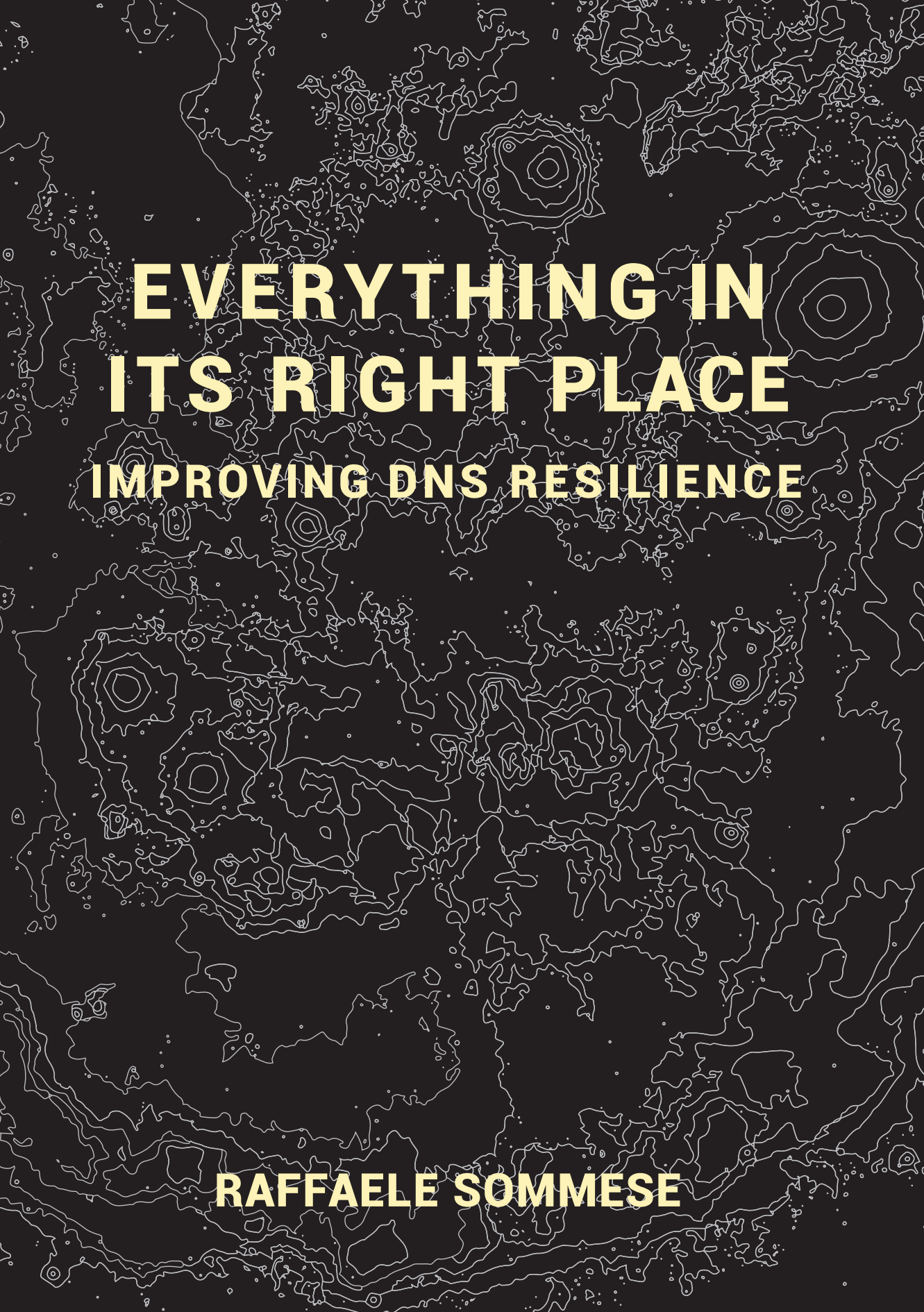

EVERYTHING IN ITS RIGHT PLACE
IMPROVING DNS RESILIENCE
RAFFAELE SOMMESE

Everything in Its Right Place: Improving DNS resilience

Raffaele Sommese

Graduation committee:

Chairman:	Prof. dr. J.N. Kok
Supervisor:	Dr. A. Sperotto
Supervisor:	Prof. dr. ir. R.M. van Rijswijk - Deij
Co-supervisor:	Dr. ir. M. Jonker

Members:

Prof. dr. ir	A.L. Varbanescu	University of Twente, The Netherlands
Prof. dr. ir.	A. Pras	University of Twente, The Netherlands
Prof. dr. ir.	C. Rossow	CISPA - Saarland University, Germany
Prof. dr.	K.C. Claffy	CAIDA, University of California, San Diego, USA
Prof. dr.	G.M. Voelker	University of California, San Diego, USA

Funding sources:
MADDVIPR (NWO/DHS Grant Agreement 628.001.031/FA8750-19-2-0004)

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

DSI Ph.D. Thesis Series No. 23-004
Digital Society Institute
P.O. Box 217
7500 AE Enschede, The Netherlands

ISBN (print)	978-90-365-5668-2
ISBN (digital)	978-90-365-5669-9
ISSN	2589-7721
DOI	10.3990/1.9789036556699
	https://doi.org/10.3990/1.9789036556699

Type set with LaTeX. Printed by Ipskamp.
Cover design by Benedetta Lusi.

Everything in Its Right Place:
Improving DNS resilience

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. ir. A. Veldkamp,
on account of the decision of the Doctorate Board,
to be publicly defended
on Wednesday 14^{th} June 2023 at 16.45 hours

by

Raffaele Sommese

born on the 25^{th} December, 1994
in Ottaviano, Italy

This dissertation has been approved by:

Dr. A. Sperotto (supervisor)
Prof. dr. ir. R. M. van Rijswijk - Deij (supervisor)
Dr. ir. Mattijs Jonker (co-supervisor)

Tuesday March 21^{st}, 2023

Dedicated to:

Vincenza Costantini

Sometimes you don't meet your family by blood,
but by heart

Abstract

In 2023, the Domain Name System (DNS) will celebrate 40 years since its creation. Despite the passing of four decades, the DNS continues to play a fundamental role in today's Internet. Specifically, the DNS provides the essential service of translating human-readable domain names (e.g., `example.org`) to IP addresses (e.g., `93.184.216.34`).

