A Matter of Degree: Characterizing the Amplification Power of Open DNS Resolvers

Ramin Yazdani(✉), Roland van Rijswijk-Deij, Mattijs Jonker, and Anna Sperotto

Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
{r.yazdani,r.m.vanrijswijk,m.jonker,a.sperotto}@utwente.nl

Abstract. Open DNS resolvers are widely misused to bring about reflection and amplification DDoS attacks. Indiscriminate efforts to address the issue and take down all resolvers have not fully resolved the problem, and millions of open resolvers still remain available to date, providing attackers with enough options. This brings forward the question if we should not instead focus on eradicating the most problematic resolvers, rather than all open resolvers indiscriminately. Contrary to existing studies, which focus on quantifying the existence of open resolvers, this paper focuses on infrastructure diversity and aims at characterizing open resolvers in terms of their ability to bring about varying attack strengths. Such a characterization brings nuances to the problem of open resolvers and their role in amplification attacks, as it allows for more problematic resolvers to be identified. Our findings show that the population of open resolvers lies above 2.6M range over our one-year measurement period. On the positive side, we observe that the majority of identified open resolvers cut out when dealing with bulky and DNSSEC-related queries, thereby limiting their potential as amplifiers. We show, for example, that 59% of open resolvers lack DNSSEC support. On the downside, we see that a non-negligible number of open resolvers facilitate large responses to `ANY` and `TXT` queries (8.1% and 3.4% on average, respectively), which stands to benefit attackers. Finally we show that by removing around 20% of potent resolvers the global DNS amplification potential can be reduced by up to 80%.

Keywords: DDoS · Reflection and amplification · DNS · Open resolvers

1 Introduction

Distributed Denial of Service (DDoS) attacks are one of the common means for causing disruption on today's Internet. In DDoS attacks, the attacker typically leverages a large number of nodes on the Internet to exhaust the resources of a target network or host. In case of a Reflection & Amplification (R&A)

O. Hohlfeld et al. (Eds.): PAM 2022, LNCS 13210, pp. 293–318, 2022.
https://doi.org/10.1007/978-3-030-98785-5_13

DDoS attack [35], the attacker issues specifically crafted requests with a spoofed source IP address to cause (unaware) servers to send large responses to the victim. R&A attacks are made possible by the existence of connection-less networking protocols such as Domain Name System (DNS) [25], Network Time Protocol (NTP) [36], and Connectionless Lightweight Directory Access Protocol (CLDAP). In September 2017, for example, the largest reported DDoS attack to date targeted thousands of IPs of Google with a traffic rate of 2.5 Tbps [1]. This was a R&A DDoS attack using a combination of DNS, CLDAP and Simple Network Management Protocol (SNMP) servers.

Open DNS resolvers, that is DNS resolvers configured to respond to requests from any address on the Internet, have been widely misused to bring about R&A attacks. Evidently, the potential strength of an attack depends on the number of resolvers misused in the attack. We argue, however, that there are more aspects that contribute to attack potential. Namely, the configuration of a resolver can affect the effective size of responses, i.e., it factors into attack strength. Existing research [19] has revealed that specific DNS query types such as `ANY` and `TXT` have been misused more frequently in real world DDoS attacks. However, an Internet-wide and detailed study of the effect of the resolver internal configuration on response amplification has not been investigated and is not yet well understood.

Contrary to existing studies, which focus on quantifying the existence of open resolvers, our paper focuses on infrastructure diversity and aims at characterizing resolvers in terms of their ability to bring about varying attack strengths. The contribution of this paper is twofold. First, we identify the main factors that determine the potential attack strength and link these factors to the internal configuration of a resolver. Second, we perform a measurement-based, Internet-wide characterization of open resolvers to quantify which of the determining factors are prevalent in the wild. This mapping stands to help network operators and the network security community by enabling them focus on more powerful reflectors first.

The remainder of this paper is organized as follows. We discuss related work in Sect. 2. In Sect. 3 we identify and explain the factors that determine attack potential. We introduce our methodology for Internet-wide characterization in Sect. 4. The results of our measurement are then presented in Sect. 5. Finally, we conclude in Sect. 6.

2 Related Work

Several studies have investigated open DNS resolvers. Kührer et al. [24] classify open DNS resolvers based on the authenticity of their responses and the software that these resolvers are running. Similarly, [32] investigates open DNS resolvers that respond with incorrect answers. A classification of amplifiers based on hardware, architecture and operating system is provided in [25]. Our paper differs from these studies. We are not concerned about how legitimate the responses returned by open resolvers are or in purely profiling the software behind them,

but rather in characterizing the amplification power that these resolvers stand to provide, when misused, to bring about R&A attacks.

Rijwsijk et al. studied the impact of DNSSEC support on providing higher DDoS amplification power at the authoritative nameserver side [34]. The authors show that DNSSEC support significantly increases the amplification power of a domain name. We extend this research by investigating DNSSEC support at the open DNS resolver side as a key aspect of R&A attacks. Moura et al. [30] conducted a study to measure the problem of large UDP DNS responses. The authors analyze DNS queries and responses at the authoritative name servers of the `.nl` ccTLD and show that large DNS responses and server-side IP-fragmentation are rare. Our research differs from this work in a couple of aspects. We focus on open DNS resolvers as they provide a misuse potential for DDoS attacks. Besides, we explore query patterns that result in large response sizes rather than generic queries issued by real clients on the Internet.

Moon et al. [29] developed a service called AmpMap to quantify the amplification risk of six UDP-based protocols, using a budget of 1.5 k queries per server in their study. As it is not feasible to apply this method to all existing open resolvers without causing disturbance, the authors limited their measurement to 10 k DNS resolvers. In addition, ten popular domains were used in their queries. Our study differs both in scope as well as in approach. We target the entire open resolver population and in our setup we observe both the DNS query generation point as well as the authoritative nameserver.

Jiang et al. [21] investigated the caching behavior of over 19 k open resolvers when a revoked (ghost) domain is queried. We explore caching pattern of open resolvers from a different angle as our aim is to determine whether or not an open resolver is capable of evading the rate limiting mechanisms when resolving frequent queries for a domain name.

Nawrocki et al. [31] studied the behavior of attackers in terms of which sets of open DNS resolvers they misuse in attacks. Their results show that attackers efficiently detect new resolvers and steadily rotate between them. The authors link the behavior to the fact that resolvers disappear, either because they change away from open status, or because they are subject to IP address churn (e.g., home routers). Our work focuses on configuration aspects of open resolvers that could factor into decision-making processes.

Open DNS resolvers also exist in IPv6 address space. Hendriks et al. [20] explored the potential provided by open resolvers running on IPv6 addresses. Our methodology can be further extended to characterize open resolvers running in IPv6 address space. We leave this as a future work.

Lack of destination side source address validation results in hosts behind a firewall to become partially accessible to externals leveraging source IP address spoofing [16,22]. DNS resolvers residing in such networks are known to be vulnerable to DNS cache poisoning attacks, but can also be misused in DDoS attacks. Our study focuses on open DNS resolvers and thus we do not cover resolvers accessible through IP address spoofing.

Finally, an important consideration is that the bandwidth of open resolvers plays an important role in determining amplification power. Leverett et al.

studied the impact of bandwith availability in DDoS attacks [27]. Our study focuses specifically on characterizing the internal configuration of resolvers and the associated, thus far not studied effects on amplification potential.

3 Factors that Determine Attack Potential

Our measurement-based characterization in Sect. 5 will show that there are millions of open resolvers on the Internet, corroborating earlier findings in the literature. This is problematic as it gives attackers plenty of choices for performing DNS-based R&A. However, our intuition is that not every open resolver is likely to be equally effective as amplifier. The amplification capabilities of a resolver depend on a number of factors. In this section we analyze typical DNS configurations and highlight how those have an impact on amplification. We focus on support of specific DNS protocol features, handling of `ANY` queries, caching behavior and TCP support.

