When More Isn't Merrier: Exploring DNS Amplification Hazards in the Dutch Digital Landscape

Wouter Jehee
Delft University of Technology
Delft, Netherlands
w.jehee@student.tudelft.nl

Ignjat Pejic
Delft University of Technology
Delft, Netherlands
i.pejic@student.tudelft.nl

Gregor Schram
Delft University of Technology
Delft, Netherlands
g.schram@student.tudelft.nl

Maarten Weyns
Delft University of Technology
Delft, Netherlands
m.b.m.weyns@student.tudelft.nl

Abstract

DNS amplification attacks allow an adversary to abuse nameservers to generate much more data for their DDoS attack than than they produce themselves. But how much the amplification is differs a lot per server. Inspired by the 2021 paper "ANYway: measuring the amplification DDoS potential of domains" by van der Toorn et al. we set out to look at the amplification potential of the nameservers that support the roughly 1800 domain names owned by the Dutch government. We calculated the amplification potential of every domain and nameserver owned or managed by the government, both with ANY meta queries and TXT records. We found that the Dutch government currently can inadvertently amplify DNS requests up to 70x. We also recommend some changes to the DNS configurations that lower the amplification potential and show a proof of concept to abuse this part of the Dutch Digital Landscape.

1 Introduction

A DNS amplification attack is a denial of service (DoS) attack using open DNS resolvers to overwhelm the target system with DNS response traffic [4]. According to a report by Nexusguard, DNS amplification attacks accounted for 23.2% of all DDoS attacks in Q1 2021 [14], making these attacks highly relevant in the cybersecurity space. Moreover, given that attacks can go up to 3.47 Tbps [3], they can create real havoc to organizations, business entities, and many more. It is a threat that should be handled very seriously, but unfortunately as of 2023 there are still many vulnerabilities that allow for such attacks to take place.

The goal of this paper is to investigate the amplification potential of Dutch DNS servers over UDP, using resource records (RR) of type ANY, something that many adversaries do before deciding which servers to abuse [9]. The secondary objective was to compare the amplification potential of ANY records to TXT records, to be able to explore the effectiveness of removing ANY records on the potency of the domains, and whether there would be any noticeable effect.

After this introduction section, the paper is structured in the following way. First, in section 2, the required background of the paper will be presented. There will be a detailed explanation of what DNS amplification attacks are and how they work, alongside the original paper based on which the idea of this paper came about. After that, section 3 will go over the current state of the art with regard to DNS servers (more specifically, Dutch DNS servers). Moving on, section 4 will go over how the experiment was set up, explaining all the steps in detail. Next, section 5 shows the results and explains them. While, section 6 discusses the limitations of the research and recommends some mitigation strategies against DNS abuse. Lastly, the conclusion in section 7 will summarize the paper and provide the closing remarks.

2 Background

This section will start off by explaining what Domain Name Systems are and how they work, before going over how amplification attacks can be executed using it. After that, it will go over the related work and elaborate on how this paper came about.

2.1 Domain Name System

Domain Name System (DNS) is a hierarchical-distributed system that provides a mapping between domain names and their corresponding IP addresses [5]. It consists of multiple DNS servers that work together to resolve domain names into IP addresses.

Once a DNS query is made by a user, it is first sent to a recursive DNS resolver. The resolver then checks its cache to see if it already has the requested data, and if so, it returns it to the query sender. Otherwise, it forwards the query to a series of DNS servers until it receives the requested data, if that data exists on the servers the resolver communicates with. Otherwise, the user can try different DNS resolvers if they are sure the requested data exists. Figure 1 provides a visual diagram for what exactly happens.

