How resilient is the Netherlands government DNS?

DINO Project Advisory Report (English Version)

dr. Giovane C. M. Moura[1,2]
Raffaele Sommese, MSc[3]
dr.ir. Mattijs Jonker[3]
1: SIDN Labs 2: TU Delft 3: Universiteit Twente

How resilient is the Netherlands government DNS?

DINO Project Advisory Report (English Version)

by

dr. Giovane C. M. Moura[1,2]
Raffaele Sommese, MSc[3]
dr.ir. Mattijs Jonker[3]

1: SIDN Labs 2: TU Delft 3: Universiteit Twente

Friday 11th March, 2022

Contents

List of Figures

List of Tables

Management Summary

Target Audience: ***[RF: A4]*** *This reported is written for senior policymakers at the Netherlands government and decisionmakers for vital infrastructure. As such, we assume that some readers do not have a strong technical background. We will therefore try to provide the relevant background information when needed.*

The Netherlands is one of the **worldwide leaders in e-government**, having more than 80% of its adult population interacting with the government online[1]. Citizens and companies can make use of various vital online services, which improves government efficiency and reduces bureaucracy and costs.

To keep these online services available at all times, it is of paramount importance that the government deploys **dependable and robust online services**, that are designed to withstand a myriad disruptive online threats. Imagine, for example, what would happen if `DigiD` were to become unavailable, even for a single day. This would stop citizens from accessing online services, disrupt numerous digital processes, and could propagate to physical infrastructures (customs-related traffic congestion).

In this report, we focus on *one part* of the Internet infrastructure associated with the e-government services: the **Domain Name System (DNS)**, which is a *vital* component of the Internet. The DNS can be seen as the "phonebook" of the Internet: suppose a citizen wants to file her tax reports on `https://belastingdienst.nl`. The DNS is responsible for *mapping* the domain name (belastingdienst.nl, which humans understand) to addresses that computers understand (IP addresses). The DNS infrastructure that is involved in reaching `https://belastingdienst.nl` needs to dependable and robust.

Failures of the DNS can have **severe consequences**, and can ultimately compromise the ability of the government provide online services. There are notorious cases in which DNS failure led to online services becoming inaccessible to users. In this context, the authors have been requested by the NCSC to perform **an evaluation on the resiliency of the DNS** associated with the Netherlands' e-government domain names, and determine if they follow current **best-practices** for DNS resiliency (robustness).

The objectives are to **evaluate** the *configuration* of DNS infrastructure used by the e-government domains, and to **identify** where such configurations can be improved to increase resiliency to disruptive threats. While we acknowledge that we are not policy experts, we present two **recommendations** on how the DNS infrastructure of e-government domains can be improved, and discuss the complexity and estimated efforts involved in following these recommendations.

Main findings:

We find that many most government domain names follow current DNS configuration best-practices, which is good news. However, we also observe that a significant number of domains –

[1] `https://joinup.ec.europa.eu/sites/default/files/inline-files/Digital_Government_Factsheets_Netherlands_2019_0.pdf`

and at various levels of government – fail to adhere to best practices. This creates risks, which we would argue are in some cases relatively straightforward to avoid.

Specifically, we identify two *critical* single points of failure, present for roughly 50% of all government domain names. (A single point of failure is a part of a system that when it fails, the entire system fails):

- First, half of the Government domains **[RF: C1]** each use only a single DNS provider, creating unnecessary risk. If these DNS single DNS providers are attacked, all e-government services that depend on it will become unavailable. (This is not hypothetical: there have been notorious cases in which deliberate attacks on exclusively depended on infrastructure has led to services being unavailable).
- Secondly, a significant percentage of domains rely on a chain of servers that operate under a single top-level domain (TLD), such as `.nl` or `.com`. As a consequence, failure in parts of DNS infrastructure could create a cascading effect, equally causing these services to also become unreachable.

To address these single points of failure, we propose in Chapter 5 two recommendations:

1. Diversify DNS providers.

   This can be done in two complementary ways:

   - by **[RF: C2]** encouraging and helping de Overheid at various levels (gemeenten, provincie, *etc.*) to add extra DNS providers to their services
   - by creating GovNL_DNS, a state-of-the-art, secondary DNS server that can be used only by Government domains as an *additional* server to provide extra redundancy and that is independently operated from the primary DNS infrastructure

2. Monitor DNS infrastructure for issues:

   - Outages can occur and human errors can be made at any time, so it is important to keep track of the health of DNS servers beyond a (our) one-time assessment
   - **[RF: C3]** Frequently assessing can provide more situational awareness and prompt responsible operators to act as issues occur, so they can be mitigated with the least effects on service

The adoptions of these measures will improve the resiliency of the DNS services of the Netherlands Government.

1

Introduction

The Domain Name System (DNS) [4] provides a core service on the Internet: every Website visit and e-mail interaction requires interaction with the DNS. The DNS is analogous to the contact list on your phone, which provides translations from names to phone numbers, where phone numbers are used to setup a communication channel. On the Internet, the DNS maps names such as `https://overheid.nl` to addresses that computers understand and use to communicate.

When the DNS fails, it can have severe consequences for services that rely on the DNS. Precedent exists in which human error, network outages, or deliberate cyber attacks have left users unable to reach services that rely on the DNS. As an example, consider the notorious 2016 attack on the large DNS provider Dyn, cutting many users off service to Twitter, Netflix, and Spotify [11]. **[RF: B3]** For roughly half a day, large portions of the US' East Coast users could not access any of these services – all because all these websites relied upon a single DNS provider, who was attacked by a large DDoS attack, renderning all these websites unreachable.

Closer to home, back in August of 2015, millions of Internet users in the Netherlands found themselves largely unable to access the Internet because of an attack involving the Ziggo DNS infrastructure on which they relied.

In the context of Governments, having a dependable, resilient DNS is paramount for the correct functioning of digital government. If the DNS of government services becomes unavailable or severely under stress, citizens will not be able to access vital online services. As a thought experiment, imagine what the unavailability of DigiD on even a single day would mean.

The DNS is complex system, designing a robust DNS infrastructure requires both technical skills and financial resources. Luckily, researchers and operators have worked on defining *best practices* for DNS operations to make the DNS more resilient. Following these practices will help *reduce* risks of unavailability, making it less likely for dependent services to fail. (We summarize these best practices in §A.3).

In this advisory report we assess the resiliency of DNS infrastructure relevant to Netherlands government domain names. This means names used for Websites (for example, `https://overheid.nl`) as well as for e-mail delivery. Our analysis covers roughly 2000 governmental domain names. These names include those provided by the Forum Standardisatie as well as DigiD entry points. We identify two critical points of failure affecting roughly 50% of the domain names. We present these findings in Chapter 3. Afterwards, in Chapter 4, we present two further important (yet not critical) points for improvement. In Chapter 5 we make recommendations on

how to improve DNS resiliency. We conclude and discuss avenues for future investigation in Chapter 6.

2

Netherlands Government Domains

The NCSC has provided us with a list of domain names associated with the Government of The Netherlands. In total, we have received a list comprising 1309 web domains and 536 e-mail domains – provided by DigiD and Forum Standaardisatie. These domains are our *input* in this research: we set out to evalute their DNS configurations.

Table 2.1 shows the distribution of these domains, according to their source (either DigiD domains or Standaardisatie) and to which government level these domains are associated (Gemeenten, Provincies, *etc.*).

Figure 2.1 shows the data from Table 2.1 in a histogram format. We see that the far majority of the government domains fall into the Gemeenten category.

To evaluate if these domains comply with current DNS best practices (§A.3), we carried out a Internet measurement study. In this report, we only cover the *actionable* results and refer the reader to the technical report for further details.

	Web	**Mail**
Domains	1460	536
Source		
DigiD	902	0
Forum Standaardisatie	558	535
Unique domains	1309	536
Gov. Level		
Gemeenten	1044	366
Uitvoerders	99	49
Rijk	84	64
Waterschappen	46	31
Provincies	30	19
Overig	6	6

Table 2.1: Datasets analyzed: 2021-10-01

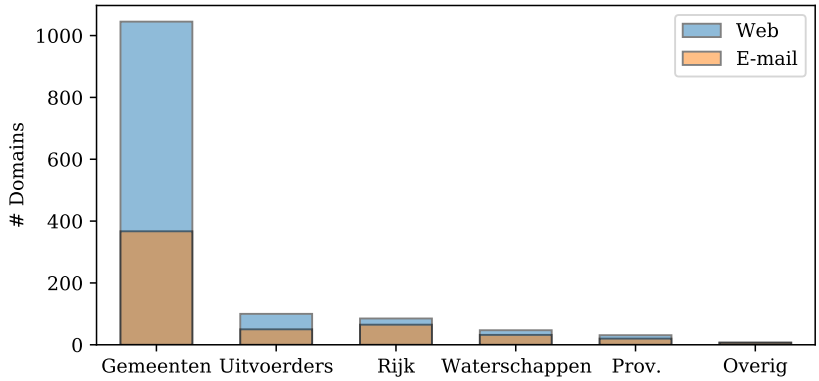


Figure 2.1: Evaluated Dutch Government Domains

2.1. [RF: A3] Scope and Limitations

Scope: This study encompasses only "public" Internet domain names belonging to different levels of government. As such, the *private* diginetwerk of the Dutch government is not included in this study.

