Amplification Detection: Determining DDoS Abuse Potential of Your Network

A Quantitative Study of the Amplification Potential of Three Popular Protocols

Piotr Politowicz

Supervisors: Georgios Smaragdakis, Harm Griffioen

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Piotr Politowicz
Final project course: CSE3000 Research Project
Thesis committee: Georgios Smaragdakis, Harm Griffioen, George Iosifidis

Abstract

Amplification Distributed Denial of Service attacks require networks that do not drop packets with spoofed IP addresses and servers open to the Internet running UDP protocols with amplification potential. These attacks have the potential to overwhelm large network links and disrupt the availability of the largest network infrastructures. While multiple studies exist addressing vulnerable protocols and exploring solutions for collaborative mitigation of such attacks, there is the unaddressed issue of stopping exploitable servers from being deployed in the first place. In this paper, we investigate such servers located in Sweden running DNS, NTP, and Memcached services to learn what makes them vulnerable. After analyzing thousands of servers, we give insights into the current state of the Swedish network and find multiple DNS recursive resolvers amplifying bandwidth more than 100 times, authoritative DNS servers hosting domains with huge amounts of data, and NTP servers patched against old attacks but still providing significant amplification. We find that important parameters that shape the large DNS amplifiers include the choice of the EDNS buffer size, the truncation of responses exceeding it, and a lack of the "ANY" query limitation. NTP servers with debug commands accessible from the Internet have amplification potential, and no vulnerable Memcached servers were found in Sweden.

1 Introduction

Amplification Distributed Denial of Service (DDoS) attacks have recently demonstrated the capability to achieve bandwidths in the Tb/s range, potent enough to disrupt the availability of large network infrastructures. In these attacks, adversaries trick publicly accessible servers into overwhelming victims with large amounts of traffic that could be legitimate, making detection difficult. The significant traffic volume is triggered by small requests amplified by the servers. This increase in volume compared to the original size of requests is measured by the amplification factor, which attackers aim to maximize. Historically, the bandwidth generated by these attacks grew significantly from 300 Gbps during the Spamhaus attack in 2013 [1] to 2.54 Tbps in the attack on Google in 2017 and 2.3 Tbps in the AWS attack in 2020 [2] [3]. This worsening danger creates the need to detect vulnerable servers before they are made public and abused by attackers, fixing the problem at its root as soon as possible. This could be done if the parameters of successful attacks are identified which could lay the foundations for automated tooling made to aid network engineers.

The purpose of this paper is to investigate how estimating the amplification factor and necessary parameters for successful amplification attacks in Swedish network infrastructures can improve the detection of exploitable systems. Sweden was randomly assigned to us as part of how the study was conducted. It will address the following sub-questions:

1. How to identify potential amplifiers in a given network and estimate the amplification factors?
2. What parameters affect the attack's success?
3. How can the identified factors help detect infrastructure prone to abuse?

We look into three protocols with a history of use in DDoS amplification attacks: Domain Name System (DNS), Network Time Protocol (NTP), and Memcached. They were attributed to large attacks and are now often used in "multi-vector attacks" where attackers combine multiple protocols to maximize amplification [4]. Our main observation is that there are still many misconfigured servers in Sweden that are ready to be used in amplification attacks. The contributions of this work can be summarized as follows:

1. We reveal numerous DNS servers in Sweden as amplifiers with a bandwidth amplification factor (BAF) exceeding 50, 75, and even 100. We show that recursive resolvers are mostly responsible for large amplification and identify parameters that make servers more likely to provide high amplification such as Extension Mechanisms for DNS (EDNS) buffer sizes of 4,096- and 4,000-bytes, the implementation of cryptographic signatures, and a lack of the "ANY" query limitation.
2. We show that "ANY", "RRSIG", and "DNSKEY" query types are responsible for the highest amplification and show correlations between servers' operation systems and versions and their amplification.
3. We find that almost no NTP servers in Sweden are vulnerable to the "monlist" DDoS attack, but other debug commands exist that make them better amplifiers than DNS servers by providing amplification ranging from 6 to 92 BAF.

The rest of the paper is structured as follows. In Section 2, the background of amplification attacks is provided along with related work in Section 3. Section 4 describes the datasets used to conduct experiments and Section 5 follows with details of the approach taken to obtain the results. Section 6 explores the ethical implications of the research. The results of this study are detailed in Section 7, followed by a discussion in Section 8. The paper concludes with a summary of conclusions and future work in Section 9.

2 Background

Amplification Denial of Service (DoS) attacks are still some of today's most common and devastating cyber-attacks [5] [6]. In an amplification attack, an attacker needs several components to bring down its victim and cause substantial damage. Two are a protocol that can be weaponized based on its amplification potential and a server running a service that uses it. This server called a reflector or an amplifier is used by the attacker to overwhelm the victim by sending requests with a spoofed IP address, pretending that they come from the victim. The amplifier responds to the requests and reflects the traffic onto the victim, overwhelming its network links and deteriorating its availability.

An amplification attack can be done with a protocol that does not require the attacker to send more than the initial request to elicit a significantly larger response. This is because any responses to the initial request will be sent to the victim of the attack, making the attacker unable to respond to them. The natural choice are protocols built on the User Datagram Protocol (UDP) which uses a simple connectionless communication model.

2.1 BAF

To measure amplification, we define the Bandwidth Amplification Factor (BAF) identically to Rossow [5]. BAF can be calculated by comparing the size of the UDP payloads in the responses that the amplifier sent to the victim to the size of the UDP payload of the request that triggered them:

$$BAF = \frac{\textit{len(UDP payload) amplifier to victim}}{\textit{len(UDP payload) attacker to amplifier}} \quad (1)$$

The headers encapsulating the UDP payload are not included in the calculation to keep the results valid across changes made to the protocols.

2.2 DNS

The Domain Name System (DNS) is a simple query-response protocol that provides a naming system for resources on the Internet. Its perhaps most common function is mapping fully qualified domain names such as "example.com" used by users to IP addresses, a format understood by routing infrastructure. There are many other resource types besides IP addresses such as text records, mail exchange servers, or cryptographic signatures. The larger the resource stored on the server, the larger the response and produced amplification. In this study, we investigate resource types producing large amplification and the "ANY" query type that, when unrestricted, asks a DNS server to return all resource records (RRs) stored for a given domain.

We focus on authoritative and recursive DNS servers. Authoritative servers know the content of their DNS zone and can respond to queries directly with requested RRs [7]. They are the main source of amplification: we require large RRs or the option to retrieve all of them with the "ANY" query. Recursive servers receive queries from clients and respond to them from a local cache or send queries to other servers to obtain the answers. They can be considered middlemen that provide large amplification given a domain name with a large amount of data assigned to it on its authoritative server.

DNS servers use a buffer size to define the maximum size of a DNS message that can be sent over UDP. If a response exceeds this size, it should be truncated and the full response should be offered over TCP [8]. The introduction of cryptographic signatures for RRs, such as those used in DNSSEC [9], has made the default buffer size of 512 bytes insufficient to contain all necessary fields. This limitation necessitated the development of mechanisms like EDNS0 (Extension Mechanisms for DNS), as specified in RFC 6891 [10]. EDNS0 allows DNS messages to exceed the 512-byte limit by specifying a larger UDP payload size in the OPT pseudo-RR.