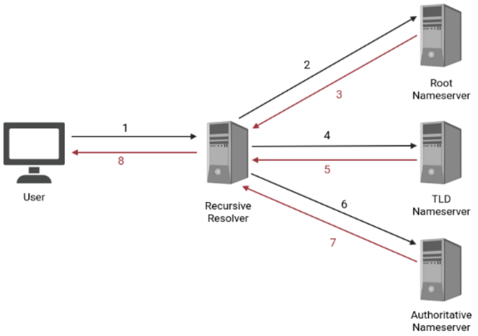

Figure 1. DNS traffic overview

Part	Size (bytes)
Header	12
Transaction ID	2
Flags	2
Questions	2
Answer RRs	2
Authority RRs	2
Additional RRs	2
Query	6 + length domain
Domain name	2 + length domain
Type	2
Class	2
EDNS additional record	11
Total	29 + length domain

Table 1. DNS Request size overview

DNS queries can request a number of records, including, but not limited to:

- A Records: IPv4 address of a domain name
- AAAA Records: IPv6 address of a domain name
- Mail Exchange (MX) records: Mail servers associated with a domain name
- Name Server (NS) records: Authoritative Name Servers of a domain name
- Text (TXT) records: Any text based data associated with a domain name

DNS can use both the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) for communication between clients and servers. UDP is a connection-less protocol, which means there is no established connection between the client and server before the communication takes place. The packets are instead immediately sent. TCP on the other hand is connection based, meaning that there is first a handshake between the client and server before the packets can be sent. In general, DNS prefers UDP due to its lower latency leading to quicker responses, especially when smaller amounts of data are requested.

The size of a DNS request can be calculated as shown in table 1. The header is 12 bytes, the query is the length of the domain name (the amount of characters) plus 2 bytes, and an additional 11 bytes are needed for additional records.

The resulting size is the size of the DNS payload. In order to get the full packet size, the Ethernet, IP, and UDP headers should still be added. These are respectively 14, 20, and 8 bytes. This results in a total packet size of 71 bytes, to which the length of the domain name should be added.

The size of a DNS reply cannot be predicted easily. It depends on the type of query and the data inside the reply. The only certainty is that the reply will at least be as large as the request, as the performed query is included in the reply.

2.2 DNS Amplification

DNS amplification attacks are a type of Denial of Service (DoS) attack that exploit vulnerabilities in the DNS protocol to flood the target server with enough traffic to prevent it from responding to normal traffic [15]. The attacks can be conducted at both the UDP and TCP levels, but the execution would differ. The idea in both of the attacks would be to amplify the message size sent by the user to the open resolver when responding to the target.

In the UDP attack, the amplification would be done by sending queries requesting for ANY records. The ANY resource record is distinct and serves a special purpose of fetching all available information regarding a domain name. Whenever a DNS resolver requests the ANY resource record, the server sends back all the obtainable resource records associated with that domain name, such as records named in the previous section. Therefore, the response would typically be very large compared to the request. The amplification potential is calculated in the following way:

$$amplification\ potential = \frac{request\ size\ (bytes)}{response\ size\ (bytes)}$$

To execute the attack over UDP, the following steps have to take place. First, the attacker spoofs the target's IP address. Then, using that spoofed IP as the source IP, it sends a DNS query message to an open resolver. The query would typically ask for a lot of data to be returned by the DNS resolver, which would send the response message to the query's source IP, which in this case is the target's IP. The attacker would do this process simultaneously with many open resolvers, which would together send a huge amount of data to the target and effectively make it too busy to respond to normal traffic. Figure 2 visualizes the attack.

A DNS amplification attack at the TCP level works in a different way that takes advantage of the fact that TCP requires a three-way handshake to establish a connection. Here, the attacker first sends a SYN packet with a spoofed source IP address to the DNS server, causing the server to respond with a SYN-ACK packet to the target's IP address.

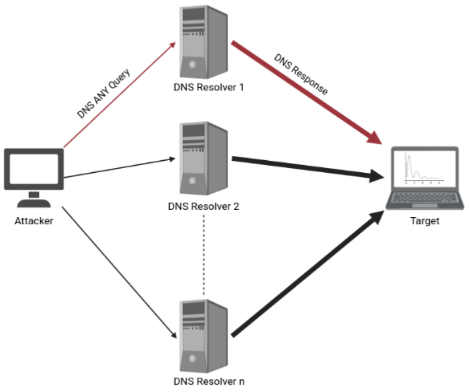

Figure 2. DNS Amplification over UDP

The SYN-ACK packet is typically larger than the original SYN packet, which causes the amplification. The attacker would do this with many servers, similarly to the UDP attack, in the attempt to overwhelm the target.

DNS amplification attacks are a growing threat to organizations, as they are relatively easy to carry out and can cause significant damage to network infrastructure. In recent years, DNS amplification attacks have become more common due to the increased availability of open DNS resolvers on the internet. These resolvers are typically misconfigured and can be used by attackers to launch large-scale attacks with little effort.

