text) + 2

A. How to estimate ANY response sizes?

A DNS response consists of the following parts: a header, the original question, and a response to the question [15]. With DNSSEC, the response can be further divided into an answer and signatures. As an 'ANY' query is "a request for all records" [8], the response can be seen as a combination of header, question, and a collection of answers and signatures, to answer for all records.

For our estimation of the response size to an 'ANY' query, we have to estimate the header, question, answers, and signatures sizes. While we base our estimates on data from OpenINTEL, any data source that exposes the number of records per type, or their standardized record type length, would work for the estimation. Our computation is summarized in Table I.

Since our computations are relatively straight-forward, implementing them in PySpark was trivial. This resulted in an approach where we could estimate all domains in OpenINTEL (for a single day snapshot) in the matter of minutes, making our method scalable to large numbers of domains.

a) Header: We combine the DNS header with the question, to get to an equation that is only dependent on the length of the domain name. The header itself is 12 bytes. The size of the query is 4 bytes plus the length of the domain name (with root label) plus one (null byte). To account for the EDNS additional record (type OPT), we add 11 bytes to our header.

b) Signature: The fixed-size parts of a signature come to 30 bytes. To this we need to add the length of the signer's name (plus one for the null byte), and the length of the signature in bytes. To perform this estimation at scale (all domains in OpenINTEL), we assume that the signer's name is the domain name. Also note that while the DNSSEC signatures (RRSIG) are part of an answer, we treat it as a separate part, because it makes evaluation at large scale simpler.

c) Answer: Estimating the size of the answer is the most complex part, due to fact that many record types require their own interpretation of fields in the packet. Each answer record has a fixed length 'header' accounting for 12 bytes per record in the answer. In an answer, we can have records with fixed length or records whose length depends on the content of the answer (variable length).

As shown in Table I, records with a fixed length are 'A' (4 bytes) and 'AAAA' (16 bytes). All other records comprise a fixed size part, that can either be the 12 bytes 'header' only or other fixed size information, and a variable length part. For 'CAA' records, indicating which certificate authorities (CAs) can issue digital certificates for a name, the variable part is the length of the CA. Similarly, for (C)DS records, the variable part is based on the length of the delegation signer. For '(C)DNSKEY' records the fixed part contains the protocol and algorithm fields, the size of the variable part, the key, we reverse-engineer due to the way OpenINTEL stores the results for this record type. 'MX' and 'NS' record sizes will depend, respectively, on the length of the mail exchange name and the length of the nameserver, while the size of a 'NSEC3PARAM' record, used to determine which 'NSEC3' records to include for DNSSEC requests for non-existing names, will depend on the length of the salt. Additionally, this record contains fixed fields for 4 bytes. A 'SOA' record has two variable fields, the 'MNAME' (primary nameserver) and the 'RNAME' (administrator email address). Additionally it has a few fields fixed in size. The fixed length fields combined account for 16 bytes. Finally, the free text record 'TXT' fully depends on the length of the text, with the addition of two bytes.

There is another aspect we have to take into consideration when estimating the return size of a DNS query. The DNS applies compression by replacing any duplicate label with a pointer (two bytes in size), to the first mention of the label elsewhere in the reply. This compression can be substantial, e.g., for record types such as MX and NS, as these each contain two names, often with repeating labels. For example, for the zone 'example.com' with NS records for 'ns1.example.com' through 'ns10.example.com', all the mentions of 'example.com' are replaced with a pointer to the label in the query, thus reducing the size of the answer. In our estimations we replace each mention of the query name with a pointer of two bytes. However, the compression applied by the DNS goes further than our estimation, *any* repeated label is replaced with a pointer. Suppose, the zone 'example.org.' has MX records 'mx-N.third-party.com.', with 'N' from 0 to 9. For 'mx-N' 1 through 9, the label 'third-party.com.' is replaced with a pointer to the label in 'mx-0'. Our estimation does not take this into account, as this is computationally expensive to do given the scale of OpenINTEL data. We have opted to implement only the replacing of the query name, and not all duplicate labels which may appear in the answer. Therefore we expect this aspect to contribute to an overestimation of the 'ANY' response size measured in practice.