3.1 Support for DNS Protocol Features

In the original DNS specification [28], DNS messages over UDP are limited to a maximum of 512 bytes, putting a cap on the amplification potential of 'classic' DNS. This means that open resolvers that only support classic DNS are moderate amplifiers at best. The Extension Mechanisms for DNS (EDNS0) were introduced in 1999 [15]. The goal of EDNS0 was to overcome a number of limitations in the existing DNS protocol that were hampering the development of new functionality. Support for DNS messages over UDP of more than 512 bytes is one of the features of EDNS0. Thus, if an open resolver supports EDNS0 it can be a much more potent amplifier. How potent depends on the specifics of the configuration and implementation of EDNS0. Algorithm 1 shows the possible variants of the DNS and EDNS0 implementation.

In an EDNS0 exchange, the client sending the query can specify the maximum DNS message size it is willing to receive over UDP ($client_{max_{UDP}}$). As the function GetEDNS0Response (line 11) shows, the server has two implementation options. It can impose its own maximum UDP message size limit and apply that to the response (*variant a*), or it can only use the client's value from the query (*variant b*). If the response size exceeds the maximum response size over UDP, the response will be truncated. Here, there are two implementation options, as the TruncateResponse procedure (line 1) shows. In *variant 1*, the server truncates to a small response, that either contains no data or a minimal number of resource records (RRset), such that the response is correct and passes DNSSEC validation. In terms of putting a cap on amplification, this is the most favorable option. In *variant 2*, the truncated response is filled with RRsets from the original response until the maximum UDP response size is about to be exceeded. In terms of amplification, this is the worst option.

The original DNS specification also lacks critical security features, which makes the protocol vulnerable to so-called cache poisoning attacks. The DNS

Algorithm 1. EDNS0 code variants

```
1: procedure TruncateResponse(response, max_UDP)
2: variant_1:
3:                                              ▷ Find minimal RRset TC_min
4:     return TC_min
5: variant_2:
6:     for all RRset ∈ response do
7:         if |TC_resp ∪ answer| > max_UDP then
8:             return TC_resp
9:         else
10:            TC_resp ← TC_resp ∪ answer
11: function GetEDNS0Response(response)
12: variant_a:
13:    max_UDP ← min(client_max_UDP, server_max_UDP)
14: variant_b:
15:    max_UDP ← client_max_UDP
16:
17:    if |response| ≤ max_UDP then return response
18:    else
19:        return TruncateResponse(response, max_UDP)
```

Security Extensions [10–12] address this vulnerability. It is known from the literature [34] that DNSSEC can be abused in amplification attacks. This is because DNSSEC responses are generally larger than unsigned DNS responses, as they include digital signatures and – depending on the message type – public key material. Reports of DDoS attacks in the news suggest that attackers increasingly use DNSSEC-signed domains in amplification attacks [14]. This is supported by observations using DDoS honeypots [23] as well as in a recent work that studies R&A attacks using IXP traces [31]. Whether or not an open resolver supports DNSSEC is thus a factor that influences its usability for amplification from an attacker's point of view. Clients can signal to a nameserver that they wish to receive DNSSEC data by setting the `DNSSEC OK` flag (`DO`) in an EDNS0 query. Resolvers can have three levels of DNSSEC support:

i. *No DNSSEC support* – the resolver does not return DNSSEC-specific record types at all;
ii. *Pass-through of DNSSEC-specific records* – the resolver returns DNSSEC-specific record types returned by upstream resolvers or authoritative nameservers, but does not set the `DO` flag in queries to upstream servers;
iii. *DNSSEC fully supported* – the resolver returns DNSSEC-specific record types and sets the `DO` flag in queries to upstream servers.

The difference between the latter two forms of DNSSEC support is subtle, yet important. In the pass-through case, responses will typically not include signatures. If an attack uses a DNSSEC-specific record type, such as `DNSKEY`,

this affects amplification. Open resolvers with full DNSSEC support will return a significantly larger response to such a query because they will include the signatures.

The final protocol feature we need to consider is processing of the optional *authority* and *additional* sections in a DNS response. According to the original DNS specification [28], a DNS response contains three sections: answer, authority and additional. The *authority* section contains records that point to an authoritative nameserver for the domain. This optional section – when included in a response – typically contains the `NS` records listing (part of) the authoritative nameservers for a domain. The *additional* section contains additional information related to the query, such as the `A` and `AAAA` records for nameservers listed in the authority section. Whether or not an open resolver includes the two optional sections in a response affects the amplification potential of that resolver, since responses will be larger if either one or both of these sections is included. If a domain is DNSSEC-signed, the authority and additional section may also include signatures. There are three implementation options regarding the optional authority and additional sections:

i *Minimal responses* – the resolver only returns the authority and/or additional section if required by the DNS specification (i.e. in a `NO DATA` or `REFERRAL` response);
ii *Pass-through* – the resolver includes authority and additional sections only if these are returned by an upstream nameserver in response to a query from the resolver;
iii *Active synthesis* – the resolver attempts to populate the authority and additional sections based on information it already has in its cache for the domain being queried.

Table 1 summarizes the impact of the DNS protocol features on amplification as just discussed. The right-hand column gives an intuition about the impact of the specific feature on amplification potential. Negative signs indicate that this variant of the feature makes an open resolver less potent as amplifier, positive signs indicate the opposite.

3.2 Handling of `ANY` Queries

The next factor that affects the amplification potential of a resolver is how it handles the so-called `ANY` query type. An `ANY` query signals to a resolver or authoritative nameserver that the sender wishes to receive all records pertaining to the name in the query. As a result, any record that exists in a zone, such as IP address records (`A` and `AAAA`) and `TXT` records (containing, e.g., domain verification tokens) are requested *at once*. Responses to `ANY` queries are therefore potentially the largest possible DNS response for a name. Because of this, `ANY` queries are frequently used for amplification attacks [37]. On the side of the DNS resolver, an `ANY` can be dealt with in three ways:

Table 1. Impact of protocol features on amplification

DNS protocol version

Short ID	Description	Impact
P1-classic-only	The open resolver only supports the classic DNS protocol [28]	−−
P2-EDNS0-server-lim	The open resolver supports EDNS0; it imposes its own maximum size on UDP messages	+
P3-EDNS0-client-lim	The open resolver supports EDNS0; it returns responses up to the maximum size requested by the client	++

Message truncation

Short ID	Description	Impact
TC1-minimal	Truncation is done to a minimal size (e.g. no data at all or a single RRset)	−−
TC2-maxfill	Truncated responses are filled up to the maximum UDP message size (as set for EDNS0 responses), by adding RRsets until adding another RRset would exceed the maximum UDP message size.	++

DNSSEC support

Short ID	Description	Impact
SEC1-DNSSEC-no	DNSSEC is not supported and DNSSEC-specific record types are not returned	−−
SEC2-DNSSEC-passthru	DNSSEC is not supported, but DNSSEC-specific record types are returned if included in a response from an upstream server	+
SEC3-DNSSEC-support	DNSSEC is supported, DNSSEC-specific record types are returned and the DO flag is set in queries to upstream servers	++

Authority and additional section processing

Short ID	Description	Impact
AA1-no	The authority and additional sections are not returned to queries, unless required	−
AA2-passthru	The authority and additional sections are returned if and as present in responses from upstream servers	=
AA3-synthesize	The authority and additional sections are actively synthesized based on data in the resolver's cache	+

i *Refuse or restrict type ANY queries* – this is uncommon at present, but we expect the number of DNS resolvers that exhibit this behavior to increase as standardized in RFC 8482 [9].

ii *Return whatever is cached for the query name* – the resolver returns whatever records it has cached for the query name. If it has no cached information, it forwards the query to an upstream resolver or authoritative nameserver and returns whatever response these give to the ANY query;

iii *Return a full ANY response* – the resolver returns a full ANY response, either from its cache, or, if not in the cache, from the response to a follow-up query to an upstream resolver or authoritative nameserver (which the resolver then also caches).