Limitations: our measurement techniques have also their intrinsic limitations – which is not exclusively to them, but to the Internet itself. For example, we cannot know if two servers are located on the same datacenter, or if they use the same physical fiber cable to connect to the Internet. We can know, however, if they are located in the same IP network. We describe these limitations in more details in §3.3 in §A.3)

3

Critical Problems: Single Point of Failure

A good design principle for any fault-tolerant system is to avoid *single points of failure*, which are parts in a system that if they fail, the entire system will stop working. For example, consider the Maeslantkering: to prevent that power failures stop it from working, it has three independent power sources. Even if two fail, one is enough to guarantee that the barrier can shut the river, preventing storm surges at the sea from flooding the port of Rotterdam and surronding areas.

We evaluate the domans from the Netherlands Government (Table 2.1) against 10 single points of failure metrics (the metrics are described in §3.1 in §A.3). We found that most domains are protected against 8 of these points of failure. Next, we present the two points of failure we found in the Government domains: single DNS provider (§3.1) and single TLD dependency (§3.2). **[RF: A2]** We classify the respective levels of criticalness as **very high** and **low**, which we will motivate later.

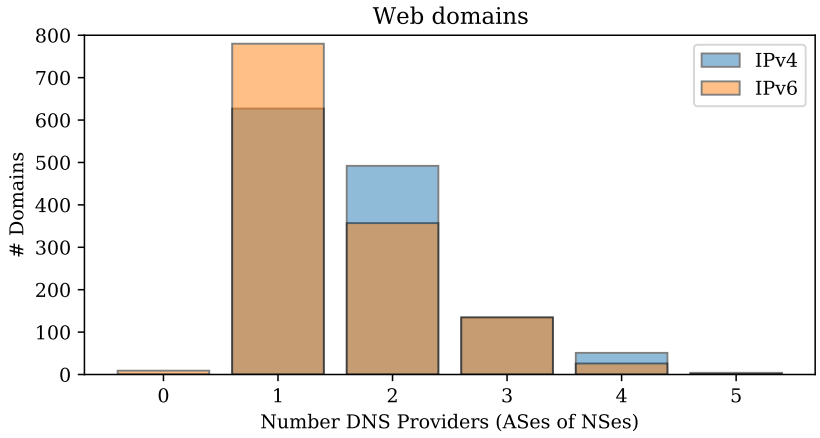
3.1. Single DNS provider

An example of single point of failure is to use a single DNS provider – analogous to putting all your eggs in one basket. **[RF: C4]** As an example, consider the previously mentioned attack on the large DNS provider Dyn in 2016 [11] (see Chapter 1), which saw interruption for services that relied exclusively on Dyn for DNS.

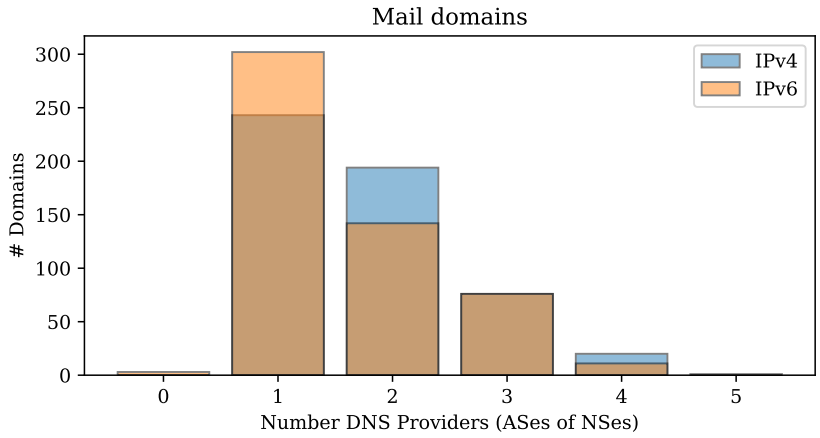
Figure 3.1 shows the number of DNS providers for both Web and E-mail datasets. We see that most domain names – for both Web and e-mail domains – have a single DNS provider, which amounts to a single point of failure. If this provider is targeted with a large distributed denial-of-service (DDoS) attack or otherwise suffers from network outage, all these domains may become unreachable, as was the case of the Dyn 2016 attack. **[RF: A2]** Because of precedent and practicability of such an attack, we classify the level of criticalness of having a single DNS server as **very high**.

Table 3.1 shows the Top 3 DNS providers for each dataset. For the Web domains, we see TransIP dominate the DNS market: 281 domain names depend *fully* on their services. For E-mail, we see that Microsoft dominates the DNS infrastructure: this is because many of these Websites use Microsoft Outlook services, which has its DNS managed by Microsoft.

We see in Table 3.2 that, **[RF: C5]** in terms of Web domain names with a single DNS provider, the results range between 30.9% – 51% for various levels of government. More concretely, 533



(a) Web



(b) E-mail

Figure 3.1: Number of DNS providers

	Web	
	IPv4	**IPv6**
Transip-AS20857	281	281
TWS-AS48365	86	86
Quality-AS12315	73	73
Rest	182	179
	E-mail	
	IPv4	**IPv6**
Microsoft-AS8075	154	0
Transip-AS20857	36	39
Amazon-AS16509	19	19
Rest	108	119

Table 3.1: Top 3 providers for single DNS provider domains

	Web		Mail	
	IPv4	IPv6	IPv4	IPv6
gemeenten	533 (51.0%)	664(63.3%)	185 (50.5%)	234 (63.9%)
uitvoerders	37(37.3%)	50(50.5%)	17(34.7%)	21(42.8%)
rijk	26 (30.9%)	25(29.7%)	20 (31.2%)	19 (29.7%)
provincies	14(46.6%)	15(50.0%)	10(52.2%)	11(57.8%)
waterschappen	14(30.4%)	23 (50.0%)	8 (25.8%)	14 (45.6%)
overig	3 (50.0%)	3 (50.0%)	3 (50.0%)	3(50.0%)

Table 3.2: Single DNS provider domains and categories **[RF: C5]**. Percentages are computed with regards total domains per category.

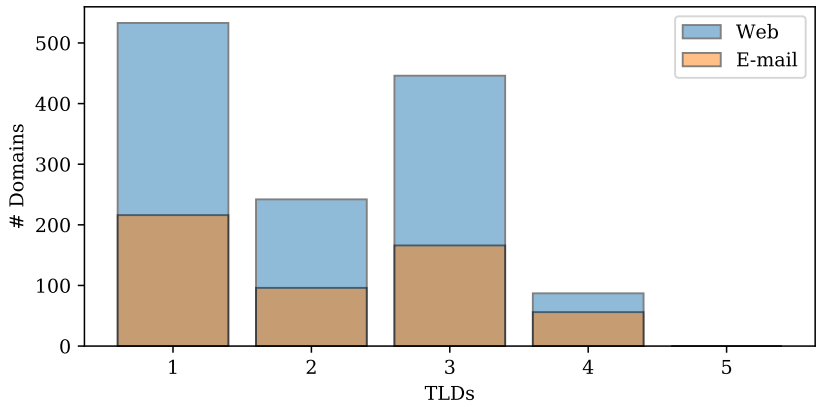


Figure 3.2: Number of TLDs used by domains

out of 1044 gemeente Web domains (51%) have a single DNS provider. Rijk sees the lowest percentage of 30.9%, but that still comes down to 26 out of 84 domain names at risk.

Our first recommendation is therefore to *mitigate* these single points of failure (wherever possible). **[RF: A2]** This mitigation should not be very difficult. A significant part of domains *already* have secondary servers deployed. In terms of effort, it fixing the problem could involve contracting a (second) commercial provider. **[RF: C6]** In §5.1 we suggest additional steps that could be taken to both help facilitate this process and add resiliency that does not depend on commercial providers per se.

3.2. Single TLD dependency

Another single point of failure is the dependence on a single top-level domain (TLD), such as `.com` or `.nl` for *all* authoritative DNS servers. The TLD servers are part of the chain of servers required to reach a domain name. With a single TLD dependency, if the TLD servers are taken down, such problems could cascade. **[RF: A2]** Note that TLDs typically use multiple servers and taking them all down less practicable, but not impossible. For this reason we classify the level of criticalness as **low** here.

Figure 3.2 shows the number of TLDs used by the authoritative servers of both Web and E-mail domains. We see that many use one, even though many others use 2 or 3. In terms of doing things well, consider `digid.nl`. In §A.2 we show how this domain name uses 4 TLDs for its authoritative servers (`.com`, `.nl`, `.eu`, `.org`).

