Exploring DDoS amplification attack vectors prevalent in the Dutch IP range

Konstantin Dimitrov[1]

Supervisor(s): Georgios Smaragdakis[1]**, Harm Griffioen**[1]

[1]**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: Konstantin Dimitrov
Final project course: CSE3000 Research Project
Thesis committee: Georgios Smaragdakis, Harm Griffioen, George Iosifidis

Abstract

This paper explores Distributed Reflective Denial-of-Service (DRDoS) attacks, a variant of Distributed Denial-of-Service (DDoS) attacks that leverage publicly accessible UDP servers to amplify traffic towards a target. These attacks, accounting for over half of all DDoS cases in 2023, are significant threats to online services due to their potential to generate traffic volumes in the Tbps range. Despite existing research on DDoS attack vectors and techniques, there remains a gap in tools for identifying potential amplification sources within specific networks. This paper aims to fill that gap by identifying and measuring amplification hazards in the Dutch IP range, focusing on DNS, NTP, and Memcached protocols. Our findings reveal significant amplification potentials, particularly within NTP and Memcached servers, and highlight the influence of factors such as EDNS0 buffer size on DNS amplification.

I. Introduction

A distributed reflective Denial-of-Service (DRDoS) attack is a distributed denial-of-service (DDoS) attack that uses publicly accessible User Datagram Protocol (UDP) servers, capable of amplification, to overwhelm the target system with UDP response traffic [1]. According to the Nexusguard DDoS trend report, such attacks accounted for 51.05% of all DDoS attacks performed in 2023 [2], highlighting their considerable relevance within the cybersecurity space. Furthermore, since this kind of attack can reach the Tbps range [3], it has the potential to cause severe disruption to online services and infrastructure, highlighting the significant importance of addressing and mitigating such threats in cybersecurity strategies.

Previous work that addresses DDoS attacks has been done by researchers like Rossow [4], who works on identifying different attack vectors and how effective they are, and Griffioen et al. [5], who work on analyzing adversarial techniques, tactics, and procedures when it comes such attacks. Research on attacks using specifically DNS servers has been done by Toorn et al. [6]. So far, however, no framework (or tool) to identify potential amplifiers in a given network and calculate their potency has been proposed.

This paper presents an effort to identify amplification hazards within the Dutch IP range and measure their potency. It also aims to identify the factors that enable adversaries to use them in attacks.

In our investigation of amplification hazards within the Dutch IP range, we uncovered critical factors influencing amplification, notably including misconfigurations in DNS servers' EDNS0 buffer sizes and the prevalence of outdated NTP and Memcached versions susceptible to exploitation. Through our analysis, we identified potential amplifiers across these protocols, some demonstrating significant potency. Our findings show the need for proactive mitigation strategies to safeguard against these vulnerabilities.

After this introduction section, the paper is structured in the following way. First of all, in Section II, the necessary background knowledge will be presented. Then, in section III we will show the state-of-the-art literature. In Section IV the datasets used in the study will be described and there will be information on how they are collected. Then, in Section V the experiments done in the study will be explained in full detail. Section VI focuses on the reproducibility of the project and the integrity of the research. Section VII shows and explains the acquired results. Section VIII is a discussion of the research. Finally, Section IX summarizes the conducted research and provides suggestions for future works.

II. Background

In this section, we will start off by explaining what DDoS attacks are, before going over the protocols that were researched and how they can be used in attacks. Finally, we will take a look at application layer traffic loops.

A. DDoS Attacks

A Denial-of-Service (DoS) attack is a malicious attempt to disrupt the normal functioning of a machine or network, rendering it inaccessible to its legitimate users. This is done by overwhelming the target with excessive traffic. An extension of this concept is the Distributed Denial-of-Service (DDoS) attack, where multiple compromised computers, often part of a botnet, are used to increase the attack traffic volume. These direct flood attacks, however, have been going down in popularity, with a decrease of 30.93% YoY in 2023, likely due to a shift toward more sophisticated methods, such as the distributed reflective denial-of-service (DRDoS) attack [2]. DRDoS attacks use third-party servers to amplify the malicious traffic toward the victim. Attackers achieve that by sending requests to servers with a forged (spoofed) source IP address. Since the third-party service cannot easily differentiate between legitimate requests and forged ones, it ends up sending its response to the supposed requester, who in this case is the victim.

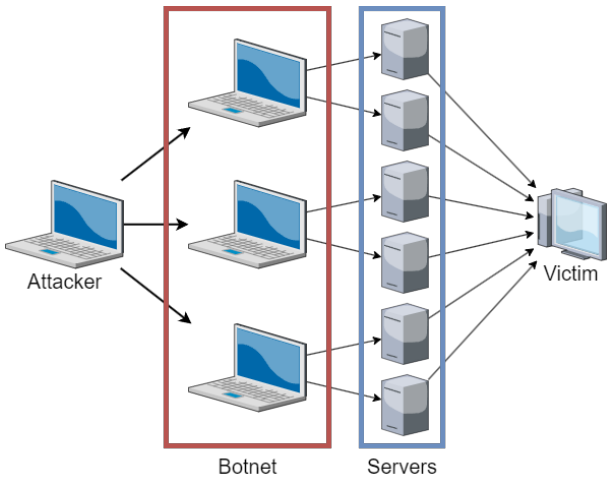

Fig. 1: Schematic Diagram of DRDoS Attack.

Amplification of the attack traffic is achieved by selecting reflectors that generate large responses to small requests, thus magnifying the attack's impact. To avoid creating a bottleneck and to maximize the attack's effectiveness, attackers use multiple reflectors, distributing the malicious requests across numerous third-party servers. To measure the amplification of the reflectors we will be using the bandwidth amplification factor (BAF) by Rossow [4]:

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

There are many protocols that can be exploited in DRDoS amplification attacks, as shown in [4]. In this paper, however, we will be focusing on 3 of them, namely DNS, NTP and Memcached.