The latter two options entail that the open resolver can return potentially large responses to ANY queries. This is not guaranteed, however. Both options have advantages as well as disadvantages for attackers. Option ii may return a smaller response if the open resolver already has cached information for the

Table 2. Impact of implementation choices for handling `ANY` queries

Short ID	Description	Impact
`ANY1-refuse`	The open resolver refuses `ANY` queries or returns a minimal response.	−−
`ANY2-return-cached`	The open resolver returns whatever it has in its cache for the queried name.	+
`ANY3-any-upstream`	The open resolver sends the `ANY` query upstream and returns whatever the authoritative nameserver responds with.	+

queried name. An attacker, however, can also prime such a resolver, by sending it one or more queries for the query name used in an attack to pre-populate the cache. Resolvers that follow implementation option iii will always return the largest possible response to an `ANY` query, provided that the authoritative nameserver also returns a full response to an `ANY` query.

Table 2 summarizes the implementation choices for `ANY` query handling. The columns are similar to Table 1.

3.3 Caching

The third major factor that affects the potential of an open resolver as amplifier is whether or not the resolver has its own cache and how this cache is implemented. If the resolver caches responses, it will typically only interact with the authoritative nameserver for the domain abused in an amplification attack infrequently. This means that any form of mitigation, such as Response Rate Limiting [39] at the authoritative nameserver, is likely to be ineffective. This is an advantage for an attacker seeking to misuse the resolver.

A second aspect to consider if the resolver has a cache is for how long the resolver caches responses from authoritative nameservers to `ANY` queries. One particular option that, e.g., the popular open source resolver Unbound[1] supports, is to use the minimum of the TTLs observed in the records received in the `ANY` response from the authoritative. This is of particular interest in case, for attack purposes, a DNSSEC-signed domain is used that, in turn, uses `NSEC3` [26] for authenticated denial-of-existence. The zone of such a domain will contain a so-called `NSEC3PARAM` metadata record that stores parameters specific to the `NSEC3` mechanism. This metadata record has a TTL of 0 by default. Consequentially, a resolver receiving an `NSEC3PARAM` record as part of an `ANY` responses may decide not to cache at all. This makes such a resolver a much less effective amplifier because of the previously explained effect of resolver caching on the response rate.

Table 3 summarizes the implementation options with respect to caching and indicates their effect on the amplification potential of a resolver.

[1] http://unbound.net/.

Table 3. Impact of cache implementation choices

Short ID	Description	Impact
`cache1-none`	The open resolver does not have a cache.	−−
`cache2-ANY-minTTL`	The open resolver has a cache but does not cache `ANY` responses with one or more records with a TTL of 0 seconds.	=
`cache3-caches`	The open resolver has a cache and also caches `ANY` responses with one or more records with a TTL of 0 seconds.	++

Table 4. Impact of TCP support

Short ID	Description	Impact
`TCP1-no`	The open resolver does not support TCP fallback when sending queries to upstream servers.	−−
`TCP2-yes`	The open resolver supports TCP fallback for queries to upstream servers.	+

3.4 TCP Support

The final characteristic that we consider is whether or not the open resolver supports TCP fallback to upstream resolvers or authoritative nameservers. If an upstream nameserver receives a query from the open resolver that it is unable or unwilling to answer over UDP, it will send back a truncated response (see also Sect. 3.1). This indicates to the initiator of the query (in this case the open resolver) that they should retry the query over TCP. If the open resolver does not support TCP fallback, this has two consequences. First, it limits the maximum size of responses it can return to the maximum size its upstreams are willing to send over UDP. Second, it means that forced truncation of queries such as `ANY` by upstream servers can be used as a mitigation mechanism to make the open resolver an ineffective amplifier. Table 4 shows the implications of TCP support.

4 Data Collection Methodology

In this section we present our data collection methodology and discuss the ethical considerations that we made towards our measurement design.

4.1 Scanning and Testing Open Resolvers

Our data collection methodology is based on two main steps: *open resolver identification* and *systematic testing of amplification power* (following the characteristics described in Sect. 3).

The *open resolver identification* step consists of scanning the entire IPv4 address space, randomly and on a weekly basis. For each contacted IP, our scan issues a DNS `A` query for a unique subdomain (by binding the IP address and the timestamp to the query name) of a domain under our own control. If we receive

Table 5. Queries issued in our *systematic testing of amplification power* step

#	Query Type	ENDS0 enabled	EDNS buffer size (B)	DO bit set	Amplification factor (×)	Description
1	A	No	–	No	1.9	0-day churn investigation
2	A	Yes	4096	Yes	8.5	DNSSEC support test
3	ANY	No	–	No	2.4	Classic ANY
4	ANY	Yes	16384	No	32.4	EDNS0 enabled ANY
5	ANY	Yes	16384	Yes	38.0	EDNS0 and DO enabled ANY
6	ANY	Yes	16384	No	153.7	ANY with 12KB response size
7	TXT	No	–	No	2.3	Classic TXT
8	TXT	Yes	16384	No	24.8	EDNS0 enabled TXT
9	TXT	Yes	16384	Yes	31.4	EDNS0 and DO enabled TXT
10	TXT	Yes	16384	No	125.9	TXT with 11KB response size
11–13	2×A & ANY	No	–	No	1.9	A and ANY cache test
14,15	2×ANY	No	–	No	1.9	0–TTL response caching

the correct answer for the query – indicating that a full resolution process has taken place – we infer that IP address is an open resolver. Our scan utilizes the ZIterate [8] tool of the Zmap library [18] to create a random permutation of the IPv4 address space for each week in which we scan. We then rely on the MassDNS tool [4] to issue the unique subdomain queries to each IPv4 address.

The second step aims at checking the internal configuration of an open resolver. To derive logical and consistent results, we note that there is a time-critical dependency between the *open resolver identification* and the *systematic testing of amplification power* steps of our data collection methodology. Previous studies have revealed that there is considerable IP churn among open resolvers [24,25], which entails that any meaningful interaction with the open resolvers that we discover needs to happen as close as possible in time to the identifying scan. We therefore run the *systematic testing of amplification power* step on the same day as the *open resolver identification*.

For the second step we issue 15 queries to each identified open resolver. Note that our goal is not to come up with a complete list of patterns that cause high amplification factors [29]. We rather aim to investigate a limited number of known high amplification patterns and explore the extent to which such patterns are supported among open DNS resolvers. A summary of the queries is provided in Table 5. The corresponding typical amplification factor for each query in this table is derived by dividing the TCP/UDP message size of the response to that of the respective query. Note that these amplification factors are given to provide a sense of typical amplification power that each query causes. This can however differ based on the implementation of a recursive resolver and domain name in question.

We use unique queries, which means that none of the queries cause a cascading interference with the following queries, as the resolver will treat each query independently. To start, we repeat an `A` query (query `#1`) to verify that an open resolver identified in the first step of our methodology is still responsive (due

to churn). Then we proceed with queries that provide answers to the following aspects. First, we focus on deriving DNSSEC support and EDNS0 buffer sizes. This is done by sending `A` queries with the EDNS0 flag enabled and an EDNS0 buffer size of 4096 bytes (query `#2`). Next, we focus on how resolvers handle `ANY` queries, as we have identified those as a critical factor for amplification. We do this by first testing for `ANY` query support with classic `ANY` queries (query `#3`), then with `ANY` queries with EDNS0 and DNSSEC enabled (queries `#4` and `#5`). Finally, we test for `ANY` queries that trigger large responses (query `#6`). While `ANY` queries are deprecated and are likely to be increasingly blocked in the future, we argue that R&A attacks might shift towards domains with large `TXT` records [38] which, due to their variable length, can also result in sizable amplification. Therefore, we decided to also include queries to test how resolvers handle `TXT` records. Similar to ANY queries, we do this for classic `TXT` records (query `#7`), EDNS0 and DNSSEC enabled `TXT` queries (queries `#8` and `#9`), and `TXT` queries with very large responses (query `#10`). For the queries with a large response size (queries `#6` and `#10`) our authoritative nameserver responds with a truncated answer which would trigger the resolvers to retry using a TCP fallback. We then test if a resolver is a caching resolver by sending two consecutive `A` queries (queries `#11` and `#12`) and checking if the resolver contacts our authoritative nameserver for both queries. This is followed by sending an `ANY` query (query `#13`) for the same domain name to investigate whether `ANY` queries with a cache entry for an `A` record are resolved using the cache. Finally we test how resolvers treat queries for a record with a zero TTL set on our authoritative nameserver. We do this by sending two identical `ANY` queries (queries `#14` and `#15`). We use the dnspython library [2] to conduct the scans in this second step. Note that multiple of our queries result in responses which are larger than common path MTU sizes, which can lead to responses getting dropped rather than fragmented [13]. We do not investigate resolvers that fail to respond to us due to such limitations in the middle-boxes on the path because we consider this out-of-scope.