Our second recommendation is to not rely on a single TLD in the chain of servers for each given domain. However, the choice of which TLDs must take in consideration the operators of

the TLD as well – for example, `.net` and `.com` are ran by Verisign on the same infrastructure – so they behave like a single TLD with regards to resilience.

By doing that, these domains will not long depend on a single TLD, adding redundancy in case of (unlikely) TLD failures.

4

Important problems

Next we cover other important problems we have found – they are not as critical as single point of failure from Chapter 3 – but they also contribute to the resiliency of the DNS.

4.1. DNS Configuration Errors

There are many types of DNS configuration errors that can occur, often caused by human mistakes. One of the errors is setting wrong IP addresses for DNS servers of domains. For example, suppose in the `digid.nl` case (Table A.3), the server 178.22.85.27 would be unresponsive, while still being listed in the DNS configuration. If a server is unresponsive, it reduces the resiliency of the domain, given that clients cannot rely on the server. This case is equivalent to having a DNS server only in "the paper", but not deployed.

For web domains, we found 17 domains with at least one IPv4 unresponsive authoritative server, and 23 domains with at least one unresponsive IPv6 server. For e-mail domains, the numbers were 3 and 4, respectively. These could readily be fixed –either updating the DNS configuration or verifying the status of these servers. **[RF: A2]** The amount of effort should be relatively low. We classify the level of criticalness as **medium** to **high**.

There are many other types of DNS errors or misconfigurations that can occur – errors that we discuss in §4 in §A.3). And many of these errors can go on for years without being noticed – we have carried a peer-reviewed academic study in [13] where we show one of these errors.

To mitigate such configuration errors, we propose in §5.2 a recommendation on how to detect these errors, via frequent monitoring. By doing that, operators are able to quickly detect (transient) errors and can notify the responsible operators to fix it.

These monitoring is an standard practice in many DNS operators, and it could help the Government to early detect errors.

4.2. Low anycast adoption

Another measure to increase resiliency is to use IP anycast [3, 10] on authoritative DNS servers of Government domain names. IP anycast is widely deployed in large DNS servers worldwide – all Root DNS servers deploy, many TLDs as well. Anycast increases the resiliency by using multiple servers distributed across the globe, instead of a single physical location. This makes taking such servers down less practicable. Anycast is readily available on major DNS providers

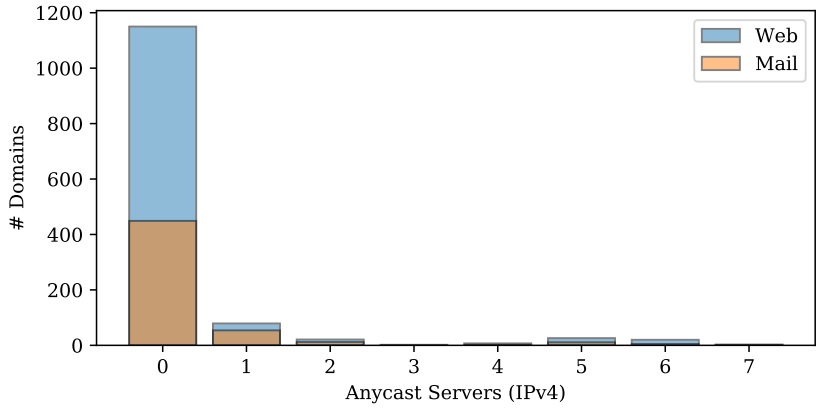


Figure 4.1: Number of domains with anycast servers.

as well. **[RF: C7]** It is important to note that anycast deployments require specific operational knowledge and Internet resources.

Figure 4.1 shows the number of domains with anycast DNS servers. We see that the majority of domains have zero anycast servers. While many domains may not need the high availability provided by anycast, **[RF: C7]** having it does add to resilience [9]. **[RF: A2]** We classify the level of criticalness as **low**. The 'easiest' way to adopt anycast is by choosing a (commercial) DNS provider that supports it. In §5.1 we suggests steps that could be taken to facilitate anycast adoption without relying on commercial parties.

5

Recommendations to improve DNS resiliency

While we acknowledge that we are not policy experts, we provide two suggestions for concrete steps that will help in increasing the resiliency of the Netherlands Government DNS.

5.1. Diversify DNS providers

The most important recommendation that we offer is to *diversify* the number of DNS providers for each domain. This applies in particular to single DNS provider domain names (§3.1). We have seen in Table 3.2 that the percentages of single provider names range from 30.9% (Rijk) to 51% (Gemeenten). We recommend adding *at least* a second provider to each domain name, thus eliminating the single point of failure.

We envision two ways that this diversification can take place, which we expand next:

5.1.1. Notify and support domain operators in adding DNS providers

Our investigation has revealed that many of the domain names with single providers are from Gemeenten (§3.1). We assume that there is large variation in the technical skills within the Gemeenten, and that not all of them have the technical expertise or know-how to correctly choose and configure an extra DNS provider to their DNS services.

A good choice would be a DNS provider that *already* supports IP anycast (§4.2) and that uses TLDs different from the ones already relied on by the primary provider (§3.2).

This is probably a task in which the Rijksoverheid could assist – especially in the case of small gemeenten. They could first notify and then support (technically) these operators in adding extra DNS providers.

[RF: A1,B2] Required effort: given that the Rijksoverheid is not directly in control of most domains with a single provider, this tasks would (probably) require Government analysts to contact every *single* government entity (Gemeente, Provincie, *etc.*) and informing them of the issue, and helping them in adding an extra DNS provider.

This contact can be done via e-mail, letter, or phone calls and video conferences –the last two being time consuming. Once the right operator is notified, he or she will need to make the right adjustments, which likely will take days and possibly weeks. In this process, the job of the

analysts is to (i) first explain the issue, and (ii) help the local operators to improve their DNS settings with extra DNS provider(s).

To confirm that the problems have been solved, the analysts would also (iii) have to *run the measurements* again to determine who has fixed, and who has not. There would be then follow-up calls to help those lagging behind, which is also time consuming.

This process can be last longer or shorter depending on the already established relationship between Rijksoverheid and the involved entities, but it will probably take months to be complete, requiring one to multiple government analysts.

We would like to emphasize that this task **must be executed very carefully**: making configuration errors during this process can actually deteriorate resiliency or create unavailability.

5.1.2. Develop a NLGov Secondary Authoritative DNS server

In addition to encouraging domain name operators to add a secondary DNS provider of choice, another possibility for the Rijksoverheid could be to design and run a robust and state-of-the-art authoritative DNS server [9]. This server could be used as a *secondary* authoritative DNS server by all government domains. This idea is not new – it is currently deployed by RIPE NCC, which runs a secondary authoritative **[RF: C8]** DNS server for many countries **[RF: B4]**top-level domains (TLDs)[1], such as Uruguay, Nepal, and Philippines. This service, however, is not available to government domains.

By deploying NLGov Secondary DNS, each government domain can keep their authoritative servers of choice and add robustness by adding one centrally operated by the Government. For example, suppose the domain `delft.nl` relies on one DNS provider. **[RF: C9]** It could add the NLGov server as a secondary DNS server to avoid a single point of failure. **[RF: raffa-isp-comment]** Besides, by running its own secondary DNS, the Rijksoverheid can *choose* which Internet Service Providers (ISPs) it will deploy individual anycast servers. For example, it may wish to deploy on every single ISP in the Netherlands, or on key Internet Exchanges (IXPs) such as AMS-IX. This would allow service to be available to client in these networks when attacks occur outside these networks.

Another advantage of running a NLGov authoritative server is that you reduce the risk of **collateral damage** from depending on large commercial DNS providers. For example, when Dyn was attacked, Netflix and Spotify, which shared this infrastructure, experienced problems. A dedicated, secondary authoritative server from the government would not bring such collateral damage risk, unless of course this infrastructure becomes itself the target of the attack. **[RF: C7]** Having such infrastructure might also create an opportunity to deploy anycast. Recall from §4.2 that this requires specific operational knowledge and Internet resources.

[RF: B4] An alternative to running the GovNL Secondary Authoritative Server directly would be to *outsource it* to a commercial DNS provider – which deploys IP anycast as well. There are several available, and some of then in the European Union (which may come in handy in case of privacy/legal aspects). Outsourcing the service provider allows for faster deployment and relief the burden of daily operations. However, it *may* increase the risks of collateral damage, given that DNS providers *share* infrastructure, *i.e.*, the same servers may be used for multiple zones – and even if one gets attacked, all of them suffer altogether (similar to what happen to Dyn in 2016). It would probably behoof policy makers at the government to weigh all these options when deciding what is the best option. **[RF: raffa-isp-comment]** Also, by outsourcing the service to a commercial party, the Rijsoverheid will have no control over *where* servers are deployed, so that would make more difficult to deploy anycast servers on each Dutch ISP, if that is intended.