The impact of a DNS amplification attack can be severe. By flooding a victim's network with traffic, the attack can cause network congestion, slowing down or even crashing the victim's servers. The attack can also be used as a smokescreen to distract security teams, allowing attackers to carry out other malicious activities undetected.

Some examples of DNS amplification attacks include:

- The 2013 Spamhaus attack [13]: This attack targeted Spamhaus, an anti-spam organization, and peaked at 300 Gbps. The attack lasted for several days and caused widespread disruption across the internet. The attackers used a botnet of compromised servers to launch the attack.
- The 2018 GitHub attack [2]: This attack targeted GitHub, a popular software development platform, and peaked at 1.3 Tbps. The attack lasted for approximately 20 minutes and was one of the largest DDoS attacks ever recorded. The attackers used a botnet of compromised Memcached servers to launch the attack.

2.3 Related Work

The idea of this paper initially came about after reading the paper by van der Toorn et al. [16]. This paper had three main goals:

- Propose and Validate a scalable method to estimate the amplification potential of a domain name based on the ANY response size
- Check whether attackers are using the most-potent domains (greatest amplification potential) for amplification attacks
- Effect of blocking ANY requests to limit DNS-based DDoS attacks

According to this paper, the response size of domains would be reduced by 57% and the size of most-potent domains decreases by 69%. However, certain domains still have a response size of over 2048 bytes to requests that are not of type ANY, still making them potent enough to be used in amplification attacks.

As we are situated in the Netherlands, it was decided to investigate the amplification potential (using ANY requests) of Dutch government DNS servers instead. Also, the amplification potential of TXT would be investigated and compared to requests of type ANY, to check how effective would blocking ANY requests be. Both our paper and the reference paper investigate the amplification potential of DNS amplification over UDP.

3 State of the art

This section contains details about the state of the art of DNS servers, and specifically the ones used by the Dutch government. Subsection 3.1 analyzes the state of ANY requests. The next subsection 3.2 shows the state of TXT records, and we finish this section with a short description of DNSSEC 3.3

3.1 ANY requests

In recent years, there has been scrutiny on the usage of ANY requests for DNS servers. There are multiple reasons why this has happened. First, ANY queries are meta queries and not a type that has an implementation specification. Second, implementing ANY responses is difficult, as aggregating data is hard when working with anycasts [11] and CDNs [1]. Third and final, the ANY type has no legitimate use case and can easily be abused. In 2015 Cloudflare started dropping support for ANY queries, later it proposed the RFC8482 spec [12] to reply with a minimal response to ANY queries to prevent abuse.

Cloudflare took the initiative on deprecating ANY requests, and not many providers have followed suit. When looking at the biggest cloud providers: Amazon Web Service (AWS), Microsoft Azure, Google Cloud Platform, we see all of them allow setting up a DNS. However, none of these seemed to support the RFC8482 spec. Both AWS and Azure do have a few protections against ANY request abuse, mainly they require it to happen over TCP, or it gets truncated, this means that the amplification potential is very low as UDP is required to get the server to send the response to a different IP. Google Cloud doesn't put any protections in place for

ANY requests, as it just sends the full data over TCP as well as UDP. The only provider we found besides Cloudflare that seems to implement RFC8483 is Dyn.

The Dutch government uses Azure DNS for some of its domains, these thus apply the general Azure protections against ANY request abuse. Most of the domains owned by the Dutch government however use servers managed by the government itself. These seem to apply no protections against ANY requests at all, however they do sometimes support DNSSEC but don't enforce it.

3.2 TXT queries

Even if all DNS providers would implement RFC8482 or other protections to reduce the ability of abusing this type of requests, there is another possible attack vector that is slightly less effective but still allows huge amplification in some cases. This attack vector is to do the same trick as done with the ANY request, spoof an IP, request a lot of data from the DNS and send it to the spoofed IP, overloading the victim, but this time using TXT records. A server often has multiple TXT records for things like DMARC validation or SSL certificate checks. Often these records are large (semi) random strings of text to uniquely identify the server and verify the person requesting a certificate or claiming an email server owns the domain.