Over the years, the Internet has become increasingly vital to our modern society. The continuous flow of information that takes place on the Internet every day cannot be stopped without catastrophic consequences. In addition, services of crucial importance for people's everyday lives, such as government services, are increasingly transitioning to digital infrastructure. Given the importance of the DNS for the functioning of the Internet and modern society, any issues that the DNS encounters nowadays would have far-reaching consequences. However, over the past 40 years, weaknesses in the DNS system have emerged. One of the most significant cybersecurity threats facing the DNS today are Distributed Denial of Service (DDoS) attacks, which can have a severe impact on the availability of the DNS ecosystem. Recent events show that targeted attacks on even a small portion of the DNS infrastructure can impact millions of services and users.

In this scenario, a comprehensive characterization of the resilience mechanisms of the DNS authoritative infrastructure, along with an analysis of threats against this resilience, is missing. This gap has led us to the main goal and contribution of this thesis. To achieve this goal, we use a mixed measurement and analytical approach, which has focused on different detractors of DNS resilience. Specifically, throughout the course of this thesis, we analyze misconfigurations and vulnerabilities resulting from miscommunication between operators, assess the choices made by these operators in creating more robust and stable deployments in the face of existing best practices, and evaluate the effectiveness of the deployed techniques in overcoming DDoS attacks.

Focusing on our contributions, we show that while the distributed nature of the DNS has enabled its scalability and success, it also presents risks to its resilience. Inconsistency in the DNS hierarchy resulting from miscommunications between stakeholders increases the attack surface and affects resilience, enabling lame delegations and hijacking with potentially severe consequences.

Later, we characterize the large-scale adoption of well-defined *best practices*, as defined by several RFCs, Internet standards, and recent self-regulatory frameworks and legislation. In the wild, we show that DNS is a robust system with good resilience properties, mainly due to choices made by large operators. However, DDoS attacks are still affecting the DNS ecosystem. To overcome them, we show that combining traditional DNS resilience techniques with newer technologies such as IP Anycast is one of the key success strategies.

For this reason, we entitle our thesis "*Everything in Its Right Place: Improving DNS Resilience*". Through this work, readers will understand that our choice of title reflects our aim to demonstrate that only a well-configured and provisioned DNS infrastructure, addressing all possible facets of DNS resilience and operating with *Everything in Its Right Place*, can withstand modern threats and continuously provide the fundamental service for the modern Internet and society.

To conclude this work, we leverage the accumulated knowledge from this thesis as well as insights from previous research efforts to provide a series of actionable best practices for network operators when configuring authoritative nameservers. With this final contribution, our aim is to enhance both the overall understanding of the effectiveness of resilience mechanisms for DNS and the overall health of the DNS ecosystem.

Samenvatting

In 2023 viert het Domain Name System (DNS) zijn 40-jarig bestaan. Ondanks dat DNS al 40 jaar oud is, speelt het nog steeds een fundamentele rol in het moderne internet. Het DNS biedt specifiek de essentiële service om leesbare domeinnamen (bijv. `example.org`) om te zetten naar IP-adressen (bijv. `93.184.216.34`). In de loop der jaren is het internet steeds belangrijker geworden voor onze moderne samenleving. De continue informatiestroom die dagelijks op het internet plaatsvindt, kan niet gestopt worden zonder catastrofale gevolgen. Bovendien maken diensten die van cruciaal belang zijn voor het dagelijks leven van mensen, zoals overheidsdiensten, steeds meer de overstap naar digitale infrastructuur. Gezien hoe belangrijk DNS is voor het goed functioneren van het internet en de moderne samenleving, zouden problemen met het DNS verstrekkende gevolgen hebben. In de afgelopen 40 jaar zijn er echter zwakke punten in het DNS-systeem naar voren gekomen. Een van de grootste cybersecurity-bedreigingen voor het DNS vandaag de dag zijn Distributed Denial of Service (DDoS) aanvallen, die een ernstige impact kunnen hebben op de beschikbaarheid van het DNS-ecosysteem. Recente gebeurtenissen tonen aan dat gerichte aanvallen op zelfs een klein deel van de DNS-infrastructuur invloed kunnen hebben op miljoenen diensten en gebruikers.

In dit scenario ontbreekt een uitgebreide karakterisering van de weerbaarheid van de *authoritative* DNS infrastructuur, samen met een analyse van de dreigingen die deze weerbaarheid in gevaar brengen. Dit gebrek aan kennis leidt ons naar het hoofddoel en de hoofdbijdrage van dit proefschrift. Om dit doel te bereiken, maken we gebruik van een gemengde meet- en analytische aanpak, die zich heeft gericht op verschillende belemmeringen van de DNS-veerkracht. In dit proefschrift analyseren we specifiek verkeerde configuraties en kwetsbaarheden als gevolg van miscommunicatie tussen DNS beheerders. We beoordelen de keuzes die deze beheerders hebben gemaakt bij het creëren van robuustere en stabielere implementaties in het licht van bestaande *best-practices*, en evalueren we de effectiviteit van de ingezette technieken bij het bestrijden van DDoS-aanvallen.

Kijkend naar onze bijdragen laten we zien dat hoewel de gedistribueerde aard van de DNS zijn schaalbaarheid en succes mogelijk heeft gemaakt, het ook risico's met zich meebrengt voor zijn veerkracht. Inconsistentie in de DNS-

hiërarchie als gevolg van miscommunicatie tussen belanghebbenden vergroot het aanvalspotentiaal en beïnvloedt de veerkracht, waardoor bijvoorbeeld domeinkapingen met potentieel ernstige gevolgen mogelijk zijn. Later karakteriseren we de grootschalige adoptie van goed gedefinieerde *best-practices*, zoals gedefinieerd door verschillende RFC's (internetstandaarden) en recente kaders voor zelfregulering en wetgeving. In de praktijk laten we zien dat DNS een robuust, veerkrachtig systeem is, voornamelijk dankzij de keuzes die door een klein aantal grote spelers op de markt is gemaakt. Toch hebben DDoS-aanvallen nog steeds invloed op het DNS-ecosysteem. Om deze aanvallen te af te slaan, laten we zien dat het combineren van traditionele DNS-weerbaarheidstechnieken met nieuwere technologieën zoals IP Anycast een van de succesvolste aanpakken is.