Once we are able to estimate the 'ANY' query size, we can use this to further characterize domains. In particular, in literature the term amplification factor [14], [18] is often used to gauge the effectiveness of a domain for DDoS purposes. Definition II.1 shows how the bandwidth amplification factor is calculated, which can be computed for DNS by dividing the response size by the query size. In the following, we use the amplification factor in two ways. First, to filter 'smaller' domains. And secondly, to rank domains (Section V).

B. Validating the estimations

OpenINTEL offers a plethora of information about a domain and its resource records, and we have shown a way to combine those to an estimation of the 'ANY' response size. Now we verify how accurate this estimation is, by comparing it with the response size of 'ANY' queries in the wild.

To validate our estimation we performed an active measurement. Since we did not want our results to be biased towards a particular resolver setup, we used Zmap [13] to find a random sample of 2,000 open resolvers on the (IPv4) Internet. Removing systems that responded with malformed or invalid DNS messages left us with 804 open resolvers, which we further probed for their 'ANY' response behaviour.

As this study focuses on domains used in DDoS attacks, we selected a random sample of 1,000 domains with an estimated amplification factor larger than eight, but with an estimated response size of fewer than 4,096 bytes. This size threshold reflects the value most EDNS0 implementations will use as an initial parameter for the EDNS0 buffer size. Responses larger than the buffer threshold are truncated, and retried over TCP for the full answer, normally. Receiving a partial response over UDP makes these domains less attractive to attackers who rely on spoofing. Furthermore, we made a uniform random selection of domains in order to make the measurement more feasible and to reduce the impact our measurement may have on the infrastructure of the Internet.

We then queried each of the resolvers in our set for all the domains in our selection in a randomized order. To keep the impact of our active measurement minimal, we ensured that resolvers would receive queries only once every 16 seconds on average. Since we were interested in obtaining maximal responses, we built our queries with an EDNS0 record setting the payload size to 4,096 bytes and requested a DNSSEC signed answer by setting the 'DO' flag, while also requesting recursive resolution by setting the 'RD' header bit.

Since we attempt to estimate the size of a *complete* 'ANY' response, we next filtered out invalid and incomplete responses. This includes resolvers employing RFC 8482 [5] as a defense mechanism as well as failing or malformed responses (a return code other than 'NOERROR', a missing 'SOA' record, or responses that had the truncate flag set). Lastly, we also filtered out partial responses by comparing the set of records returned for a domain over all resolvers. Note that such filtering can easily be employed by dedicated attackers when selecting resolvers to use in subsequent attacks.

On the other hand, resolvers may also include additional records in 'ANY' responses as part of the additional and authority sections. Since we cannot possibly predict their size, we chose to ignore these in the following analysis, given that these records will only *enlargen* the overall response size. We averaged the measured response size per domain before comparing the measured size to our estimated response size.

C. Validation results

After making sure our measurements of ANY queries were comparable to our estimations, we could evaluate how accurate