The problem lies in the fact that these records should be deleted after the verification is done, however most system administrators don't do this. This means that we can request all the TXT records from a DNS and get quite a big response. This response will generally be smaller than the response to an ANY request, but those are getting blocked or truncated. TXT requests can't easily be blocked as they have very legitimate use cases as described before. The only way to make this attack less feasible is to educate system administrators to configure their DNS correctly and remove unnecessary TXT entries [16]. The Dutch government luckily seems to have some policy for cleaning up TXT records, as we found very few DNS servers under their control that contain many TXT records.

3.3 DNSSEC

DNSSEC was developed to provide an added layer of security to the DNS process. The primary reasons for its existence are:

- Data integrity: DNSSEC aims to ensure the integrity of the data received from DNS queries by using digital signatures. This protects users from receiving manipulated or false information, which could be the result of an attacker modifying DNS records to redirect users to malicious websites.
- Authentication: DNSSEC enables DNS resolvers to verify that the DNS data they receive comes from a legitimate source. It does this by using a chain of trust established through cryptographic keys. This helps prevent attacks like DNS spoofing, where an attacker intercepts and alters DNS responses to redirect users to fraudulent websites.
- Non-repudiation: Because of the digital signatures used in DNSSEC, it becomes difficult for an attacker to deny their involvement in an attack. This non-repudiation property helps trace the origin of attacks and provides better accountability.

This sounds great, and it is truly an advancement in the security of the internet, however the attack vector we are looking at: DNS amplification attacks, was not taken into consideration when creating DNSSEC. DNSSEC includes cryptographic keys in its headers, because these keys have to be long to be secure this increases the amplification potential of servers that support DNSSEC [17]. In order to support these larger responses, an extension to DNS was introduced called EDNS [6], allowing for even more amplification potential as it also allows administrators to use it for other purposes than DNSSEC.

When we focus on the Dutch government's domain infrastructure, we see that most of the domains (1664 out of 1814) support DNSSEC thus allowing us to abuse quite some servers for DNS amplification attacks. The only way to protect against this would be to either force all traffic over TCP, which might be difficult considering accessibility from low end (IOT) devices would be limited or do rate limiting, which can cause difficulties for benign traffic.

4 Methodology

The research done can be split in a few different steps. These steps are visually represented in Figure 3 and explained in this section.

4.1 Acquiring domains

The first required step was to find a list of domain names owned by the Dutch government. Luckily, the Dutch government has published this list on their website [1] available as an Excel download.

In order to work with this file, we parsed it in such a way to extract a list of all domains from it.

4.2 Getting authoritative nameservers

From the previously obtained list of domain names, the authoritative nameservers should be extracted. This was done by writing a script that queries all domains and retrieves the authoritative nameservers for them. This was done with the dig command in a bash script:

```
# Get authorotive nameserver for DOMAIN
dig +short NS DOMAIN
```

[1] Available on https://www.communicatierijk.nl/vakkennis/rijkswebsites/verplichte-richtlijnen/websiteregister-rijksoverheid

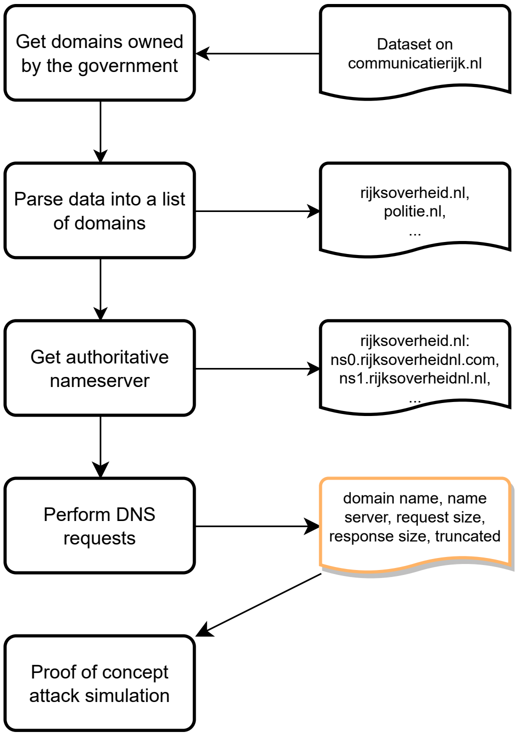
Get domains owned by the government
Dataset on communicatierijk.nl
Parse data into a list of domains
rijksoverheid.nl, politie.nl, ...
Get authoritative nameserver
rijksoverheid.nl: ns0.rijksoverheidnl.com, ns1.rijksoverheidnl.nl, ...
Perform DNS requests
domain name, name server, request size, response size, truncated
Proof of concept attack simulation

