Estimating the Amplification Factors in the Network Infrastructure of France

Defining factors that affect amplification DoS attacks

Panayiotis Hadjiioannou

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: Panayiotis Hadjiioannou
Final project course: CSE3000 Research Project
Thesis committee: Georgios Smaragdakis, Harm Griffioen, George Iosifidis

Abstract

Amplification Denial of Service (DoS) attacks have been a persistent challenge in network security, with the consequences ranging from causing minor disruptions to substantial financial losses and irreparable damage to reputation.

In today's network environment, many infrastructures are not primary targets of amplification attacks but unwittingly aid them by sending large responses generated by spoofed packets to the potential victims. The ever-growing number of servers makes manual detection of vulnerable components impractical, emphasizing the urgent need for automated tools, which are currently lacking.

This paper investigates factors that affect amplification DoS attacks on three UDP-based protocols, DNS, NTP, and Memcached. Our analysis indicates that for DNS, factors such as the buffer size, replying to ANY queries, Resource Records (RR), and Name Servers (NS) per domain significantly impact the amplification potential. For Memcached, the key and value lengths substantially affect the amplification factor. Regarding NTP, the magnitude of amplification is influenced by the number of recently contacted clients, with the version being a critical determinant for the likelihood of attack success for both NTP and Memcached.

By incorporating these parameters, we propose the development of an automated tool capable of identifying such vulnerable components within network infrastructures.

1 Introduction

Amplification DoS attack aims to overwhelm the target's system with a massive flood of traffic making it unavailable to its users. Being relatively easy to perform and costly to mitigate, makes it favorable among the attackers community. As a result, there has been a dramatic increase in the frequency and intensity of amplification DoS attacks over the years [1].

Many servers unintentionally aid such attacks by returning larger responses to small requests. Despite the extensive research conducted on amplification attacks, there remains a significant gap in automated methods and tools for identifying such vulnerable components within specific network infrastructures that could be exploited in amplification attacks. This research seeks to address this gap by identifying different parameters that affect the success and magnitude of amplification by estimating the amplification factor produced by vulnerable servers. Understanding these elements allows for the development of automated tools, which can assess the level of resistance against aiding amplification attacks.

In this study the approach to solving the problem involved stepping into the attacker's perspective through a three-step process. First, we aimed to understand the attacking playfield, including the protocols and parameters typically exploited. Second, we identified open servers running these protocols in France. Finally, we sent crafted packets to these servers to measure the amplification factor, the key metric for assessing the success and magnitude of such attacks. This approach enabled us to derive the following main contributions:

- We measured the amplification factor on DNS, NTP, and Memcached servers located in France.
- We identified 4 factors affecting the likelihood of success of amplification attacks on these protocols.
- We defined 8 parameters that can increase the amplification factor.
- We discovered that a large portion of DNS servers claim a smaller buffer size than the one implemented in reality.
- We identified 130 servers running DNS or NTP potentially vulnerable to the traffic-loop vulnerability on the application layer.

The remainder of this paper is organized as follows. Section 2 provides background information, while Section 3 discusses previous research in the field. Section 4 outlines the datasets utilized for the experiment, followed by a detailed description of the methodology employed in Section 5. The results of the experiment are presented in Section 6. Considerations regarding the ethicality and reproducibility of the research are addressed in Section 7. In Section 8, we present observations based on our results and discuss the limitations of this research. Finally, Section 9 offers suggestions for future research directions and concludes the paper.

2 Background

In this section a brief explanation on the underlying principles of amplification DoS attacks is given.

Amplification DoS attack is a reflected and volumetric Denial of Service (DoS) attack. In this attack, the adversary directs traffic toward the victim through an intermediary amplifier by spoofing the source IP address to appear as the victim's IP address. An amplifier is a vulnerable server that returns significantly larger responses to small requests. As a result, a massive volume of traffic is directed towards the victim, leading to disruption of services.

User Datagram Protocol (UDP) is a connectionless protocol, meaning that it does not require a formal handshake between client and server. This property makes it a fast and efficient communication protocol. However, this also makes it susceptible to being used in various types of cyber attacks, such as Amplification DoS.

Domain Name System (DNS) plays a vital role in translating user-friendly domain names like www.google.com into machine-readable IP addresses. *Authoritative DNS servers* maintain the Resource Records (RR) sets for specific domain names, providing crucial information for resolving queries. Resource Records are entries in the DNS database that contain information about a domain name and can have multiple types, such as A, AAAA, NS, TXT, etc. A Resource Records Set (RR set) is a group of RR that share the same record type. When a client initiates a request asking for a specific RR set, *Recursive DNS resolvers* typically serve as the initial point of contact. These resolvers employ recursive resolution, by

traversing the DNS hierarchy to fulfill the request. The process begins by querying a root name server, which directs the resolver to the Top-Level Domain (TLD) server responsible for managing the specific top-level domain (e.g., .fr). The resolver then contacts one of the authoritative servers for that domain name, as indicated by the TLD DNS server. The selected authoritative DNS server returns the requested RR set for that domain name.

Network Time Protocol (NTP) facilitates the clock synchronization between computer systems. It is widely used in various applications such as financial transactions, timestamping in cookies, logging events in network systems, and coordinating scheduled tasks in distributed systems, among many other uses.

Memcached servers are designed to achieve faster database retrieval by storing data in memory as key-value pairs. Each value can store up to 1MB of data, significantly reducing database load and enhancing application performance.

3 Related work

Considerable research has been conducted on amplification attacks. Rossow's study [2], for instance, identified the 14 most exploited UDP-based protocols and parameters by attackers. One proposed mitigation strategy to reduce the response sizes involves DNS servers refusing to reply to ANY requests, with Rossow et al [3] examining the potential impact of this measure. A subsequent study in 2017 by Ercan and Selçuk [4] assessed the implementation of recommended hardening measures across DNS, NTP, and Memcached servers in 41 countries, including France. These measures included refusing ANY requests for DNS, disabling monlist for NTP, and deactivating UDP ports for Memcached servers.