[1] `https://www.ripe.net/publications/docs/ripe-663`

[RF: A1,B2] Required effort: just like §5.1.1, deployment would require contacting each operator, which may take some time.

Then, the rest of the effort would involve setting up the procedures for each client entity (say a Gemeente) to inform the government of their DNS configurations and whenever they change them – so the DNS servers of the government can also be updated in time. While the task is relatively simple, coordinating each among so many parties may become a problem. We recommend analysts to think carefully on how to best design this process.

If the Rijksoverheid decides to run its own, dedicated anycast secondary server, that will take far more effort in engineering, designing, deploying and maintaining. It will require multiple persons to make it operational, and several people to keep it running. It will provide the advantage of having full control over where is deployed – for example, on each single ISP in the Netherlands. The drawback, however, is having to run to be responsible for the service operations. The number of required servers (or virtual machines) would depend on how many ISPs and locations the Government whish to cover. The example from `digid.nl` (Table A.3) shows this knowledge is already *available* within the Rijksoverheid. This task could be **sped up** by leveraging this in-house knowledge.

5.2. Monitoring the DNS

In the same way that the Rijkswaterstaat continuously monitors the state of the dikes in the Netherlands, we posit that continuously monitoring of the DNS is *crucial* to keep it resilient. Networks can become under attack or suffer outages at any time, servers can fail, and configuration errors can be introduced. For this reason, it is important to continually monitor infrastructure. Doing so often enough can test responsiveness, configuration and resilience properties, and provide notifications to act in case of problems. This could be executed in automated fashion, multiple times a day.

[RF: C10] There are various commercial network intelligence providers that offer parties the means to monitor their own infrastructure, for example for availability and performance, from a global users' perspectives. These services can be costly and are not necessarily NL or EU-based. We posit that monitoring the Netherlands DNS infrastructure can be done with a tailored solution, built largely on top of widely available and open source tools, but requiring plumbing work and an operator. We imagine this is something that should be done centrally and could possibly be integrated with development and operations teams that are already monitoring other vital infrastructure.

[RF: A1,B2] Required effort: we can assume that such tasks would require several analysts that can, in part-time, monitor the network and analyze the results. This is far easier than deploying and running a dedicated secondary DNS server, but also requires precision and correctness.

To simplify things, the monitoring could be done centrally, say by the Rijksoverheid, which would *continually* monitor every single DNS server responsible for government domains. Whenever problems occur, analysts could notify the responsible operators and involved entities.

There are several open-source tools that could possibly help in automating these monitoring tasks, and they can also be configured to send alerts.

6

Conclusions and Future Work

The main conclusion that accompanies our report is that the Netherlands government domain names fulfill most of the critical metrics for DNS resiliency. However, there are still two critical metrics that leave roughly 50% of the domains susceptible to problems, in case an attack or outage occurs. Such a situation may never occur, but prevention is better than curing, especially when prevention can be relatively straightforward.

Our assessment data suggests a high level of freedom within the government divisions to choose which DNS providers to choose – which led to concentration in local players. Taking into account the importance of online government to the citizens of the Netherlands, we believe that the government should support their local and provincial governments in using a more robust DNS, and frequently monitoring for errors. In this way, errors or events are less likely to negatively affect citizens.

As future work, we propose to repeat this study and compare the results to test if there have been significant changes in the resiliency of the DNS of government domain names.

Authors

dr. Giovane C. M. Moura is a Data Scientist with SIDN Labs, the research arm of SIDN, the Netherland's .nl top-level domain operator. His research focus on bringing academic rigor to network operations, to improve performance, security and stability of networked systems. I am also a research guest at TU Delft's CyberSecurity group. He obtained his Ph.D. in 2013 from the University of Twente.

dr.ir. Mattijs Jonker is assistant professor and research scientist at University of Twente. His research is on network security in a broad sense and involves extensive data science and Internet measurement. He is one of two architects of the award-winning OpenINTEL project, which measures sizable parts of the domain name system for security research. Mattijs earned his Ph.D. at the University of Twente in 2019, cum laude, and also holds a MSc specialization in Cyber Security.

Raffaele Sommese, MSc is a PhD candidate at the University of Twente. His research focuses on analyzing and characterizing DNS vulnerabilities and misconfiguration in order to improve protection against and the prevention of DNS DDoS attacks. Raffaele received his Master's degree in Computer Engineering from Politecnico di Torino in 2018.

A

Appendices

A.1. Tables of DNS providers

Table A.1 and Table A.2 shows the number of DNS providers for web and mail domains, respectively. They are the numbers behind Figure 3.1.

A.2. Reference examples

Before we dive into the results, we select two domains from the dataset that are on the opposite extremes of DNS resiliency: `digid.nl` and `slachtofferportaal.nl`. Table A.3 shows the DNS configurations for the first, and Table A.4 for the latter.

We then set out to compute the *critical* best metrics we have defined in our best-practices document. We show the results for the critical best practices both domains in Table A.5, while Table A.5 shows the recommended best practices scores. Cells in red denote where a domain does not meet the best practices requirements.

Major issues critical metrics: the major issue with `slachtofferportaal.nl` is that it has a single AS (29311), a single IPv4 and a single IPv6 prefix. As such, if anything were to happen to these route announcements ro this particular AS, `slachtofferportaal.nl` would become *unreachable*. A solution would be to add a second AS, which can add an extra layer of redundancy. `digid.nl`, for example, is announced by three different ASes.

Major issues recommended metrics: we see in Table A.6 that `slachtofferportaal.nl` only has one TLD in its NS records. If something were to happen to the TLD, this domain would become unreachable.

#Providers	IPv4	IPv6
0	0 (0.0%)	9 (0.6%)
1	627(47.9%)	780 (59.6%)
2	492(37.6%)	357 (27.3%)
3	134(10.2%)	135(10.3%)
4	51(3.9%)	26(2.0%)
5	4(0.3%)	1(0.1%)
Total	1308	1308

Table A.1: DNS Providers Distribution for Web domains

.

#Providers	IPv4	IPv6
0	0(0.0%)	3(0.56%)
1	243(45.34%)	302(56.34%)
2	194(36.19%)	142(26.49%)
3	76(14.18%)	76(14.18%)
4	20(3.73%)	11(2.05%)
5	1(0.19%)	0(0.0%)
Total	534	534

Table A.2: DNS Providers Distribution for Mail domains

NS	IPv4	Prefix	IPv6	Prefix	AS IPv4	AS IPv6
ns0.rijksoverheidnl.com	185.136.96.82	185.136.96.0/24	2a06:fb00:1:0:0:0:1:82	2a06:fb00:1::/48	203391	203391
ns1.rijksoverheidnl.nl.	178.22.85.27	178.22.84.0/22	2a00:d00:3:6:0:0:0:130	2a00:d00::/32	41887	41887
ns2.rijksoverheidnl.eu	94.228.142.136	94.228.142.0/23	2a00:d01:3:1:0:0:0:20	2a00:d01::/32	41887	41887
ns3.rijksoverheidnl.org	145.100.177.67	145.100.0.0/15	2001:610:188:203:3:1:0:67	2001:610::/29	1103	1103

Table A.3: `digid.nl` DNS configurations

NS	IPv4	Prefix	IPv6	Prefix	AS IPv4	AS IPv6
ns1.minvenj.nl	159.46.194.11	159.46.192.0/22	2a04:9a04:18ad:8a04:0:0:2:0	2a04:9a04::/32	29311	29311
ns2.minvenj.nl	159.46.194.12	159.46.192.0/22	2a04:9a04:18ad:8a04:0:0:3:0	2a04:9a04::/32	29311	29311

Table A.4: `slachtofferportaal.nl` DNS configurations

Metric	**Description/Reference**	**Reference**	`digid.nl`	`slachtofferportaal.nl`
nNSes	Number of NS records for a zone/[5]	>=2	4	2
nIP(NSv4)	Number of Unique IP addresses for NSes (IPv4) [5]	>=2	4	2
nIP(NSv6)	Number of Unique IP addresses for NSes (IPv6) [5]	>=2	4	2
ResponsiveNSesV4	All authoritative servers are responsive for the domain/[12]	`True`	True	True
ResponsiveNSesV6	All authoritative servers are responsive for the domain/[12]	`True`	True	True
nPrefixes(NSv4)	Number of unique BGP prefixes for NSes (IPv4/[2])	>=2	4	1
nPrefixes(NSv6)	Number of unique BGP prefixes for NSes (IPv6)/[2])	>=2	3	1
nAses(NSv4)	Number of unique ASes for NSes (IPv4) [1]	>=2	3	1
nASes(NSv6)	Number of unique ASes for NSes (IPv6) [1]	>=2	3	1
nGeoDiverseNSes	Number of NS distinct geographical locations [2]	>=2		

Table A.5: Critical metrics scores

Metric	**Description/Ref.**	**Value**	`digid.nl`	`slachtofferportaal.nl`
nTLDs	Use more than one TLD for NS records/[1]	2	3	1
NS TTL	TTL values of NS records/[7–9]	>=3600s	3600	3600
A(NS) TTL	TTL values for A (NS) records[7–9]	>=1800s	14400	3600
AAAA(NS) TTL	TTL values for AAAA (NS) records[7–9]	>=1800s	14400	3600
nAnycastIPv4	Number of Anycast Auth Servers IPv4/[6]	>=1		
nAnycastIPv6	Number of Anycast Auth Servers IPv6/[6]	>=1		

Table A.6: Recommended Best Practices Metrics

A.3. Best Practices for DNS operators

In the next pages we describe the best practices for the DNS operators.