Figure 3. Methodology flowchart

4.3 Performing ANY requests

The nameservers obtained with the above command were then queried for ANY and TXT requests. This was also done with the dig command:

```
# Make a request of type TYPE for DOMAIN on SERVER
dig -4 +notcp +ignore +bufsize=4096 @SERVER DOMAIN
    TYPE
```

This command includes some parameters required to get the correct response from the server:

- The -4 parameters specifies that the request should be done over IPv4.
- The +notcp parameter specifies that the request should be made over UDP, rather than TCP.
- The +ignore parameter then additionally specifies that a query receiving a truncated reply should not be re-queried over TCP.
- The +bufsize=4096 parameter tells the nameserver that the UDP buffer size of the client is 4096 bytes (EDNS, briefly described in subsection 3.3)

The output from this script is then put into a CSV file, containing the following information upon script completion:

- The domain name
- The nameserver
- The IPv4 address of the nameserver
- The request size in bytes
- The response size in bytes
- A boolean indicating whether the response is truncated or not

4.4 Calculating packet sizes

As mentioned in section 2, the DNS query size can be calculated by taking 71 bytes and adding the length of the domain name.

The response size is obtained with the help of the output of the dig command. This prints the size of the DNS payload, so just the sizes of the headers (14 + 20 + 8 = 42 bytes) is added to this.

Taking all the headers into account gives a more accurate representation of the sizes, as this is the way the packets are sent over the network between clients.

4.5 Attack Simulation

In order to test the viability of a DNS amplification attack, we implemented a proof of concept attack [10] in the Go programming language [7] with the GoPacket library [8]. The program first reads an input file in the same format as was produced by previous scripts, each line in the file must include the domain name to query for and the IP address of the nameserver to use. Furthermore, the program requires a few important command line arguments to run as well. These include the IP address of the attack target, the number of threads to use and the duration of the attack (there are more possible arguments, but they are less relevant to explain the inner workings of the code).

With all the required data present, the labor will be divided among the threads (if more than 1 is used) by giving a subset of the domain, nameserver pairs to each of the threads. Each thread will then start sending packets to each of the nameservers with the domain name as the query, asking for ANY records until the specified duration of the attack runs out. Once a thread has sent a packet for each of the domain, nameserver pairs, it will start reusing the oldest one. All the packets are sent with a spoofed IP address specified to be the attack target. The questions are comparatively much smaller than the answer, thus the attacker achieves their goal of increasing the amount of data sent without doing it them self.

The provided Docker Compose file allows us to simulate an attack on the same machine in two environments, one running the attack and the other running tcpdump to record the traffic to the file. Afterward, this file can be analyzed to show the size of the attack over the duration of the experiment. There are however some limitations to this approach, which are touched upon in the discussion section.

5 Results

This section contains details about the results of our experiments. Subsection 5.1 shows the results of our queries to each of the nameservers and the domains associated with them.

The results of our attack simulation are shown in subsection 5.2.

5.1 Worst offenders in different query types

In each plot, we make a distinction between the part of the amplification caused by truncated responses and a part by non-truncated responses. Note that each graph has the total amplification potential for the domain or nameserver. This means that, for example, a domain's amplification might consist of 3 different nameservers and vice versa. We think this is a more valuable result than showing the average amplification, as truncated responses will pull those down and an adversary is likely to skip servers that truncate while abusing all the other available servers and not just one. Another important factor to be aware of is that the graphs exclude domains and nameservers that had incomplete data (e.g. missing requests or response size). The full data files can be found alongside our code on GitHub [10].