B. DNS

Domain Name System (DNS) is the protocol used to translate human-readable domain names into their corresponding numerical IP addresses [7]. It operates as a distributed database organized into a hierarchical tree-like structure. At the top of the hierarchy are the root DNS servers, which delegate authority to top-level domain (TLD) servers responsible for specific domains (like .com, .org, .net). Beneath the TLD servers are authoritative name servers responsible for individual domain names.

Once a DNS query is made to a resolver, it first checks its local cache to see if it already knows the IP address associated with the domain name. If it does not, it sends a recursive DNS query to one of the root DNS servers, asking for the IP address of the domain name. The root DNS servers then refer the resolver to the appropriate TLD server. The resolver then sends a query to the TLD server, which either provides the IP address or refers the resolver to the authoritative name server for the specific domain. Finally, the resolver sends a query to the authoritative name server, which responds with the IP address associated with the domain name.

DNS maintains various types of resource records (RRs) associated with domain names, such as A records (IPv4 address), AAAA records (IPv6 address), MX records (mail exchange), CNAME records (canonical name), and TXT records (text information).

A client can learn all the RRs on a particular server, they can use the ANY query, which returns all, or a subset of all RRs. This type of query can be particularly useful for attackers as it can lead to high BAF, depending on the state of the cache.

The original DNS specification limits DNS message sizes to a maximum UDP payload of 512 bytes. However, Domain Name System Security Extensions (DNSSEC) needs more in order to work. This is why it relies on Extension Mechanisms for DNS, colloquially known as "EDNS0" [8]. It allows for a maximum UDP payload of 4096 bytes, making the protocol even more potent for amplification attacks.

C. NTP

Network Time Protocol (NTP) is a protocol used to synchronize the clocks on computer networks with very high precision. It uses a hierarchical system of time resources to achieve this task. NTP can be used for amplification attacks because it allows users to request a list of hosts which the NTP server has communicated with recently. This list called "monlist" can contain up to 600 IPs and can result in a BAF of 5500 if it is full. While in ntp versions after 4.2.7p26 the "monlist" feature has been disabled by default, there are still a lot of hosts running older versions and are therefore susceptible to being used in amplification attacks.

D. Memcached

Memcached is a UDP-based high-performance memory caching system, commonly used to speed up dynamic web applications. It functions as an in-memory key-value store, designed to cache small pieces of arbitrary data such as strings or objects. It typically stores data retrieved from database calls, API requests, or page rendering processes [9]. By default, the protocol allows a specific key value to store 1 MB of data.

Attackers can utilize vulnerable servers by first injecting a key-value pair with short key length, and a value that has 1 MB of data, before the attack. Then, during the attack, adversaries send a GET request, with a spoofed source IP address, for one or more of the previously injected keys. This can result in a BAF of over 506,720 [10].

E. Application layer traffic loops

Application layer traffic loops are an attack primitive that can be abused to launch DoS attacks [11]. They happen if two servers indefinitely respond to each other's messages, creating an infinite loop of traffic without requiring continuous traffic from the attacker. Attackers can exploit this vulnerability by sending a single IP-spoofed packet to trigger the loop. These loops are prevalent across various popular and legacy UDP-based application-layer protocols, including NTP and DNS, affecting millions of servers and numerous networks globally. Finally, the traffic loops can be abused as an almost infinitely-amplifying attack primitive.

III. Related work

Amplification attacks have been extensively studied due to their significant impact on network security and service availability. Microsoft reported one of the largest DDoS attacks peaking at 3.47 Tbps [12], demonstrating the severe potential impact of such attacks. Starting with Paxson [13] in 2001, researchers have begun to conduct comprehensive analyses of DRDoS attacks. More recently, Rossow [4] has worked on identifying different attack vectors that can be used for amplification attacks, providing 14 of them.

Subsequently, there has been a lot of research focusing on amplification using specific protocols, such as DNS [6, 14, 15, 16, 17], NTP [18, 19] and Memcached [20]. In contrast to these well-studied amplification techniques, the concept of application layer traffic loops has only recently gained attention and has been explored in depth by Pan et al. [11].

While significant progress has been made in understanding the mechanics of DRDoS amplification attacks, there remains a critical need for practical tools and frameworks to

identify and mitigate potential amplifiers within specific network environments. This paper aims to address this gap by presenting a systematic approach to identifying amplification hazards within the Dutch IP range, measuring their potency, and uncovering the factors that lead to higher amplification and enable adversaries to exploit them in DRDoS attacks.

IV. Datasets

The Technical University of Munich (TUM) database [21] has a list of the top 10 million websites. The 4691 most popular ".nl" domains on the 16th of April 2023 were filtered out and used in the study. A Python script that queries all domains and retrieves the authoritative nameservers for them was then written. This left us with a file consisting of the IPs of 13703 authoritative DNS servers and the domains they are authoritative for.

Censys Search [22] was utilized in order to gather the IPs of NTP and Memcached servers located in the Netherlands. The collection of the servers happened in May 2024. For NTP, Censys indicated close to 90,000 servers, however out of those only 30,000 were used in the study due to availability limitations. For Memcached, there were 341 results and all of them were used. Furthermore, Censys was also used to gather 30,000 IPs of DNS servers for application layer traffic loops testing. All queries used can be found in Appendix A.

V. Methodology

The experimental, quantitative research done can be split into a few different steps for each protocol studied. These steps are visually depicted in Fig. 2 and elaborated upon in this section.

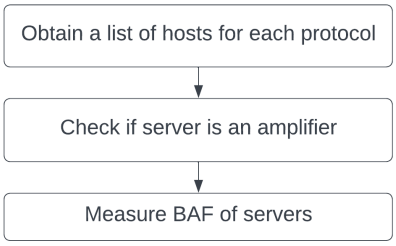

Fig. 2: The steps of our methodology.