Om deze reden hebben we dit proefschrift de titel *"Everything in Its Right Place: Improving DNS Resilience"* (*"Alles op de juiste plek: Verbetering van de veerkracht van het DNS"*) gegeven. Lezers van dit werk zullen na het lezen begrijpen dat onze keuze van de titel ons doel weerspiegelt om aan te tonen dat alleen een goed geconfigureerde en beheerde DNS-infrastructuur, die alle mogelijke facetten van DNS-weerbaarheid aanpakt, en met *"Alles op de juiste plek"*, moderne bedreigingen kan weerstaan en voortdurend de fundamentele diensten kan bieden voor het moderne internet en de maatschappij.

Om dit werk af te sluiten, maken we gebruik van de opgebouwde kennis uit dit proefschrift, evenals inzichten uit eerdere onderzoeksinspanningen, om een reeks bruikbare *best-practices* te bieden voor beheerders bij het configureren van *authoritative* nameservers. Met deze laatste bijdrage streven we ernaar zowel het algemene begrip van de effectiviteit van de veerkrachtmechanismen voor DNS als de algehele gezondheid van het DNS-ecosysteem te verbeteren.

Sommario

Il 2023 segna il quarantenario dalla creazione del Sistema di Risoluzione Nomi a Dominio, conosciuto come DNS. Nel corso del tempo, il DNS ha sempre svolto un ruolo fondamentale nell'ecosistema di Internet: possiamo considerarlo come un elenco telefonico, in quanto il DNS traduce nomi di dominio facili da ricordare in indirizzi IP (similarmente alle nostre pagine gialle per i numeri di telefono). Negli anni, Internet ha assunto sempre più importanza e i suoi quarant'anni iniziano a pesare sul DNS. Ogni giorno, infatti, assistiamo ad un flusso costante di informazioni che scorrono sulle autostrade digitali e che non possono essere fermate senza conseguenze catastrofiche. Pensiamo, infatti, a tutti i servizi della persona e delle pubbliche amministrazioni che sono legati indissolubilmente al mondo digitale. Vista l'enorme importanza del DNS, è chiaro che una qualsiasi sua debolezza potrebbe avere conseguenze catastrofiche.

In questo scenario, ciò che manca è una caratterizzazione completa dei meccanismi di difesa e di resilienzació dell'infrastruttura globale del DNS, nonché un'analisi delle possibili minacce, come possono per esempio essere gli attacchi Distributed Denial of Service (DDoS). Studi recenti hanno dimostrato come attacchi mirati, anche se in piccola scala, ad una parte dell'infrastruttura DNS possono influenzare milioni di servizi e utenti. Per poter arrivare ad una caratterizzazione completa dei meccanismi di resilienza, nel corso di questa tesi, utilizzeremo un approccio misto che combina analisi e misurazioni. Prenderemo in considerazione i diversi fattori che possono influire sulla resilienza del DNS e analizzeremo, in particolare, le diverse misconfigurazioni e vulnerabilità che possono essere causate da una cattiva comunicazione tra i vari attori del sistema e le loro scelte, al fine di incrementare la stabilità del sistema DNS. Tutto ciò verrà eseguito seguendo le best practices esistenti e valutando l'efficacia delle scelte adottate durante gli attacchi DDoS.

Mostreremo poi come nonostante la natura distribuita del DNS gli abbia permesso di diffondersi, abbia anche introdotto dei rischi per la sua resilienza. Informazioni inconsistenti all'interno della sua gerarchia possono, infatti, facilmente portare ad un aumento dei possibili attacchi e contemporaneamente ad una diminuzione della resilienza globale del sistema. Dopo aver analizzato questo aspetto, andremo ad esplorare l'adozione delle best practices in correlazione con i diversi standard di Internet e dei framework legislativi e vedremo come

il DNS risulti un sistema che al giorno d'oggi ha buone proprietà di resilienza, dovute alle scelte dei grandi provider operanti sul mercato. Nonostante però l'indomita resilienza del sistema, gli attacchi al DNS continuano a rappresentare un problema rilevante per la sua infrastruttura e l'unico modo per cercare di superare il problema è la combinazione di tecnologie tradizionali e nuove tecniche (es. IP Anycast).

Sfogliando questa tesi, i lettori comprenderanno che *Tutto al suo giusto posto* è il motto fondamentale per il DNS, in quanto solo un'infrastruttura ben configurata può sopravvivere alle moderne minacce informatiche e continuare a fornire il suo fondamentale servizio. Come conclusione di questo lavoro di tesi, faremo tesoro di tutta la conoscenza accumulata e forniremo agli operatori una serie di best practices per la configurazione dei sistemi DNS, tenendo conto dell'efficacia delle varie tecniche di resilienza.

Contents

CHAPTER 1

Introduction

Street Sign, by Meadow Marie

Since the beginning of humankind, at the dawn of modern languages, humans have had the need to describe classes, categories, and single objects with names. Establishing names provided the power to modern languages of conceptualizing complex objects or situations with single words [1]. For example, we gave names to cities and streets to be able to recognize and distinguish them.

Similarly, at the dawn of the Internet, the pioneers of this technology found the necessity of creating a system to identify complex objects connected to this global network (e.g. servers, mainframes) with names. This necessity led to the development of the Domain Name System (DNS), a system that identifies devices connected to the Internet using a mnemonical name instead of a complex set of numbers (IP addresses).