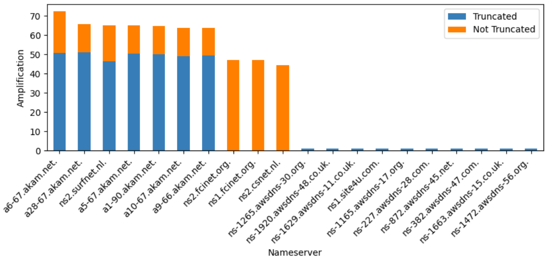

Figure 4. The top and bottom 10 amplifying nameservers for ANY requests, attributed by truncated and non-truncated parts

Looking first at Figure 4 we see that the nameservers hosted by Akamai are responsible for 6 out of 10 of the worst offenders for ANY requests while the bottom 10 is dominated by servers hosted by AWS. What is interesting to see is that the same cannot be said for TXT record amplification shown in Figure 5. Here we see quite a changing list of worst offenders, and also the bottom 10 is not dominated by any provider in particular. This can mostly be attributed to the fact that the response size for ANY requests is largely decided by the configuration of the server as a whole, this is often not configurable. On the other hand, TXT records are added and managed by the people who run a website and use the DNS. This can very much come down to certain teams within the Dutch government cleaning up after using TXT records while others don't or the fact that sometimes a nameserver needs to have a bunch of TXT records for email setups and the likes.

What is also interesting is that one would expect the worst offenders for TXT queries and ANY queries to correlate, but they don't. This indicates that the ANY queries either send records we didn't manage to measure, like DNSSEC, or that they are used for multiple domains and return data on all of those domains. That last hypothesis is debunked quickly by the missing correlation between Figure 6 and 7.

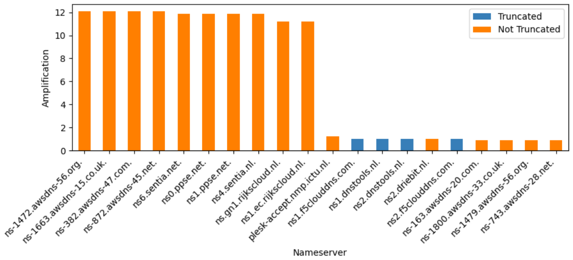

Figure 5. The top and bottom 10 amplifying nameservers for TXT requests, attributed by truncated and non-truncated parts

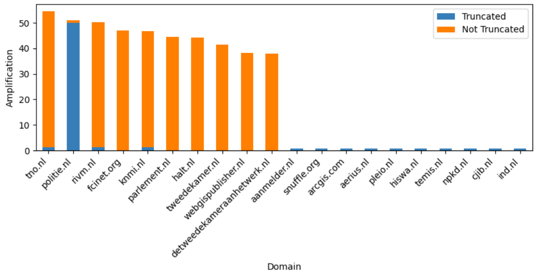

Figure 6. The top and bottom 10 amplifying domains for ANY requests, attributed by truncated and non-truncated parts

When we look at the figures for the domain name amplification, we see that there are some organizations within the Dutch government that seem to have more expertise either in-house or externally to properly configure their DNS. Both for the TXT and ANY records, we see lower amplification potential than for the nameservers. This means that there are most likely a few very bad apples in the nameserver list that aren't used by all domains. And the fact that every domain uses multiple nameservers causes the number to be averaged, while especially for ANY requests the response size for domain x or y from a nameserver will be the same. For the ANY requests, we can't really give much of a recommendation or make observations about certain domains, as this is most likely not the fault of the sysadmin at these organizations. What is however something within their control is to limit the TXT amplification potential, as we see that there are quite a few domains that score consistently higher than individual nameservers.

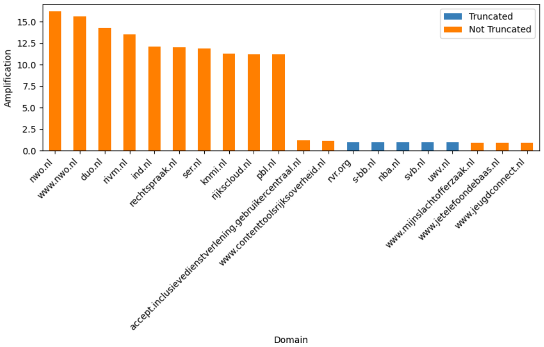

Figure 7. The top and bottom 10 amplifying servers for TXT requests, attributed by truncated and non-truncated parts