A. DNS

Step 1: Obtain a list of authoritative DNS servers

In order to obtain a list of authoritative DNS servers, the first step involved sourcing a list of ".nl" domains. This dataset, as detailed in Section IV, was acquired from the TUM database. The next part of the process involved finding the authoritative DNS servers for each domain in the previously acquired list. This was achieved through a Python script designed to query each domain and retrieve its authoritative nameservers. The dnspython [23] library was used for the task.

Step 2: Check if server is an amplifier

Since for DNS the focus is on ANY queries, we classify a server as an amplifier if it responds to such queries over UDP. In order to test for this we had to query the obtained DNS server from the previous step for ANY request. Those queries (Appendix B) were crafted using the Scapy Python library [24]. They were then sent using the sr1() method from the library, which sends the query and only returns one packet that answered. The hosts, from which we did not receive a response were then filtered out.

Step 3: Measure BAF of servers

To measure the BAF of the DNS servers, we continued to use the same ANY queries crafted in the previous step. However, instead of using the *sr1()* method, we used *send()* as it allowed us to send the queries without waiting for an immediate response, enabling us to capture responses that might contain multiple packets.

After sending the queries, we used the *sniff()* function to capture the incoming packets for a duration of 2 seconds. This was necessary because some DNS responses, especially for ANY queries, can be long and may not fit into a single packet. By sniffing we were able to collect all parts of the responses. Finally, the total UDP payload size of all responses for a specific query was calculated and the BAF was computed.

B. NTP

Step 1: Obtain a list of NTP servers

As described in Section IV, Censys Search was utilized in order to gather IPs of NTP servers in the Netherlands. All in all, 30,000 IPs were collected and used in the research.

Step 2: Check if server is an amplifier

As for NTP the focus falls on the monlist query, we classify a server as an amplifier if it responds to such a query. Similarly to our approach with DNS, Scapy was utilized to craft the requests (Appendix B), which were then sent using the *sr1()* method. The hosts that did not respond were then filtered out.

Step 3: Measure BAF of servers

Again following the same approach as with DNS, the same query was sent to the remaining NTP servers, this time using the *send()* method. Then the procedure of using *sniff()* to collect all responses to a specific request within 2 seconds and then calculating the total UDP payload size of all responses to compute the BAF was used.

C. Memcached

Step 1: Obtain a list of Memcached servers

Following the same approach, the Censys Search tool was employed to gather the IPs of Memcached servers in the Netherlands, as detailed in Section IV. A total of 341 IPs were collected for Memcached servers, which were utilized in the research.

Step 2: Check if server is an amplifier
We classify a Memcached server as an amplifier as long as it responds to any request over UDP. To identify such servers, the gathered server IPs were first queried for the *stats* command (Appendix B) to filter out ones that did not respond. This left us with 12 amplifiers.

Step 3: Measure BAF of servers
Since the BAF that can be achieved with Memcached servers without injecting our own keys and data is highly dependent on the keys already stored in the server, this step is mostly proof of concept. Nevertheless, in order to collect keys that were already in the servers we used *stats items* and *stats cachedump* commands. GET queries (Appendix B) for popular keys such as ftce and ftcd were then sent to these servers to find out what BAF we could get with what is already available. As before, we sniffed for responses for two seconds and then calculated the BAF. After those experiments, as a proof of concept, SET queries for keys with 1KB data and a lifetime of a minute were sent to the servers and then retrieved using the GET command.

D. Application layer traffic loops

In order to find servers susceptible to application layer traffic loops we followed the methodology from Pan et al. [11] and used their code available on GitHub [25]. The same lists of NTP server IP addresses were used as in the previous experiments. However, for DNS a different list of 30,000 hosts, obtained from Censys was used. It has to be noted that in this paper we skip the final, loop verify, step as we can not do the setup within our time limits. Furthermore, in the fourth step, we do not drop any loops.

VI. Responsible Research

This section of the report emphasizes the ethical approach taken in conducting network scanning and vulnerability analysis. It also outlines the measures taken to make sure our research is reproducible.

A. Ethics

When conducting research involving network scanning and analysis of potential vulnerabilities, it is crucial to adhere to ethical guidelines to ensure the integrity of the study and mitigate any potential harm to network infrastructures or systems. In this section, we outline the ethical considerations observed in our approach.

First of all, no active scanning was performed during the research, Instead, we opted to use passive network scanning, meaning that we got our data from tools such as Censys Search and Shodan [26]. This approach minimizes the impact on the target networks and reduces the risk of inadvertently causing disruptions.

Additionally, all response traffic generated throughout the research was directed back to our machines, meaning that no IP address spoofing was done. We also put limitations on the frequency of packet sending. Specifically, 10 queries were sent every 2 seconds, making sure that we did not overload any connections.

It is also important to emphasize that our research solely focused on the identification and measurement of potential vulnerabilities. At no point did we engage in, or perform any actual attacks.

Lastly, no subjective interpretation was done of the results. The analysis focused solely on objective data metrics.

B. Reproducibility

In addition to the ethical considerations outlined previously, we also took specific measures to ensure the reproducibility of our research methodology. We have provided detailed information on the queries used to obtain servers in the "Appendix" section of our paper. To facilitate easy replication of our experiments, we have made the code used in our research available on GitHub [27]. While due to the nature of the experiments, we can not ensure result reproducibility, we provide the information required for process reproducibility.

VII. Results

This section provides an overview of the findings obtained from our experiments. We first take a look at DNS, then NTP, Memcached, and finally application layer traffic loops.

A. DNS

Out of the 13,703 queried authoritative DNS servers, the highest recorder BAF was 86.6 for the ANY query. 1,718 servers did not respond to the query and further 623 had a BAF less than 1. We first show the average BAF for all the DNS amplifiers, for the 50% worst and the 10% worst hosts in Table I.