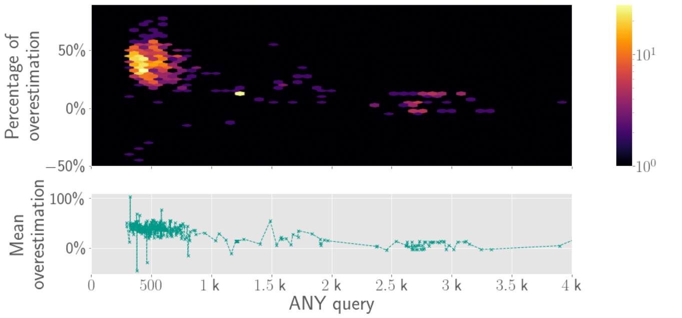


Fig. 1: Correlation overestimation and estimated size

our estimations were. For this evaluation we calculate how much we over- or underestimate, by taking the difference between our estimation and the measured size and dividing the outcome by the measured size. Fig. 1 shows the amount of overestimation compared to the estimated ANY response size, with at the bottom the mean overestimation corresponding to the estimated ANY response size. For ‘smaller’ domains (<1,000 bytes) our estimations are roughly 20%-60% larger than the measured size. This is mainly due to our imperfect implementation of DNS compression. We apply compression based on the query name alone, while the DNS compresses any duplicate label. These domains contain multiple ‘MX’ or ‘NS’ records to the same (third-party) zone, resulting in a smaller response size than our estimation. The three domains exceeding 100% overestimation contained record types in the estimation that were not present in the measured result, resulting in a much larger estimation. Our overestimation drops as we increase in size. It even crosses the origin, which means that the measured response is larger than the estimated response size. There are 20 domains where the measured result is larger than the estimated size. Four domains returned an NSEC3 record with signature while they didn’t support DNSSEC, because of this our estimation missed these signatures. For 16 domains the underestimation was due to record types which are not measured by OpenINTEL, predominantly SPF records. SPF records were deprecated in favor of TXT-based SPF records. Because of this, OpenINTEL does no longer measure this type of record, and therefore we are not able to use this in our estimation, causing a lower estimate (especially if the domain is using DNSSEC, as we would also miss the signature). On average the overestimation for larger domains (>2,048 bytes) is 5%.

Key takeaway: Estimating ANY response sizes from active DNS measurements leads to a size overestimation, for large domains, of 5%, making it a viable solution to identify DDoS potent domains.

V. Ranking domains

Using our estimation of the ANY response size we now rank domains in OpenINTEL by their amplification factor. Such a ranking allows us to evaluate how large the domains from the AmpPot dataset are compared to all the other domains in OpenINTEL. Suppose the domains we observed in attacks are not the largest available, we want to investigate why attackers would choose these domains over larger domains.

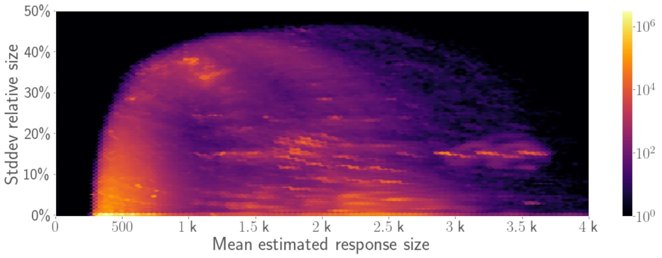
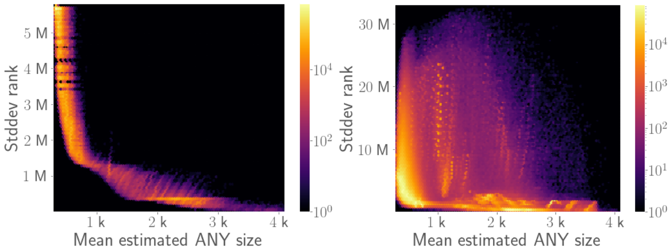


Fig. 2: Stability of the estimated size



(a) Standard deviation of domains without changes in size

(b) Standard deviation of domains with changes in size

Fig. 3: Comparison of the effect of size changes on the ranking, for domains with stable (left) / unstable (right) sizes.

A. Methodology for ranking domains

For all domains in OpenINTEL we estimate the ANY response size. Based on the estimation we calculate the amplification factor for each domain. Then we rank domains from the largest amplification factor to the smallest.

In order to analyze the stability of size and rank of domains, we sample a ranking every first of the month. We take samples for the period covering January 2019 to December 2020. For each sample, for scalability reasons, we select domains with an amplification factor higher than eight and an estimated ANY response size of below 4,096 bytes. Recall that OpenINTEL measures more than 235 million domain names, and we are mainly interested in larger ones (without truncation), as those are more attractive for DDoS attacks.

For our rank and size stability analysis we select domains that were present in all 24 samples. The rank stability of domains which are present for a long time may be influenced by two aspects. First, the size of the domain itself, and secondly the size of the other domains in the dataset. Using 24 months worth of samples allows us to evaluate, if the rank changes, how much of this is due to the domain itself changing in size and how much other domains attribute to this.