4.2 Ethical Considerations

We designed our methodology with the following aspects in mind. First, we want to minimize the number of queries per host and open resolver. For this reason, we run our scans on a weekly basis, with the exception of a short dedicated measurement to quantify churn (Sect. 5), which was done on a daily basis for open resolvers discovered during the *open resolver identification* phase. Second, we took care of distributing the scan randomly across the input space (by using ZIterate). This diffuses our queries sent towards a specific network over time and reduces bursts. We also offer additional information about our research and an opt-out mechanism. Finally, we use a specific query name in our scans that makes it easy for network operators to discover the purpose behind our scans. We have full control over the authoritative nameserver for the domain name that we use to measure. The `PTR` record of the IP address of our scanning host also points to a domain name on which we host an information page.

During the study period we received seven opt out requests for our scans. Two requests directly reached us using our contact info on the webpage of our project. The others were forwarded to us through our ISP. Two out of seven complaints did not respond to our request to provide us with the IP address ranges that they wished to be excluded from being scanned. For the remaining five we stopped further scanning.

One could argue that other projects that monitor the existence of open DNS resolvers exist and that they could be used as input for our data collection. While such projects do exists, and we initially did consider them as data providers. Projects such as the OpenResolverProject [5] and The Measurement Factory [6] have already stopped their measurements at the time of our study. There exist other projects which do not share their data set. Thus, we ultimately decided on running our own IPv4 open resolver scans. This is furthermore essential due to considerations such as keeping the amount of time between open resolver identification and systematic testing as short as possible, but also to address the unpredictability and difficulty of quantifying the effects of scanning from different vantage points.

5 Results

5.1 Open Resolvers over Time

Figure 1 shows the number of open DNS resolvers that correctly resolved our queried domain name over time. We observe the population of open resolvers to gradually decrease from roughly 3 millions at the start of our measurement period to roughly 2.6 millions towards the end. As a reference point we also include the numbers reported by the Shadowserver project. For dates with a missing data-point in the Shadowserver dataset we use the nearest data-point.[2] A substantial difference is initially noticed comparing our results to those published by the Shadowserver DNS scanning project: 38% more detected through our scans, on average. The Shadowserver project – as we infer from their webpage – considers a host to be a recursive DNS resolver only if the response is issued by the queried host. However, it has been shown [25] that a large body of DNS forwarders (roughly 800k hosts in our scans), due to potential misconfigurations, fail to correctly change the IP address in forwarded DNS queries. This results in answers being returned from non-contacted hosts and is the main source of difference between our results and the numbers reported by the Shadowserver project. Also the Shadowserver project has been running on a daily basis and for a longer period compared to our scans. This would increase the chance of networks asking to be excluded from their scans as well as more networks blocking the IP addresses of the scanning hosts. The vantage point of a measurement also has an impact on the visibility of hosts to the scanner. These differences have been investigated earlier for scanning other protocols such as HTTP(S) and

[2] The discontinuity seen on the plot for our scans on 2021-06-28 was the result of a one-day measurement failure.

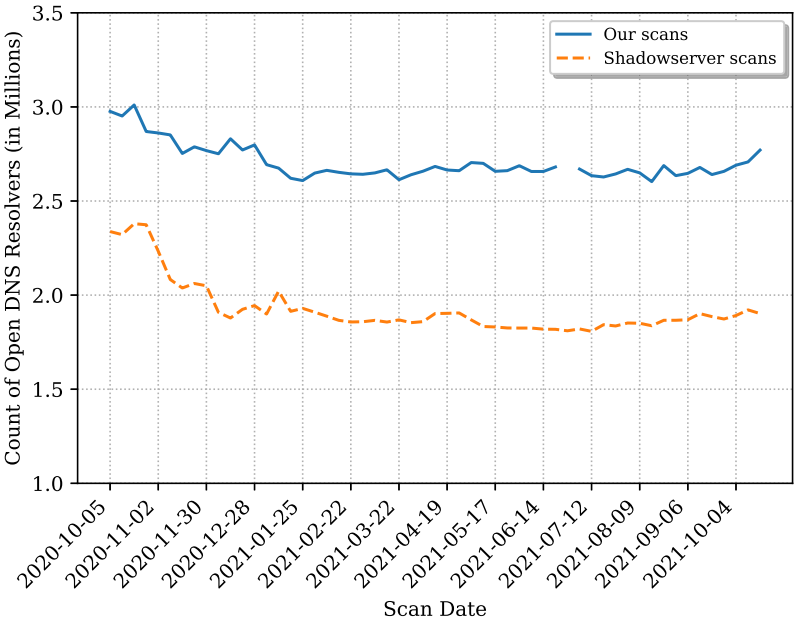

Fig. 1. Number of open resolvers over our study period that correctly resolve queries

SSH [17,40]. Scanner configuration parameters such as scanning rate and time of the day would also cause differences in measurement results. Since we don't have access to the raw data of the Shadowserver project, it is not possible for us to further investigate the potential sources of differences. Overall, a negative growth in the number of open resolvers is visible, which might be due to the efforts of researchers and operators in patching open DNS resolvers. Verifying this is, however, out of the scope of our paper.

Key Takeaway: Despite a decreasing trend, open resolvers still exist in the magnitude of millions.

5.2 IP Address Churn

A share of open resolvers become unresponsive after our initial discovery (i.e., on the same day still). This can be due to the IP address churn caused by DHCP lease of IP addresses as well as honeypot type hosts that apply rate limiting and do not respond to subsequent queries. We do not discern these two cases in this paper. However, we send extra probes at the beginning of our followup scans to quantify the percentage of hosts that are already no longer responsive. Previous studies have explored the churn of open DNS resolvers on a weekly scale [24,25]. In order to investigate this in a more fine-grained way, we sent daily queries to the open resolvers that correctly resolved our main scan query. According to [24], 52.2% of open resolvers disappear in the first week. Our results represent

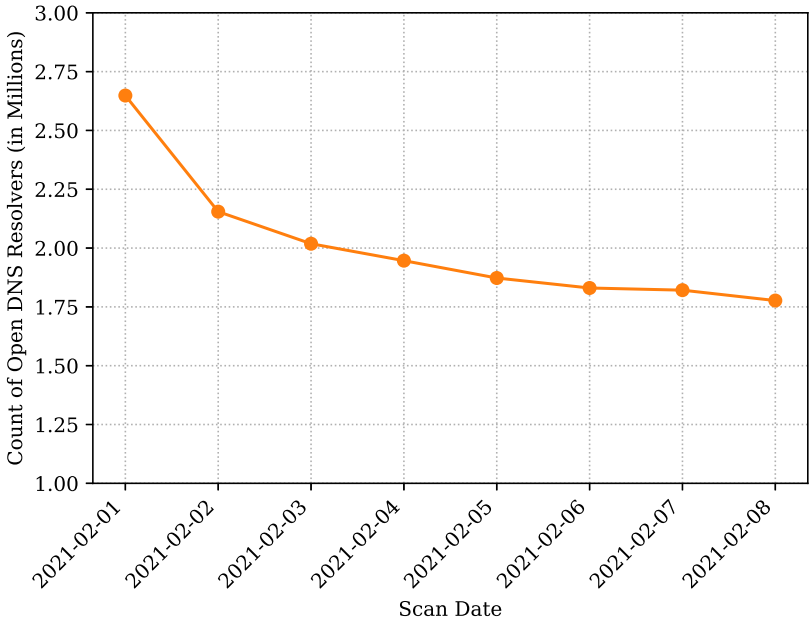

Fig. 2. Count of open resolvers discovered on 2021-02-01 which were still responsive on the consecutive days till 2021-02-08

a lower IP churn (32% of open resolvers are not responsive after a week, as seen in Fig. 2). This difference may be caused by the large reduction in the magnitude of open DNS resolvers in the time period between two studies. We observed a 19% IP churn one day after our main scan.