TABLE I: Average BAF for DNS ANY.

all	50%	10%
6.72	11.04	32.56

The overall average BAF was found to be 6.72, which is 8.125 times smaller than the one observed by Rossow in 2014. This indicates improvement in DNS server management practices during the last 10 years.

In Fig. 3 we show the distribution of the BAF for DNS ANY queries in increments of 10. The horizontal axis represents the amplification factor and the vertical axis represents the number of hosts. As seen in the figure, the majority of the servers had BAF of under 10. Furthermore, the servers with BAF over 50 were a very small minority consisting of only 30 servers.

Fig. 4 shows the average of the BAF for different EDNS0 buffer sizes. The horizontal axis represents the average bandwidth amplification factor and the horizontal axis represents the different buffer sizes. As we can see the majority of the servers (8,379) have a buffer size of 1,232, which is the recommended one by DNS flag day 2020 [28] as it does not cause fragmentation. The average BAF for this size is 4.68 and in Fig. 5 we see that the variance is low. However, interestingly there is also a considerable amount of outliers. This is unexpected as the low buffer size does not allow BAF as high as recorded. This means that some

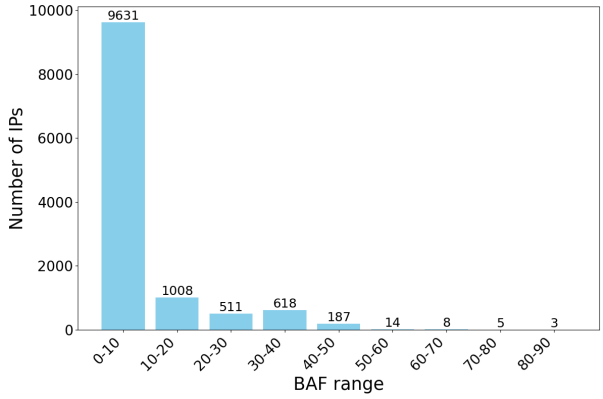

Fig. 3: Distribution of BAF for DNS ANY responses.

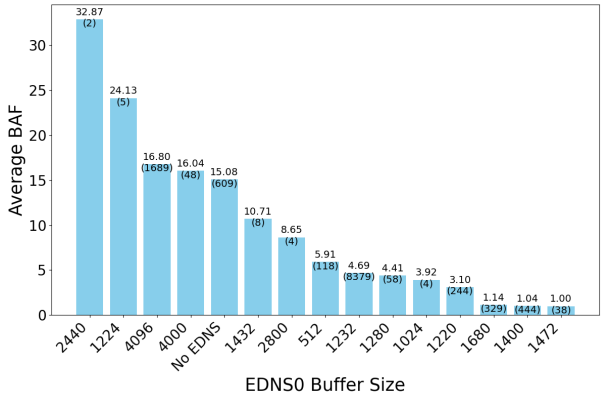

Fig. 4: Average BAF for different EDNS buffer sizes.

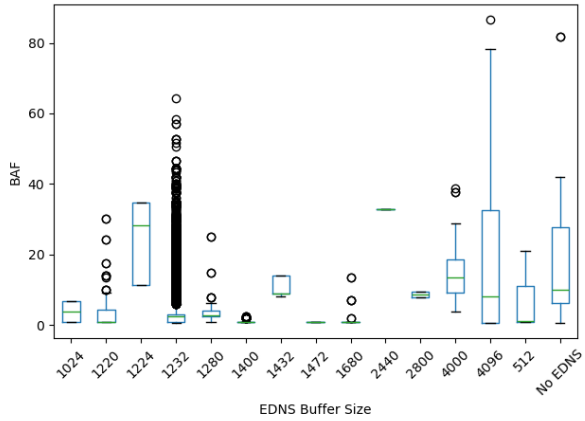

Fig. 5: Box plot of BAF for different EDNS buffer sizes.

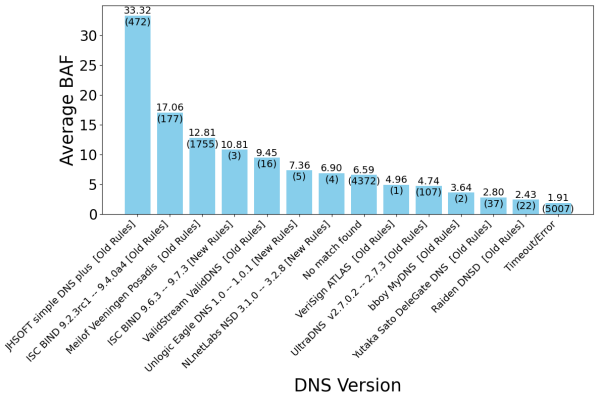

Fig. 6: Average BAF for different DNS versions.

servers advertise one buffer size but use another. In fact, after checking the results 101 of the servers with 1,232 advertised EDNS0 buffer size were found to actually have a bigger one. Furthermore, 98 of those were part of the Signet B.V. autonomous system (ASN = 20857) and the other 3 were all authoritative servers for the same domain, showing that the issue is centralized. If we look at the buffer size for buffers with more than 10 servers we see that the highest average is for 4096, which is the highest possible. The highest recorder BAF is also from a server with such a buffer size. Looking at the figures we see that there is a correlation between EDNS0 buffer size and BAF with a lower buffer relating to lower BAF.

Another factor that we look into is the DNS version the servers are running. Fig. 6 shows the average BAF for the different versions. As we can see JHSOFT simple DNS plus [Old Rules] has the highest average BAF with 33.32, which is considerably higher than others. It has to be noted that the fingerprinting software that was used, namely fpdns [29], only matched a DNS version for 2,655 of the 11,985 responding hosts or roughly 22%. Furthermore, Meilof Veeningen Posadis [Old Rules], is not a valid DNS version, but a default one that newer ISC BIND implementations get matched to.