In the study on NTP DDoS attacks by Yu et al [5], they analyzed extensive data from five unique sources, including a DDoS mitigation vendor, ISPs, and a darknet. This comprehensive analysis provided insights into the nature of NTP DDoS attacks, including amplifiers, their overall potential, and the victims of previous attacks. An experiment to measure the amplification potential of Memcached servers was done in a closed environment by Akamai. They measured this amplification factor on a Memcached server created in the lab, by sending UDP packets requesting the server statistics, and the respective values of one or multiple keys [6]. In the paper by Griffoen et al [7] the deployment of honeypots provided insights into attackers' Tactics, Techniques, and Procedures (TTPs), leading to the definition of the phases of amplification attacks.

Despite significant efforts, a notable gap in knowledge persists: there are currently no studies specifically focused on identifying the underlying factors influencing the amplification factor on DNS, NTP, and Memcached servers. Bridging this gap is crucial for enhancing our understanding and mitigation of amplification attacks. Ultimately, this will enable the development of automated tools capable of detecting vulnerable components within infrastructures.

4 Dataset

The collection of servers located in France was done with **Censys** [8] from the 6th until the 11th of May 2024, except for authoritative DNS servers. Censys is a passive network scanning tool that is constantly scanning the internet. The results of these scans are stored in a database, which is publicly available for retrieval. Apart from the most important information which is the IP address, Censys stores other information, such as AS, vendor, and more.

Our account was granted a non-commercial academic membership by Censys, allowing for a higher monthly quota for queries and IP addresses compared to a free account. However, there was still an upper limit on the number of IP addresses that could be queried each month. For the purposes of this research, we determined that the proportion of collected servers as seen in the subsequent Table 1, was sufficient to draw meaningful conclusions.

Protocol	Collected	Total
DNS	5,000	170,000
NTP	5,000	180,000
Memcached	790	790

Table 1: Number of collected servers per protocol using Censys CLI

The results were obtained using the Censys Command Line Interface (CLI), with the main filters targeting servers located in France and running one of the researched protocols on the corresponding port. The exact queries performed are detailed in Appendix A.

5 Methodology

To effectively understand the factors influencing amplification attacks, we adopted the perspective of the attacker in our methodology. The TTPs described and the attack phases defined in the paper by Griffioen et al [7] provided a comprehensive framework that informed and guided the research process. By simulating the TTPs commonly used by adversaries, we aimed to observe and analyze the same elements they encounter, allowing for the identification of the underlying factors of amplification attacks.

5.1 Attacking Grounds

In the first phase, the primary objective was to acquire insights into well-known vulnerabilities in servers, which attackers commonly exploit in amplification attacks. This entailed determining the UDP-based services running on the servers, identifying the parameters that yield a higher amplification factor, and understanding the underlying mechanics.

High amplification factor on **DNS** servers [9] can be achieved by requesting open DNS recursive resolvers or authoritative DNS servers to return all RR sets for a domain name using the *ANY* parameter. Since servers might deploy some mitigation measures against ANY queries, the second best parameter was used according to [3], which was *TXT*.

NTP servers [10] running a version prior to 4.2.7 allowed to be queried with the *monlist* command. It was intended to be a debug feature that returned the last 600 IP addresses of

the devices that queried that particular server. However, this introduced high overhead in the response which can reach up to 5,500×, and later aided attackers in performing amplification attacks.

A recent disclosure has unveiled a traffic-loop vulnerability on the application layer [11]. This vulnerability enables attackers to create an infinite loop between two servers by sending a single trigger packet with a spoofed IP address. The remarkable aspect of this attack lies in its minimal bandwidth requirement coupled with its ability to generate a substantial amplification factor. Consequently, in addition to assessing the susceptibility of servers running DNS or NTP protocols to amplification attacks, we also aimed to determine their vulnerability to traffic loops at the application layer.

A high amplification factor on **Memcached** servers can be achieved by requesting the server's statistics. However, the true amplification potential can be achieved by sending a *GET* request with a valid key to receive the corresponding data. Since Memcached servers can store up to 1MB of data per key-value pair, the amplification magnitude can reach up to 65,000× for a single key-value pair [12].

5.2 Amplifier Discovery

In this phase, potential amplifiers in France's network infrastructure were identified. A potential amplifier is a server running one of the three protocols we were researching on the respective port and replies to UDP requests.

Server Collection. The process began by collecting authoritative DNS servers by identifying the nameservers for popular French domain names. These domain names were gathered by extracting the top 1,000 unique domain names from each of the three files listed in Appendix A.1. All of which, belong to the French country-code top-level domain (ccTLD) .fr, resulting in a total of 3,000 French domain names. This resulted in a dictionary of reverse DNS names and their corresponding domain names. The remaining servers, including DNS recursive resolvers, NTP, and Memcached servers, were obtained through Censys as explained in section 4.

Amplifier Classification. The subsequent step involved classifying the collected servers as potential amplifiers. This was accomplished by sending a simple protocol-specific UDP packet to each server and retaining those that responded without errors. For the **authoritative DNS servers**, we requested the IPv4 address for each reverse DNS name, retaining only the servers that replied. These servers underwent further filtering to ensure their location within France, utilizing IPinfo [13], which provides detailed geolocation information for IP addresses.

For the collected DNS servers via Censys, we filtered them to isolate only those functioning as **recursive DNS** resolvers. This filtering process involved requesting the IPv4 address of google.com, resulting in a list of open DNS recursive resolvers. Regarding **NTP** servers, we sent an NTP packet in mode 3 (client mode) and, retained only those servers that responded with an NTP packet in mode 4 (server mode). Lastly, for **Memcached** servers, we filtered out those that did not respond to a UDP packet requesting the server's statistics. More details regarding the simple UDP packets can be found in B.1. As seen in the following Table 2 the filtering allowed for shrinking the testing surface for the next phase of the methodology.

Protocol	Open	Collected
Authoritative DNS	1,123	3,050
Recursive DNS	307	5,000
NTP	4,622	5,000
Memcached	25	790

Table 2: Open servers located in France per protocol

5.3 Computing the Amplification Factor

By leveraging the information from the previous two phases we crafted tailor-made UDP packets for each potential amplifier. The key metric for computing the amplification factor is defined as follows:

$$\text{BAF} = \frac{\text{len(UDP payload of response)}}{\text{len(UDP payload of request)}} \quad (1)$$

The experiments were performed between the 11th of May and the 7th of June 2024, with the crafting, sending, and capturing of the packets done with Scapy [14], which is a powerful packet manipulation library. Appendix B.2 contains all the crafted packets used during the experiment.