The following extension of the buffer size default to 4,096 bytes inflated response sizes significantly. It raised the chance of IP fragmentation since their size can exceed the common 1,500-byte MTU on the Internet [11]. This heightens the risk of transmission failures and potential DNS cache poisoning attacks, where an attacker corrupts the cache by providing incorrect DNS responses, which are then cached and served to clients by DNS servers [12]. To counter it, DNS Flag Day 2020 proposed that the buffer size defaults in DNS software should reflect the minimum safe size of 1,232 bytes [13]. This prevents fragmentation on most of the network while preventing truncation of most responses. Naturally, a smaller buffer size reduces the amplification potential of a server, thereby diminishing the risk of exploitation by attackers. However, some servers do not switch to TCP when the response to a user's query exceeds the configured buffer size. When this buffer is set to 1,232 bytes, the configuration does not fully comply with DNS Flag Day 2020 recommendations, which specify that authoritative servers must not send responses larger than the advertised buffer size over UDP.

2.3 NTP

Network Time Protocol (NTP) is used to synchronize time between time servers and clients on the network. In a usual packet exchange, it is not a source of large amplification. NTP defines 8 values for association modes that can be set in the NTP header with mode 7 specified to be "reserved for private use" [14]. This mode allows server administrators to troubleshoot or gather diagnostic information about NTP servers, but some commands can produce large responses and create amplification risk. Two queries of interest are "monlist" and "peerlist" which ask the server to return the list of its peers and the list of the previous 600 clients that contacted the server, respectively. Both have the potential to return a lot of data in response to just a single packet, but only "monlist" has been widely reported in attacks [15].

2.4 Memcached

Memcached is an in-memory data store used to speed up applications by decreasing the load placed on databases [16]. It is used to store and retrieve items from its memory. Publicly accessible Memcached servers operated using UDP can become a source of amplification. An attacker can implant a large payload of data on the server and then retrieve it with a spoofed IP address to redirect large responses to the victim.

3 Related Work

Various UDP-based protocols could be used in an amplification DDoS attack. An overview of 14 protocols with amplification potential is described by Rossow [5]. It turns out that multiple protocols, both commonly used nowadays and legacy ones, can be used to obtain the average amplification factor ranging from 4.9 up to an astonishing 4,670. However, 10 years after the paper was released, more security measures have been implemented, and the number of vulnerable servers decreased drastically. Besides DNS and NTP covered by Rossow, this paper also analyzes the Memcached protocol that rose to its fame in 2018 [17], 4 years after the paper was published.

Looking closer at DNS, one of the most widespread protocols and the protocol with natural amplification, Van der

Toorn et al. [18] estimated the potential of DNS amplifiers when querying different domain names and evaluated the effectiveness of solutions proposed to block DNS-based DoS attacks. Limiting or blocking "ANY" queries significantly reduces the size of responses for the majority of servers, but it is still possible to find "next-best" query types such as "DNSKEY" or "TXT" that yield smaller, but notable amplification. We consider these query types when scanning servers and searching for the highest possible amplification.

Herzberg and Shulman [12] presented DNS cache poisoning attacks circumventing widely deployed defenses and showed the significance of minimizing response sizes to avoid the fragmentation of IP packets. They describe how DNSSEC was implemented to authenticate DNS responses, but because of the large size of cryptographic signatures, they significantly enlarge them and enable other attacks. The resulting consensus in the IP and DNS communities to avoid IP fragmentation in DNS [19] gave rise to recommendations proposed by the DNS Flag 2020 document [13]. It is advised to configure DNS servers to limit UDP responses to 1,232 bytes and switch to TCP when a larger size is needed. Moura et al. [19] gives insights into the state of implementation of these recommendations. They show that the 4,096-byte buffer is still the most common, but the number of servers implementing the recommended 1,232-byte buffer is growing. They further show that the large majority of responses are smaller than 1,232 bytes and their sizes depend on more than just resource records.

Finally, Wagner et al. [6] investigated the mitigation potential of distributed attack-mitigation platforms that collaborate with each other. While there is a global effort to detect and prevent DoS attacks on a network layer in peering locations such as Internet Exchange Points, more research needs to be done to develop a systematic approach to detecting amplifiers. This could help build tooling to facilitate the detection of vulnerable servers in network infrastructures and fix the problem at its root whenever possible.

4 Datasets

For our study, we used the Censys Internet Map [20] with research access to collect IP addresses of servers in Sweden with open ports assigned to protocols researched in this paper. Next to this, we complemented the list of DNS servers with addresses resolved from Swedish domain names.

4.1 Censys

We got access to thousands of IP addresses scanned by Censys that were identified to be running a DNS, NTP, or Memcached service. Censys regularly scans the whole IPv4 space to present the most accurate representation of the Internet's current state [21]. Since services with observation older than 24 hours are rescanned, we can assume that the provided data is up-to-date and represents most of the publicly accessible servers located in Sweden.

We used the Censys Python Library [22], a wrapper for Censys API, to obtain a list of machines matching the search queries. The query consists of two parts - a location: we specify that we are only interested in servers with a geographic location in Sweden, and a service name: a different protocol name is specified for each query. This resulted in 42,131 DNS, 48,814 NTP, and 675 Memcached servers. For an unknown reason, these numbers are slightly lower than the number of results presented directly on the website. Nevertheless, they represent at least 96% of identified servers which is a large majority, but it is possible that some vulnerable servers were missed because of it.

4.2 Resolved Domain Names

Two sources of domains were used to identify authoritative name servers and their domains. Firstly, four different lists representing the top one and the top ten million most popular domains [23]: The Majestic Million, Cisco Umbrella Top 1 Million, Alexa Top 1 Million, and Open PageRank Top 10 Million. Secondly, we retrieved domains with the "se" country code top-level domain (ccTLD) from the Common Crawl [24] dataset (crawl "CC-MAIN-2023-17") using the CDX Index Client tool [25]. Both sources combined resulted in a list of 440,811 .se domains. However, because of the age of data, some of the domains were not active anymore. To find the authoritative servers behind these domains, the NS record lookup was performed. The NS record stands for "nameserver" and indicates which DNS servers are authoritative for the queried domain [8].

One domain can have multiple authoritative servers located in different parts of the world. Because of this, the geolocation of each IP address had to be estimated which was done with IPInfo [26]. Initially, 15,413 unique DNS authoritative servers used by .se domains were identified. Interestingly, only 3,699 of them (24%) were located in Sweden according to IPInfo. It is important to note that the geolocation of IP addresses is sometimes inaccurate [27], thus this number may not be entirely precise.

5 Methodology

The common factor when looking for a good source of amplification is finding request types that make servers produce large responses. The approach is therefore similar but differs between protocols as described in this section. Each server is queried with specific requests to estimate BAF and retrieve additional information found in the responses. Additionally, the Autonomous System Number (ASN) and operating system information of every server is saved if found in data provided by Censys. After measuring BAF, we define amplifiers as servers that produced BAF larger than 1 for at least one of the requests. These servers are then analyzed in the results section. All requests made to measure BAF were crafted with Scapy, a "powerful interactive packet manipulation library written in Python" [28]. All code used to perform experiments is available in the public repository [29].