We then look for the EDNS0 buffer size implemented by different DNS versions. Fig. 7 represents a heatmap of the different DNS versions and the EDNS0 buffer size they use with the average BAF for each combination. Immediately we see that all but 6 servers running JHSOFT simple DNS plus [Old Rules] use a 4,096 buffer size. Looking at the website of the implementation [30] it can be seen that JH Software recommends 1,280 as a good starting point for the buffer, but as we can see from the figure, only 4 of the servers use that recommendation. For Meilof Veeningen Posadis [Old Rules] we observe high variation in the EDNS0 buffer size used. This can be due to the aforementioned fact that this is the default response for newer ISC BIND versions by the fingerprinting software. Still, we see a high number of servers (440) that are configured to use 4,096 buffer size.

Next, we try to figure out if there are particular autonomous systems (AS) with servers that achieve high BAF. Fig. 8 shows the 10 ASs, that have more than 5 servers, with the highest average BAF. ASN 21342 (Akamai International B.V.) sticks out with a very high average amplification factor, but also a relatively high number of hosts.

Fig. 9 and Fig. 10 represent heatmaps of the different EDNS0 buffer sizes and DNS versions used by servers in the five autonomous systems with more than 5 hosts with

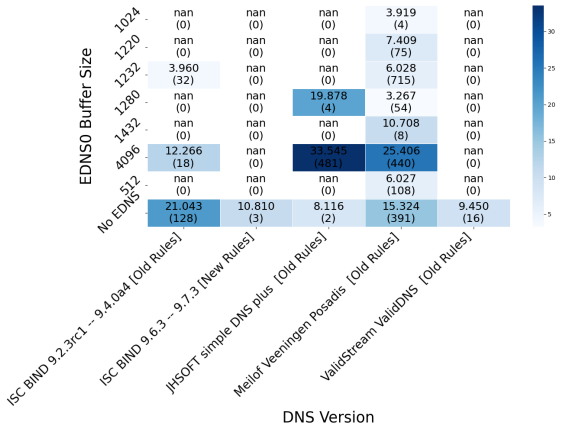

Fig. 7: Different DNS versions and the EDNS0 buffer size they use.

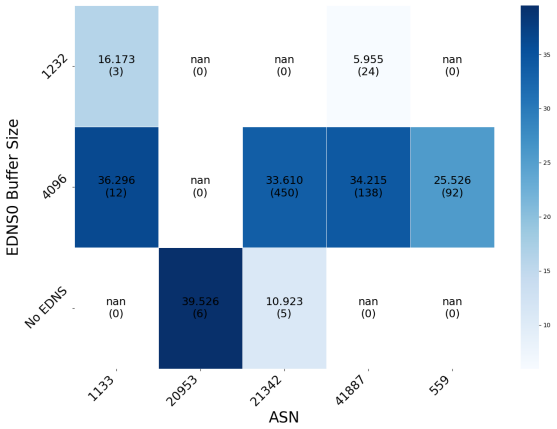

Fig. 9: Autonomous systems with highest BAF and the EDNS0 buffer size their hosts use.

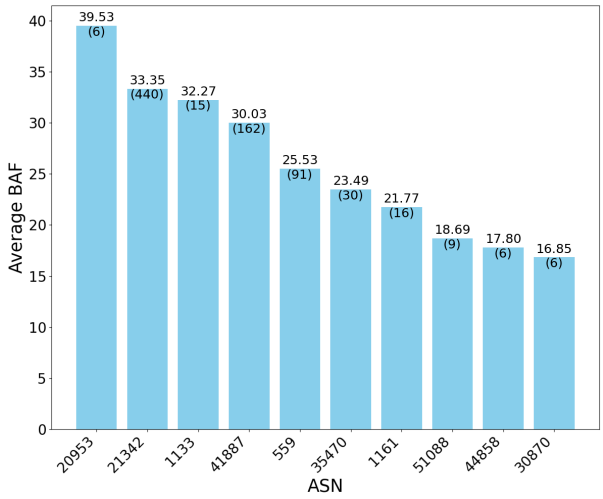

Fig. 8: Autonomous systems with highest BAF.

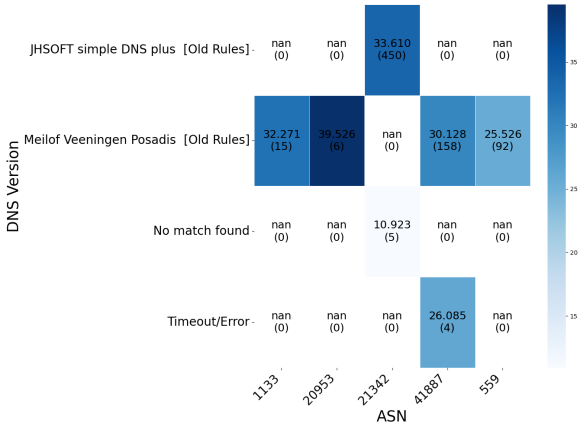

Fig. 10: Autonomous systems with highest BAF and the DNS version their hosts use.

the highest average BAF. After further analysis, using these figures, we find that all but 5 servers have EDNS0 buffer size set to 4096 and all but 5 are running the same DNS version, namely JHSOFT simple DNS plus [Old Rules]. Another ASN with high average BAF and number of hosts is 41887 (Prolocation B.V.). Similar to Akamai 138 out of 162, or roughly 85%, have EDNS0 buffer size set to 4,096, with the other 24 running the recommended 1,232. However, as anticipated the ones running 1,232 have 5.75 times lower average BAF. This shows that servers within an autonomous system are managed in a very similar, or identical, way, and bad management can lead to the creation of a hotspot for high amplification.