Key Takeaway: The IP churn rate of resolvers has significantly decreased compared to the previous studies. This could 'benefit' attackers, as their list of open resolvers to misuse needs to be renewed less frequently.

5.3 DNSSEC Support and Supported EDNS0 Buffer Sizes

To determine DNSSEC support by open resolvers, we sent, to the list of open resolvers collected during the main scan, `A` queries with EDNS0 enabled and DNSSEC OK (DO) bit set to 1. The EDNS0 buffer size of our queries was set to 4096 bytes. Based on the presence of an `RRSIG` record in the DNS response that we get back, we can infer DNSSEC support for each open resolver. On average 59% of open resolvers during the study period lack DNSSEC support (see Fig. 3). For open resolvers for which we inferred DNSSEC support, we have extracted the advertised EDNS0 buffer sizes that resolvers return to our scanner (see Table 6). 53.24% of resolvers advertise an EDNS0 buffer size of 512 bytes (also the default value for BIND DNS software which is one of the widely deployed DNS implementations). Although these open resolvers support

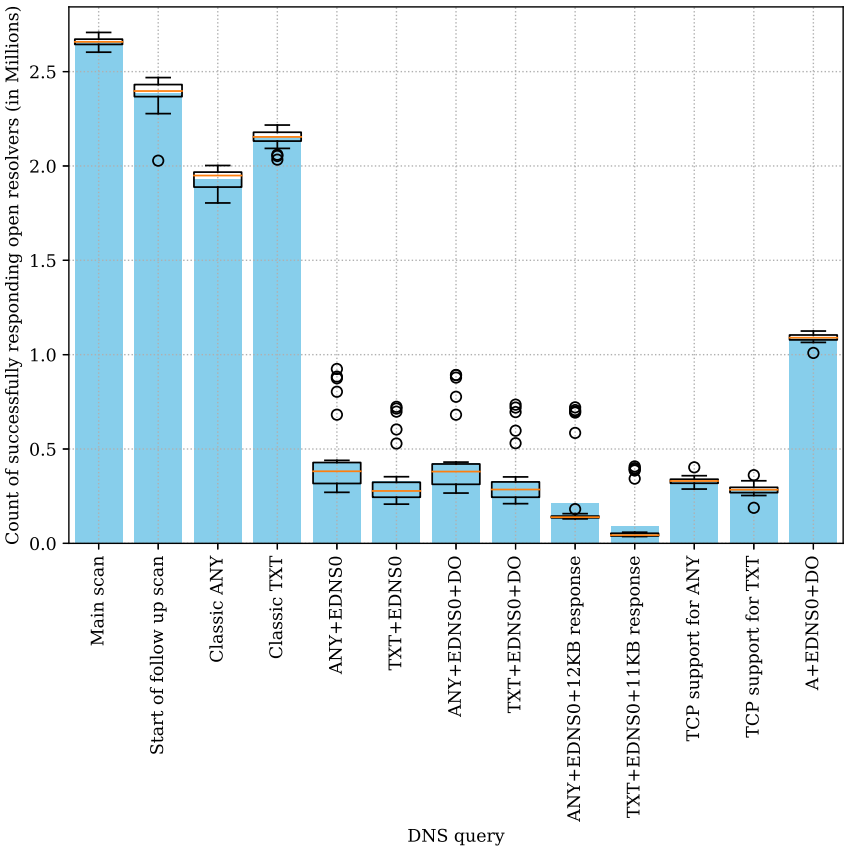

Fig. 3. Distribution of responsive open resolvers for each query category in the period of 2021-01-25 till 2021-10-11

DNSSEC, they only offer a limited amplification power. 30.31% of resolvers advertise a size of 4096 bytes which is the RFC recommended value [15]. This considerably large group of resolvers are potentially dangerous as they advertise to handle pretty large responses. Roughly 16% contribute to the next eight common values which cover a range of EDNS0 buffer sizes with the value of 8192 bytes standing out. Considering the way EDNS0 buffer sizes are negotiated between DNS clients and servers (as we discussed in Sect. 3), an attacker still needs to select authoritative nameservers that do not impose a limit on EDNS0 buffer support to be able to leverage this feature.

Key Takeaway: A large group of open resolvers lack DNSSEC support. This stands to substantially limit attackers when misusing DNSSEC.

Table 6. Average EDNS0 buffer size distribution between 2021-01-25 and 2021-10-04 (Top 10)

EDNS buffer size (B)	#Resolvers	Percentage
512	525.3k	53.24%
4096	299.0k	30.31%
4000	56.3k	5.71%
1232	54.8k	5.55%
1280	33.6k	3.41%
1220	8.7k	0.88%
8192	3.1k	0.31%
1224	2.5k	0.25%
1472	1.6k	0.16%
1460	0.6k	0.07%

5.4 `ANY` Query Handling

We used multiple variants of `ANY` queries to explore how open resolvers handle such query types. Our results show that roughly 72.6% of open resolvers successfully return an answer for a classic `ANY` query, for which the answer fits within a 512-byte packet. Only 16.2% of open resolvers successfully respond to EDNS0 enabled `ANY` queries (without the DO bit set). Setting the DO bit further decreases the resolution success to 15.9%. Finally, 8.1% of resolvers are capable of responding to an `ANY` query that has an answer of approximately 12 KB.

Key Takeaway: Only a limited number of open resolvers are capable of handling ANY queries with very large response sizes.

5.5 `TXT` Query Handling

Multiple `TXT` queries were sent to each open resolver to explore the consequences of changing various DNS protocol fields in the way open resolvers react to our queries. Our experiments reveal that on average approximately 80.9% of open resolvers during the study period successfully resolve a classic `TXT` query for which the answer fits a 512-byte packet. This is surprising as classic `TXT` queries have been part of the DNS standard from its beginning. We further observe that only 12.3% of open resolvers successfully respond to EDNS0 enabled `TXT` queries. Setting the DO bit doesn't have a noticeable impact in this case. Finally, only 3.4% of resolvers are capable of responding to a `TXT` query that has an answer of approximately 11 KB. Comparing the behavior of resolvers when dealing with `TXT` and `ANY` queries, we observe a higher success rate for classic `TXT` queries, while when it comes to EDNS0 enabled requests, `ANY` queries have a higher success rate.

Given the amplification potential of open resolvers supporting large `TXT` and `ANY` queries, we also investigate if these resolvers are concentrated in specific

Table 7. Distribution of resolvers at the start of followup scans on 2021-03-29 (Top 10) with the network type field categorized as: ISP (Fixed Line ISP), ISP/MOB (Dual service ISPs with fixed line and mobile), DCH (Data Center/Web Hosting/Transit), COM (Commercial), MOB (Mobile ISP), EDU (University/College/School), CDN (Content Delivery Network), SES (Search Engine Spider), ORG (Organization), GOV (Government)

Network type		AS distribution	
Type	#Resolvers	ASN	#Resolvers
ISP	40.49%	AS4134	7.68%
ISP/MOB	33.97%	AS4837	5.10%
DCH	11.45%	AS4766	2.84%
COM	9.97%	AS45090	2.69%
MOB	1.83%	AS47331	2.40%
EDU	1.21%	AS5617	2.29%
CDN	0.33%	AS12389	1.67%
SES	0.29%	AS3462	1.53%
ORG	0.21%	AS209	1.36%
GOV	0.20%	AS9318	1.18%

Table 8. Distribution of resolvers successfully resolving a `TXT` query with a response size of 11 KB on 2021-03-29 (Top 10)