B. Domain ranking results

Before we present the ranking of domains observed in attacks, we study what affects the ranking the most. For domains present all 24 months, the ranking can be affected

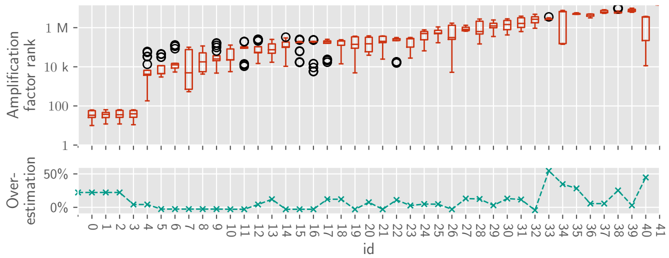


Fig. 4: Spread of rank per (attacked) domain

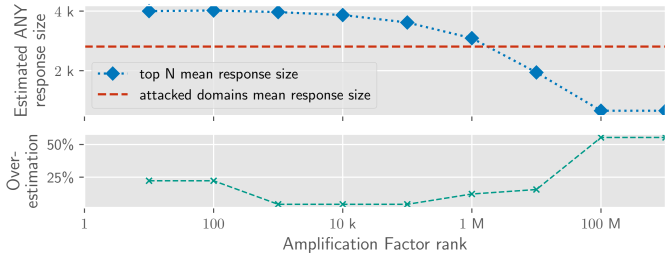


Fig. 5: Comparison mean estimated response size top-N against domains used in attacks

either by a change in their own size, or by changes in the size of other domains. In Fig. 2 we plot the standard deviation of each domain's relative size, here the relative size is their size compared to the maximum size we observed over 24 months. As the graph shows, the domains are spread out between zero and 50%, with no clear correlation between estimated response size and standard deviation. In Fig. 3 we compare the standard deviation of the rank for two groups. The first group, in Fig. 3a, consists of domains with a stable size during the 24 months (zero percent standard deviation in relative size, Fig. 2). The second group, consists of domains with variable sizes during the 24 months (a standard deviation in relative size unequal to zero). Comparing the standard deviation in rank of these two figures, shows a five times larger deviation for group two versus group one. This suggests changes in the size of other domains affect the ranking much less than changes in the size of the domain in question.

Selecting domains used in attacks from our ranking data shows the amplification rank spread as in Fig. 4. Here, we have ordered the domains by their mean amplification rank over the 24 months. Each domain was present for at least 12 samples out of the 24. To put these results into perspective we provide the percentage of overestimation. We determine the percentage by matching the mean estimated size per domain to their closest size match of Fig. 1.

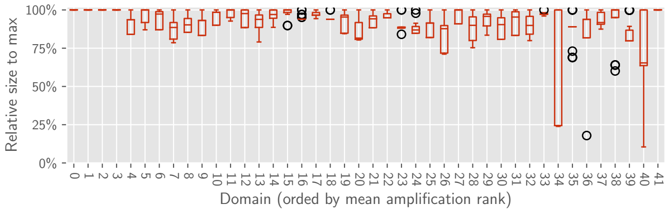


Fig. 6: Fluctuation in size per domain.

Four domains observed in attacks have reached ranks ten, eleven and twelve, during the 24 months of observation. However, Fig. 4 shows many domains used in attacks with much lower ranks, with a range of ranks roughly between 10^3 and 10^7. The mean estimated ANY size of these domains used in attacks is 2,809 bytes. In Fig. 5 we compare this mean with the mean ANY response size of domains in the top N (X-axis). From this we find that a selection of domains used in attacks are among the largest in our ranking, but there are many large domains which we have not observed in attacks at all.

Fig. 6 shows the fluctuation of the relative size of domains used in attacks. The figure shows sizes relative to the maximum size of each domain over the 24 months of observation. For most of these domains the spread is around 20%, with a few outliers, domain 34 in particular. This suggests that these attack domains have been relatively stable in size over the two years of measurements. The changes that did occur may simply have been related to normal operations of the domain, rather than DDoS related activities (e.g., purposeful inflation of the domain before attack).

Key takeaway: Domains observed in attacks are among the largest domains available. However, our ranking shows that there are still a sizable number of domains larger than the ones used so far that could easily be exploited.

VI. The Impact of Dropping ANY

With a growing dissatisfaction in the operational community for using ANY queries, we now want to know what the impact of dropping them would be on the response sizes of domains known to be used in attacks. Subsequently, we evaluate the positive effects of disabling the ANY response type.