The fundamental role played by names in our society makes the effects of their unavailability catastrophic. Imagine if tomorrow someone would spray-paint all the street signs with black ink. Making a street sign unreadable will not make a street disappear, but people will effectively be unable to locate it. Similarly, making the DNS unavailable will not make Internet services disappear, but people will be unable to locate them. From a security standpoint, this means that an attacker only needs to compromise a portion of the DNS to render millions of services inaccessible. For this reason, ensuring the availability of the DNS ecosystem is one of the most relevant challenges of cybersecurity nowadays.

1.1 The Domain Name System

The first Request for Comments (RFC) defining the Domain Name concept (RFC882 [2]) is dated November 1983. Despite its age, we still use DNS every day. We perform DNS queries every time we visit a website, open an app on our smartphone, or turn our thermostat on remotely. Nearly any interaction we perform on today's Internet leverages the DNS for its original role: translating names into IP addresses. Like the Internet Protocol, the DNS has not changed its original core working mechanisms. However, over 40 years after its introduction, the world has changed. While its significance in modern Internet infrastructure remains intact, if the DNS were to encounter issues nowadays, it would have far-reaching consequences for society.

The Internet has progressively become fundamental to the operation of modern society, and the continuous stream of information that flows over it every day cannot be stopped without catastrophic consequences. The COVID pandemic of the last three years increased the relevance of the Internet for our society [3, 4, 5, 6]. Working from home has become the norm for a large part of society to reduce the spread of the virus, and the Internet has become our primary source of daily information. Likewise, 40 years after its inception the DNS still plays a fundamental role in delivering information following the events and news of the world. New domains are continuously being created on any topical subject. See, for example, new domain names containing `ukraine`as a keyword after the start of the Ukraine-Russia conflict (Figure 1.1).

The DNS grew organically with the Internet itself. While at the start the primary use was only the translation of names into IP addresses, many RFCs extended the protocol adding functionalities to support, for example, IPv6 support [8]. Also, increased security played a part in the evolution of DNS. The introduction of DNSSEC provided authentication of the records [9], the SPF [10]

Figure 1.1: Newly registered domains including Ukraine as a keyword - DZDB[7]

and DKIM [11] records reduced the chances of email phishing, and finally, the newest TLSA [12] records opened the way to a synergetic integration between the X.509 certificates and DNS ecosystem.

The DNS was designed as a global hierarchical distributed database with the Internet Assigned Numbers Authority (IANA) as the central authority in charge of managing the root of the infrastructure. IANA delegated the operation of the Top Level Domains (TLDs) to different countries for ccTLDs (e.g. `nl.` to SIDN) and companies for the gTLDs (e.g. `com.` to Verisign). Those entities, usually identified as *registries*, delegated further the operation of selling and managing domain names to third parties called *registrars* (e.g. GoDaddy).

This distributed nature of the DNS proved to be one of its key mechanisms for success, allowing a multitude of companies to operate in the field. DNS grew not only in traffic and features but also in the number of stakeholders responsible for managing the infrastructure. As a consequence, the DNS exploded over the years and became a multi-billion dollar industry [13].

This complex market created a scenario where thousands of stakeholders started to play a fundamental role in this structural pillar of the Internet, leading to further complexity.

But can this complexity have an impact on the resilience? If the DNS itself is under attack, can this complexity amplify the consequences?

1.2 Towards a System Resilient to Attacks

October 21, 2016: A series of Distributed Denial of Service (DDoS) attacks were launched against the DNS provider Dyn. As a consequence, major Internet websites and services, such as Airbnb, Twitter, and others, became unreachable to a myriad of users in Europe and North America for several hours [14, 15]. The Dyn attack showed for the first time to the general public the relevance of the DNS ecosystem and its fragility in the face of DDoS attacks. To understand what happened, recall the example of the street names we made in the previous section. Similarly to that case, during the Dyn attack websites and services were online, but the translation service of mnemonic names into IP addresses failed. Therefore users were unable to connect to those services. The Dyn case proved a simple concept:

If you can stop the DNS, you can effectively stop most Internet communication.

There are a plethora of attacks against the DNS that can affect its functioning, such as, for example, nameserver hijacking and DDoS attacks. DDoS attacks, in particular, have become increasingly relevant in the last decade. The

number of devices connected to the network, their availability, and their vulnerabilities made these attacks one of the most critical threats on the Internet nowadays [16]. DDoS attacks became cheap, effective, and growing in intensity. Furthermore, the level of knowledge required for performing these attacks dropped significantly since the introduction of DDoS-as-service tools [17]. Attackers use DDoS attacks to disrupt network operations and deny users access to online services. Over the years, we saw increasing trends of DDoS attacks used for different purposes (e.g., protests, political reasons, cyber-wars)[18].

In this scenario, the DNS, as one of the most critical services on the Internet, was not an exception. The DNS ecosystem becamed a relevant target of severe DDoS attacks over the last decade [19]. To protect the DNS against these attacks, our attention must be directed towards increasing its resilience [20]. This thesis endeavors to investigate this topic.

There is not just one single threat to DNS resilience

What does increasing DNS resilience mean? To answer this question, we need to understand which are the detractors of this resilience. In the context of this thesis, two primary detractors were identified: internal and external. Internal detractors, such as misconfigurations and Single Points of Failure, are caused by operator choices and mistakes and are within the control of the organization. Overprovisioning the DNS service by adding multiple nameservers is an example of an internal way to increase resilience. If one nameserver becomes unavailable, the other nameservers will continue to provide resolution service, ensuring the continued functioning of the system.

External detractors, such as DDoS attacks and third-party failures, are outside the control of operators. Recall the case of the Dyn attack, all the services that relied on a single operator (Dyn) suffered catastrophic consequences when Dyn became the target of large attacks. To increase external resilience, customers should diversify their service providers to avoid that operation or business disruption of a single company can entirely disrupt the activites of the customers. Customers should ensure that their chosen providers follow best practices in configuring their services. While internal resilience can be easily assessed through adherence to best practices, external resilience may require policies to mandate their implementation.