Network type		AS distribution	
Type	#Resolvers	ASN	#Resolvers
ISP/MOB	58.51%	AS5617	32.29%
ISP	26.31%	AS4134	6.86%
DCH	8.11%	AS3462	2.17%
COM	4.71%	AS4766	1.33%
MOB	1.49%	AS131090	1.26%
EDU	0.39%	AS5384	1.24%
CDN	0.32%	AS56044	1.23%
GOV	0.07%	AS53006	1.01%
ORG	0.05%	AS3269	0.97%
SES	0.05%	AS37671	0.94%

Table 9. Distribution of resolvers successfully resolving an `ANY` query with a response size of 12 KB on 2021-03-29 (Top 10)

Network type		AS distribution	
Type	#Resolvers	ASN	#Resolvers
ISP/MOB	41.59%	AS5617	14.51%
ISP	37.56%	AS12389	9.80%
DCH	8.68%	AS4134	7.00%
COM	5.17%	AS4538	2.56%
EDU	3.35%	AS4812	2.29%
MOB	2.87%	AS4837	1.82%
SES	0.28%	AS3462	1.28%
CDN	0.24%	AS6805	1.24%
GOV	0.12%	AS3352	0.85%
ORG	0.11%	AS9269	0.82%

networks/ASes. We use IP2Location data [3] to determine the network type of resolvers. Moreover, we use RouteViews data [7] to map resolver IP addresses to autonomous systems. Table 7 shows the distributions of mappings in the entire open resolvers set. Tables 8 and 9 show the distributions for resolvers that successfully resolve queries with large responses (i.e., the 12 kB `ANY` and 11 kB `TXT` cases, respectively). These resolvers appear to be concentrated in a couple of ASes, while type of the networks that they are located does not deviate too much from the distribution for the entire open resolvers set.

Key Takeaway: TXT queries have the potential to take over ANY based amplification. Although not all open resolvers are capable of handling TXT queries, due to the legitimate use cases of TXT records (e.g., to publish domain verification tokens), TXT-based amplifications could be harder to mitigate than ANY-based ones.

5.6 TCP Fallback

In Sect. 3 we discussed the implications of TCP support between open DNS resolvers and authoritative nameservers. To explore this we send `TXT` and `ANY` queries, to which our authoritative nameserver is set to respond with a truncated answer. We then investigate queries for which we see (followup) TCP DNS queries on our authoritative nameserver. Our results show that, on average, 10.7% of open resolvers over the study period fallback to TCP for `TXT` queries when the answer is truncated. For `ANY` queries, 12.4% of queries result in a TCP fallback, as shown in Fig. 3. These numbers are way lower than those reported by Moura et al. [30] (80% of TCP fallback in IPv4). We suspect the main reason behind this difference to be the resolver set under inspection. While we focus on open resolvers which typically are not well configured, their study investigates DNS queries arriving at the authoritative nameservers which are not necessarily issued by open resolvers.

Key Takeaway: While truncation can be used as a mechanism to avoid returning large DNS responses during attacks, there is a small yet non-negligible number of resolvers that can still be misused to bring about harm as they fall back to TCP.

5.7 Caching

`A` Query Caching. To test how `A` queries are cached, we sent two consecutive `A` queries to open resolvers. We then investigated whether open resolvers contact the authoritative nameserver under our control for these queries. 37.77% of open resolvers on 2021-03-29 contact our authoritative nameserver only once while correctly resolving the two queries. We infer the presence of cache for this group of resolvers. Note that this is a lower bound estimation for caching open resolvers due to the existence of complex caching implementations (e.g., in case of public DNS resolvers) [33].

Table 10. TTL values returned by resolvers for a zero-TTL response on 2021-03-29 (Top 10)

TTL	Count
0	1647.0k
60	74.9k
1	43.7k
30	34.1k
5	27.7k
300	14.8k
3600	8.7k
600	6.0k
59	2.7k
10	1.9k

Key Takeaway: The fact that majority of open resolvers do not respond from cache provides an opportunity to deploy response rate limiting on authoritative nameservers or upstream resolvers as a measure to dampen the impact of amplification attacks.

`ANY` Query Caching. We examined caching behavior for `ANY` queries in two different ways. Initially we sent an `ANY` query for the same domain name sent in the `A` query cache test of the previous step. We did this to investigate how `ANY` queries are handled when a potential response exists in the cache. Our analysis reveals that only 6.9% of open resolvers (also on 2021-03-29) rely on their cache to resolve such a query and others contact our authoritative nameserver.

We performed another test to explore how open resolvers deal with a zero TTL response. To this end, we sent two `ANY` queries for which our authoritative nameserver is configured to return a response with a TTL value of zero. 26.22% of open resolvers contact our authoritative nameserver only once in order to respond to these queries. Thus, we infer that these resolvers use their cache in order to respond such queries, despite the zero TTL value. As discussed in Sect. 3.3, this would make a large group of open resolvers less effective when there is a record with a 0-TTL set for a zone. In Table 10 we show the distribution of TTL values returned by these caching resolvers. 88% of resolvers return the TTL value of zero, as set on our authoritative nameserver for the domain name.

Key Takeaway: While resolvers might respond to ANY queries from their cache, the vast majority of them do recursive resolution to respond to these queries, which makes response rate limiting an effective measure to limit their amplification power when ANY queries are misused.

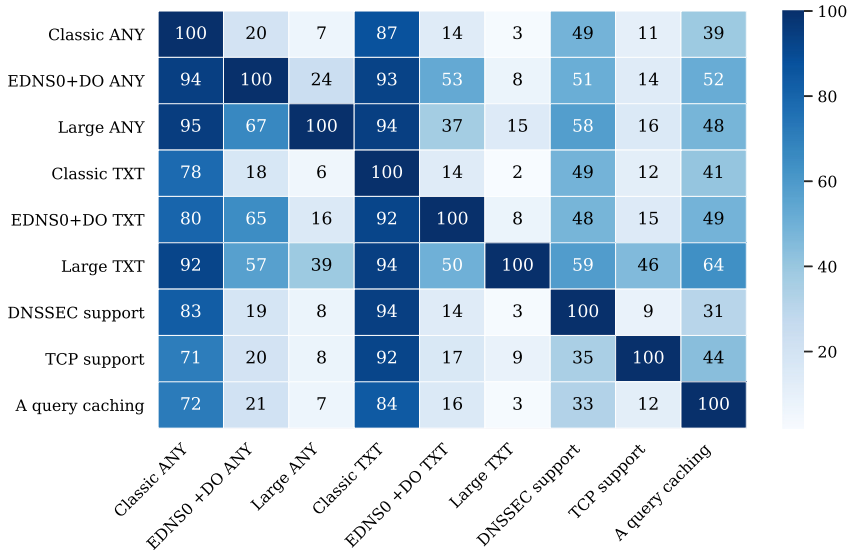

Fig. 4. Correlations between support for various DNS features on 2021-03-29

5.8 Feature Overlaps and Common Resolver Configurations

Now that we have experimented with all DNS features in question, we check which supported features mutually overlap among open resolvers that support them. We do this to get an initial insight into DNS protocol features that might be commonly supported together if we disregard all other features. A heatmap is given in Fig. 4. Each entry of this table represents the percentage of open resolver supporting the feature of the respective row as well as the corresponding column. At first glance, we see that the majority of open resolvers that support one of the features, also support Classic `ANY` and `TXT` queries. Note that the asymmetry seen in the table for mutual features is due to the difference in absolute number of open resolvers supporting each feature. For example, while 39% (roughly 21K out of 53K) of open resolvers that support large `TXT` queries also support large `ANY` queries, this fraction drops to only 15% (roughly 21K out of 140K) if we consider the reverse order. A similar pattern is seen when looking at EDNS0 enabled `ANY` and `TXT` queries with DO bit set. Surprisingly, this behavior is swapped when exploring classic `ANY` and `TXT` queries. In this case a larger portion of open resolvers that support classic `ANY` queries, also support classic `TXT` queries if we compare it to the reverse situation.