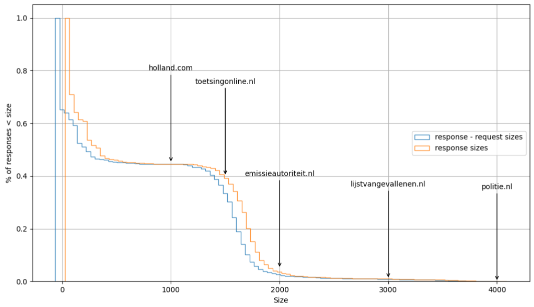

Figure 8. CDF of response sizes for different domains

Figure 8 shows a reverse cumulative density function for the response sizes of different domains. The blue line before the 0 on the x-axis indicates a negative amplification factor, since the request size is greater than the response size. The top 3350 domains have an amplification factor of 2 or more, the top 45% of domains have a response size of greater than 1000 bytes, the top 53 domains have response sizes greater than 3000, with the largest response size being 4091 bytes. The figure also shows a sample of domain names around certain response sizes to show some of the worst offenders.

5.2 Attack simulation

Three different files were used in the simulation, differing in what nameservers and queries are used in the attack:

1. Original: the original file with all the Dutch government's domains.
2. Big: the 50 entries with the greatest response sizes.
3. Small: the 50 entries with the lowest response sizes greater than 100.

During the attack simulation, 1756166, 1792169, and 1741543 packets were sent over the span of 10 seconds for each of the files respectively, as can be seen in table 2. The average response size over all the domains used by the Dutch government is 839 bytes as can be seen in table 3, this resulted in an attack of 147 MB/s (11x amplification). The attack using the file with the largest responses results in an attack of 644 MB/s on average, while the attack with the smallest response sizes results in only 19.3 MB/s. This shows that the choice in which domains and nameservers are used by the attacker can make a very significant difference in the size of the attack. There is even the possibility for an adversary to create a "DNS reduction" attack if the wrong servers are chosen for the attack, as the response can be smaller than the original request in some cases.

File	Packets	Attacker	Amplified	Amp †
Original	1756166	13 MB/s	147 MB/s	11x
Big	1792169	13 MB/s	644 MB/s	49x
Small	1741543	12 MB/s	19.3 MB/s	1.6x

† Amplification factor.

Table 2. Attack simulation sizes per file used

File	Request size	Response size
Original	78 B	839 B
Big	76 B	3594 B
Small	69 B	111 B

Table 3. Average request and response sizes for each file

6 Discussion

Given our findings, the current DNS configuration of the Dutch government's nameservers allow for quite large DNS amplification attacks. The amplification potential can be significantly reduced by introducing some relatively simple countermeasures, which are described in the recommendation section below.

6.1 Recommendations

The first and most important recommendation is to implement RFC8482 and block DNS ANY requests (at least over UDP) as that reduces the maximum amplification from around 70x to approximately 12x if implemented across all the servers that were evaluated. This would require updates to the current configuration of the DNS servers that currently allow ANY requests, especially those that currently produce the greatest response sizes. As was discussed in the results, some providers for government services already have some protections in place against ANY requests, as such, transferring services to those cloud providers with better protection can reduce the amplification potential. This holds especially true if the configuration regarding ANY requests for current DNS servers is difficult to change or cannot be altered at all.

Secondly, even without ANY requests, TXT records alone still allow for a significant amount of amplification, it is therefore important to reduce the number and size of TXT records as much as possible. The degree to which this can be applied heavily depends on the context however, as TXT records have various legitimate use cases.

Lastly, some form of rate limiting could be applied, such that the same IP cannot request as many packets per second. The difficulty with this approach is that there are many nameservers, and if the attacker cycles through all these nameservers just like in our proof of concept attack, the rate limiting must be quite strict in order to significantly reduce the size of such an attack. However, such strict rate limiting could impact the performance for non-malicious users and thus be undesirable.

6.2 Limitations

The main limitation during this research has been the ability to simulate a full attack, we had limited access to devices and thus could not accurately measure the true amount of incoming data caused by an attack. We believe the measurements of the results on the victim machine during our attack simulation show incorrect measurements about the incoming number of packets and bandwidth used. Some of the samples taken using this approach show bandwidth usage of around 4 MB/s to the victim machine, yet the victim machine is completely unable to receive any other traffic. However, when simply downloading a file using around 12 MB/s, the victim machine can operate just fine. These findings indicate that the measurements should be taken by a machine other than the victim machine in order to obtain a more accurate picture of the attack, perhaps a better way to measure this can be explored and tested in further research.