A. How do we estimate the impact of dropping ANY?

We can adapt our size estimation to a specific query type by summing the sizes of the fixed header, the query, and a single signature and answer for the given query type. This gives us the ability to compare the response size of an ANY query with responses to the most common record types. To gauge the size reduction when dropping ANY queries we consider two approaches. The first approach selects the 'next-best type', namely whichever record type delivers the largest response for a certain domain. In the second approach the query type is fixed. This gives insight into what areas require focus when dealing with DNS-based DDoS attacks.

B. Is dropping ANY requests effective?

In Fig. 7 we see a CDF of the amount of reduction when going from ANY to the 'next-best type'. We have annotated with vertical dashed-lines the mean reduction. Ideally we would want the largest part of the CDF to be pushed towards the righthand side of the plot, as this would mean that most domains are reduced in size by a hundred percent. The other extreme is if the line stays mostly on the lefthand side of the

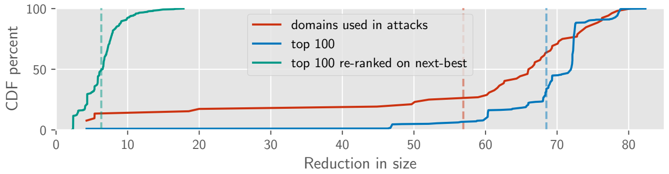


Fig. 7: Reduction in size by dropping ANY

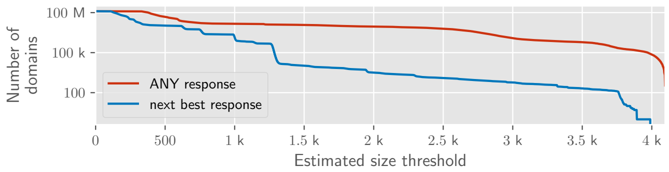


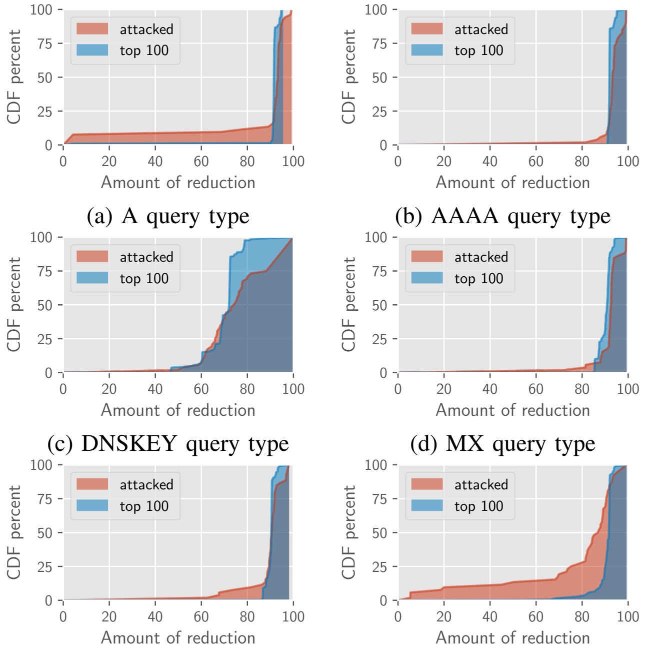
Fig. 8: Number of domains exceeding the estimated size threshold

plot, as this means that none of the domains see any reduction in size when moving away from ANY.

For domains used in attacks we see a mean reduction of 57%, with 75% being reduced by 52% or more. The situation with domains in the top 100 is even better. The mean reduction in size is 69%, with 75% of domains being reduced by 68% or more. However, if we compare this to a new top 100, ranked on the next-best type, the story changes. The mean reduction of this set of domains is 6%, with 75% of domains being reduced by 8% or *less*. In other words, the *currently* largest domains indeed substantially reduce in size when removing 'ANY'. However, there is a substantial number of large domains that are not part of the current top-100 which would be only marginally affected by removing 'ANY' support.

Next, we quantify the number of bad domains. Fig. 8 shows the number of domains which exceed a given estimated size threshold (x-axis). While the number of large domains ($> 2,048$ bytes) drop dramatically when moving from 'ANY' to the next-best type, it is worrying that there are still around a thousand domains which are larger than 2,048 bytes without the use of 'ANY' queries. This means that, while for the current top 100 domains and domains observed in attacks, the dropping of support for 'ANY' queries is effective, it is not the end-all be-all solution to the DDoS problem.