Increasing resilience: A technical and policy effort

Already in the original specification of the DNS protocol, the authors provided means for increasing resilience of the DNS infrastructure through the possibility of expressing multiple servers responsible for name resolution of a specific

domain. In 1997, with RFC2182 [21], the authors defined a set of best practices to operate secondary nameservers to increase DNS resilience. However, the deployment of those technique was left to the single operators of authoritative nameservers for the different domains. This freedom of choice created a complex scenario, where different parts of the DNS namespace are operated with different levels of resiliency.

However, since these decisions can impact the security of billions of Internet users, recently ICANN proposed an initiative to codify best practices into a set of global norms to improve the security and resilience of DNS: the Knowledge-Sharing and Instantiating Norms for DNS and Naming Security (KINDNS) [22][1]. A similar effort for routing security – Mutually Agreed Norms for Routing Security – provided inspiration for this initiative. In the MANRS program, operators are encouraged to voluntarily commit to a set of practices that will improve collective routing security [23]. Likewise, in KINDNS operators commit to adhere to the defined best practices and self-asses their compliance. At the time of writing of this thesis, the KINDNS conversation has just started, and stakeholders are still debating what should be included in the set of practices.

Next to these existing best practices and self-regulation initiatives, the path towards increased resilience also garnered the attention of policymakers. In the last years, national and international governments woke up and realized how critical DNS is for today's Internet and are now attempting to regulate it.

In this context, the European Union is discussing the Network and Information Security (NIS) 2 directive. The goal of the NIS2 directive is to improve the resilience of critical infrastructure, including the DNS, in the European Union. The NIS2 directive aims to reach this goal by providing a series of policies for enhancing the security of network and information systems, improving the cooperation and the data sharing among member states, and promoting the development of a resilient digital ecosystem in the European Union [24]. The NIS2 directive also aims to develop a framework for certifying critical infrastructure providers and raising attention to third-party dependencies. Finally, for the data sharing and collection part, the directive imposes obligations to operators to maintain DNS registration data ("Know Your Customer") to tackle abuse.

Independently, national governments also increased their attention toward the resilience aspect of their e-government services. An example of this is our collaboration with the Netherlands National Cyber Security Centre (NCSC-NL) on an initiative to assess the resilience of the DNS infrastructure serving Dutch e-government domains.

[1] `https://kindns.org`

1.3 Goal, Research Questions and Approach

From the observations of previous sections, it is clear that studying DNS resilience became fundamental to understand *if* and *how* the old DNS ecosystem can overcome modern attack threats.

To build this knowledge to better stand against those attacks, a comprehensive scientific characterization of the resilience of the global DNS ecosystem is needed.

In this thesis we want to contribute to this charachterization and thus we define the following goal:

> ***Goal:*** *to perform a comprehensive real-world measurement of resilience mechanisms of DNS authoritative infrastructure and to assess their mitigative effects in the face of DDoS attacks.*

To achieve this goal we defined five research questions (RQs). In the following section we will list them and describe for each how we address them.

Research Questions

In our research goal, we identify as one of the objectives of this thesis the assessment of the resilience properties of the DNS. An obvious peril to the resilience of a distributed system is consistency. Consequently, our first research question aims to identify consistency issues in the DNS ecosystem.

> ***RQ 1:*** *What types of inconsistency exist between different entities in the DNS and to what extent do they occur in practice?*

We address RQ 1 in chapters 3 and 4 of this thesis.

Having identified the major inconsistency issues in the DNS ecosystem, the obvious consequential research question is to understand the harm of those inconsistencies for DNS resilience. This leads us to the following research question:

> ***RQ 2:*** *What is the harm of these inconsistencies?*

We address RQ 2 alongside RQ 1 in chapters 3 and 4.

Shifting the focus from interaction between different operators to choices of single players, we will now focus on the adoption of the different resilience techniques available for the DNS ecosystem. Considering the defined research goal, the third research question will focus on assessing the available infrastructure resilience techniques for DNS operators and analyzing their adoption at scale. This leads to the following research question:

> ***RQ 3:*** *Which are the key strategies to enable DNS resilience and to what extent are they deployed?*

We address RQ 3 in chapters 5 and 8 of this thesis.

Assessing infrastructure in the wild provides a general overview of the resilience of the DNS ecosystem. However, some deployments are more relevant and critical towards modern society. We try to understand if these sensitive deployments are more eager to adopt better resilience mechanisms for their DNS infrastructure. We focus on e-government deployments, given their valuable nature for citizens and society. This leads to our fourth research question:

> ***RQ 4:*** *How do societally critical setups, such as government services, adopt resilience techniques?*

RQ 4 is addressed in chapter 7.

Finally, we want to investigate the effectiveness of the adopted resilience technique against DDoS attacks. This leads us to the final research question for this thesis:

> ***RQ 5:*** *To what extent do resilience mechanisms mitigate the effect of DDoS attacks?*

RQ 5 will be addressed in chapter 6.

Approach

To address the research questions, we take a mixed measurement and analytic approach. In most cases, we leverage available datasets such as: active DNS measurements (OpenINTEL [25]), zone files (OpenINTEL, DZDB [7]), BGP data (Prefix2AS [26]) and telescope DDoS attack feeds (RSDoS UCSD NT [27]). We combine these massive datasets on the DNS and real-world attacks to synthesize new knowledge. This approach allows us to assess the resilience properties of the DNS ecosystem at scale and their effectiveness against attacks. We corroborate our results with additional insights and factual evidence of choices of network operators, obtained in public and private communications. Furthermore, in case of unavailable datasets, as for the case of anycast, we developed techniques and run active measurements to fill the gap. We use the collected data for our studies and share them with other researchers.