7 Conclusion

This paper delved into the DNS amplification potential of domain names owned and managed by the Dutch government.

In order to research this, a few different steps were taken. First, a list of domain names owned by the Dutch government was obtained. From this list, the authoritative nameservers were discovered, which were then queried with ANY and TXT records. The resulting request and reply sizes were saved and analyzed.

Additionally, based on these results, a simulation of a DNS amplification attack was created. The goal of this simulation was to test the viability of such an attack. Using the 50 domains with the largest amplification potential led to a total amplification factor of just under 50.

The results obtained from querying the authoritative DNS servers show us that the Dutch DNS infrastructure can indeed be used to perform DNS amplification attacks, with domains even reaching amplification factors of more than 70x. This large amplification potential combined with the fact that there is no observable rate limiting in place makes the Dutch government's DNS servers even more interesting for executing such attacks.

The amplification potential of the Dutch DNS ecosystem could be greatly reduced by introducing some relatively simple countermeasures, such as blocking ANY requests or implementing rate limiting. These countermeasures are becoming more and more common in big cloud providers such as Cloudflare and Azure.

However, until then, the Dutch digital landscape remains an attractive player for a DNS amplification attack.

References

[1] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. 2010. Comparing DNS resolvers in the wild. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 15–21.

[2] Catalin Cimpanu. 2018. GitHub Survived the Biggest DDoS Attack Ever Recorded. *BleepingComputer* 22 (2018).

[3] Catalin Cimpanu. 2021. *Microsoft: Here's How We Stopped the Biggest Ever DDoS Attack.* https://www.zdnet.com/article/microsoft-heres-how-we-stopped-the-biggest-ever-ddos-attack/

[4] Cloudflare. 2023. *DNS Amplification DDoS Attack.* https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/

[5] Cloudflare. 2023. *What is DNS?* https://www.cloudflare.com/learning/dns/what-is-dns/

[6] Joao Damas, Michael Graff, and Paul Vixie. 2013. *Extension mechanisms for DNS (EDNS (0)).* Technical Report.

[7] Go Developers. 2023. *The Go Programming Language.* https://go.dev/

[8] GoPacket Developers. 2023. *GoPacket.* https://pkg.go.dev/github.com/google/gopacket

[9] Harm Griffioen, Kris Oosthoek, Paul van der Knaap, and Christian Doerr. 2021. Scan, test, execute: Adversarial tactics in amplification ddos attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 940–954.

[10] Wouter Jehee, Ignjat Pejic, Gregor Schram, and Maarten Weyns. 2023. *GitHub repository containing the code and data used for this paper.* TU Delft. https://github.com/gregor160300/IN4253ET-HackingLab/

[11] Marek Majkowski. 2015. *Deprecating the DNS ANY Meta Query Type.* Cloudflare Blog. https://blog.cloudflare.com/deprecating-dns-any-meta-query-type/

[12] Marek Majkowski. 2019. *RFC8482 - Saying goodbye to ANY.* Cloudflare Blog. https://blog.cloudflare.com/rfc8482-saying-goodbye-to-any/

[13] Elinor Mills. 2013. Spamhaus under DDoS attack. *CNET* 27 (2013).

[14] Nexusguard. 2021. Q1 2021 Threat Report. https://www.nexusguard.com/threat-report-q1-2021 Accessed: April 14, 2023.

[15] Ray K. L. Tso and Cheng Wang. 2017. An Overview of DNS Amplification Attack Defense via Flow-Based Analysis and SDN. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1873–1894. https://doi.org/10.1109/COMST.2017.2683541

[16] Olivier van der Toorn, Johannes Krupp, Mattijs Jonker, Roland van Rijswijk-Deij, Christian Rossow, and Anna Sperotto. 2021. ANYway: measuring the amplification DDoS potential of domains. In *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 500–508.

[17] Roland van Rijswijk-Deij, Anna Sperotto, and Aiko Pras. 2014. DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. 449–460.