5.1 DNS

With the list of IP addresses of authoritative servers, we query different record types of 10 randomly selected domains per server (or fewer if 10 is not available) and calculate the amplification factor received for each. To identify record types with the largest amplification potential, we performed a smaller

experiment where we checked which resource records produce the largest response per server. From there, we identified 5 types of interest: "RRSIG", "TXT", "DNSKEY", "NS", and "MX". Naturally, if a server does not limit the "ANY" query, it will return all of the types and produce the largest response. Otherwise, we query the identified types. Moreover, the "DNSKEY" and "TXT" query types were found to produce large responses and be the next "best choice" when looking for amplification when "ANY" is limited [18].

Next, we identify recursive resolvers among all DNS servers found. We start by sending a simple "A" query with a known domain name such as "google.com" to elicit a response from a DNS server. If the "recursion available" ("ra") flag is set in the response, we mark this server as recursive and pass it to the next stage. The idea is to attempt to resolve a domain with a lot of data assigned to it and produce a large amplification. With a list of ten "best domains" identified in the experiment with authoritative servers, we choose two whose all authoritative servers return a large amplification, randomly select 1 out of 8 remaining, and send queries of types "ANY", "DNSKEY", and "TXT" for every selected domain to the recursive server. Since multiple servers were found to need more than 2 seconds to respond, in a second experiment we resend queries that did not receive a response with a timeout of 6 seconds and merge the results. Finally, we select the largest response to calculate the BAF for this server.

Besides BAF, for both types of servers, we save the indicated ENDS buffer size if specified in the DNS response in the OPT psuedo-RR that contains information about EDNS relevant to the communication [10]. These buffers are used to verify if servers truncate responses exceeding the buffer size. For this, we calculate and compare the theoretical maximum amplification of a limited response with the actual one. This limited amplification is computed using Equation 1, assuming the buffer size is the maximum payload from the amplifier to the victim. We are also interested in fingerprinting the servers by their versions which we do in two different ways:

1. Firstly, using the DNS fingerprinting tool called "fpdns" [30] with version 0.10. It attempts to determine versions of DNS servers by sending different queries and comparing their responses. However, since version 0.10 was released at least 10 years ago, it may be inaccurate.
2. The second approach is to query the TXT record of Chaosnet (CH) class named "version.bind" which should be answered with a string containing information about the server. Many operators wisely choose not to do this and erase the record or change it to meaningless information, which also impacts our results.

5.2 NTP

We investigate if NTP servers respond to selected queries sent using packets with mode 7 which is used for debugging commands. A preliminary scan on a sample of 1,000 servers revealed five queries listed in Table 1 to which servers responded with a variable amplification. We use these queries together with the two "monlist" commands to check the highest possible BAF that could be obtained from every server.

Query	Description
peerlist	list of configured peers
peerlist summary	summarized version of peerlist
sys info	operational summary of the server
peerlist stats	peer memory usage statistics
restrict list	access control list of the server

Table 1: A list of identified queries for mode 7 NTP requests

Besides the mode and command, we set the version in the NTP header to 2 which should allow for interaction with servers running versions 2, 3, and 4.

Given a list of amplifiers, we attempt to retrieve their versions with the "ntpq" tool [31]. We use its "rv" control message command to retrieve all variables from the system variables namespace. If the "version" variable is found in the returned set, we save it as the server's version.

5.3 Memcached

Memcached servers support two main formats of communication between clients and servers: text-based and binary. They also support both TCP and UDP communication protocols, with TCP preferred in newer versions. To perform the experiments, we first attempt to connect to every server over TCP and request its version. This check could help identify outdated servers serving users over UDP by default.

To measure amplification, we send the "stats" command over UDP to retrieve statistics and other internal data [32] as this query is known to elicit large responses. These responses, if present, are used to calculate the BAF for this server. If the server turns out to be operable, we could further attempt to maximize the amplification by implanting large entries and retrieving them in subsequent requests.

6 Responsible Research

Adhering to strict ethical guidelines and emphasizing reproducibility is crucial to the integrity and credibility of our research. This approach ensures our findings are reliable and can be validated by other researchers.

6.1 Ethics

We do not perform active scanning of the entire internet range, which could disturb the operation of some systems. Instead, we use data obtained from Censys, a reputable source that conducts its own scanning activities responsibly. Furthermore, we do not perform any form of attacks or exploitations; our work is purely analytical. To avoid overwhelming servers or network links, we implement rate limiting in our requests. This ensures that our research does not disrupt the normal operation of the systems we study, adhering to the principle of minimizing harm as outlined in the Netherlands Code of Conduct for Research Integrity [33]. This approach aligns with the principles of responsibility and scrupulousness, ensuring that our research is conducted ethically.

6.2 Reproducibility

The ability to reproduce results is key to scientific validation and further study. Consequently, we have included additional

Protocol	n	%	BAF		
			all	50%	10%
DNS_{NS}	3,476	N/A	15.13	25.28	61.48
DNS_{OR}	2,372	5.63%	36.46	59.20	113.79
NTP	445	0.91%	95.01	173.48	772.20

Table 2: Bandwidth Amplification Factors per protocol. *n* is the number of servers that provided amplification larger than 1 and *%* represents the fraction of these servers compared to all active ones. *all* shows the average BAF of all amplifiers, *50%* and *10%* represent the average BAF of servers that were in the top 50% and top 10%, respectively, when compared for the largest amplification.

information on how we conducted our experiments in the appendix section of this paper. This includes the specific queries used to obtain IP addresses from Censys and snippets of code employed to perform our experiments.

The full version of code provided in the public repository [29] is well-documented and designed to be user-friendly, requiring only the input of IP addresses and domains for DNS servers to replicate our experiments. By following the instructions in the repository, other researchers can reproduce our findings and verify the results. However, subsequent results will likely differ from the ones described in this paper because of the rapidly changing state of the Internet. This commitment to transparency ensures that our research process is clear and verifiable, supporting the principles of transparency and honesty as specified in the Code of Conduct mentioned previously.

7 Results

In this section, we follow the per-protocol approaches described in Section 5 with datasets from Section 4 and present the results of experiments. All of the experiments were performed from May 15 to June 15 2024 with the newest results presented here if an experiment was repeated. Further details are given in the following subsections.

Table 2 gives an overview of amplification obtained from scanned servers: authoritative name servers (DNS_{NS}), open resolvers (DNS_{OR}), and NTP servers. Since attackers can choose to use only servers with the highest BAFs, we include the average amplification for two subsets of all servers. The largest set of potential amplifiers, servers that produced BAF larger than 1, is in the authoritative name servers for DNS. The average BAF, however, is smaller than that of other services. Open resolvers that indicated readiness to resolve our requests are a small part of retrieved servers (5.63%). Due to the strategy in choosing domain names to resolve, they produce about twice as large amplification as DNS_{NS}. The protocol with the smallest number of amplifiers is NTP, but its produced amplification is significantly larger than that of other servers. When choosing the top 10% servers ranked by produced BAFs, an attacker can obtain an amplification almost 7 times larger than with open resolvers.

Next, we proceed to discuss the findings specific to each protocol. Each has parameters that may help us identify the weak points and characteristics that adversaries could look for when collecting servers for their attacks.