Open resolvers having similar software configurations, would intuitively exhibit similar behavior. To study common behaviors we group open resolvers that react similarly to our set of DNS queries in Table 5. To do so we use the set of features given in Table 11. Using these features, we put open resolvers with a common behavior (feature support) into the same group. We summarize 10

Table 11. Features considered for grouping open resolvers

Feature	Description
1	Resolver being still responsive at the start of the followup scans
2	Resolver responding to classic ANY queries
3	Resolver responding to EDNS0 enabled ANY queries
4	Resolver responding to EDNS0 enabled ANY queries with DO bit set
5	Resolver responding to EDNS0 enabled ANY queries with a large response
6	Resolver responding to classic TXT queries
7	Resolver responding to EDNS0 enabled TXT queries
8	Resolver responding to EDNS0 enabled TXT queries with DO bit set
9	Resolver responding to EDNS0 enabled TXT queries with a large response
10	Resolvers that support DNSSEC
11	Resolver that implement caching for A queries
12	Resolvers that fallback to TCP for TXT queries with a large response
13	Resolver that fallback to TCP for ANY queries with a large response

Table 12. Common groups of open resolvers (Top 10)

Group	Description	Count	Percentage
1	Resolver being still responsive at the start of the followup scans and support classic TXT and ANY queries and support DNSSEC	457.9k	17.3%
2	Resolver being still responsive at the start of the followup scans and support classic TXT and ANY queries	228.5k	8.7%
3	Resolver being still responsive at the start of the followup scans and support classic TXT and ANY queries and implement caching	227.0k	8.6%
4	Resolvers that disappear after our initial scan and are not responsive anymore	166.3k	6.3%
5	Resolver being still responsive at the start of the followup scans and implement caching	142.6k	5.4%
6	Resolver being still responsive at the start of the followup scans but do not support additional features	138.2k	5.2%
7	Resolver being still responsive at the start of the followup scans and support classic TXT and ANY queries, DNSSEC and implement caching	106.0k	4.0%
8	Resolver being still responsive at the start of the followup scans and support DNSSEC	78.7k	3.0%
9	Resolver being still responsive at the start of the followup scans and support classic TXT and ANY queries, implement caching and fallback to TCP for large ANY responses	44.8k	1.7%
10	Resolvers that disappear after our initial scan but are responsive to classic TXT and ANY queries later on	44.2k	1.7%

most common groups of resolvers in Table 12 which covers roughly 62% of open resolvers in our study.

Key Takeaway: The groups of open resolvers with the most common behavior offer a limited amplification factor and thus can be assigned a lower priority to be rooted out by operators.

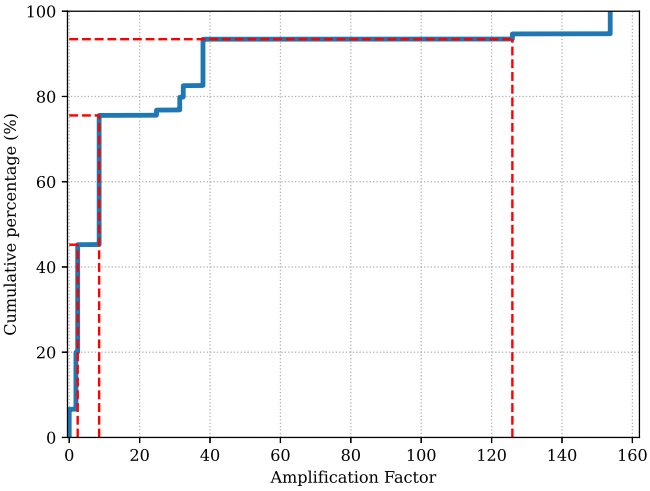

Fig. 5. Amplification factor CDF (considering relevant queries indexed in Fig. 4) for open resolvers on 2021-03-29

5.9 Ranking Open DNS Resolvers

As we have previously shown, each resolver reacts differently to amplification. To study the worst-case scenario, we assign to each open resolver the highest amplification factor that is supported by that specific resolver when responding to our set of queries (indexed in Table 5). This gives us an upper bound to the amplification potential for a given resolver. Figure 5 shows the CDF plot for supported amplification factors. We observe that roughly 6.5% of open resolvers offer an amplification factor of 125× or more. On the other hand, around 75% of open resolvers offer an amplification factor of 8.5× and a bit less than half of all open resolvers (45%) provide an amplification factor of only 2.4×. A non-negligible group of open resolvers (around 6%) do not respond to our followup queries, for which we assign an amplification factor of 0×. We are of course aware that these numbers depend on the configuration of the zone we set up for testing. However, we do not aim to present sharp borderlines on the amplification factor of open resolvers, but rather to differentiate among them.

We then proceed with ranking resolvers based on their highest amplification factor. At the basis of this study is the intuition that not all open resolvers will be equally potent in a DDoS attack. To quantify the usefulness of such a ranking, we group all open resolvers in our dataset based on their maximum amplification factor. We then calculate the maximum possible attack traffic contribution of each group of resolvers by multiplying the maximum amplification factor with the number of resolvers in that group. In Fig. 6 we derive the residual DNS amplification power if resolvers are rooted out, starting with the most powerful ones. Figure 6 shows that removing the top 6.5% of open resolvers halves the cumulative amplification power, while removing the top 25% of resolvers further

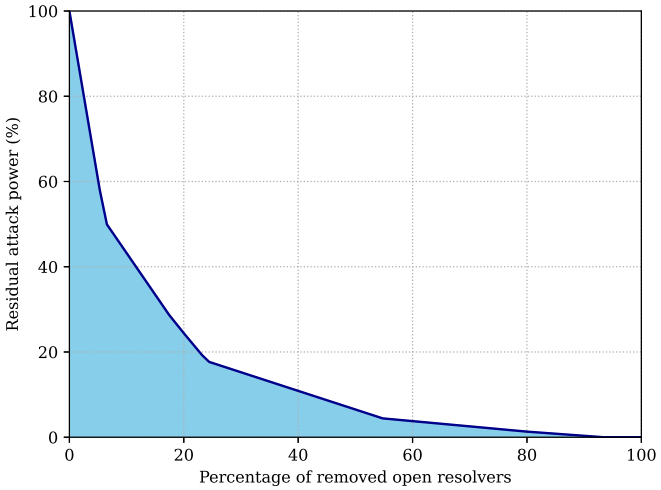

Fig. 6. Residual attack power when rooting out open resolvers based on their amplification rank

reduces the amplification power to only 18%. This confirms our initial intuition that not all resolvers are equally potent. Moreover, it clearly indicates that a priority-based strategy in open resolvers takedown could be very effective in taking the edge off of DNS-based R&A attacks.

Key Takeaway: The majority of open resolvers only offer limited amplification power, while a small group of resolvers is causing very large amplification. This creates an opportunity for operators to prioritize their efforts in discriminately eradicating open resolvers starting with the most-powerful ones.

6 Conclusion

To this day, DNS (R&A) attacks remain one of the most-used forms of DDoS attack. In this type of attack, misconfigured open DNS resolvers are misused to typically send large DNS responses to victims. Despite community efforts, going back well over a decade, to reduce the number of open DNS resolvers on the Internet, millions remain online and open today. This led us to ask: *are all open resolvers equal, in terms of how they can be abused for attacks, or can we identify traits of open DNS resolvers that make them more potent attack vectors?*

By using domain knowledge about how the DNS works, and what features can be misused in R&A attacks, we created a measurement setup that allows us to identify the attack potential of open resolvers. Our results show that – like many phenomena on the Internet – attack potential shows a long-tailed distribution, with a fraction of open resolvers responsible for the vast majority of the attack potential. With this outcome, operators can prioritise takedowns of open resolvers and focus on the most potent ones first. With the scarce time

that they have, if they were to focus on just the 20% most potent amplifiers, they could reduce the Internet-wide attack potential by up to 80%, making it much more challenging for attackers to bring about crippling DNS R&A attacks.

Acknowledgments. We would like to thank the anonymous PAM reviewers for their valuable feedback on our paper. This research is funded by the EU H2020 projects CONCORDIA (#830927) and partially funded by SIDNfonds.