We began the experiment by iterating through the list of open **authoritative DNS servers**. We sent UDP packets requesting the ANY and TXT resource record (RR) types for their corresponding domain names. The response from a **recursive DNS server** consists of five sections: header, question, answer, authority, and additional. The largest section is typically the answer section, which contains the RR set(s) returned by the authoritative DNS server for the queried domain name. To obtain larger amplification factors, we requested the recursive DNS servers to resolve the two domain names that produced the highest amplification factor for each RR type from the experiment on authoritative DNS servers. For the experiment on open **NTP servers**, we sent a UDP packet with the monlist command. If a server was vulnerable to monlist, it responded with multiple packets, up to 100, each containing information for a maximum of 6 IP addresses. The BAF was calculated as the sum of the UDP payloads of all response packets.

For **Memcached** servers, some preliminary steps were required before computing the amplification factor. Initially, we sent a TCP request for the stats slabs and parsed the results to obtain the slab IDs. For each slab ID, we extracted keys with a significant ratio, ensuring the ratio of bytes stored for each key was at least 100 times larger than the key size. If this resulted in a large number of keys, we selected the top 100 that provided the best ratio. These steps were essential to avoid overwhelming the servers with requests yielding a low amplification factor. The keys were gathered over TCP to circumvent potential rate-limiting on UDP requests. Using TCP allowed for faster retrieval of keys without impacting the realism of a potential attack scenario, as they could also be collected over UDP.

Subsequently, we computed the amplification factor for all collected keys for each server by sending a get request with the key to retrieve the stored value. The maximum UDP payload for each response packet from Memcached servers is defined as 1400 bytes, resulting in multiple packets. Similar to the NTP experiment, the BAF was calculated as the sum of the UDP payloads of all packets.

5.4 Loopy attack

The methodology outlined in the paper was employed to identify DNS and NTP servers potentially vulnerable to the traffic-loop vulnerability at the application layer. This methodology consists of five key steps: "Discovery Probes", "Response Clustering", "Loop Probe", "Loop Graph and Loop Search", and finally "Loop Verify".

The provided code [15] was utilized when conducting the experiments with minor adjustments, leading to the following four steps:

1. **"Discovery Probes"**: We sent packets to the collected servers from Censys as seen in Table 1. The packets were sent with Zmap [16], which is a network scanning tool.
2. **"Response Clustering"**: We clustered the responses from the servers based on semantics differences.
3. **"Loop Probe"**: From each cluster, we sampled five responses and sent them to every server to obtain a response. The minimum responses required for a cluster to be sampled were set to 2, with servers offering only one response being disregarded. We deviated from the default value of 100, opting for 2 due to the smaller dataset compared to that in the referenced paper, where such a high threshold would have led to discarding all clusters. Then, we reclustered the responses based on the same semantic differences.
4. **"Loop Graph and Loop Search"**: We constructed a directed graph with the clusters as nodes and the IPs as edges. Cycles were identified using the graph, indicating the potential presence of a loop between the servers. Here, we made a minor adjustment to include cycles with at least two IP addresses, as opposed to the default value of 100. This adjustment was necessary due to our smaller dataset, where the default threshold would have resulted in discarding all cycles.

The "Loop Probe" step included an additional step for evaluating clustering performance, which was considered satisfactory if most servers replied to all five probes. However, due to our small sample size and the fact that most clusters contained fewer than 5 unique responses, we could not fairly evaluate the clustering performance. Furthermore, the "Loop Verify" step involved the creation of a proxy to verify the existence of the loops, which was omitted due to the risk of potentially damaging the servers.

6 Results

This section showcases the results obtained from sending UDP packets to potential amplifiers across DNS, NTP, and Memcached protocols. Additionally, we present the number of servers identified as potentially vulnerable to the traffic-loop vulnerability at the application layer.

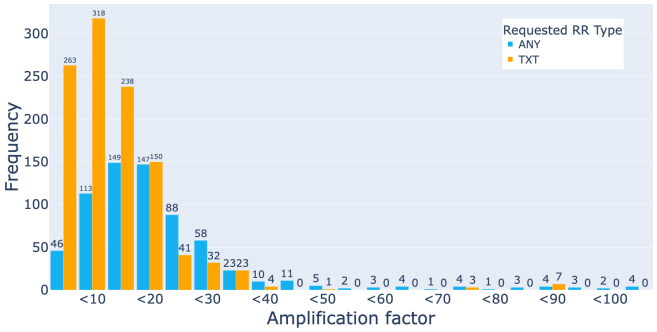

Figure 1: Maximum Amplification factor per Authoritative DNS server requesting ANY and TXT RR types for their domain names

The amplification factor was calculated using the Formula 1. Notably, only amplification factors larger than 1 were considered in our analysis. An amplification factor of 1 indicates that the response was the same size as the request, resulting in no amplification. Factors smaller than 1 were disregarded as they produced negative amplification, rendering them insignificant for our analysis.

6.1 DNS

Authoritative DNS. Figure 1 illustrates the frequency distribution of amplification factors for authoritative DNS servers when queried their domain names with ANY and TXT RR types. For authoritative DNS servers with multiple domain names, we retained only the highest amplification factor for each RR type across all domain names. The horizontal axis represents the amplification factor, while the vertical axis indicates the frequency of each amplification factor. The blue bars correspond to the ANY RR type, and the orange bars correspond to the TXT RR type.

The graph reveals that the majority of authoritative DNS servers exhibit lower amplification factors, predominantly clustered around the ranges of 1 to 20. As amplification factors increase beyond 20, the frequency of occurrences sharply declines for both RR types, with the largest being 102.

We noticed that domain names on an authoritative DNS server yielded different amplification factors. When querying with ANY RR types, we essentially request all RR types available for a domain name. Each domain name can have various RR types, and certain RR types are inherently lengthier, such as DNSSEC records compared to A records. Additionally, RRs of the same type can differ in both length and quantity for each domain, as seen in the results from TXT queries. Therefore, the RR sets and the length of each RR can significantly impact the amplification factor.

Importantly, most of the occurrences of amplification factors above 40× were caused by ANY queries. This suggests that the main cause of high amplification factors is responding to requests with ANY parameter without implementing any of the recommended mitigation strategies as mentioned in RFC 8482 [17], which includes:

- Refuse to reply or respond with a single or small subset of the RR set.