Finally, we want to understand which 'next-best' type is most problematic. To this end, Fig. 9 shows the results of our second approach. Here we fix the query type to evaluate the amount of reduction when moving from ANY to that query type. We performed this analysis to get more insight where the reduction is smallest, and conversely, where there is still work to be done by operators. The query types standing out from this analysis are 'DNSKEY' and 'TXT', as these show the smallest mean reduction. Domains used in attacks show a mean reduction of 76% when moving from type ANY to type DNSKEY, and 79% when moving from ANY to TXT. For domains in the top 100, the mean reduction, when moving to type DNSKEY, is 70%, and going from ANY to TXT, the reduction is 76%. Since the DNSKEY record contains large keys this smaller size reduction is understandable. The relative small size reduction when moving to TXT queries is less intuitive, and prompted us to investigate what makes these TXT records large.



(a) A query type

(b) AAAA query type

(c) DNSKEY query type

(d) MX query type

(e) NS query type

(f) TXT query type

Fig. 9: Reduction by moving from ANY to a specific query type

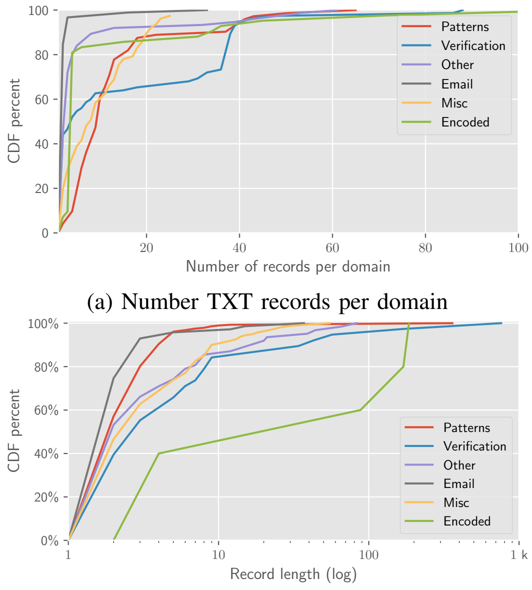
TABLE II: DNS TXT record categories on 2020-12-31.

Label	# of Records	% of Total	Plot
DNS TXT Records	3793	100%	🟩🟨🟥⬜🟦🟪
Verification	1168	31%	🟩
Patterns	890	23%	🟨
Miscellaneous	698	19%	🟥
Encoded	451	12%	⬜
Other	432	11%	🟦
Email	154	4%	🟪

C. Categorization of TXT records

We selected the domains from the re-ranked top 100 for this case-study as their reduction, when moving to TXT queries, was only 32%. In the past, by matching each TXT record against a regular expression, we were able to categorize roughly 99% of all TXT records in OpenINTEL [7]. This technique quickly shows what kind of records are present for a given population. We perform this analysis to shed light on the TXT records responsible for the small reduction in size. Table II shows a breakdown of the categories present in the TXT records of domains in the (new) top 100.

The Verification category contains records related to the domain ownership verification, such as Google or Facebook domain verification. Records in the Email category are related to email, like SPF records. The Miscellaneous category contains recognizable keywords to identify companies, advertising, etc.



(a) Number TXT records per domain

(b) TXT record length

Fig. 10: Number of and length distributions of TXT records

Pattern records are records containing dates, IP-addresses, or other clear patterns. Encoded records are, for example, base-64 records, base-32 records, or hashes. Records in the Other category do not fit the categories discussed previously. This category formed the core of the research in our previous work, and we refer to the paper [7] for details.

Compared to the numbers for the whole population of OpenINTEL on 2018-12-31, the 'Email' category dropped significantly, 4% from 69%. While the 'Verification', 'Other' and 'Patterns' categories have grown, 31% from 14%, 11% from 1%, and 23% from 5%, respectively. Suggesting that these domains are large due to TXT records from the categories 'Verification', 'Other' or 'Patterns'.

However, the distribution of TXT categories paints only part of the picture. The table does not tell the number of records per type per domain. Nor, does it say anything about how large the records of each category are. In Fig. 10a, and Fig. 10b, we look at the number of records per domain, and length of the records, of each category, respectively.