References

1. 2.5Tbps DDoS Attack on Google. https://cloud.google.com/blog/products/identity-security/identifying-and-protecting-against-the-largest-ddos-attacks. Accessed 11 Jan 2022
2. dnspython. https://www.dnspython.org/. Accessed 11 Jan 2022
3. IP2Location. https://www.ip2location.com/. Accessed 11 Jan 2022
4. MassDNS, A high-performance DNS stub resolver. https://github.com/blechschmidt/massdns. Accessed 11 Jan 2022
5. Open Resolver Project. https://web.archive.org/web/20200603050044/http://openresolverproject.org/. Accessed 11 Jan 2022
6. The Measurement Factory. http://dns.measurement-factory.com/surveys/openresolvers.html. Accessed 11 Jan 2022
7. University of Oregon Route Views Project. http://www.routeviews.org. Accessed 11 Jan 2022
8. ZIterate, ZMap IP permutation generator. https://github.com/zmap/zmap/blob/main/src/ziterate.1.ronn. Accessed 11 Jan 2022
9. Abley, J., Gumundsson, Ó., Majkowski, M., Hunt, E.: Providing minimal-sized responses to DNS queries that have QTYPE=ANY. RFC 8482, January 2019. https://doi.org/10.17487/RFC8482, https://rfc-editor.org/rfc/rfc8482.txt
10. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: RFC 4033 - DNS security introduction and requirements (2005). http://tools.ietf.org/html/rfc4033
11. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: RFC 4034 - resource records for the DNS security extensions (2005). http://tools.ietf.org/html/rfc4034
12. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: RFC 4035 - protocol modifications for the DNS security extensions (2005). http://tools.ietf.org/html/rfc4035
13. Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., Gont, F.: RFC 8900 - IP fragmentation considered fragile (2020). https://www.rfc-editor.org/info/rfc8900
14. Constantin, L.: Attackers use DNSSEC amplification to launch multi-vector DDoS attacks (2016). http://www.computerworld.com/article/3097364/security/attackers-use-dnssec-amplification-to-launch-multi-vector-ddos-attacks.html
15. Damas, J., Graff, M., Vixie, P.: RFC 6891 - extension mechanisms for DNS (EDNS(0)) (2013). http://tools.ietf.org/html/rfc6891
16. Deccio, C., Hilton, A., Briggs, M., Avery, T., Richardson, R.: Behind closed doors: a network tale of spoofing, intrusion, and false DNS security. In: Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, pp. 65–77 (2020). https://doi.org/10.1145/3419394.3423649
17. Durumeric, Z., Bailey, M., Halderman, J.A.: An internet-wide view of internet-wide scanning. In: 23rd USENIX Security Symposium (USENIX Security 2014), pp. 65–78 (2014)

18. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: fast internet-wide scanning and its security applications. In: Proceedings of the 22nd USENIX Security Symposium, pp. 605–619 (2013)
19. Fachkha, C., Bou-Harb, E., Debbabi, M.: Fingerprinting internet DNS amplification DDoS activities. In: 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–5. IEEE (2014)
20. Hendriks, L., de Oliveira Schmidt, R., van Rijswijk-Deij, R., Pras, A.: On the potential of IPv6 open resolvers for DDoS attacks. In: Kaafar, M.A., Uhlig, S., Amann, J. (eds.) PAM 2017. LNCS, vol. 10176, pp. 17–29. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54328-4_2
21. Jiang, J., Liang, J., Li, K., Li, J., Duan, H., Wu, J.: Ghost domain names: revoked yet still resolvable (2012)
22. Korczyński, M., Nosyk, Y., Lone, Q., Skwarek, M., Jonglez, B., Duda, A.: Don't forget to lock the front door! inferring the deployment of source address validation of inbound traffic. In: Sperotto, A., Dainotti, A., Stiller, B. (eds.) PAM 2020. LNCS, vol. 12048, pp. 107–121. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44081-7_7
23. Krämer, L., et al.: AmpPot: monitoring and defending against amplification DDoS attacks. In: Bos, H., Monrose, F., Blanc, G. (eds.) RAID 2015. LNCS, vol. 9404, pp. 615–636. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26362-5_28
24. Kührer, M., Hupperich, T., Bushart, J., Rossow, C., Holz, T.: Going wild - large-scale classification of open DNS resolvers. In: Proceedings of the 2015 ACM Internet Measurement Conference - IMC 2015, pp. 355–368. ACM Press, New York (2015). https://doi.org/10.1145/2815675.2815683, http://dl.acm.org/citation.cfm?doid=2815675.2815683
25. Kührer, M., Hupperich, T., Rossow, C., Holz, T.: Exit from hell? Reducing the impact of amplification DDoS attacks. In: 23rd USENIX Security Symposium (USENIX Security 2014), pp. 111–125 (2014)
26. Laurie, B., Sisson, G., Arends, R., Blacka, D.: RFC 5155 - DNS security (DNSSEC) hashed authenticated denial of existence (2008). http://tools.ietf.org/html/rfc5155
27. Leverett, E., Kaplan, A.: Towards estimating the untapped potential: a global malicious DDoS mean capacity estimate. J. Cyber Policy **2**(2), 195–208 (2017)
28. Mockapetris, P.: RFC 1035 - domain names - implementation and specification (1987). http://tools.ietf.org/html/rfc1035
29. Moon, S.J., Yin, Y., Sharma, R.A., Yuan, Y., Spring, J.M., Sekar, V.: Accurately measuring global risk of amplification attacks using AmpMap. Technical report, Technical report CMU-CyLab-19-004 (2020)
30. Moura, G.C.M., Müller, M., Davids, M., Wullink, M., Hesselman, C.: Fragmentation, truncation, and timeouts: are large DNS messages falling to bits? In: Hohlfeld, O., Lutu, A., Levin, D. (eds.) PAM 2021. LNCS, vol. 12671, pp. 460–477. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72582-2_27
31. Nawrocki, M., Jonker, M., Schmidt, T.C., Waehlisch, M.: The far side of DNS amplification: tracing the DDoS attack ecosystem from the internet core. In: Proceedings of the 2021 ACM Internet Measurement Conference (IMC 2021) (2021). https://doi.org/10.1145/3487552.3487835
32. Park, J., Khormali, A., Mohaisen, M., Mohaisen, A.: Where are you taking me? Behavioral analysis of open DNS resolvers. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 493–504. IEEE (2019)

33. Randall, A., et al.: Trufflehunter: cache snooping rare domains at large public DNS resolvers. In: Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, pp. 50–64 (2020). https://doi.org/10.1145/3419394.3423640
34. van Rijswijk-Deij, R., Sperotto, A., Pras, A.: DNSSEC and its potential for DDoS attacks. In: Proceedings of ACM IMC 2014. ACM Press, Vancouver (2014). https://doi.org/10.1145/2663716.2663731
35. Rossow, C.: Amplification hell: revisiting network protocols for DDoS abuse. In: Proceedings of the 2014 Network and Distributed Systems Security Symposium (NDSS 2014), no. February, pp. 23–26. Internet Society, San Diego (2014). http://www.internetsociety.org/sites/default/files/01_5.pdf
36. Rudman, L., Irwin, B.: Characterization and analysis of NTP amplification based DDoS attacks. In: 2015 Information Security for South Africa (ISSA), pp. 1–5. IEEE (2015)
37. Santanna, J.J., et al.: Booters - an analysis of DDoS-as-a-service attacks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 243–251. IEEE, Ottawa, May 2015. https://doi.org/10.1109/INM.2015.7140298
38. van der Toorn, O., Krupp, J., Jonker, M., van Rijswijk-Deij, R., Rossow, C., Sperotto, A.: ANYway: measuring the amplification DDoS potential of domains. In: 2021 17th International Conference on Network and Service Management (CNSM) (2021)
39. Vixie, P., Schryver, V.: DNS response rate limiting (DNS RRL). Technical report (2012). https://web.archive.org/web/20160307112057/, http://ss.vix.su/~vixie/isc-tn-2012-1.txt. Accessed 11 Jan 2022
40. Wan, G., et al.: On the origin of scanning: the impact of location on internet-wide scans. In: Proceedings of the ACM Internet Measurement Conference, pp. 662–679 (2020)