We also look at the autonomous systems with the lowest average BAF. Fig. 11 shows a heatmap of the 10 ASs, with more than 5 hosts, with the lowest BAF and the EDNS0 buffer sizes their servers are running. It has to be noted that all these ASs have an average BAF of 1. Unsurprisingly, the heatmap shows that all hosts have low buffer sizes. Only 20, or 11%, have ones higher than the recommended 1,232. Furthermore, 19 of them are in the same AS. This further shows how beneficial good DNS server management can be.

We also tried to see if any of the low BAF autonomous systems were running a specific DNS version, however, we did not have the full data for each host for any of them. This might be due to them running newer versions that the fingerprinting software we used cannot detect yet, though cannot be certain that is the case.

B. NTP

As of the time of the experiments, out of the 30,000 NTP servers that were queried for the monlist command, 209 responded as shown in Fig. 12. Error messages with BAF of 1.75 made up the majority of those responses and there was 1 error message with BAF of 6. However, there were 3 servers with a BAF of 5500, which is also the maximum possible for the monlist query. Finally, one server had a BAF of 4950. Two of these four servers were within the same AS, again hinting toward bad management on AS level.

All the servers that responded were found to be running

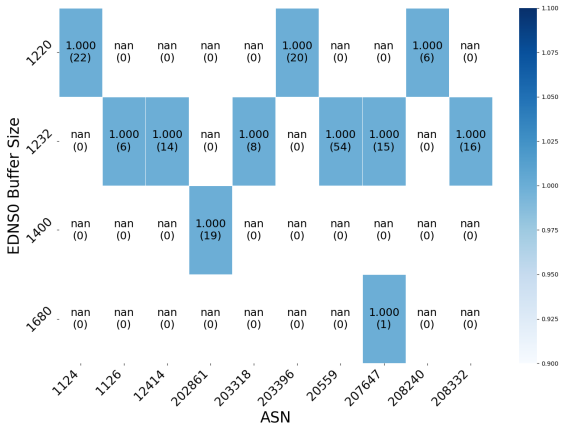

Fig. 11: Autonomous systems with lowest BAF and the EDNS0 buffer size they have.

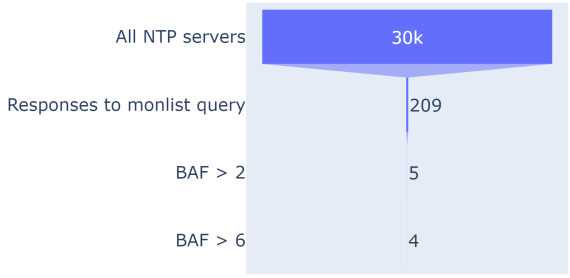

Fig. 12: Distribution of NTP monlist responses.

NTP versions older than 4.2.7-p26. The four servers with very high BAF were searched on Shodan to see if they had signs of being honeypots, however, no conclusive information was found.
Percentage-wise, only about 0.7% of the queried servers can be used for amplification attacks, and close to 1.9% of those have a considerable BAF.

C. Memcached

Out of the 341 Memcached servers in the Netherlands, only 12 of them were found to be responding to UDP requests. Using the STATS command they had an average BAF of 69.8. All these servers also responded to the GET command and by querying for the key *ftce* the highest BAF achieved was 58,509 for one of the servers. All of the servers were found to have existing keys for which the GET command had a BAF of over 10,000. Finally, multiple-key queries were made to the servers, and for the majority of them, a BAF of over 20,000 was achieved.

When looking at the Memcached version the 12 vulnerable hosts were running, we found that all but 2 of them were using version 1.4.15 from 2012. Interestingly this version is noted as "a more experimental release than usual" [31] from the creators. Another host was running 1.4.4, an even more outdated version. Finally, the host for which the highest BAF was observed, was running version 1.6.14, released in 2022. This is unexpected because as of Memcached version 1.5.6, UDP, which is required for amplification, is disabled by default. Shodan was used to check if the host exhibited signs of being a honeypot, however, no conclusive evidence was found.

For the 10 servers running version 1.4.15 we found that we did not always get the precalculated BAF and we got variable results for the same keys within 10 seconds. We suspect that this has to do with the version as it did not happen for the other 2 hosts.

We also checked what autonomous systems the vulnerable servers belonged to, finding that they belonged to 9 different ASs. Three of these had two hosts and six had one each. We also checked to see if all Memcached servers within an AS were vulnerable. This was the case for 7 out of the 9. For the eight one we found that only 1 of 12 was vulnerable and for the last one only 2 out of 6.

D. Application layer traffic loops

Using the methodology from [11] on the list of NTP servers in the Netherlands, 1 pair of servers was found that is susceptible to this kind of attack. After checking for DNS, NTP-DNS, and DNS-NTP loops no other pairs of vulnerable hosts were discovered. After cross-referencing with the results for NTP servers it was found that neither of the two hosts open to application layer traffic loop attacks were amplifiers, therefore we can rule out the correlation between the two.

VIII. Discussion

In our study, we systematically evaluated the amplification potential of DNS, NTP, and Memcached protocols within the Dutch IP range. We identified potential amplifiers across DNS, NTP, and Memcached protocols, with different degrees of amplification potency. The high BAF values observed, particularly in NTP and Memcached protocols, show the critical importance of addressing these vulnerabilities in cybersecurity strategies.

A. Findings

For DNS we found that the EDNS0 buffer size is an important factor when it comes to the BAF of a server with high buffer sizes correlating with a higher amplification factor. However, we also revealed a critical issue: while the majority of servers adhered to the recommended buffer size of 1,232 bytes, a subset of servers advertised this buffer size but used another. Since the majority of these servers were in the same AS, this finding indicates centralized misconfigurations of the buffer size.