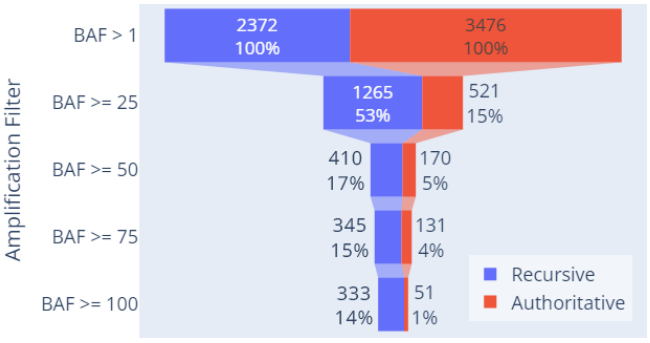


Figure 1: DNS - Distribution of servers based on the BAF produced.

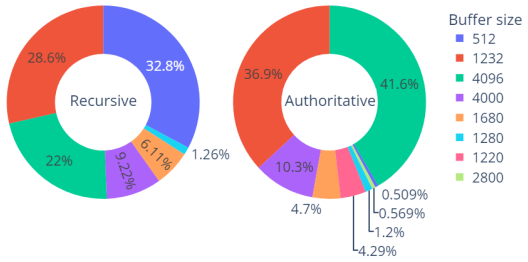


Figure 2: DNS - Distribution of EDNS buffer sizes indicated by at least 10 servers.

7.1 DNS

Running scans on authoritative and recursive DNS servers revealed the distribution shown in Figure 1. For authoritative servers, only 15% produced amplification of at least 25. Most resolved domains return minimal data, which is of little use to attackers. A smaller number of servers achieve higher BAFs, with 170 producing amplification of at least 50, and 51 reaching a BAF of 100 and more. Conversely, recursive resolvers present a different scenario, with 410 servers producing significant amplification of at least 50. These servers resolve and return most records associated with queried domains. Notably, 333 of 2,372 servers produced a BAF of at least 100, indicating that a substantial number of open resolvers return all data associated with large domains identified in the authoritative server scans.

The pie charts in Figure 2 provide a comparison of buffer sizes used by recursive and authoritative DNS servers. For recursive servers, the most common buffer size is 512 bytes, the default before the EDNS0 extension, making up 32.8% of the total. This is followed by buffer sizes of 4,096 bytes (22%) and 1,232 bytes (28.6%), indicating support for EDNS and implementation of recommendations from the DNS Flag Day 2020. 292 recursive servers (12.3%) did not return a buffer size. On the other hand, authoritative servers were identified to use 8 distinct buffer sizes, the most common being 4,096 bytes accounting for 41.6% of the total. This is followed by 1,232 bytes (36.9%), showing a significant preference for one of these sizes. 122 authoritative servers (3.5%) did not return a buffer size. Interestingly, the usage of the old buffer size of 512 bytes (0.509%) is tiny in contrast to recursive servers where it is the most common. For both types of servers, we find instances implementing the recommended buffer size of 1,232, but different configurations still outnumber them.

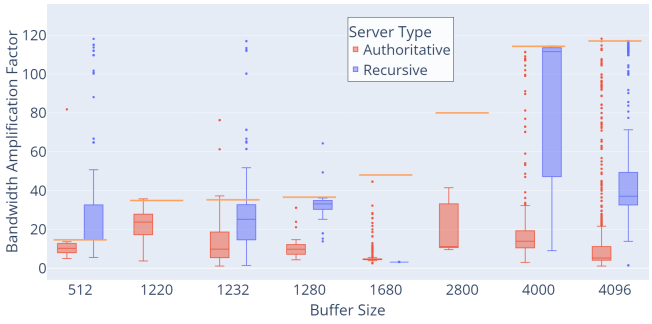


Figure 3: DNS - BAFs recorded per indicated EDNS buffer size. Orange lines show the theoretical maximum BAF if responses are truncated at the buffer size.

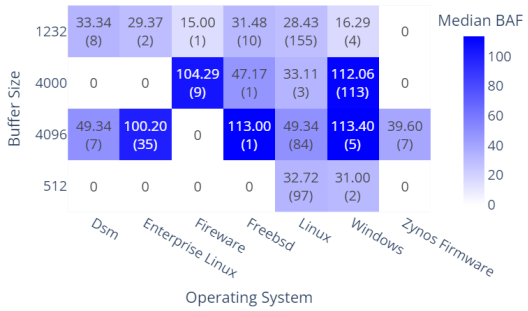


Figure 4: DNS - Heatmap showing the Median BAF of recursive resolvers across EDNS buffer sizes and operating systems. The color intensity indicates the magnitude of BAF, with darker blue representing higher values. In parenthesis, the number of servers matching the filters.

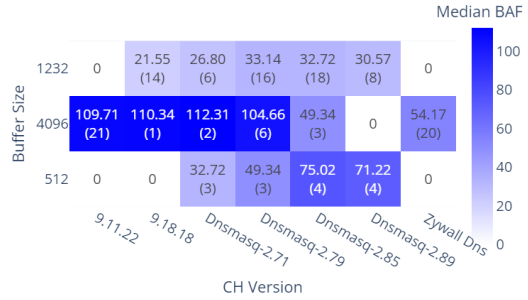


Figure 5: DNS - Heatmap showing the Median BAF of recursive resolvers across EDNS buffer sizes and versions retrieved from the TXT record of CH class. Versions 9.11.22 and 9.18.18 belong to the BIND suite.

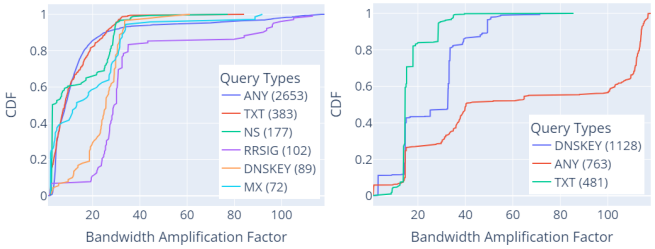
We next investigate whether the buffer size indicated by the server provides any hints about the expected amplification. Looking at Figure 3, for recursive servers, the majority of large amplifiers indicate buffer sizes of 4,000 and 4,096 bytes. Over 75% of servers with a buffer size of 4,000 produce a BAF higher than 40, with a median BAF of around 111. Similarly, for a buffer size of 4,096, almost 50% of servers achieve amplification of at least 40 BAF. For all other buffer sizes, more than 75% of servers do not exceed a BAF of 40. Notably, 126 servers with the 1,680-byte buffer consistently returned amplification in the narrow range of 3.17 to 3.24. It was discovered that they were all hosted within the same autonomous system and always returned a single Start of Authority (SOA) record with the name `a.misconfigured.powerdns.server.hostmaster`. This indicates that these servers were likely not properly configured, as this is the default value for the SOA record in PowerDNS 4.0 Authoritative Servers [34]. Consequently, we will exclude the 1,680-byte buffer from further analysis due to these misconfigurations.