Most categories are seen with relatively few records per domain, generally below 20 records. With the exception of Verification records, roughly 31% of domains have 30 records, or more, in the category Verification. In Fig. 10b we look at the length of the TXT records per category. We have chosen to use a log-scale on the x-axis, since we have a mix of shorter and longer classes of records. The Encoded and Verification categories are among the longer record categories. On average the category with the longest records is the Encoded category with a length of 75 characters, followed by the Verification category with an average length of 30 characters.

In Table III we change the perspective slightly, instead of looking at the number of records per domain, or the length of individual TXT records, we look at the total size contribution per category to a domain. The table shows the average length of each category as it is returned for a domain. For example, while TXT records in the category Verification are, individually, not very long, due to their number per domain they result in an average length of 1,066 bytes per domain. And the category is responsible for, on average, 32% of the total TXT response size. Meaning that the full TXT response consisted of more than just Verification tokens.

TABLE III: DNS TXT record contributions.

Label	Average Length (bytes)	% of TXT response	# of Domains
Patterns	2,239	65%	73
Verification	1,066	32%	76
Email	1,010	35%	92
Miscellaneous	888	26%	78
Encoded	475	14%	43
Other	389	13%	76

From these two CDFs, and the table, we conclude that TXT records in the category Verification, Patterns, and Encoded, are among the 'worst' offenders, when it comes to inflating a domain. Either because of their relatively long length, and therefore response size, or the number of records per domain.

Key takeaway: Dropping responses to ANY queries is an effective way of reducing the response size of domains observed in DDoS attacks and of top ranked domains. However, the RR composition of several domains is such that, even when dropping ANY, a large response (>2,048 bytes) can easily be reached with another record type. Therefore dropping ANY might be only a temporary solution in the fight against DDoS.

VII. Operational Considerations

In the sections above we show that domains observed in attacks are among the largest domains measured by OpenINTEL. However, the picture is incomplete. OpenINTEL measures second level domains (e.g. example.com.). Lower level domains (third level and beyond, e.g. subdomain.example.com.) are not measured, and as a consequence we cannot estimate the size of those lower level domains. While these lower level domains are less enticing to attackers – lowered amplification factor due to a longer domain name – and account for less than 5% of attacks in the AmpPot dataset, it might mean there are more 'DDoS-suitable' domains than those studied in this paper.

Dropping full responses to 'ANY' queries through RFC 8482 [5] helps reduce the DDoS problem, as we show in Section VI. We fear, when the RFC gains more adoption, domains created for DDoS attacks will shift their efforts to a single record type to obtain a size near equal to their 'ANY' response size. Likely, this record type will be 'TXT', as we saw with the top 100 of domains ranked on the next-best type. Additionally, top ranked domains, ranked on the next-best type, typically consisted of many 'TXT' records.

When we classified the 'TXT' records of these top-ranked domains, we found many verification tokens per domain. Typically, these verification tokens can be removed once the domain has been verified. Here, we urge operators to take a critical look at these kinds of TXT records, and ask themselves if these are still required. For example, we have seen domains with unnecessarily many Google domain verification tokens. This implies that there is need for better guidance to users to avoid such misconfigurations. By keeping records up to date, and as minimalistic as possible, they help keeping the Internet cleaner. On the one hand, by reducing the volume of traffic needed for answers from a domain, and on the other, by making a domain less attractive for attackers.

Clearly, RFC 8482 does not help with large domains primarily consisting of 'TXT' records. What, then, can we do to make the DNS less attractive for DDoS attacks? A careful selection of the EDNS buffer size on resolvers [19] may help reduce the amplification factor, as answers above the buffer size are truncated and retried over TCP. However, legitimate responses (e.g. DNSSEC signed) may experience additional latencies when the buffer size is too small. Another approach is allowing zone operators to suspend domains on grounds of having a too large amplification factor. Of course, such decisions are highly sensitive and political.

VIII. Conclusion and Future Work

DNS is abused for tens of thousands of DDoS attacks daily and—unfortunately, as it is at the core of the Internet—stands out as a long-lasting driver for powerful amplification attacks. In fact, in contrast to other amplifying UDP-based protocols with fixable vulnerabilities, the amplification vulnerabilities in DNS lie in its core principles. That is, the fact that DNS responses are significantly larger than requests cannot be completely fixed. In this work, we studied *the* main culprit for DNS amplification at the moment: ANY responses. Our work shed light on the amplification potential of millions of domains, and illustrated to what extent ANY can be held accountable for the overall DDoS capabilities of DNS.