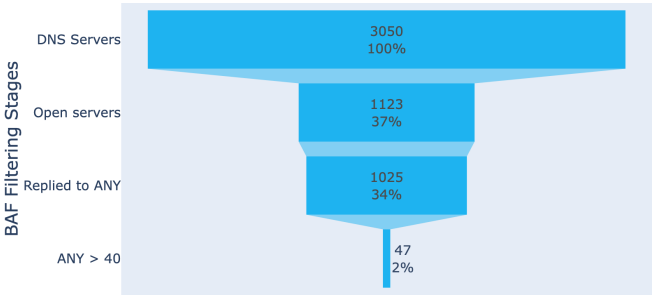

Figure 2: Authoritative DNS server filtering based on responses on ANY request

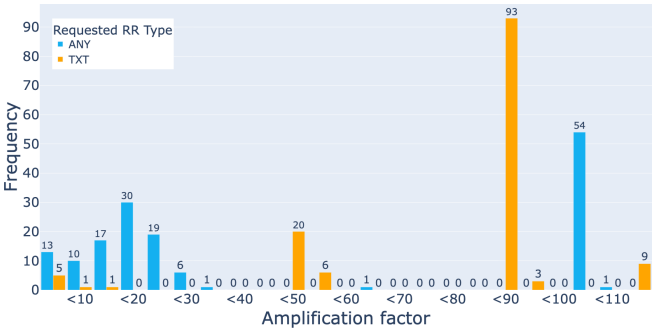

Figure 3: Amplification factor of Recursive DNS servers using ANY and TXT parameter on 2 domain names

- Return whatever is deemed relevant to what the client may have requested.
- Provide a synthesized HINFO RR set

Using Figure 2 we aimed to get an overview of how many servers adhere to the recommendations of the RFC 8482 for returning minimal ANY responses. It was observed that 91% of the open authoritative DNS servers replied to ANY requests. This also included responses that yielded an amplification factor of 1 which, most likely indicated that the responses were truncated and sent over TCP. That means that the server could reply, but due to the implementation of a smaller buffer size, the response was sent over TCP. Furthermore, 4,5% of the servers replied to a request with the ANY parameter yielded a BAF of at least 40×.

However, it is important to note that dropping ANY queries is not the only solution. While there is a tendency to focus on ANY queries, other RR sets, such as TXT, can also be significant. For example, the highest amplification factor obtained by asking for the TXT records was 87×.

Recursive DNS. The succeding Figure 3 illustrates the difference in amplification factor when querying 307 recursive DNS servers with 2 different parameters, namely ANY and TXT. We used the two domain names that yielded the highest amplification factors for ANY and TXT, which were 102× and 87× respectively.

The amplification factors with the highest frequencies were 87× and 102×, which was anticipated. However, we detected that 9 servers when queried with TXT parameter, yielded an amplification factor of 110×, which is significantly larger than the expected of 87×. Upon further investigation, we discovered that the authority section in the response from

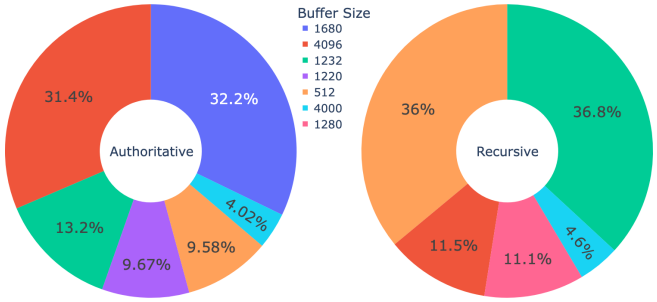

Figure 4: Buffer sizes of Authoritative and Recursive DNS servers

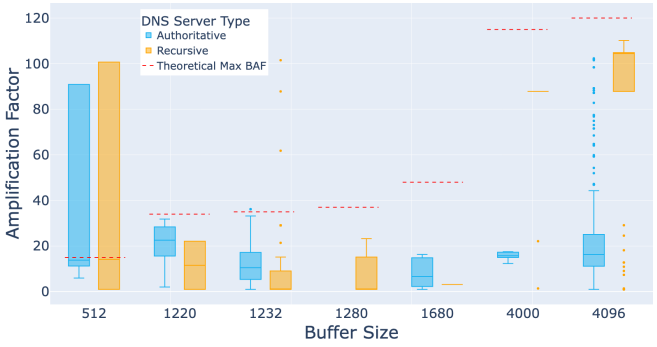

Figure 5: Amplification factors per buffer size of Authoritative and Recursive DNS servers

these servers contained the 13 root name servers, instead of the two nameservers responsible for that domain name. The additional section of the response of the DNS server includes glue records, which essentially comprise the IP addresses of the nameservers found in the authority section of a DNS response. While these records usually consist of IPv4 addresses, if a nameserver supports IPv6, its corresponding IPv6 address is also included. Thus, the amplification factor may increase based on the number of nameservers returned and whether they support IPv6 addresses.

Buffer Size. One crucial element in configuring DNS servers is determining the buffer size, typically ranging from 512 to 4096 bytes. Administrators must strike a balance, setting a buffer size large enough to handle most requests via UDP rather than TCP, yet not excessively large to avoid triggering larger responses. The recommended buffer size as of DNS Flag Day 2020 was 1232 bytes [18]. In Figure 4, we showcase the distribution of buffer sizes that were implemented by at least 5 DNS servers, omitting those that were underrepresented. It was observed that approximately 23% of authoritative and 47% of recursive DNS servers adhere to this recommendation, employing a buffer size of approximately 1232 bytes. However, a significant proportion, about 35% of authoritative and 16% of recursive servers implement a buffer size of 4000 bytes or higher.