Almost all authoritative servers, with a few exceptions, provide amplification smaller than 40 BAF. Specifically, buffer sizes 512, 1,232, and 1,680 have only four servers exceeding a BAF of 40. Examining the two most common buffer sizes, 1,232 and 4,096, reveals a clear distinction: servers that produce significant amplification set the buffer size to 4,096. Similarly, large amplifiers also use buffer sizes of 4,000 bytes. The buffer size of 1,680 stands out again, with half of the servers's BAFs between 4.36 and 4.82. This time it was the result of not implementing DNSSEC.

Investigating whether authoritative servers switch to TCP when their response exceeds the advertised buffer size reveals distinct differences between authoritative and recursive servers. Among authoritative servers advertising a buffer size of 1,232 bytes, only 4 out of 1230 (0.3%) returned responses exceeding this size. In contrast, recursive servers revealed that 72 out of 589 (12.2%) exceeded it. Additionally, buffer size 512 is very popular among recursive resolvers, and the vast majority, 561 out of 676 (83%), returned responses larger than this size. This is to be expected with such a small buffer size but it raises a question as to why it is set to this value in the first place. Servers whose responses do not exceed the buffer size may not have domains with sufficient resource records set to do so. This means that the numbers presented here are the maximum and more servers may not truncate their responses.

Shifting our focus to Figure 4, we examine median BAFs produced by recursive servers categorized by their operating systems and advertised buffer sizes. The highest amplification is produced by servers advertising buffer sizes of 4,000 and 4,096 bytes. Windows has the largest number of servers, 113, with a high median BAF of 112.06 at a buffer size of 4,000 bytes. Following Windows, Enterprise Linux has 35 servers advertising a 4,096-byte buffer with a median BAF of 100.20. Operating systems with fewer servers but high median amplification include Fireware and FreeBSD. The 1,232-byte buffer was present across all operating systems, while the popular 512-byte buffer among recursive resolvers was advertised almost exclusively by servers running on Linux. The median BAF, however, exceeds the theoretical maximum if responses were truncated at 512 bytes.

Next, in Figure 5, we evaluate median BAFs of selected recursive servers, categorized by buffer sizes and versions retrieved from the TXT record of the CH class as described in Section 5. Servers included in the heatmap produced amplification larger than 50 in at least one of the categories. BIND 9.11.22 and Zywall DNS were discovered on 21 and 20 servers, respectively, both advertising only the 4,096-byte buffer and producing high median BAFs of 109.71 and 54.17. This buffer size was the default for BIND servers up to version 9.16.7. [19] Similarly, BIND 9.18.18 and Dnsmasq 2.71 and 2.79 include high amplifiers with fewer servers for the



(a) Authoritative servers (b) Recursive servers

Figure 6: DNS - CDFs for query types of responses that produced largest BAFs per server with the number of servers in parenthesis.

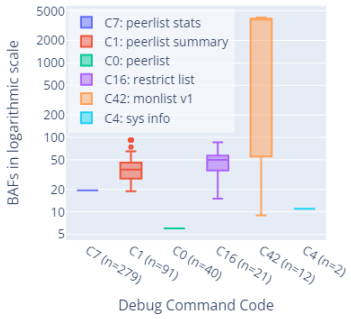


Figure 7: NTP - Recorded BAFs larger than 1 per debug command.

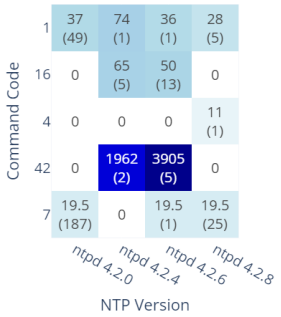


Figure 8: NTP - Heatmap showing the Median BAF across command codes and NTP versions.

Version	n	%
ntpd 4.2.0	236	80.0%
ntpd 4.2.4	8	2.7%
ntpd 4.2.6	20	6.8%
ntpd 4.2.8	31	10.5%
Total	295	100%

Table 3: Distribution of identified NTP server versions. The table shows the count (n) and fraction (%) of each version in the results.

Command	Code
peerlist sum.	1
restrict list	16
sys info	4
monlist	42
peerlist stats	7

Table 4: Mapping of NTP commands to their corresponding codes displayed in the heatmaps.

same buffer size. Lastly, all identified Dnsmasq versions advertised the 512-byte buffer with their median BAF exceeding the theoretical maximum.

Moving on to query types that produced the largest amplification, Figures 6a and 6b show that "ANY" and "DNSKEY" are the most popular types for authoritative and recursive servers, respectively. For authoritative servers, "DNSKEY" and "RRSIG" typically produced amplification ranging from 20 to 30 BAF, while other types generally resulted in BAFs below 20. Only "RRSIG" and "ANY" exceeded 100 BAF, with other types rarely surpassing 60 BAF. For recursive servers, more than 80% of "TXT" and "DNSKEY" responses had BAFs below 20 and 40, respectively. When these query types produced amplification higher than "ANY", it largely stayed under 60 BAF. "ANY" queries triggered the largest response on 763 servers, with the top 40% yielding BAF of more than 100.

Ending with Autonomous Systems (ASes) identified in both DNS scans, we found 328 and 139 ASes with authoritative and recursive amplifiers, respectively. They sum up to 363 unique ASes identified by Autonomous System Numbers (ASNs). Among them, as shown in Figure 10, 69 ASes (21%) had authoritative servers with a BAF of at least 40, and 28 (8.5%) had servers that exceeded 100 BAF. For recursive resolvers, 93 ASes (66.9%) had instances that returned amplification of at least 40, while the level of 100 BAF was exceeded by 76 ASes (54.7%). Most networks were found to contain some large amplifiers with BAF $\geq$ 40 among other properly configured servers producing substantially smaller amplification or none at all. Our analysis reveals that the main configuration differences include the buffer size, truncating responses at the buffer size, and DNSSEC implementation. Some networks were found with behaviors that limited amplification. One AS with recursive servers truncated all responses at 512 bytes. While this increases the need to switch to TCP, adding some round-trip latency, it drastically decreases the amplification to $\approx$ 14.6 BAF. Another AS with recursive servers limited the "ANY" response by selecting which RRs to return, thus limiting the maximum BAF but still keeping it above 40. Lastly, an AS with authoritative servers did not support DNSSEC, resulting in small responses that did not reach the maximum buffer size.

7.2 NTP

We analyzed the responses of 48,814 NTP servers to various queries described in Section 5. Servers in the analysis responded to at least one query with a BAF greater than 1. The largest group, comprising 279 servers, responded to the "peerlist stats" command with a consistent amplification factor of 19.5, as shown in Figure 7. Other commands with consistent amplifications include "peerlist", with 40 servers producing a BAF of 6, and "sys info", with 2 servers showing a BAF of 11. The "peerlist summary" and "restrict list" queries yielded responses with amplification factors ranging from 19 to 92 and 15 to 86, respectively. Notably, the "monlist" command stands out with a maximum recorded BAF of 4,070 and a median of 3,905, despite only 12 servers responding to this request.

From the group of 445 NTP amplifiers, 295 (66.3%) responded to our version requests, identifying themselves with one of the five versions listed in Table 3. The majority of servers (79.2%) reported running version 4.2.0, making it the most common version by a significant margin. Additionally, versions 4.2.6 and 4.2.8 were identified in 6.7% and 10.7% of the servers, respectively. Notably, 89.3% of the servers are running versions earlier than 4.2.7p26, in which the "monlist" functionality was disabled [35].