The investigation into DNS server versions showed two implementations with considerable numbers of badly configured EDNS0 buffer sizes. However, no conclusive evidence of a direct correlation between specific DNS software versions and higher BAF was found. This is partly due to the limited matching success rate of the fingerprinting software used, which only identified versions for 22% of the responding hosts.

The identification of autonomous systems (AS) with high average BAFs, such as Akamai International B.V. and Prolocation B.V., highlights the role of centralized management practices in amplification attack susceptibility. These findings suggest that targeted interventions at the AS level, promoting better practices for DNS server configuration, could reduce the overall amplification potential within these networks.

For NTP we found that a very small subset (0.7%) of the tested servers responded to the monlist command. Still, risks remain as servers with BAF of up to 5500 were found. Notably, all responding servers were running outdated NTP versions (older than 4.2.7-p26).

Finally, for Memcached, only 12 servers were found to be exploitable. However, due to their severe amplification potential of over 20,000 BAF they remain a threat in the network security landscape.

B. Mitigations

For DNS we propose following the recommendations from DNS flag day 2020 and running EDNS0 buffer size of 1,232. Furthermore, we recommend implementing RFC 8482 and blocking ANY request over UDP. This requires updates to current DNS server configurations that allow ANY requests, focusing on servers with the largest response sizes. Finally, rate limiting can be applied to hosts in order to restrict the number of requests from a single IP address. While challenging due to the potential impact on legitimate users and the ability of attackers to cycle through nameservers, careful rate limiting can help reduce attack sizes.

As for NTP, we suggest encouraging updates to the latest NTP versions and deprecate outdated versions that are susceptible to amplification attacks. If that is not possible, we advise disabling the monlist command manually.

Finally, for Memcached we recommend updating to a version newer than 1.5.6, as UDP is disabled by default. If UDP cannot be disabled, consider fully blocking external access or restricting access to a designated set of necessary source IPs.

C. Comparison with other countries

This research was conducted within a group of 5 students, each focusing on the vulnerabilities within different countries. The countries studied were The Netherlands, Greece, France, Sweden, Belgium and Luxembourg. Overall, similar trends were observed within each of the aforementioned countries with higher EDNS0 buffer size correlating with higher BAF. However, for DNS the Netherlands was found to be the least susceptible to abuse, providing lower amplification factors. For NTP we found servers providing the maximum BAF of 5,500 in each country. Finally, for Memcached the Netherlands and France appear to be the most vulnerable with the two countries having a considerable amount of abusable servers, though higher BAF was achieved using the dutch servers.

D. Limitations

Our study was limited to the Dutch IP range and may not reflect global trends. Furthermore, we could not get a list of all servers in the Netherlands as we didn't have full access to Censys and we didn't utilize active scanning. Moreover, the fingerprinting software(fpdns) used to get the DNS version of the servers was limited in the number of servers it returned a match for.

Another limitation of the study is the fact that for the DNS hosts tested for ANY request, it was only assumed that they were located in the Netherlands because they are authoritative for ".nl" domains. While this should be the case for most, if not all, there could be exceptions. Moreover, DNS servers were only tested for ANY request, even though there are other types of requests such as DNSKEY and TXT. There are also other NTP queries that can lead to high amplification, but only *monlist* was studied.

IX. Conclusion

In this paper, we conducted a comprehensive study on the amplification potential of DNS, NTP, and Memcached protocols within the Dutch IP range. We uncover significant issues, pinpointing factors like misconfigured EDNS0 buffer sizes in DNS servers and the use of vulnerable, outdated NTP and Memcached versions. We have identified potential amplifiers in these protocols, some showing significant amplification potential. This highlights the urgent need to implement mitigation strategies as soon as possible. Furthermore, we revealed widespread misconfigurations within autonomous systems, indicating the importance of targeted interventions at the AS level to promote better practices for server configuration.

While conducting the study we also created a methodology that can be used as a systematic approach to identifying amplification hazards within the Dutch IP range and measuring their potency. By addressing the identified vulnerabilities and implementing suggested mitigations, network administrators can enhance the resilience of their infrastructure against DRDoS attacks.

In conclusion, our research provides valuable insights into the amplification hazards within the Dutch IP range, emphasizing the need for proactive measures to enhance cybersecurity. By identifying and quantifying these vulnerabilities, we contribute to the broader understanding of DRDoS attacks.

There are several areas for further exploration that can build on our findings. Expanding the study to include a global analysis of amplification hazards would provide a more comprehensive understanding of the threat landscape. Additionally, investigating other protocols that may be susceptible to amplification attacks, beyond DNS, NTP, and Memcached, could help identify new vulnerabilities. Conducting longitudinal studies to track changes in amplification potential over time and assess the effectiveness of mitigation strategies would also be beneficial. By addressing these areas and improving our methodology, future research can build on our findings and contribute to the ongoing efforts to mitigate the risks associated with DRDoS attacks.

References

[1] Cybersecurity and I. S. Agency, "UDP-Based Amplification Attacks," 2019. https://www.cisa.gov/news-events/alerts/2014/01/17/udp-based-amplification-attacks, Last accessed on 2024-05-08.

[2] Nexusguard, "DDoS Trend Report 2024," 2024. https://www.nexusguard.com/threat-report/ddos-trend-report-2024, Last accessed on 2024-05-08.

[3] UPX, "Largest DDoS attacks ever recorded," 2023. https://upx.com/en/post/5-largest-ddos-attacks/, Last accessed on 2024-05-08.

[4] C. Rossow, "Amplification hell: Revisiting network protocols for ddos abuse," 01 2014.

[5] H. Griffioen, K. Oosthoek, P. van der Knaap, and C. Doerr, "Scan, test, execute: Adversarial tactics in amplification ddos attacks," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, (New York, NY, USA), pp. 940–954, Association for Computing Machinery, 2021.

[6] 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)*, pp. 500–508, 2021.