Figure 5 presents the amplification factors obtained for both Authoritative and Recursive DNS servers across various buffer sizes. The red dashed line indicates the maximum theoretical amplification factor achievable for each buffer size. It is observed that the median amplification factor generally ranges between 5 and 25, without significant differences between buffer size pairs for most buffer sizes. However, no-

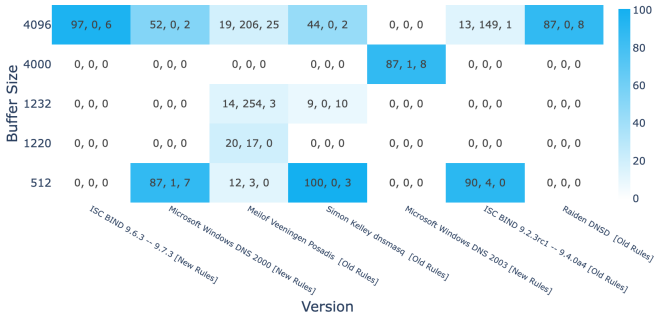

Figure 6: Median amplification factor and the number of Authoritative and Recursive DNS servers for each version-buffer size pair

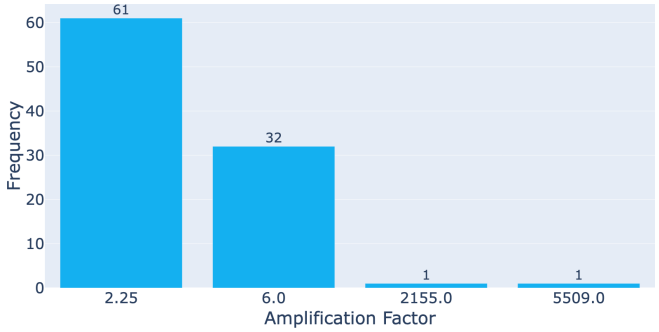

Figure 7: Amplification factor of NTP servers using the monlist command

table differences are evident at buffer sizes of 4000 and 4096 bytes, where recursive DNS servers exhibit a median amplification factor of approximately 100, which is significantly higher compared to authoritative DNS servers.

Another critical observation is that numerous servers provided responses larger than the specified supported buffer size. For instance, a substantial portion of servers claiming a buffer size of 512 bytes resulted in amplification factors greater than 15, which is around the maximum theoretical amplification factor for that buffer size. Additionally, instances of servers reporting smaller buffer sizes were identified for the buffer size of 1232 bytes, indicating discrepancies between the reported and actual buffer sizes.

Furthermore, the relation between buffer size and DNS version was investigated. Figure 6 illustrates the median amplification factor, along with the number of authoritative and recursive DNS servers for each version-buffer size pair, excluding those that were underrepresented. The most commonly implemented version among the servers was Meilof Posadis. The median amplification factor for this version across all buffer sizes ranged between 12 and 20×. In contrast, other versions exhibited significantly larger amplification factors, ranging from 87 to 100×. The highest median amplification factors were observed on buffer sizes of 512, 4000, and 4096 bytes. BIND is the most widely used DNS software, with ISC BIND version 9.16 and later offering the option to configure authoritative DNS servers to comply with RFC 8482. This compliance involves providing minimal ANY responses that arbitrarily return a single RR set matching the query name. However, it was noted that the tested authoritative servers were running older versions, specifically between 9.2.3rc1 and 9.4.0a4.

Authoritative DNS selection. Domain names have multiple authoritative servers holding their RRs. Each authoritative server can have different properties, such as buffer size. It has been observed that for the same domain name, one authoritative DNS server might respond over TCP because the response size exceeds its buffer size, while another might respond over UDP due to the support of a larger buffer size, such as 4096 bytes.

When a recursive server resolves a client's request, it must select one of the authoritative servers for the domain. As noted by Yu et al [19], there are different algorithms for selecting the authoritative server. These algorithms include choosing the server estimated to provide the fastest response based on previous queries or selecting a server randomly to balance the load. Depending on this selection, the response might be truncated and sent over TCP, yielding no amplification factor, or it might not be truncated and sent over UDP, resulting in some amplification. Therefore, the selection of an authoritative DNS server by the recursive resolver plays a crucial role in determining the likelihood of a successful attack.

6.2 NTP

The results shown in Figure 7 were obtained by sending UDP packets to 4,622 NTP servers with the monlist command. A total of 95 servers responded to the monlist command, representing only 2% of the tested servers. We categorize the servers that replied into three categories: Error message (63%) with a BAF of 2.25, Normal response, server mode (34%) with a BAF of 6, and Replied to monlist command with at least 1 IP address (3%) with a BAF greater than 6.

Significantly, these servers were running either version 4.2.0 or 4.2.6, which have the monlist command enabled by default. However, only two servers returned a proper response to the monlist command, while others running such versions did not respond at all. Thus, while the server version can influence the likelihood of a successful attack, it is not a definitive indicator of vulnerability. Running a version prior to 4.2.7 does not necessarily mean a server is susceptible to the monlist command.

Further investigation revealed that the difference in response sizes from these servers was due to the number of IP addresses returned. Specifically, the server with an amplification factor of 2,155 returned 240 IP addresses, while the server with an amplification factor of 5,509 returned 600 IP addresses. Consequently, when querying vulnerable NTP servers with monlist command, the amplification factor depends on the number of clients that recently contacted the server.

6.3 Memcached

Figure 8 showcases the amplification factors obtained by sending UDP packets requesting the statistics of 790 Memcached servers. In total 25 servers returned the server's statistics which accounts for 3% of the tested servers. The vast majority of servers yielded an amplification factor between 70 and 80 with the average being 71.3×. Two servers returned a slightly smaller amplification factor of 54 and 59.

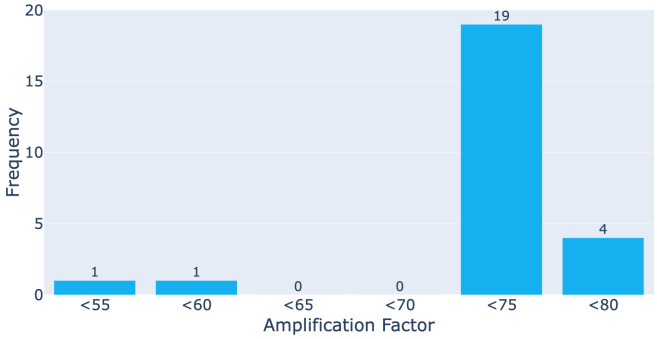

Figure 8: Amplification factor of Memcached servers asking for the server statistics

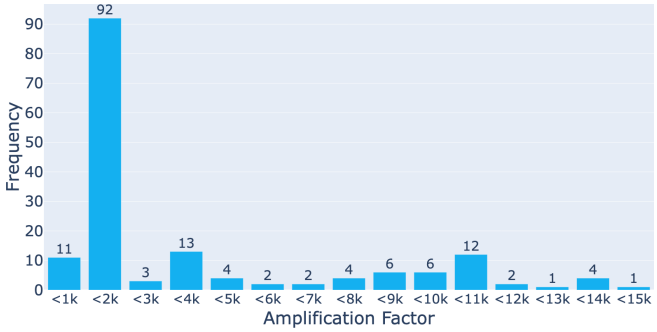

Figure 9: Amplification factor of Memcached servers using get key

Memcached server STATS is essentially a dictionary of predefined fields and values that are dynamically defined by the server administrator or based on the interaction of the clients with the server. Therefore, the fields are constant in the response and the value size is a changing variable. It was observed that the higher the value size of each field, the higher the amplification. Subsequently, the cumulative length of the values can affect the magnitude of amplification. This was also observed in the experiment results where the values of the two servers with the lowest BAFs were significantly smaller compared to the rest servers, which explains the difference of around 20× between them.