We investigate whether server versions influence our results with Figure 8. The codes of discussed commands are in Table 4. Starting with the notorious command 42 "monlist", we observed that servers running versions ntpd 4.2.4 and 4.2.6 did not block this command. This aligns with the assumption that servers running version 4.2.7p26 or above typically block this query. These versions were also the only

ones that responded to the "restrict list" command. Next, the "peerlist summary" command, was prevalent across all identified versions, showing it is commonly implemented regardless of the server version. The "peerlists stats" command, which provides a constant amplification of 19.5, was observed on all versions except ntpd 4.2.4. Lastly, the "sys info" command, was only observed on a server running the newest version 4.2.8.

Lastly, we explore the insights provided by Figure 9. Given that only 25 amplifiers (5.6%) were identified with an operating system, these results may not fully represent reality. However, some patterns are noticeable. Machines responding to the "monlist" command ran exclusively on the Linux OS. Similarly, the "peerlist summary" command was only present on servers running Freebsd OS. Three operating systems - Comware, Junos, and Windows Server 2012 R2 - were only identified on servers responding to the "peerlist stats" command. The VMware Esxi Server OS was identified on a machine responding to the "restrict list" command.

7.3 Memcached

Starting with a list of 675 servers, only 12 responded to the "version" query sent over TCP. Five of the servers were running versions below 1.5.6, which disabled the use of UDP by default [36]. Next, no servers responded to the "stats" command sent over UDP. Most servers accepted the connection and did not return any data, while a few servers rejected it immediately.

8 Discussion

The study analyzes servers in Sweden running DNS, NTP, or Memcached services to identify possible amplification and parameters that make them possible. For DNS, we found that open resolvers provide larger amplification than authoritative servers contrary to findings from Rossow [5]. However, our authoritative servers produce smaller amplifications due to the implementation of DNS Flag 2020 recommendations and the limitation of the "ANY" query. Similarly, results for NTP are significantly lower because of the global effort in disabling the "monlist" command and the debug mode [37]. All servers running Memcached did not respond to our queries over UDP, making them worthless to the attackers.

DNS: Our study finds that "ANY" request type is still commonly allowed and produces the largest amplification. It is followed by cryptography-related requests such as "RRSIG" and "DNSKEY", which confirms the next-best types identified by Van der Toorn et al. [18]. Requests for the "TXT" RR also commonly produced the largest amplification on servers, but the triggered responses were usually smaller, and only three of them yielded a BAF larger than 80. The main culprit of high amplification is implementing DNSSEC without limiting "ANY" requests or configuring a small buffer size with mandatory truncation. This is supported by our findings which reveal multiple amplifiers using 4,096- or 4,000-byte buffers while supporting cryptographic signatures. This suggests that the lack of proper configuration to comply with the newest recommendations is why such servers exist, despite large efforts to make settings such as the 1,232-byte buffer or mandatory truncation default as shown by the updated defaults for the BIND DNS system [38]. We identified several operating systems and DNS versions installed on large recursive amplifiers with buffers set to 4,000 and 4,096. Interestingly, multiple recursive resolvers use the 512-byte buffer but do not truncate responses exceeding this size, leading to large amplification. Large amplifiers mostly specified these three buffers. Most Autonomous Systems researched in this paper contained large amplifiers. More than half of those hosting recursive resolvers had servers producing amplification of at least 100, suggesting misconfiguration problems in these types of servers.

NTP: We conclude that the number of servers answering the "monlist" command is minimal and confirm that the quick community response to the NTP amplifier threat [37] successfully mitigated it. However, other commands from the debug mode were found to be active across all identified server versions. Their amplification is not as large as for the "monlist" command, but the total average is larger when compared with DNS servers making NTP still a good amplification source for attackers. Next, most amplifiers ran on versions released before the popularity of the "monlist" attack. This prevalence of old versions in the dataset suggests that most amplifiers run on outdated software and their number will gradually decrease. We also found that some commands were only answered by servers running a specific operating system, which suggests that there could be a difference in default settings across different OSes.

Limitations: The project's time constraints allowed us to perform only a limited number of scans and made capturing a consistent state of Swedish networks difficult. Some servers may have been inactive or misrepresented the usual state during our scans. Moreover, performing scans from one vantage point may have provided a biased or limited view of the network. Another thing to note is that servers and their networks commonly implement DDoS prevention mechanisms which are not reflected in this study. Next, the two methods used to estimate DNS servers' versions may be inaccurate. The "fpdns" tool was developed more than 10 years ago and the "TXT" record with the version of its DNS server may be freely modified by its administrator which could lead to misleading results. Lastly, not all servers were identified with a buffer size, operating system, or version, meaning some results may be biased.

Peers' outcomes: We identified many similarities by comparing our study with the peers in the same research group performing similar experiments on networks in different countries. Starting with DNS, in all researched countries authoritative servers were found to produce smaller amplification than recursive resolvers. Moreover, buffers of sizes 4,000 and 4,096 bytes were present on most amplifiers, which aligns with our results in Sweden. Another similarity is that multiple servers were found to not truncate their responses upon exceeding the advertised buffer size, especially buffer 512 which is popular among recursive servers. The Netherlands had a surprisingly high presence of buffer 1,232 among DNS servers, something other countries should follow. For NTP, a very scarce number of servers answered the "monlist" command which aligns with our study and confirms a

good mitigation of this vulnerability. These servers were also identified as running on Linux with versions below 4.2.7-p26, where "monlist" may not be disabled by default [35]. Ending with Memcached, 1, 12, and 25 servers communicating over UDP were found in three countries showing some, but minimal, opportunity for attacks.

Recommendations: for DNS server administrators, we recommend two solutions to minimize the risk of servers being used in amplification attacks: setting the buffer size to 1,232 bytes with truncation and switching to TCP for responses exceeding this size and limiting or disabling the "ANY" query. Seeing a problem with recursive resolvers being used to resolve huge domains, we want to emphasize the need for their better configuration. A system of selecting which authoritative server to query based on the minimal response size could also be implemented to prioritize well-configured options. Ending with NTP servers, the debug mode should be completely disabled or allowed for authorized requests only. This mode is of no usage for a regular client and only the minimum needed for an NTP server to operate should be enabled.

9 Conclusion

In this study, we investigated the current state of servers in Sweden providing amplification and attempted to identify parameters that make them amplifiers. Our work revealed the amplification potential of authoritative and recursive DNS servers and NTP servers in Sweden. By finding factors responsible for amplification, we hope that our study will facilitate the detection of vulnerable servers and help correctly configure new servers before their deployment.

DNS servers advertising large buffer sizes such as 4,096 or 4,000 are more likely to be large amplifiers. With DNSSEC supported, these servers are able to produce amplification exceeding 100 BAF. The lack of truncation for large responses exceeding smaller buffer sizes and the absence of limitations on the "ANY" query can also be attributed to significant amplification exceeding 40 BAF. These parameters should be considered when setting up a new server, and more attention should be paid to servers running specific operating systems and protocol versions due to default settings not representing the latest recommendations. NTP servers, despite being hardened against the "monlist" attack, were still found to provide amplification with other debug commands. This shows the significance of fully disabling the debug mode instead of solely focusing on the "monlist" command. The large presence of old server versions suggests that the number of amplifiers will gradually decrease in the next year due to improved security in newer releases. However, vendors and network administrators should still learn the weak points of their servers and ensure they are properly protected.