1.4 Organisation and Key Contributions

Figure 1.2 shows a schematic outline of the structure of this thesis. The figure suggests a possible reading order of the chapters and the relationship between

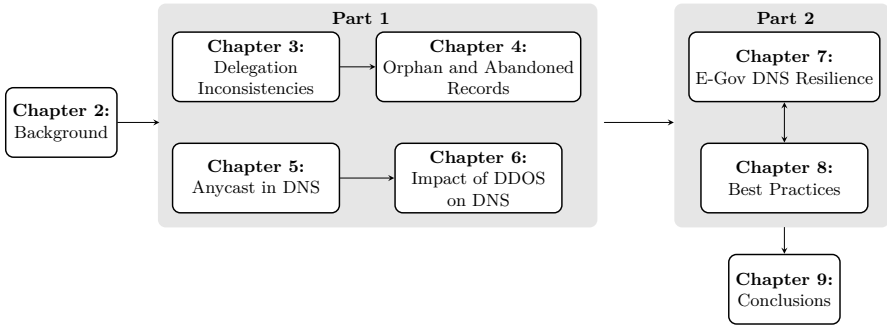


Figure 1.2: Schematic thesis outline

different chapters. The thesis consists of two main blocks: Part 1 and Part 2. In the first block, we build the knowledge torwards defining a set of norms and best practices that we enumerate and validate in the particular case of e-government deployments in the second block. We suggest the readers follow an ordered approach from chapter 2 to chapter 9 to build familiarity with the content of this thesis. In this section, we will provide a summary of each chapter, listing the key contributions and the reference to the original publication(s) on which the chapter is based.

Chapter 2: Background

In this chapter we will provide background information on the DNS, the possible resilience mechanisms, and DDoS attacks. We will describe the theoretical working principles of the DNS, its application in practice, and how the interactions between different components of the ecosystem take place. Furthermore, we will discuss IP Anycast, one of the most used resilience techniques for the DNS. Finally, we will briefly discuss the DDoS attacks ecosystem. The main contributions of this chapter are:

- An introduction to the DNS's working principles.
- An overview of IP Anycast.
- A brief categorization of DDoS attacks.

Chapter 3: DNS delegation inconsistencies

With this chapter, we introduce one of the first problems we identify affecting the resilience of the DNS ecosystem: delegation inconsistency. Delegation information (i.e., who is responsible for a specific domain) is duplicated along the

DNS hierarchy in both parent and child zones. While this information should be "consistent and remain so"[28], we found evidence this is not always the case. In the chapter, we analyze this issue by characterizing these inconsistencies at scale and shedding light on the practical consequences from a security and resilience point of view. We show 13 million domain names (8% of our dataset) exhibit forms of delegation inconsistency, representing a large-scale issue of the DNS ecosystem. The main contributions of this chapter are:

- Providing a broad-scale characterization of the delegation inconsistency issue over 166M domains, showing a relevant portion of the DNS space ($\approx 8\%$) is affected.
- Shedding light on the consequences of these delegation inconsistencies, with real world cases of reduced resilience and potential security risks.
- Evaluating the behavior of users and software resolution when facing delegation inconsistencies, unveiling inconsistent resolution behaviors that are resolver-dependent.

We base this chapter on the following peer-reviewed publication:

- R. Sommese, G.C.M. 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 Proceedings of the Passive and Active Measurement Conference, 2020 [29].

The results of the work discussed in this chapter were also shared with the DNS and network operator community at a technical conference (RIPE 80). This work influenced the discussion on "Delegation Revalidation by DNS Resolvers" draft[2]. To further help the community, we created a tool to automate the testing of resolver behaviors: SuperDNS[3].

Chapter 4: Orphan and Abandoned Records

In this chapter, we focus on another misconfiguration caused by miscommunication between two DNS entities: the registry, and the registrar. This misconfiguration takes the name of *orphan records*. Orphan records are former A/AAAA glue records, used in the resolution of the DNS delegation path (i.e., the process that determines which is the authoritative nameserver to contact for a specific domain). As soon as the parent domain expires, those records should be removed from the zone. Our analysis shows that this is not always the case, with

[2] `https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-ns-revalidation-03`
[3] `https://superdns.nl/`

a maximum of ≈ 36% orphans records on the total number of glue records in some zones. This work reproduces and extends an analysis performed 12 years ago by Kalafut *et al.* [30], showing that orphans are still a consistent threat to the DNS ecosystem. In the chapter, we expand the original study with an analysis of the possible causes. We also investigate risks induced by orphan records on DNS stability and security. The main contributions of this chapter are that we:

- Show, a decade after the original analysis, what the phenomenon of the orphan record looks like.
- Prove some TLDs adopted hygiene techniques to solve the orphan problem after the original work.
- Highlight another possible misconfiguration: *abandoned records*, a potential precursor of the creation of orphan records.
- Shed light on the possible risks of removing orphan records in breaking the resolution of other domains or reducing their infrastructural resilience.

This chapter is based on the following peer-reviewed publication:

- R. Sommese, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy and A. Sperotto. "The Forgotten Side of DNS: Orphan and Abandoned Records" in Proceedings of the Workshop on Traffic Measurements for Cybersecurity, 2020 [31].

We also presented this work at DNS-OARC 33 to DNS network operators and worked in synergy with Afilias, one of the most affected registry operators, to mitigate the problem[4].

Chapter 5: Anycast Adoption in the DNS Authoritative Infrastructure

In this chapter, we shed light on the different resilience techniques for increasing DNS infrastructural resilience. As mentioned before, one of the most effective techniques is IP Anycast. Measuring IP Anycast at scale is a challenging task. We will outline in the chapter, how we approached and overcame this challenge, performing a complete anycast census of the entire IPv4 space in less than 24 hours. We will then leverage this collected data providing the first large-scale characterization of anycast deployment in authoritative nameserver deployment of Top Level Domains (TLDs) (e.g., `.org`.) and Second Level Domains (SLDs)

[4] `https://circleid.com/posts/20200811-afilias-to-protect-tlds-against-potential-orphan-glue-exploits/`

(e.g., `example.org.`). We will also present evidence of organizational centralization related to anycast adoption. We conclude the chapter with reccomendations for network operators of anycast deployments. The main contributions of this chapter are:

- A methodology to perform anycast census at scale.
- A large scale characterization of anycast adoption in the DNS ecosystem, comparing 2017 and 2021 adoption. We show an increased adoption over time, with 98% of the TLDs adopting anycast and $\approx 50\%$ of the SLDs.
- An investigation of root causes of anycast adoption, showing they relate mainly to operator-driven choices.
- An overview of the increased organizational centralization of anycast deployments.
- An analysis of the relationship between anycast and *classical* DNS resilience technique adoption, showing that anycast deployments are less likely to deploy *classical* resilience mechanisms.