[7] Cloudflare, "What is DNS?." https://www.cloudflare.com/learning/dns/what-is-dns/, Last accessed on 2024-05-08.

[8] O. van der Toorn, M. Müller, S. Dickinson, C. Hesselman, A. Sperotto, and R. van Rijswijk-Deij, "Addressing the challenges of modern dns a comprehensive tutorial," *Computer Science Review*, vol. 45, p. 100469, 2022.

[9] "Memcached." https://memcached.org/, Last accessed on 2024-05-24.

[10] Akamai, "State of the Internet Security Report (Attack Spotlight: Memcached)," 2018. https://www.akamai.com/site/en/documents/state-of-the-internet/soti-summer-2018-attack-spotlight.pdf, Last accessed on 2024-05-18.

[11] Y. Pan, A. Ascheman, and C. Rossow, "Loopy Hell(ow): Infinite Traffic Loops at the Application Layer," 4 2024.

[12] M. Azure, "Azure DDoS Protection—2021 Q3 and Q4 DDoS attack trends." https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/, Last accessed on 2024-06-01.

[13] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 31, p. 38–47, jul 2001.

[14] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "Dns amplification attack revisited," *Computers & Security*, vol. 39, 12 2013.

[15] D. C. MacFarland, C. A. Shue, and A. J. Kalafut, "The best bang for the byte: Characterizing the potential of dns amplification attacks," *Computer Networks*, vol. 116, pp. 12–21, 2017.

[16] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "Dnssec and its potential for ddos attacks: a comprehensive measurement study," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, (New York, NY, USA), p. 449–460, Association for Computing Machinery, 2014.

[17] R. Burton, "Unsupervised learning techniques for malware characterization: Understanding certain dns-based ddos attacks," *Digital Threats*, vol. 1, aug 2020.

[18] 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*, IMC '14, (New York, NY, USA), p. 435–448, Association for Computing Machinery, 2014.

[19] L. Rudman and B. Irwin, "Characterization and analysis of ntp amplification based ddos attacks," in *2015 Information Security for South Africa (ISSA)*, pp. 1–5, 2015.

[20] N. Mishra, S. Pandya, C. Patel, N. Cholli, K. Modi, P. Shah, M. Chopade, S. Patel, and K. Kotecha, "Memcached: An experimental study of ddos attacks for the wellbeing of iot applications," *Sensors*, vol. 21, no. 23, 2021.

[21] Technical University of Munich Database. https://toplists.net.in.tum.de/archive/openpagerank/.

[22] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *22nd ACM Conference on Computer and Communications Security*, Oct. 2015.

[23] Dnspython. https://www.dnspython.org/.

[24] Scapy. https://scapy.net/.

[25] Repository for application-layer loop DoS. https://github.com/cispa/loop-DoS.

[26] Shodan. https://www.shodan.io/.

[27] Konstantin Dimitrov, "Repository for code used in writing the paper.." https://github.com/konstantindimi/ddos-amplification-nl-cse3000.

[28] "DNS flag day 2020." https://www.dnsflagday.net/2020/#:~:text=The%20next%20DNS%20Flag%20Day,systems%20before%20the%20set%20date., Last accessed on 2024-06-01.

[29] fpdns. https://github.com/kirei/fpdns.

[30] J. Software, "Options dialog - DNS - Miscellaneous." https://simpledns.plus/help/options-dialog-dns-miscellaneous.

[31] dormando, "ReleaseNotes1415." https://github.com/memcached/memcached/wiki/ReleaseNotes1415.

Appendix

A. Appendix A - Censys Queries

NTP Query

This is the Censys Search query used to find NTP hosts. It specifies that the servers should run NTP on port 123 and that they should be located in the Netherlands.

```
services : (service_name: NTP and port: 123) and
    location . country : "Netherlands"
```

Memcached Query

This is the Censys Search query used to find Memcached hosts. It specifies that the servers should run Memcached on port 11211 and that they should be located in the Netherlands.

```
services : (service_name: MEMCACHED and port:
    11211) and location.country: "Netherlands"
```

DNS Query
This is the Censys Search query used to find DNS hosts for application layer traffic loops. It specifies that the servers should run DNS on port 53 and that they should be located in the Netherlands.

```
services: (service_name: DNS and port: 53) and
    location.country: "Netherlands"
```

B. Appendix B - Request Packets

DNS ANY Request Packet
This is the packet sent to DNS servers in order to test their BAF for ANY requests. It specifies that it should be sent over UDP, using EDNS0, and that it should be an ANY request.

```
dns_request = IP(dst=ip)/UDP(dport=53)
/DNS(ad=1, qd=DNSQR(qname=domain, qtype=255), ar=
    DNSRROPT(z=1, rclass=4096))
```

NTP monlist Request Packet
This is the packet sent to NTP servers in order to test their BAF for monlist command. It specifies that it should be sent over UDP.

```
ntp_packet = IP(dst=ip)/UDP(dport=123)
/NTPPrivate(mode=7, implementation="XNTPD",
    request_code="REQ_MON_GETLIST_1")
```

Memcached stats Request Packet
This is the packet sent to Memcached servers in order to test their BAF for *stats* requests. It specifies that it should be sent over UDP and it should be sent from port 11211.

```
memcached_packet = IP(dst=ip)/UDP(dport=11211, sport
    =11211)
/Raw(load="\x00\x00\x00\x00\x00\x01\x00\x00stats\r
    \n")
```

Memcached get Request Packet
This is the packet sent to Memcached servers in order to test their BAF for *get* requests. Here *key* is the particular key or keys we want to retrieve from the server.

```
memcached_packet = IP(dst=ip)/UDP(dport=11211, sport
    =11211)
/Raw(load="\x00\x00\x00\x00\x00\x01\x00\x00get■
    key\r\n")
```