Get key. The frequencies of amplification factors obtained by request to get the value for a specific key for 25 Memcached servers are presented in Figure 9. High variability in the amplification factors was observed, with values represented uniformly across different ranges. However, a notable exception is observed between the 1k-2k BAF, which had the highest frequency of 92. It was observed that 20 queries yielded an amplification factor higher than 10,000×, topping at 14,110×. This substantial amplification potential could cause severe disruptions and considerable network congestion.

The analysis of the components of the responses allowed for the definition of the subsequent Formula 2 for the theoretical computation of the amplification factor when performing a get request with one key. The response size is simplified by keeping only the term that affects the size of the response the most, which is the value stored for the key:

$$\text{Get Key BAF} = \frac{\text{len(value)}}{15 + \text{len(key)}} \quad (2)$$

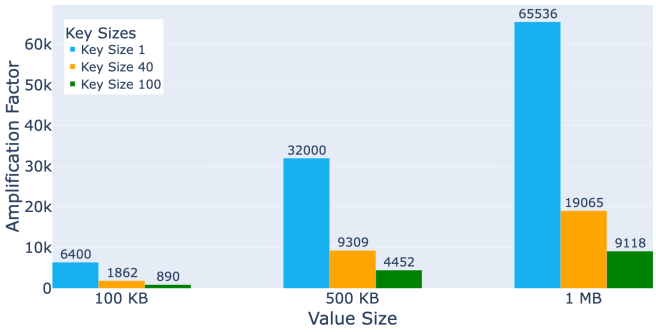

Figure 10: Theoretical experiment on the amplification factor for different keys and value sizes on Memcached

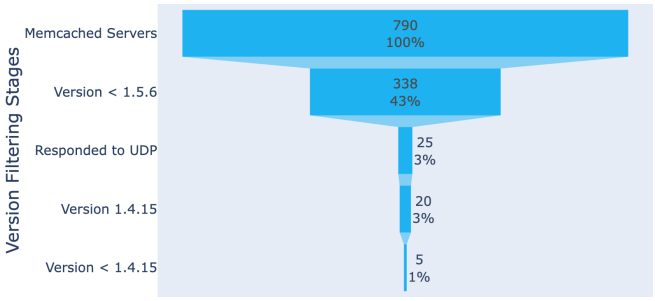

Figure 11: Funnel chart of the versions of Memcached servers

where the key length can range from 1 to 250 bytes and the value size can vary from 1 byte to 1 MB.

Figure 10 illustrates the effect of the parameters in the formula. The theoretical amplification factor was computed for three distinct value sizes of 100 KB, 500 KB, and 1 MB, with each value being associated with three distinct key sizes of 1, 40, and 100 characters. When the key size is held constant, increasing the value size leads to a substantial rise in the amplification factor. For example, with a key size of 1 byte, the amplification factor increases from 6,400× at a 100 KB value size to 32,000× at 500 KB, and further to 65,536× at 1 MB. Conversely, for a given value size, smaller keys result in higher amplification factors. For a 1 MB value size, the amplification factor is 65,536× for a 1-byte key, 19,065× for a 40-byte key, and 9,118× for a 100-byte key. Hence, a 1-byte key exhibits an amplification factor 3.4 times greater than a 40-byte key and 7.2 times greater than a 100-byte key.

Version. The determinant parameter for the success of amplification DoS attacks using Memcached servers as reflectors was whether they had the UDP port open. Memcached servers running version 1.5.6 and later have the UDP port disabled by default. Figure 11 illustrates the filtering stages of the versions of the servers. Our findings revealed that 40% of the servers were running versions with the UDP port enabled by default. Importantly, all servers that responded to UDP requests were within this category. This indicates that the server version significantly influences the likelihood of successful amplification attacks. However, it is crucial to note that the server version alone does not guarantee vulnerability. The servers that responded account for only 7% of those running versions prior to 1.5.6, suggesting that the majority have manually disabled the UDP port. Remarkably, 80% of the servers that responded to UDP requests were running version

1.4.15, an experimental version known for its extremely fast read capabilities.

6.4 Loopy Attack

The subsequent Table 3 outlines the number of e DNS and NTP servers potentially vulnerable to the traffic-loop vulnerability on the application layer:

Metric	Cycles	IPs
DNS responses on DNS servers	1	5
NTP responses on NTP servers	7	47
NTP responses on DNS servers	2	78
DNS responses on NTP servers	0	0

Table 3: Number cycles and IP addresses identified for same and cross-protocol servers

Based on the results, it was observed that the majority of potential loops were created by sending NTP responses to either NTP or DNS servers. In contrast, a significantly smaller number of DNS servers were identified as potentially involved in loops when sending DNS responses. Furthermore, sending DNS responses to NTP servers did not result in any potential loops. Notably, all identified cycles were within the same cluster, indicating that servers with semantically similar responses were responsible for causing the loops. Detailed results are provided in Appendix C.

7 Responsible Research

This section aims to argue how the ethicality of the research was questioned at each step and why it can be easily reproduced.

Prevention from damaging the servers. To collect potential amplifiers in France's network infrastructure, we opted for passive scanning instead of active scanning. This approach is more ethical, as it avoids direct network scanning. Rather than scanning the network ourselves, we obtained results from precomputed scans by querying Censys' database.