More work could be done in the area of fingerprinting servers and identifying vendors who release their software without complying with best practices. Given more time, regular scans spanned over a longer period of time could reveal servers that we potentially missed. By regularly monitoring the state of servers and identifying vulnerable instances, we can continuously patch the weak points that make servers large amplifiers and strengthen the network against future amplification attacks.

References

[1] F. J. Ryba, M. Orlinski, M. Wählisch, C. Rossow, and T. C. Schmidt, "Amplification and DRDoS Attack Defense – A Survey and New Perspectives," May 2016, arXiv:1505.07892 [cs]. [Online]. Available: http://arxiv.org/abs/1505.07892

[2] "Google says it mitigated a 2.54 Tbps DDoS attack in 2017, largest known to date," Accessed: Jun. 19, 2024. [Online]. Available: https://www.zdnet.com/article/google-says-it-mitigated-a-2-54-tbps-ddos-attack-in-2017-largest-known-to-date/

[3] "Famous DDoS attacks | Biggest DDoS attacks," Accessed: Jun. 19, 2024. [Online]. Available: https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/

[4] "The rise of multivector DDoS attacks," Nov. 2018, Accessed: Jun. 19, 2024. [Online]. Available: https://blog.cloudflare.com/the-rise-of-multivector-amplifications

[5] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proceedings 2014 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2014. [Online]. Available: https://www.ndss-symposium.org/ndss2014/programme/amplification-hell-revisiting-network-protocols-ddos-abuse/

[6] D. Wagner, D. Kopp, M. Wichtlhuber, C. Dietzel, O. Hohlfeld, G. Smaragdakis, and A. Feldmann, "United We Stand: Collaborative Detection and Mitigation of Amplification DDoS Attacks at Scale," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 970–987. [Online]. Available: https://dl.acm.org/doi/10.1145/3460120.3485385

[7] P. E. Hoffman, A. Sullivan, and K. Fujiwara, "DNS Terminology," Internet Engineering Task Force, Request for Comments RFC 8499, Jan. 2019, num Pages: 50. [Online]. Available: https://datatracker.ietf.org/doc/rfc8499

[8] "Domain names - implementation and specification," Internet Engineering Task Force, Request for Comments RFC 1035, Nov. 1987, num Pages: 55. [Online]. Available: https://datatracker.ietf.org/doc/rfc1035

[9] S. Rose, M. Larson, D. Massey, R. Austein, and R. Arends, "DNS Security Introduction and Requirements," Internet Engineering Task Force, Request for Comments RFC 4033, Mar. 2005, num Pages: 21. [Online]. Available: https://datatracker.ietf.org/doc/rfc4033

[10] J. d. S. Damas, M. Graff, and P. A. Vixie, "Extension Mechanisms for DNS (EDNS(0))," Internet

Engineering Task Force, Request for Comments RFC 6891, Apr. 2013, num Pages: 16. [Online]. Available: https://datatracker.ietf.org/doc/rfc6891

[11] R. Bellis, "DNS Transport over TCP - Implementation Requirements," Internet Engineering Task Force, Request for Comments RFC 5966, Aug. 2010, num Pages: 7. [Online]. Available: https://datatracker.ietf.org/doc/rfc5966

[12] A. Herzberg and H. Shulman, "Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org," in *2013 IEEE Conference on Communications and Network Security (CNS)*. National Harbor, MD, USA: IEEE, Oct. 2013, pp. 224–232. [Online]. Available: http://ieeexplore.ieee.org/document/6682711/

[13] P. Špaček and O. Surý, "DNS Flag Day 2020," Oct. 2020, Accessed: Jun. 12, 2024. [Online]. Available: https://www.dnsflagday.net/2020/

[14] J. Martin, J. Burbank, W. Kasch, and P. D. L. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," Internet Engineering Task Force, Request for Comments RFC 5905, Jun. 2010, num Pages: 110. [Online]. Available: https://datatracker.ietf.org/doc/rfc5905

[15] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. Vancouver BC Canada: ACM, Nov. 2014, pp. 435–448. [Online]. Available: https://dl.acm.org/doi/10.1145/2663716.2663717

[16] "memcached - a distributed memory object caching system," Accessed: Jun. 02, 2024. [Online]. Available: https://memcached.org/about

[17] "Memcrashed - Major amplification attacks from UDP port 11211," Feb. 2018, Accessed: Jun. 19, 2024. [Online]. Available: https://blog.cloudflare.com/memcrashed-major-amplification-attacks-from-port-11211

[18] O. van der Toorn, J. Krupp, M. Jonker, R. van Rijswijk-Deij, C. Rossow, and A. Sperotto, "ANYway: Measuring the Amplification DDoS Potential of Domains," in *2021 17th International Conference on Network and Service Management (CNSM)*, Oct. 2021, pp. 500–508, iSSN: 2165-963X. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9615596

[19] G. C. M. Moura, M. Müller, M. Davids, M. Wullink, and C. Hesselman, "Fragmentation, Truncation, and Timeouts: Are Large DNS Messages Falling to Bits?" in *Passive and Active Measurement*, O. Hohlfeld, A. Lutu, and D. Levin, Eds. Cham: Springer International Publishing, 2021, vol. 12671, pp. 460–477, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-030-72582-2_27

[20] "Censys," Accessed: Jun. 12, 2024. [Online]. Available: https://censys.com/

[21] "Censys Internet Scanning Introduction," May 2024, Accessed: Jun. 02, 2024. [Online]. Available: https://support.censys.io/hc/en-us/articles/25692846962708-Censys-Internet-Scanning-Introduction

[22] C. Inc, "Censys Python Library," Accessed: Jun. 12, 2024. [Online]. Available: https://pypi.org/project/censys/

[23] "Toplists TUM," Accessed: Jun. 12, 2024. [Online]. Available: https://toplists.net.in.tum.de/archive/

[24] "Common Crawl - Open Repository of Web Crawl Data," Accessed: Jun. 12, 2024. [Online]. Available: https://commoncrawl.org/

[25] I. Kreymer, "ikreymer/cdx-index-client," Jun. 2024, Accessed: Jun. 19, 2024. [Online]. Available: https://github.com/ikreymer/cdx-index-client

[26] "IPinfo," Accessed: Jun. 12, 2024. [Online]. Available: https://ipinfo.io

[27] M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, and C. Papadopoulos, "A look at router geolocation in public and commercial databases," in *Proceedings of the 2017 Internet Measurement Conference*. London United Kingdom: ACM, Nov. 2017, pp. 463–469. [Online]. Available: https://dl.acm.org/doi/10.1145/3131365.3131380

[28] "Scapy," Accessed: Jun. 18, 2024. [Online]. Available: https://scapy.net/

[29] P. Politowicz, "pwicz/research-project-amplification-attacks," Jun. 2024, Accessed: Jun. 23, 2024. [Online]. Available: https://github.com/pwicz/research-project-amplification-attacks