This chapter is based on the following peer-reviewed publication:

- R. Sommese, G. Akiwate, M. Jonker, G.C.M. 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" in Proceedings of the Network Traffic Measurement and Analysis Conference, 2021 [32].

and partly based on the following peer-reviewed publication:

- R. Sommese, L.M. Bertholdo, G. Akiwate, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy and A. Sperotto. "Manycast2 – Using Anycast to Measure Anycast", in Proceedings of the ACM Internet Measurement Conference, 2020 [33].

One of the publications on which this chapter is based has received:

- The 2021 TMA Best Paper Award

This work was also presented at DNS-OARC 39 and is part of an APNIC blogpost[5].

[5] `https://blog.apnic.net/2021/12/22/how-widely-adopted-is-anycast-in-the-dns/`

Chapter 6: Impact of DDoS attacks on DNS infrastructure

After assessing the adoption of resilience techniques in the wild, in this chapter, we will focus on their effectiveness against DDoS attacks. We provide a large-scale analysis of one year and five months of DDoS attacks on DNS authoritative infrastructure, analyzing their impact on the resolution in terms of disruption of services and performance impairments. We will show several prominent cases of attacks causing disruption to millions of domain names for several hours. We also show evidence of geo-politically motivated attacks related to the Russia-Ukraine conflict. To conclude, we assess the efficacy of the different DNS resilience techniques against DDoS attacks, showing anycast as the most effective technology to overcome devastating attacks. The main contributions of this chapter are:

- A longitudinal analysis of the DDoS attack ecosystem against DNS authoritative infrastructure, showing insights and opportunities of joining several datasets together.
- An assessment of the efficacy of different DNS resilience techniques, proving anycast as the most effective.
- A post-mortem characterization of root causes of devastating attack against DNS authoritative deployments.
- The potential usage of DNS reactive measurements in monitoring DDoS attack impact.

This chapter is based on the following peer-reviewed publication:

- R. Sommese, K. Claffy, R. van Rijswijk-Deij, A. Chattopadhyay, A. Dainotti, A. Sperotto and M. Jonker. “Investigating the Impact of DDoS Attacks on DNS Infrastructure”, in Proceedings of the ACM Internet Measurement Conference, 2022 [34].

Chapter 7: e-Government DNS Resilience

In this chapter, we will summarize the knowledge we built up of DNS resilience to investigate e-government domain deployments. Governments are increasingly using electronic communication to communicate with citizens, and these services have become an important strategic asset to protect national security. In the previous chapter, we demonstrated that attacks against DNS infrastructure might have catastrophic consequences. Consequently, especially e-government DNS services should be resilient against these attacks. Conversely, our analysis

shows a worrying picture of low-resilience deployments. E-government deployments of four analyzed countries exhibit high infrastructural and organizational centralization. Furthermore, we show that a single company, Microsoft, dominates the mail deployment of all four countries. The main contributions of this chapter are:

- An assessment of infrastructural resilience of e-government domain deployments shows a peak of 80% of domains in `.gov` relying on a single provider.
- An investigation of the organizational centralization of e-government deployments shows that the top 5 local providers dominate the market for each country.
- An analysis of the adoption of best practices, in regards to anycast deployment and caching configuration.
- An overview of e-government mail-provider market share, showing an dominant presence of Microsoft Outlook.

This chapter is based on the following peer-reviewed publication:

- R. Sommese, M. Jonker, J. van der Ham, G.C.M. Moura. "Assessing e-Government DNS Resilience", in Proceedings of the International Conference on Network and Service Management, 2022 [35].

Chapter 8: Best Practices for Critical Services

Before concluding the thesis, in this chapter, we will provide a summary of the best practices for DNS network operators to provide actionable intelligence out of the studies presented in the previous chapters. Therefore, the main contribution of this chapter is:

- A list of best practices for DNS network operators to increase the resilience of their deployments.

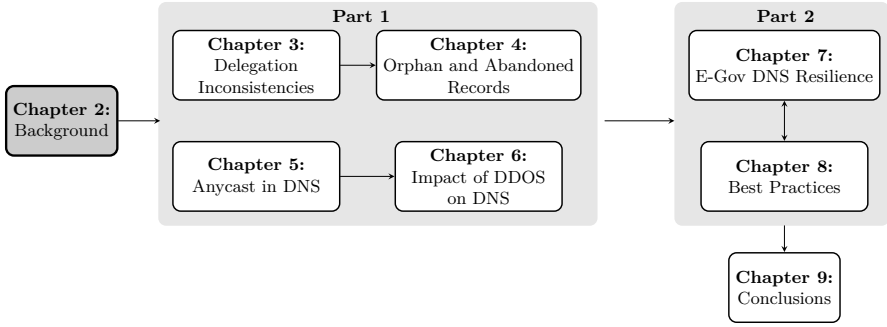
This chapter is based on a technical report delivered to the Dutch Government.

Chapter 9: Conclusions

In the last chapter we discuss the overall conclusion of this thesis based on the other chapters. Furthermore, we will outline future research directions.

CHAPTER 2

Background



This chapter provides an overview of the Domain Name System. It discusses the DNS's administrative and technical structure as well as its core concepts. It also describes DDoS attacks and strategies for improving DNS resilience, such as IP Anycast.

2.1 Reading Guide

This chapter offers comprehensive background information about the DNS and DDoS attack environments. Those seeking a fundamental understanding of the subject matter for the subsequent chapters are advised to read section 2.2, which explores the core DNS concepts.

To fully grasp the concept of how DNS resolution works and the entities involved, we suggest readers familiarize themselves with subsection 2.2.6 before proceeding to chapter 3. For a deeper understanding of how those different entities interact via the EPP protocol, please refer to subsection 2.2.8, as this lays the groundwork for the content presented in chapter 4. Prior to delving into chapters 5,7, and 8, we recommend readers become familiar with the concept of anycast, explained in section 2.4. Finally, to fully understand the mechanics of DDoS attacks discussed in chapter 6, we suggest readers refer to section 2.3.