Each step of the methodology involved querying a large number of servers for specific purposes. Conducting these experiments manually would have been impractical, thus we developed automated scripts to expedite the process. However, without appropriate precautions, this could have resulted in server damage. To mitigate the risk of overloading the servers with traffic and potentially causing harm, we implemented a delay of several milliseconds between each request. Additionally, instead of estimating the amplification factor immediately after collecting the servers, we introduced an intermediary step. This step filtered out servers that were not open and, therefore, would not yield any amplification. This approach reduced the number of servers involved in the experiments, effectively avoiding unnecessary stress on any server.

The final phase in the methodology of the traffic-loop vulnerability, required to verify the identified cycles. Typically, this involves setting up a proxy to control the flow of packets between the servers in the cycle, preventing them from being overwhelmed. However, we opted to omit this step due to concerns about the risk of server damage from potential misconfiguration of the proxy.

Reproducability. To ensure the reproducibility of this research, all queries for server collection from Censys and the UDP packets were documented in Appendix A and Appendix B. Additionally, we have published all the code used in this research in a public GitHub repository. The code is configured to run in a Docker container, facilitating easy execution on any device. A comprehensive README with step-by-step instructions is included, and the Python files are thoroughly documented.

However, it is important to note that server availability and data can change over time. Servers that are accessible today may become unavailable in the future and the opposite. Additionally, the data stored on these servers may be updated or altered. Consequently, even if the methodology is followed precisely as outlined in this research, the results may differ.

8 Discussion

This section aims to entail observations we made based on the obtained results as well as the limitations of this research.

Overall, only 4.5% of open authoritative DNS servers yielded an amplification factor above 40× when queried with the ANY RR type. Furthermore, 0.43% of NTP servers responded to the monlist command, and 3.2% of Memcached servers replied to the stats request. This indicates that some hardening measures against participation in amplification DoS attacks have been implemented by server administrators. However, the identified vulnerable servers still exhibit a high amplification potential. Recursive DNS servers generally yielded higher amplification factors, with the highest reaching 110×, compared to 102× for authoritative DNS servers. Extreme amplification factors were observed for NTP and Memcached servers, reaching 5,509× and 14,110×, respectively. If leveraged in an attack scenario, such servers could cause substantial damage to the targets.

Upon discussion with my peers, who also investigated other European countries, we identified many similarities. DNS servers often report a smaller buffer size than what is implemented. Importantly, a significant portion of DNS servers employed large buffer sizes of 4000 and 4096 bytes. In the Netherlands and Sweden, the identified vulnerable NTP servers were also running versions prior to 4.2.7, including versions 4.2.0, 4.2.4, and 4.2.6. Furthermore, the majority of vulnerable Memcached servers in the Netherlands were found to be running the experimental version 1.4.15.

Limitations. The collection of potential amplifiers via passive scanning was deemed as a more ethical approach. However, this method was not without its limitations. A notable constraint was encountered regarding the total number of IP addresses that could be gathered. Censys enforces a maximum limit on the number of IPs that can be collected per month, consequently constraining the number of servers accessible for this research. This affected the results for DNS and NTP servers when computing the amplification factor and identifying loop pairs between the servers that are vulnerable to the traffic-loop vulnerability on the application layer.

Additionally, it is important to emphasize that the servers

identified in section 6.4 are marked as potentially vulnerable to the traffic-loop vulnerability. This label arises because the final verification step of our methodology which included the implementation of a proxy was not performed. Consequently, we cannot conclusively determine that the servers were in fact vulnerable.

The amplification factor for authoritative servers is influenced by the selection of domain names, as each domain name corresponds to different RRs stored per server. This, in turn, impacts recursive DNS servers, as domain names with the highest amplification factor on the authoritative server are utilized to query recursive DNS servers.

During the research, fpdns [20] was utilized to perform DNS server fingerprinting. It is important to note, however, that the accuracy of these results may be questionable. Fpdns is a tool specifically designed for server fingerprinting, yet the versions of DNS servers it identifies might not always be precise. Consequently, any conclusions drawn from this data should be considered with caution, acknowledging the potential for inaccuracies.

9 Conclusion

In this paper, we identified factors that affect amplification DoS attacks by adopting the methodologies of the attackers. Our findings demonstrate that factors such as not returning minimal responses to ANY queries, the RR sets, and the number of nameservers per domain significantly impact the response size of a DNS server. Additionally, the buffer size of DNS servers influences the likelihood of a successful attack. For Memcached servers, the lengths of the key and value significantly affect the magnitude of amplification. Similarly, for NTP servers, the number of clients is the main determinant of the amplification factor. Lastly, the version of both NTP and Memcached servers affects the likelihood of success for amplification attacks.

Future research could involve gathering a larger sample of servers to enable more generalizable observations. For DNS, a deeper investigation into querying with the ANY parameter could be conducted. This includes parsing the responses to assess the number of servers that adhere to the recommended measures outlined in RFC 8482, as well as their effectiveness. Furthermore, the identified factors for all protocols could be formalized into formulas to represent the likelihood of a server being exploited by attackers. Additionally, a comparison between the experimental version 1.4.15 of Memcached servers can be performed with other versions to measure if the attackers have an advantage or disadvantage.

Availability

The code used in our project can be found at: https://github.com/panayiotishad04/Amplifier-Collector.

References

[1] C. Sparling and M. Gebhardt, "The relentless evolution of ddos attacks," https://www.akamai.com/blog/security/relentless-evolution-of-ddos-attacks.

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

[3] 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.

[4] E. M. Ercan and A. A. Selçuk, "A study on exploitable drdos amplifiers in europe," *International Journal of Information Security Science*, vol. 10, no. 2, p. 26–41, 2021.

[5] 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*, ser. IMC '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 435–448. [Online]. Available: https://doi.org/10.1145/2663716.2663717

[6] Akamai, "Attack spotlight: Memcached," https://www.akamai.com/site/en/documents/state-of-the-internet/soti-summer-2018-attack-spotlight.pdf.

[7] H. Griffioen, K. Oosthoek, P. van der Knaap, and C. Doerr, "Scan, test, execute: Adversarial tactics in amplification ddos attacks," ser. CCS '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 940–954. [Online]. Available: https://doi.org/10.1145/3460120.3484747

[8] Censys, "Censys search engine," https://search.censys.io/.

[9] Cloudflare, "Dns amplification attack," https://www.cloudflare.com/en-gb/learning/ddos/dns-amplification-ddos-attack/.