[30] "fpdns repository," Mar. 2024, Accessed: Jun. 12, 2024. [Online]. Available: https://github.com/kirei/fpdns

[31] "ntpq - standard NTP query program," Accessed: Jun. 12, 2024. [Online]. Available: https://www.ntp.org/documentation/4.2.8-series/ntpq/

[32] "memcached/doc/protocol.txt at master · memcached/memcached," Accessed: Jun. 03, 2024. [Online]. Available: https://github.com/memcached/memcached/blob/master/doc/protocol.txt

[33] "Netherlands Code of Conduct for Research Integrity | NWO," Accessed: Jun. 12, 2024. [Online]. Available: https://www.nwo.nl/en/netherlands-code-conduct-research-integrity

[34] "PowerDNS: List of Authoritative Server Settings," Accessed: Jun. 16, 2024. [Online]. Available: https://doc.powerdns.com/md/authoritative/settings/#default-soa-name

[35] "NVD - CVE-2013-5211," Accessed: Jun. 12, 2024. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2013-5211

[36] "ReleaseNotes156," Accessed: Jun. 07, 2024. [Online]. Available: https://github.com/memcached/memcached/wiki/ReleaseNotes156

[37] "Good News: Vulnerable NTP Servers Closing Down," Feb. 2014, Accessed: Jun. 17, 2024. [Online]. Available: https://blog.cloudflare.com/good-news-vulnerable-ntp-servers-closing-down

[38] "BIND 9 Significant Features Matrix," Accessed: Jun. 17, 2024. [Online]. Available: https://kb.isc.org/docs/aa-01310

A Appendix

In this section, we provide additional information on the methods used to obtain IP addresses from Censys and the process of crafting DNS, NTP, and Memcached packets with Scapy to measure returned amplification. All code presented here is available in the public repository [29]. Lastly, we present additional plots that could not be included in the main body of the paper due to space constraints.

A.1 Retrieving Data From Censys

The queries that we used to retrieve IP addresses with contain two arguments:

1. "location.country": the country in which servers should be located. We set it to Sweden.
2. "services.service_name": the name of the service that should be present on the servers. We used DNS, NTP, and Memcached.

The resulting three queries are the following:

```
location.country=Sweden and
    services.service_name=DNS
```

```
location.country=Sweden and
    services.service_name=NTP
```

```
location.country=Sweden and
    services.service_name=MEMCACHED
```

We used our Python script with the Censys Python Library [22] to retrieve large numbers of servers with the queries.

A.2 Crafting Packets with Scapy

We start with crafting requests that we will send to servers collected in previous steps. We used the following Python code to create **DNS requests**:

```
udp_payload = DNS(
    id=random.randint(0, 0xFFFF),
    rd=1,
    qd=DNSQR(qname=target_domain,
        qtype=record_type),
    ar=DNSRROPT(rclass=4096)
)
source_port = random.randint(10000, 65000)
query = IP(dst=dns_server_ip) /
    UDP(sport=source_port, dport=53) /
    udp_payload
```

In the "udp_payload", we set the recursion desired ("rd") flag and specify the target domain such as "example.com" with the chosen record type such as 255 that represents the "ANY" query. Then, we randomly pick a source port that we will listen on for incoming packets and put the UDP payload together with UDP and IP headers. The complete request is saved in the "query" variable.

Similarly, we create **NTP requests**:

```
version = '\x17'
source_port = random.randint(10000, 65000)
query = IP(dst=server_ip) /
UDP(sport=source_port, dport=123) /
Raw(load=str(version + "\x00\x03" + code) +
    str("\00")*4)
```

The main difference is that we manually create the UDP payload according to specifications described in the RFC 5905 [14]. "code" is one of the command codes presented in Table 4.

Lastly, since **Memcached requests** can be sent in binary and Ascii modes, we composed the requests with two different UDP payloads. Starting with binary mode:

```
binary_stats = b'\x80\x10' + b'\x00' * 22
source_port = random.randint(10000, 65000)
query = IP(dst=server_ip) /
    UDP(sport=source_port, dport=11211) /
    binary_stats
```

The opcode 0x10 following the Magic code 0x80 indicates the stat request. The Ascii mode request is more straightforward:

```
ascii_stats = b"stats\r\n"
source_port = random.randint(10000, 65000)
query = IP(dst=server_ip) /
    UDP(sport=source_port, dport=11211) /
    ascii_stats
```

With the requests ready, we **send them and sniff** for the incoming packets with the "valid response" functions that match packets by the transport mode and the source and destination ports:

```
send(query, verbose=0)
responses = sniff(lfilter=lambda x:
    valid_mc_response(x, server_ip,
    source_port), timeout=timeout)
```

```
def valid_mc_response(packet, ip: str,
    dport: int):
    return IP in packet
        and packet[IP].src == IP
        and UDP in packet
        and packet[UDP].sport == 11211
        and packet[UDP].dport == dport
```

We follow Equation 1 to **calculate the BAF** of the received responses. First, we sum the lengths of the encapsulated UDP datagrams, subtracting the UDP header size from each packet to obtain the total UDP payload received from the server. We then divide the sum by the length of the request's UDP payload and round it to two decimal places. This process is implemented with the following function:

```
def measure_BAF(request, responses):
    req_size = len(request[UDP].payload)
    resp_size = 0
    UDP_header_size = 8  # bytes
    for resp in responses:
        resp_size += resp[UDP].len -
            UDP_header_size
    return round(resp_size / req_size, 2)
```

A.3 Additional Plots

The following plots are referred to in Section 7.

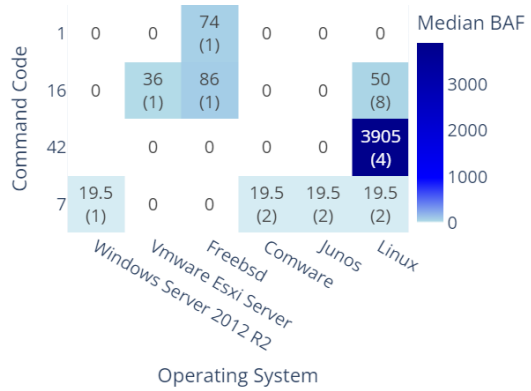


Figure 9: NTP - Heatmap showing the Median BAF across command codes and operating systems.

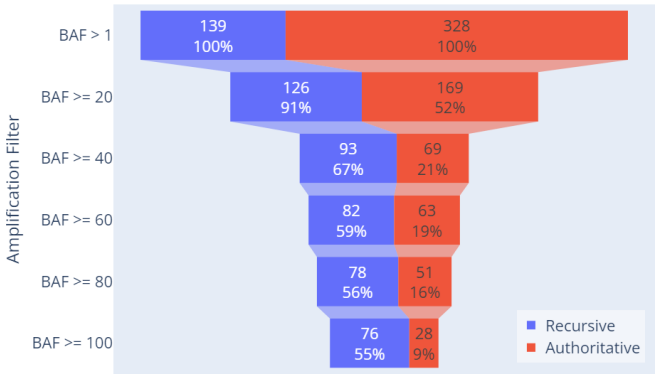


Figure 10: DNS - Distribution of Autonomous Systems based on the maximum BAF produced by their servers.