Best Practices for Resilience of Authoritative DNS Servers DINO project

Giovane C. M. Moura (1) Mattijs Jonker (2) Raffaele Sommese (2)
1: SIDN Labs 2: University of Twente

November 23, 2021

Abstract

This document fulfills Task 1 (T1) from the plan van aanpak (PvA). We identify and describe best practices that, if implemented by DNS operators, bring about resilience for authoritative nameservers. These best practices will be used as a starting point in a later task, in which we investigate the extent to which these best practices are currently adhered to by operators of DNS infrastructure associated with governmental services and therefore vitally important to the Netherlands society.

1 Introduction

The Internet Domain Name System (DNS) [1] is one of the core services on the Internet. It maps servers, resources, and services to IP addresses. Every web page visit requires a series of DNS queries, and large-scale DNS failures can have global, cascading effects. DNS-related incidents can make the front pages of prominent news outlets, as in the case of denial-of-service (DDoS) attack agains Dyn DNS in 2016. In this particular incident, the Mirai botnet [2] was used to overload the Authoritative servers of Dyn, compromising the reachability of various proeminent websites, such as Netlifx, Spotify, Reddit and the New York times [3].

2 Background

We provide brief background information here on the DNS and its components to help put some of the best practices that we later identify in context.

2.1 Types of DNS servers

The DNS is a distributed and hierarchical system. It can be seen as a distributed database, in which the management and operation of parts of that database can be delegated for technical and administrative scalability. In general terms, the DNS involves two types of servers, as we show in Figure 1. Authoritative DNS servers, which are the focus of this work and in green in the figure, are

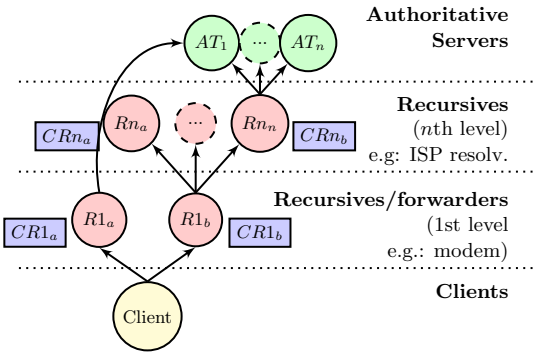


Figure 1: Relationship between clients (yellow), recursive resolvers (red) with their caches (blue), and authoritative servers (green).

servers that are – as the name suggests – authoritative for a part of the global DNS hierarchy. These servers know the contents they are responsible for from memory [4]. As an example, `ns1.dns.nl` is one of the authoritative servers for the `.nl` zone. This server knows where to find other authoritative servers that are responsible for smaller parts of the `.nl` zone, i.e., domain names further *down* the DNS hierarchy.

DNS resolvers, in turn (salmon color in Figure 1), are servers that, on behalf and users and applications, perform the task of looking up information in the DNS. As an example task, consider resolving a domain name to an IP address. Because of the hierarchical approach, such resolvers *recursively* query the DNS. That is, they potentially reach out to authoritatives in various layers of the DNS hierarchy.

As a concrete example, if a user (shown as stub in the figure) wants to visit `wikipedia.org` in their browser, she first needs to use one of her DNS resolvers to retrieve the IP address of this domain name. The resolver, in turn, will attempt to resolve the domain and ultimately obtain a response from the authoritative DNS server for `wikipedia.org` (`ns[1--3].wikimedia.org`), which will then send the requested IP address back to the user.

2.2 Authoritative DNS servers setup and redundancy

Any DNS zone (such as `example.org`) must be configured with authoritative DNS servers, which are the servers that can respond DNS queries from resolvers. These authoritative servers are defined in so-called NS records [1] in the DNS.

Replication of a DNS service is important to support high reliability and capacity and to reduce latency. The DNS has two complementary mechanisms to replicate service. First, the protocol itself supports *nameserver replication* of DNS service for a zone, by supporting multiple NS records for a given zone. Figure 2 shows the setup of the Root DNS zone (`.`), which has 13 authoritative DNS servers (`[a--m].root-servers.net.`). Each of these NS records have their own IPv4 and IPv6 addresses, defined as A and AAAA resource records.

Second, each of these authoritative servers can run in multiple physical locations while using *IP anycast* [5, 6]. This is different from the aforementioned replication through multiple NS records, because in the anycast case the same IP

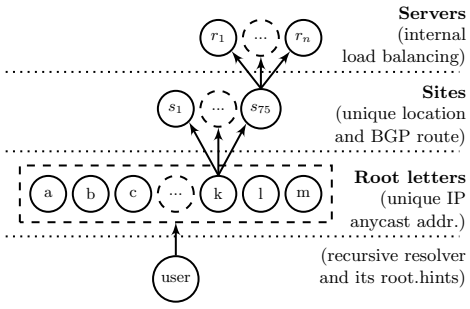


Figure 2: Root DNS structure, terminology, and mechanisms in use at each level.

address is shared between physical locations, while the Internet routing (BGP) is leveraged to direct clients to the nearest anycast site. Note that a combination between both mechanisms – multiple nameservers and multiple physical locations for each nameserver – is also possible.

Nameserver replication is recommended for all zones, and IP anycast is used by most large zones such as the DNS Root and most top-level domains [7, 8]. IP anycast is also widely used by *public resolvers*, which are DNS resolvers that are open for use by anyone on the Internet. As examples, consider Google Public DNS [9], OpenDNS [10], Quad9 [11], and `1.1.1.1` [12]. In the root zone (Figure 2), we show that `K-ROOT`, one of the root authoritative servers ran by RIPE NCC, has 75 anycast sites (S_n). BGP [13] then *maps* the IPv4 and IPv6 clients to individual sites and, in this way, a DDoS attacks can have limited effect by overwhelming *some* of the sites while leaving others active [8].

Finally, the last level of replication is per anycast site, in which each *site* can have multiple servers behind a load balancers (r_n) in Figure 2. (Unicast servers can also have load balancers, but they have a single site).

3 Best Practices

In the section we present best-practices on how to configure DNS authoritative servers. It summarizes the conclusions from these research efforts and offers specific, tangible advice to operators when configuring authoritative DNS servers.

We divide the best practices into three categories: *critical* and *recommended*, and *immeasurable*. Criticial (§3.1) refers to practices that are a *must* to overcome *single points-of-failure* (SPoF) – analogous to "don't put all your eggs in the same basket". Single points-of-failure cause total unreachability of domain names when they fail.

The second category of best practices are *recommended* (§3.2), which means that they *help* to improve the resilience of DNS, but not following them does not lead to single points-of-failure.

The last category are best practices that we consider out-of-scope of this study (§3.2). There are practices that cannot be measured using traditional Internet Measurements (layer 3 and above), such as physical and link layer practices. We however list them given their importance, although we cannot access them in this study.

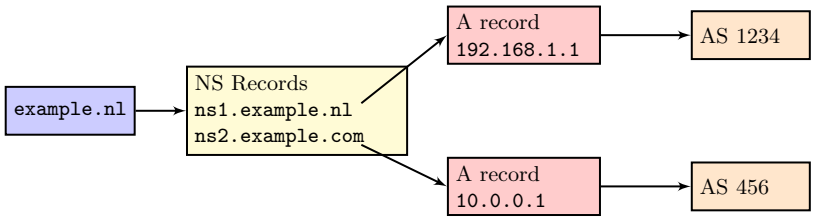


Figure 3: Example to illustate best practices for an example domain name (`example.nl`)

Metric	**Description/Reference**	**Value**
nNSes	Number of NS records for a zone/[15]	>=2
nIP(NSv4)	Number of Unique IP addresses for NSes (IPv4) [15]	>=2
nIP(NSv6)	Number of Unique IP addresses for NSes (IPv6) [15]	>=2
ResponsiveNSesV4	All authoritative servers are responsive for the domain/[16]	`True`
ResponsiveNSesV6	All authoritative servers are responsive for the domain/[16]	`True`
nPrefixes(NSv4)	Number of unique BGP prefixes for NSes (IPv4/[17])	>=2
nPrefixes(NSv6)	Number of unique BGP prefixes for NSes (IPv6)/[17])	>=2
nAses(NSv4)	Number of unique ASes for NSes (IPv4) [18]	>=2
nASes(NSv6)	Number of unique ASes for NSes (IPv6) [18]	>=2
nGeoDiverseNSes	Number of NS distinct geographical locations [17]	>=2

Table 1: Critical Best Practices Metrics

We note that following these practices may imply more financial costs. For example, hosting authoritative DNS servers on multiple Autonomous Systems may cost more than hosting on a single AS. We, however, do not take *costs* into consideration, but will mention where they could be significantly higher.