Using our proposed methodology, DNS zone operators, OpenINTEL or other parties with access to zone data now have a systematic and scalable way to gauge the amplification potential of registered domains. This not only serves as an easy-to-use early-warning system, but hopefully also helps steering the discussion on the future of ANY support. For example, we showed that dropping support for ANY responses would decrease the response sizes by $\approx 70\%$ for 75% of the largest 100 domains. This strong reduction clearly shows that the ANY type is one of the key enablers for DDoS attacks abusing DNS. At the same time, we also show that there is a significant but manageable number of domains that result in bad amplification even if ANY responses were disabled. This novel insight fosters future work that looks into how the composition of large TXT records can be reduced.

By linking these measurements with observations from DDoS honeypots, we found that attackers are already trying to optimize for the worst domains. Having said this, we also saw that there is "room for improvement", indicating that attackers are indeed not acting optimal. Our ranking allows zone operators to proactively approach owners of domains that stand out with particularly dangerous configurations. This also spurs future research that explores to what extent the configured DNS resource records are actually required, or can be dropped. Ultimately, we envision that our work will help to reliably identify, flag, and block *crafted* amplification domains.

References

[1] Cloudflare, "Famous DDoS attacks | The largest DDoS attacks of all time," 2021. [Online]. Available: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/

[2] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *NDSS 2014*, 2014.

[3] L. Constantin, "Attackers use DNSSEC amplification to launch multi-vector DDoS attacks," https://www.computerworld.com/article/3097364/security/attackers-use-dnssec-amplification-to-launch-multi-vector-ddos-attacks.html, 2016. [Online]. Available: https://www.computerworld.com/article/3097364/security/attackers-use-dnssec-amplification-to-launch-multi-vector-ddos-attacks.html

[4] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks," in *Proceedings of IMC 2014*, 2014.

[5] J. Abley, O. Gudmundsson, M. Majkowski, and E. Hunt, "Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY," Internet Requests for Comments, RFC Editor, RFC 8482, 2019. [Online]. Available: https://tools.ietf.org/html/rfc8482

[6] Cloudflare, "RFC8482 - Saying goodbye to ANY," 2019. [Online]. Available: https://blog.cloudflare.com/rfc8482-saying-goodbye-to-any/

[7] O. v. der Toorn, R. van Rijswijk-Deij, T. Fiebig, M. Lindorfer, and A. Sperotto, "TXTing 101: Finding Security Issues in the Long Tail of DNS TXT Records," in *WTMC 2020*, 2020.

[8] P. Mockapetris, "Domain Names - Implementation and Specification," RFC Editor, RFC 1035, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1035

[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC Editor, RFC 4033, 2005. [Online]. Available: https://tools.ietf.org/html/rfc4033

[10] J. Damas, M. Graff, and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))," RFC Editor, RFC 6891, 2013. [Online]. Available: https://tools.ietf.org/html/rfc6891

[11] A. Büscher and T. Holz, "Tracking DDoS Attacks: Insights into the Business of Disrupting the Web," in *LEET 2012*, 2012.

[12] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *USENIX Security 2017*, 2017.

[13] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *USENIX Security 2013*, 2013.

[14] D. C. MacFarland, C. A. Shue, and A. J. Kalafut, "The best bang for the byte: Characterizing the potential of DNS amplification attacks," *Computer Networks*, 2017.

[15] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC Editor, RFC 1034, 1987. [Online]. Available: https://tools.ietf.org/html/rfc1034

[16] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "AmpPot: Monitoring and Defending Against Amplification DDoS Attacks," in *Research in Attacks, Intrusions, and Defenses*, 2015.

[17] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE JSAC*, 2016.

[18] D. Kopp, C. Dietzel, and O. Hohlfeld, "DDoS Never Dies? An IXP Perspective on DDoS Amplification Attacks," in *Passive and Active Measurement*, 2021.

[19] jelu, "flag day 2020: Recommended EDNS buffer size," 2020. [Online]. Available: https://github.com/dns-violations/dnsflagday/issues/125