[10] ——, "Ntp amplification ddos attack," https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[11] Y. Pan and C. Rossow, "Loope hell(ow): Infinite traffic loops at the application layer," 2024, pp. 4–6. [Online]. Available: https://doi.org/10.60882/cispa.25470952.v1

[12] Cloudflare, "Memcached ddos attack," https://www.cloudflare.com/learning/ddos/memcached-ddos-attack/.

[13] IPinfo, "IPinfo," https://ipinfo.io/.

[14] Scapy, "Scapy," https://scapy.net/.

[15] C. Rossow and Y. Meui, "Traffic-loop vulnerability code," https://github.com/cispa/loop-DoS.

[16] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *22nd USENIX Security Symposium*, 2013.

[17] J. Abley, O. Gudmundsson, M. Majkowski, and E. Hunt, "Providing minimal-sized responses to dns queries that have qtype=any," https://datatracker.ietf.org/doc/html/rfc8482.

[18] DNS Flag Day, "Dns flag day 2020," https://www.dnsflagday.net/2020/#dns-flag-day-2020.

[19] Y. Yu, D. Wessels, M. Larson, and L. Zhang, "Authority server selection in dns caching resolvers," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 2, p. 80–86, mar 2012. [Online]. Available: https://doi.org/10.1145/2185376.2185387

[20] D. Wessels, S. Jobe, J. Schlyter, O. Johansson, O. Surý, and E. Betts, "Fpdns," https://github.com/kirei/fpdns.

[21] Domain Names Archive, "Domain names archive," https://toplists.net.in.tum.de/archive/, accessed on 08 of May 2024.

A Censys Queries

This appendix enlists the queries used to gather servers for each protocol using Censys CLI.

Query used to collect 5,000 **DNS** servers in France:

```
censys search "location.country_code: FR AND services.service_name: DNS AND services.port: 53" --pages 20 > dns_query_results.json
```

Query used to collect 5,000 **NTP** servers in France:

```
censys search "location.country_code: FR AND services.service_name: NTP AND services.port: 123" --pages 20 > ntp_query_results.json
```

Query used to collect 790 **Memcached** servers in France:

```
censys search "location.country_code: FR AND services.service_name: MEMCACHED AND services.port: 11211" --pages 20 > memcached_query_results.json
```

A.1 Top French Domains

The collection of the top French domains was done using the subsequent three files, found in [21] with the following order:

- alexa-top1m-2009-01-29.csv;
- cisco-umbrella-top1m-2016-12-15.csv;
- quantcast-top-sites-2018-05-22_2200_UTC.txt

B Packet Crafting

This appendix contains code snippets to demonstrate the exact parameters used for the UDP packet construction using Scapy for each server.

B.1 Simple UDP Packets

This subsection details the packets used to determine if a server is open.

DNS A RR

```
The following packet uses the dnspython library to resolve and retrieve the IPv4 addresses (A records) for a specified domain name:
a_records = dns.resolver.resolve(domain, 'A')
```

DNS NS RR

The following packet uses the dnspython library to resolve and retrieve the nameserver (NS) records for a specified domain name:

```
ns_records = dns.resolver.resolve(domain, 'NS')
```

NTP Client Request

The code snippet constructs a simple NTP packet in client mode (mode 3) using the scapy library:

```
request_packet = IP(dst=ntp_server) / UDP(sport=RandShort(), dport=123) / NTPHeader(mode=3)
```

Memcached STATS

The code snippet constructs a packet to request server statistics from a Memcached server using the scapy library:

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

B.2 Amplifying Packets

This subsection presents the packets used to produce large responses.

DNS ANY RR

The code snippet constructs a DNS query packet using the scapy library to request all available record types (ANY) for a particular domain name:

```
request_packet = IP(dst=dns_server) / UDP(sport=RandShort(), dport=53) / DNS(ad=1, qd=DNSQR(qname=domain_name, qtype=255), ar=DNSRROPT(rclass=4096, z=1))
```

DNS TXT RR

The code snippet constructs a DNS query packet using the scapy library to request TXT records for a particular domain name:

```
request_packet = IP(dst=dns_server) / UDP(sport=RandShort(), dport=53) / DNS(rd=1, ad=1, qd=DNSQR(qname=domain_name, qtype='TXT'), ar=DNSRROPT(rclass=4096, z=1))
```

NTP Monlist

The code snippet constructs an NTP packet to request the "monlist" command using version 3 of the NTP protocol in private mode:

```
request_packet = IP(dst=ntp_server) / UDP(sport=RandShort(), dport=123) / Raw(load="\x17\x00\x03\x2a" + "\x00" * 4")
```

Memached Get Key

The code snippet constructs a packet to request the value associated with a specific key from a Memcached server using the scapy library:

```
request_packet = IP(dst=memcached_server) /
    UDP(sport=RandShort(), dport=11211) / Raw
    (load="\x00\x01\x00\x00\x00\x01\x00\
    x00get {}\r\n".format(key))
```

C Loopy Attack Results

This appendix aims to show the results of the loopy attack for each experiment.

Table 4 demonstrates the number of identified cycles when probing DNS servers with DNS responses:

Cycle	Number of Involved IPs	Min Edge IP Amount	Min Edge	Number of Pairs
[7, 7]	5	5	[7, 7]	10

Table 4: DNS on DNS traffic-loop cycles

Table 5 outlines the number of identified cycles when probing NTP servers with NTP responses:

Cycle	Number of Involved IPs	Min Edge IP Amount	Min Edge	Number of Pairs
[13, 13]	5	5	[13, 13]	10
[11, 11]	11	11	[11, 11]	55
[10, 10]	8	8	[10, 10]	28
[4, 4]	2	2	[4, 4]	1
[2, 2]	11	11	[2, 2]	55
[1, 1]	12	12	[1, 1]	66
[3, 3]	2	2	[3, 3]	1

Table 5: NTP on NTP traffic-loop cycles

Table 6 presents the number of identified cycles when probing DNS servers with NTP responses:

Cycle	Number of Involved IPs	Min Edge IP Amount	Min Edge	Number of Pairs
[8, 8]	46	46	[8, 8]	1035
[6, 6]	46	46	[6, 6]	1035

Table 6: NTP responses on DNS servers traffic-loop cycles