These best practices concern *availablity* of a DNS zone and not its integrity. In this sense, we focus on metrics and properties that could improve the dependability and availablity of authoritative servers. We do not, however, focus on best practices not related to availablity, such as use of DNSSEC [14] that guarantees DNS messages authenticity and integrity and best pratices to reduce latency between clients and authoritative servers (performance).

3.1 Critical Best Practices

Table 1 summarizes the critical best practices for authoritative DNS servers operators. We define each practice as individual metrics, which will use in the second phase of this study – to measure them for the websites related to the Government of the Netherlands.

Next we expand of each individual metric and practice. For that, we use the

example show in Figure 3, for a sample DNS zone: `example.nl`.

3.1.1 nNSes: number of NS records for a zone

Description: each domain name is required to have at least two authoriative DNS servers [15], *i.e.*,, two distinct NS records, in order to guarantee *some* level of redundancy, as having a single NS would be a single point of failure. In our example from Figure 3, this is shown by having two NS records: `ns1.example.nl` and `ns2.example.com`. Each NS record, in turn, may be ran by a different organization and using IP anycast, which provides extra redundancy (§3.2.5).

Reference: this best practice has been proposed on the original DNS standard [15], so we do not expect to find many domains names that do not follow it.

How to measure it: A `dig` command line tool equivalent of: `dig ns $domain_name`

3.1.2 nIP(NSv4): number of unique IP addresses for all NSes

Description: This metric consists in determing how many unique IPv4 addresses *host* the authoritative DNS servers. In our Figure 3, that would be the number of unique IPv4 addresses assocaited with both NS records (`ns1.example.nl` and `ns2.example.com`).

Notice that a single domain may have multiple NS records (fullfiling in this way, the §3.1.1). However, all of these NS records may have the *same* A records (for example, all pointing to 192.168.1.1, which would still create a single point of failure. Thus, the metric from §3.1.1, if analyzed alone, could provide a false sense of security.

Reference: This best practice is document on RFC2182 [17].

How to measure it: For each NS record, retrieve its A record(s) that must be publicly *routable*, *i.e.*,, valid and reachable IP address space. Then, count the number of unique records for all.

3.1.3 nIP(NSv6): number of unique IP addresses for all NSes

Description: Same as in §3.1.2, except it measures AAAA records (IPv6) instead of A records (IPv4).

3.1.4 ResponsiveNSesV4 :All authoritative servers are responsive for the domain

Description: In our example domain in Figure 3, a registrant (who owns the domain) sets two NS records for its domain (`ns1.example.nl` and `ns2.example.com`). However, these servers may not be active, may be not be authoritative for the zone in question (referred to as lame delegation [16]), and ultimately may not been able to provide authoritative information for the domain.

For example, if a user would ask data about Japan's DNS zone `.jp` to a `.nl` authoritative server (*e.g.,*: `dig ns example.jp @ns1.dns.nl`), the `.nl` would *refuse* to answer the question, indicating the NL server is not authoritative for `.jp`.

Reference: Lame delegations are defined in RFC1713 [16] and evaluated in [19].

How to measure it: This involve a series of steps.

1. Get the IP addresses of all NS records
2. For each address, send a SOA query or A or NS query about the domain name in question. If the response is OK (RCODE=0 [1]), then the server is properly configured. If not, then the server has an issue.

3.1.5 ResponsiveNSesV6 :All authoritative servers are responsive for the domain

Same as §3.1.4, except for IPv6 addresses.

3.1.6 nPrefixes(NSv4) Number of unique BGP prefixes for NSes (IPv4)

Description: IP addresses are announced on the Internet in blocks called "BGP prefixes" [20]. These prefix announcements contain information that help routers determine where address space can be reached. For example, suppose a telecom company announces a IP block address 192.168.0.0/24 (which covers 256 /32 addresses). This annoucement is received by neighboring routers, which propagate it even further.

For resilence, it is better to have, for a given DNS zone, *distinct* prefixes for all IP addresses of the NS records. This provides some isolation in case one of the route announcements experiences issues.

In our example in Figure 3, we see two addresses that *likely* belong to two different prefixes.

Reference: This falls in the category of having dissimilar infrastructure of authoritative servers. This is defined in [17].

How to measure it: For that, we have to analyze BGP prefix announcements in public sources of BGP data, such as RIPE RIS [21] and RouteViews [22]. To measure it, we must:

1. Get the IP addresses of all NS records
2. Determine which prefix announcements cover these addresses
3. Count the number of unique prefixes

3.1.7 nPrefixes(NSv6) Number of unique BGP prefixes for NSes (IPv4)

Same as §3.1.6, except for IPv6.

3.1.8 nAses(NSv4): Number of unique ASes for NSes (IPv4)

Description: As discussed in §3.1.6, IP addresses are announced in BGP using prefixes. This annoucement also contains what *Autonomous Systems*(ASes) are in the path to the prefix, and its *origin* AS. Ultimately, it's the origin AS that *hosts* the IP addresses in questions.

To improve resilience, it is recommended to have the IP addresses of your authoriative server in more than one AS, to avoid single points of failure if something goes wrong with a particular AS.

Metric	Description/Ref.	Value
nTLDs	Use more than one TLD for NS records/[18]	2
NS TTL	TTL values of NS records/[25, 26, 27]	>=3600s
A(NS) TTL	TTL values for A (NS) records[25, 26, 27]	>=1800s
AAAA(NS) TTL	TTL values for AAAA (NS) records[25, 26, 27]	>=1800s
nAnycastIPv4	Number of Anycast Auth Servers IPv4/[8]	>=1
nAnycastIPv6	Number of Anycast Auth Servers IPv6/[8]	>=1

Table 2: Recommended Best Practices Metrics

Reference: This falls in category of having dissimilar infrastructure of authoritative servers. This is defined in [17].

How to measure it: For that, we have to analyze BGP route announcements from public sources, such as RIPE RIS [21] and RouteViews [22].

1. Get the IP addresses of all NS records
2. Determine which announced prefixes cover the addresses
3. Determine what origin AS announces the BGP prefix
4. Count the number of unique origin ASes

3.1.9 nAses(NSv6): Number of unique ASes for NSes (IPv4)

Same as §3.1.8, except for IPv6 addresses.

3.1.10 nGeoDiverseNSes: Number of NS distinct geographical locations

Description: Authoritative nameservers shoud be placed in different geographical location in order to avoid that a physical disaster in a location (e.g. fire) can affect all the servers.

To improve resilience, it is recommended to put the nameservers in different cities.

Reference: RFC2182 [17] states that secondary servers should be at geographically distant locations.

How to measure it: For that, we have to analyze IP geolocation databases such as Maxmind [23] or run active measurements to identify locations using RIPE Atlas [24].

3.2 Recommended Best Practices

Table 2 summarized the *recommended* best practices. Recommended refers to practices that *improve* the dependability of the DNS, but not following them does not lead to a single point-of-failure, which, in turn would imply total unreachability.

Next we expand these recommended best practices.

3.2.1 nTLDs: number of unique TLDs used in the NS records

Description: this metric refers to the number of top-level domains (TLDs) used in the NS records. In the example of Figure 3, we see that the two NS records user different TLDs: `.com` and `.nl`. That means is one of these two TLDs would become unreachable, the `example.nl` zone could still be reachable via the `.nl` TLD.

Similarly, the criticial domain `digid.nl` has 4 NS records, from four different TLDs: `.nl`, `.eu`, `.org`, and `.com`.

Note that if resolvers do not already have NS records for `example.nl`, then the `.nl` authoritative servers must be reachable.

Another solution is to provide glue records for all the NS records, in that case the domain will use in-bailiwick records that will require only the `.nl` TLD to be reachable.

Reference: This practice has been long been known by the community, and is also documented by [18].

How to measure it: extract all NS records for a given domain, and count the distinct number of TLDs.

Caveat: note that many TLDs share the same DNS infrastructure, so one has to choose carefully which TLDs to host. For example, `.com` and `.net` use the same infrastructure, and are run by a single company (Verisign).

3.2.2 NS TTL value

Description: DNS record, such as the NS records in Figure 3, always have a time-to-live field (TTL), which tells DNS resolvers the maximum time the DNS responses should be kept in the DNS cache of the servers. DNS caches, as CR_n in Figure 1, are the cornerstone of DNS performance [25, 26, 27]: having a cached response drastically reduces the response time to clients. Moreover, in case of DDoS attacks, having *longer* TTLs (say minimum an hour) would allow clients behind resolvers with hot caches to *still be able to reach* the destination website, even though the DNS authoritative servers may be completely unreachable. Caching can therefore be seen as a *ephemeral* resilience.

Given these considerations, the proper choice for a TTL depends in part on multiple external factors – no single recommendation is appropriate for all scenarios. Organizations must weigh these trade- offs and find a good balance for their situation. Still, some guidelines can be reached when choosing TTLs:

- For general DNS zone owners, [27] recommends a longer TTL of at least one hour, and ideally 8, 12, or 24 hours. Assuming planned maintenance can be scheduled at least a day in advance, long TTLs have little cost and may, even, literally provide a cost savings.

- Users of DNS-based load balancing or DDoS-prevention services may require shorter TTLs: TTLs may even need to be as short as 5 minutes, although 15 minutes may provide sufficient agility for many operators. There is always a tussle between shorter TTLs providing more agility against all the benefits listed above for using longer TTLs.

Reference: We have previously investigated the role of caching in DDoS attacks in DNS in several studies [25, 26, 27].

How to measure it: To measure the TTL value of a record, one must obtain an authoritative answer by asking *direclty* the authoritative servers, and bypass local resolvers which may have a hot cache and decremented TTL values.

Caveat: There is some level of duplication in DNS: NS records can be found in both *parent* and *child* DNS zones. For example, the NS records for `example.nl` in Figure 3 can be found at the `.nl` authoritative servers (which are the "parent"), as well as in the "child" authoritative servers (`ns1.example.nl` and `ns2.example.com`). These values, however, may differ [28], given that these zones are typically managed by different organizations. However, most resolvers in the wild tend to follow the *child* authoritative server TTL [29]. For this reason, we will consider only the child TTL value.

3.2.3 A(NS) TTL

Description: in §3.2.2, we analyze the TTL of NS records for a given domain. These NS records, in turn, need to have A and/or AAAA addresses to be reachable – these are the IP addresses that are used to route packets. In Figure 3, that refers to the TTL value of the A records (192.168.1.1).

The TTLs for A/AAAA records should be shorter to or equal to the TTL for the corresponding NS records for in-bailiwick authoritative DNS servers, since [27] finds that once an NS record expires, their associated A/AAAA will also be re-queried when glue is required to be sent by the parents. For out-of-bailiwick servers, A, AAAA and NS records are usually all cached independently, so different TTLs can be used effectively if desired. In either case, short A and AAAA records may still be desired if DDoS-mitigation services are required.

Reference: We have previously investigated the role of caching in DDoS attacks in DNS in several studies [25, 26, 27].

How to measure it: To measure the TTL value of a record, one must obtain an authoritative answer by asking *directly* the authoritative server and bypassing local resolvers, which may have a hot cache and decremented TTL values.

Caveat NS records can be in or out of zone (in or out of bailiwick in DNS terminology). For example, the IP address of `ns1.example.nl` must be placed as a glue record in the parent DNS zone (`.nl`) for `example.nl`, given they share the same second-level domain (`example.nl`). This is different from `ns2.example.com`, which uses another TLD. In this case, the IP address (A record) is only available at the child authoritative server. (Most of zones, however, are out-of-bailick [28]). So we measure them accordingly to their setup.

3.2.4 AAAA(NS) TTL

Same as §3.2.3, except for AAAA (IPv6) records.

3.2.5 nAnycastIPv4: number of anycast-based authoritative server

Description IP anycast consists of annoucing the same IP prefixes from multiple locations [5]. Anycast is largely used in DNS [30], especially by operators of prominent authoritative servers. For example, all the root DNS servers use IP anycast.

IP anycast *fragments* the IP address space, and maps each fragment into a different anycast site. For example, in Figure 2, we see that `K-ROOT` has 75

Metric	Description/Reference	Value
nPhysicalLocations	number of unique physical locations hosting the authoritative name servers	>= 2
nPhysicalLines	number of distinct physical lines connecting authoriative name servers	>= 2
nPhysicalServers	number of unique baremetal servers	>= 2
KeyServersOnClients	Place anycast sites or authoritative servers on key clients	NA

Table 3: Immeasurable Best Practices Metrics

anycast sites: the entire IPv4 is distributed among the 73 sites – which is done by BGP [8], where clients are mapped to nearby sites (nearby in BGP terms, and not necessarily geographical distance [31]). This distribution is not necessarily uniform, some sites may see far more clients than others.

In case of DDoS attacks against an authoritative server, we see that some sites experience the attack diferently [8]: some sites may remain up while others remain down. That behavior has been observed in the Root DNS Events on November 2015 [8]. As such, operators can, on-the-fly, configure their authoritative anycast DNS to try to steer DDoS traffic to one of few sites, while others may remain up.

Our goal is to determine which the A/AAAA addresses of the authoritative servers use anycast.

Reference: IP anycast is documented in [5]. Its DNS usage in [30]. Its relation to DDoS in [8]. And how to measure anycast in the wild is documented in [32, 33].

How to measure it: We will use the procedure described in [32] using the Anycast Testbed from SIDN. In short: we will use active measurements from an anycast network to measure the IP addresses from the government networks.

3.2.6 nAnycastIPv6: number of anycast-based authoritative server

Same as §3.2.5, except for AAAA (IPv6) records.

3.3 Immeasurable best practices

Our methodology can only account for metrics that can be measured on the IP layer (layer 3) and above. As such, any single-point-of-failure mitigation metric that is located *below* layer 3 is, in most cases, *immeasurable*. Although they are essential for the resilience of authoriative DNS servers, we consider them as out-of-scope in this study, given we cannot measure them.

Table 3 summarizes them. Next we detail each of them..

3.3.1 nPhysicalLocations

Authoritative servers – either virtual or bare metal, should be placed in *distinct* physical locations, to avoid that any local related failures (attacks, power outages, etc.) affects the authoriative DNS servers altogether.

Consider the worst-case scenario, in which three authoritative servers are hosted in different IP address space, using different upstream providers, but all

being physically hosted on the same, single datacenter: no matter how much redundancy is added, this setup still has a single point-of-failture, which is a single location.

As such, we recommend operators to use multiple physical locations to host their services.

3.3.2 nPhysicalLines

Similar to the number of physical servers, there must be multiple lines that connect authoriative servers to the Internet – not for each of them, but for all of the combined. The goal is to avoid a single point-of-failure.

3.3.3 nPhysicalServers

The last metric the number of physical servers hosting the authoriative DNS servers . One could run multiple authoritative DNS servers on a single bare metal server, ultimately removing redundancy. The goal of this metric is to avoid this.

3.3.4 KeyServersOnClients

For specific services, such as `digid.nl`, it may be worth to add *anycast sites* of authoritative servers on key client networks – for example, the networks of major ISPs and where most clients come from.

Depending on the type of attack, this setup may provide DNS services to clients while other parts of the network may be under attack. For example, suppose a particular DDoS attacks the networks on a IXP. Clients can still be able to resolve the domain if they have access to servers on their ISP's network. This pratice only improve resilience in cases the client's network are not able to reach the networks of the authoritative servers.

In addition to that, we intend to write a *speculative* scenario, in which The Netherlands is "disconnected" by some reason (DDoS attack, for example), from the global Internet. In this scenario, we will estimate how much of the domain names related to the government will still be able to be resolved.

4 Next steps

The metrics discussed here will be implemented in our tooling to measure resilience of the DNS of the Netherlands government. We will first draft a measurement plan, and then share it with our colleagues at the government.

References

[1] P. Mockapetris, "Domain names - concepts and facilities," IETF, RFC 1034, Nov. 1987. [Online]. Available: http://tools.ietf.org/rfc/rfc1034.txt

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet,"

in *Proceedings of the 26th USENIX Security Symposium*. Vancouver, BC, Canada: USENIX, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

[3] N. Perlroth, "Hackers used new weapons to disrupt major websites across U.S." *New York Times*, p. A1, Oct. 22 2016. [Online]. Available: http://www.nytimes.com/2016/10/22/business/internet-problems-attack.html

[4] P. Hoffman, A. Sullivan, and K. Fujiwara, "DNS Terminology," IETF, RFC 8499, Nov. 2018. [Online]. Available: http://tools.ietf.org/rfc/rfc8499.txt

[5] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," IETF, RFC 1546, Nov. 1993. [Online]. Available: http://tools.ietf.org/rfc/rfc1546.txt

[6] J. Abley and K. Lindqvist, "Operation of Anycast Services," IETF, RFC 4786, Dec. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4786.txt

[7] Root Server Operators, "Root DNS," May 2020, http://root-servers.org/.

[8] G. C. M. Moura, R. de O. Schmidt, J. Heidemann, W. B. de Vries, M. Müller, L. Wei, and C. Hesselman, "Anycast vs. DDoS: Evaluating the November 2015 root DNS event," in *Proceedings of the ACM Internet Measurement Conference*. Santa Monica, California, USA: ACM, Nov. 2016, pp. 255–270. [Online]. Available: https://www.isi.edu/%7ejohnh/PAPERS/Moura16b.html

[9] Google, "Public DNS," https://developers.google.com/speed/public-dns/, Nov. 2020. [Online]. Available: https://developers.google.com/speed/public-dns/

[10] OpenDNS, "Setup Guide: OpenDNS," https://www.opendns.com/, Mar. 2021. [Online]. Available: https://www.opendns.com/

[11] Quad9, "Quad9 — Internet Security & Privacy In a Few Easy Steps," https://quad9.net, Jan. 2018.

[12] 1.1.1.1, "The Internet's Fastest, Privacy-First DNS Resolver," https://1.1.1.1/, Apr. 2018. [Online]. Available: https://1.1.1.1/

[13] J. Scudder, R. Fernando, and S. Stuart, "BGP Monitoring Protocol (BMP)," IETF, RFC 7854, Jun. 2016. [Online]. Available: http://tools.ietf.org/rfc/rfc7854.txt

[14] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Protocol Modifications for the DNS Security Extensions," IETF, RFC 4035, Mar. 2005. [Online]. Available: http://tools.ietf.org/rfc/rfc4035.txt

[15] P. Mockapetris, "Domain names - implementation and specification," IETF, RFC 1035, Nov. 1987. [Online]. Available: http://tools.ietf.org/rfc/rfc1035.txt

[16] A. Romao, "Tools for DNS debugging," IETF, RFC 1713, Nov. 1994. [Online]. Available: http://tools.ietf.org/rfc/rfc1713.txt

[17] R. Elz, R. Bush, S. Bradner, and M. Patton, "Selection and Operation of Secondary DNS Servers," IETF, RFC 2182, Jul. 1997. [Online]. Available: http://tools.ietf.org/rfc/rfc2182.txt

[18] M. Allman, "Comments on DNS Robustness," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 84–90. [Online]. Available: https://doi.org/10.1145/3278532.3278541

[19] G. Akiwate, M. Jonker, R. Sommese, I. Foster, G. M. Voelker, S. Savage, and K. Claffy, "Unresolved issues: Prevalence, persistence, and perils of lame delegations," in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 281–294.

[20] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF, RFC 4271, Jan. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4271.txt

[21] Reseaux IP Europeens Network Coordination Centre (RIPE NCC), "Routing Information Service (RIS)," 2021. [Online]. Available: http://www.ripe.net/data-tools/stats/ris

[22] University of Oregon , "Route Views Project," 2021. [Online]. Available: http://www.routeviews.org/

[23] Maxmind, "Maxmind," 2012. [Online]. Available: http://www.maxmind.com/

[24] RIPE Network Coordination Centre, "RIPE Atlas," https://atlas.ripe.net, 2020.

[25] G. C. M. Moura, W. Hardaker, J. Heidemann, and M. Davids, "Considerations for Large Authoritative DNS Servers Operators," Internet Engineering Task Force, Internet-Draft draft-moura-dnsop-authoritative-recommendations-09, Aug. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-moura-dnsop-authoritative-recommendations-09

[26] G. C. M. Moura, J. Heidemann, M. Müller, R. de O. Schmidt, and M. Davids, "When the dike breaks: Dissecting DNS defenses during DDoS," in *Proceedings of the ACM Internet Measurement Conference*. Boston, MA, USA: ACM, Oct. 2018, pp. 8–21. [Online]. Available: https://www.isi.edu/%7ejohnh/PAPERS/Moura18b.html

[27] G. C. M. Moura, J. Heidemann, R. de O. Schmidt, and W. Hardaker, "Cache me if you can: Effects of DNS Time-to-Live," in *Proceedings of the ACM Internet Measurement Conference*. Amsterdam, the Netherlands: ACM, Oct. 2019, pp. 101–115. [Online]. Available: https://www.isi.edu/%7ejohnh/PAPERS/Moura19b.html

[28] R. Sommese, G. Moura, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, k. claffy, and A. Sperotto, "When parents and children disagree: Diving into DNS delegation inconsistency," in *Passive and Active Measurement Conference (PAM)*, 2020-03.

[29] G. C. M. Moura, S. Castro, W. Hardaker, M. Wullink, and C. Hesselman, "Clouding up the Internet: How Centralized is DNS Traffic Becoming?" in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 42–49.

[30] D. McPherson, D. Oran, D. Thaler, and E. Osterweil, "Architectural Considerations of IP Anycast," IETF, RFC 7094, Jan. 2014. [Online]. Available: http://tools.ietf.org/rfc/rfc7094.txt

[31] R. d. O. Schmidt, J. Heidemann, and J. H. Kuipers, "Anycast latency: How many sites are enough?" in *Proceedings of the Passive and Active Measurement Conference*. Sydney, Australia: Springer, Mar. 2017, pp. 188–200. [Online]. Available: http://www.isi.edu/%7ejohnh/PAPERS/Schmidt17a.html

[32] R. Sommese, L. Bertholdo, G. Akiwate, M. Jonker, van Rijswijk-Deij, Roland, A. Dainotti, K. Claffy, and A. Sperotto, "MAnycast2—using anycast to measure anycast," in *Proceedings of the ACM Internet Measurement Conference*. Pittsburgh, PA, USA: ACM, Oct. 2020.

[33] R. Sommese, G. Akiwate, M. Jonker, G. C. Moura, M. Davids, R. van Rijswijk-Deij, G. M. Voelker, S. Savage, K. Claffy, and A. Sperotto, "Characterization of Anycast Adoption in the DNS Authoritative Infrastructure," Mar. 2020.

References

[1] Mark Allman. "Comments on DNS Robustness". In: *Proceedings of the Internet Measurement Conference 2018*. IMC '18. Boston, MA, USA: Association for Computing Machinery, 2018, pp. 84–90. ISBN: 9781450356190. URL: https://doi.org/10.1145/3278532.3278541.

[2] R. Elz et al. *Selection and Operation of Secondary DNS Servers*. RFC 2182. IETF, July 1997. URL: http://tools.ietf.org/rfc/rfc2182.txt.

[3] D. McPherson et al. *Architectural Considerations of IP Anycast*. RFC 7094. IETF, Jan. 2014. URL: http://tools.ietf.org/rfc/rfc7094.txt.

[4] P.V. Mockapetris. *Domain names - concepts and facilities*. RFC 1034. IETF, Nov. 1987. URL: http://tools.ietf.org/rfc/rfc1034.txt.

[5] P.V. Mockapetris. *Domain names - implementation and specification*. RFC 1035. IETF, Nov. 1987. URL: http://tools.ietf.org/rfc/rfc1035.txt.

[6] Giovane C. M. Moura et al. "Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event". In: *Proceedings of the ACM Internet Measurement Conference*. Santa Monica, California, USA: ACM, Nov. 2016, pp. 255–270. DOI: http://dx.doi.org/10.1145/2987443.2987446. URL: https://www.isi.edu/%7ejohnh/PAPERS/Moura16b.html.

[7] Giovane C. M. Moura et al. "When the Dike Breaks: Dissecting DNS Defenses During DDoS". In: *Proceedings of the ACM Internet Measurement Conference*. Boston, MA, USA: ACM, Oct. 2018, pp. 8–21. DOI: https://doi.org/10.1145/3278532.3278534. URL: https://www.isi.edu/%7ejohnh/PAPERS/Moura18b.html.

[8] Giovane C. M. Moura et al. "Cache Me If You Can: Effects of DNS Time-to-Live". In: *Proceedings of the ACM Internet Measurement Conference*. Amsterdam, the Netherlands: ACM, Oct. 2019, pp. 101–115. DOI: https://doi.org/10.1145/3355369.3355568. URL: https://www.isi.edu/%7ejohnh/PAPERS/Moura19b.html.

[9] Giovane C. M. Moura et al. *Considerations for Large Authoritative DNS Servers Operators*. Internet-Draft draft-moura-dnsop-authoritative-recommendations-11. Work in Progress. Internet Engineering Task Force, Jan. 2022. 21 pp. URL: https://datatracker.ietf.org/doc/html/draft-moura-dnsop-authoritative-recommendations-11.

[10] C. Partridge, T. Mendez, and W. Milliken. *Host Anycasting Service*. RFC 1546. IETF, Nov. 1993. URL: http://tools.ietf.org/rfc/rfc1546.txt.

[11] Nicole Perlroth. "Hackers Used New Weapons to Disrupt Major Websites Across U.S." In: *New York Times* (Oct. 2016), A1. URL: http://www.nytimes.com/2016/10/22/business/internet-problems-attack.html.

[12] A. Romao. *Tools for DNS debugging*. RFC 1713. IETF, Nov. 1994. URL: http://tools.ietf.org/rfc/rfc1713.txt.

[13] Raffaele Sommese et al. "When Parents and Children Disagree: Diving into DNS Delegation Inconsistency". In: *Passive and Active Measurement*. Cham: Springer International Publishing, 2020, pp. 175–189.