2.2 The DNS

In this section, we will provide a brief explanation of the DNS ecosystem, the labeling structure, the resolution process and the intercommunication between different parties.

2.2.1 A Hierarchical Database

The DNS is a global decentralized and hierarchical naming system for computers, services and Internet resources. The prominent role of the DNS is to translate domain names, which can be easily memorized by humans, to the numerical IP addresses needed for the purpose of locating and identifying computer services and devices with the underlying network protocols. As an example, names such as `google.com` or `example.org`, can be translated using the DNS into numerical IP addresses to easily communicate on the Internet.

```
$ host google.com
google.com has address 142.251.36.14
google.com has IPv6 address 2a00:1450:400e:80f::200e
```

The DNS hierarchy is organized with a set of domains into a tree-like structure. Each domain in the hierarchy is handled by a set of nameservers. Those nameservers are responsible for storing the information about the domain and possibly its subdomains. All the nameservers can be queried by using the same protocol: the DNS.

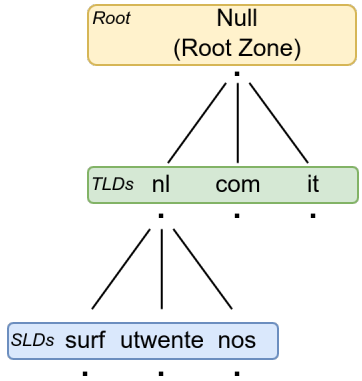


Figure 2.1: DNS hierarchy example: different levels of the hierarchy are separated by the . character

In the following sections, we will delve deeper into the core concepts of the DNS ecosystem and the various entities involved. Readers interested in a more detailed explanation of the DNS are advised to read sections 2, 3, 6 and 7 of *Addressing the Challenges of Modern DNS: a Comprehensive Tutorial* by Van der Toorn and Müller *et al.* [36].

2.2.2 DNS Concepts

A **domain name** is is a unique and human-readable string of characters that identifies a particular machine or other online resource on the Internet. Domain names are used, for example, in URLs to identify and locate websites.

The structure of a domain name consists of a series of **labels** separated by dots. Each label composing domain names is limited to a maximum of 63 ASCII characters. Only letters (`A-Z`, `a-z`), digits (`0-9`) and the hyphen (`-`) are allowed. [1] Labels are *case insensitive*, this means that `example.com.` and `EXAMPLE.COM.` identify the same resource. The right-most label, between the two last dots, is called the **top-level domain (TLD)**. The next label to the left is the **second-level domain (SLD)**, and so on. In the domain name `example.com`, `com` is the TLD and `example` is the SLD. The last dot is usually implicit in the DNS resolution and in most of the cases can be omitted. In the

[1] The limitation of ASCII-only characters had as impact that names containing non-Latin alphabet characters could not be used in the DNS. To overcome this issue, Internationalized Domain Names (IDN) were introduced in RFC3490 [37]. Nowadays, the user types the IDN names and the resolver software takes the responsibility of performing an ASCII conversion with the algorithm specified in RFC3490.

resolution process the DNS resolver software adds the final dot and appends a null caracter after it.

The left-most label in a domain name is usuallly called the **hostname**. This term is also used to establish a local name for a machine. In some cases the local hostname of a machine and its DNS hostname can coincide.

A **Fully Qualified Domain Name (FQDN)** is a domain name that specifies the complete path. It includes a hostname, a domain name, as well as a top-level domain (TLD) that identifies the specific domain within which the hostname resides. For example, `www.example.com.` is an FQDN because it specifies the hostname `www`, the domain name `example`, the TLD `com` and the last dot (indicating the root). In Figure 2.1, we provide a visual example of the DNS hierachy, with SLDs (i.e., `surf`, `utwente`, `nos`), TLDs (i.e., `nl`, `com`, `it`)) and the root.

2.2.3 DNS Hierarchical Entities

In this section, we highlight the actors responsible for managing and performing tasks within the DNS ecosystem. The **Internet Corporation for Assigned Names and Numbers (ICANN)** is a non-profit organization that is responsible for coordinating the maintenance and operation of the DNS. One of ICANN's functions is, for example, to manage the process of introducing new generic top-level domains (gTLDs) into the DNS hierarchy.

Top-level domains are the second highest level of the DNS hierarchy after the root. They are used to identify the type of organization or activity associated with a particular domain name. There are two main types of TLDs: **generic top-level domains (gTLDs)** and **country-code top-level domains (ccTLDs)**.

GTLDs are typically used to identify the type of organization or activity associated with a particular domain name and they are not associated with a specific country or region. Common gTLDs are `.com` (for commercial organizations), `.org` (for non-profit organizations), `.net` (for network infrastructure organizations), and `.edu` (for educational institutions). GTLDs are operated by different companies and are regulated by ICANN.

The ccTLDs, on the other hand, are two-letter codes that are associated with a specific country or territory. For example, `.nl` is the ccTLD for the Netherlands, `.it` is the ccTLD for Italy, and `.de` is the ccTLD for Germany. Generally, ccTLD abbreviations are aligned with ISO 3166 two-letter country codes, with a few exceptions such as `.eu`. The ccTLDs are often used to indicate the country or region in which the resource is physically or administratively based and are typically managed by organizations that are either government-affiliated or non-profit. These organizations are chosen by the relevant government or In-

1	$TTL 86400 ; Default time-to-live (TTL) value for all records			
2	example.org.	IN	SOA	ns1.example.org. admin.example.org. (
3				2022121011 ; serial number
4				7200 ; refresh
5				900 ; retry
6				604800 ; expire
7				86400 ; minimum
8				)
9	example.org.	IN	A	192.0.2.1
10	example.org.	IN	AAAA	2001:DB8::1
11	redirect.example.org.	IN	CNAME	foo.com.
12	example.org.	IN	MX	10 mail.example.org.
13	example.org.	IN	MX	20 mail.example.com.
14	example.org.	IN	NS	ns1.example.org.
15	example.org.	IN	NS	ns2.example.org.

Table 2.1: An example of a zone file for `example.org`

ternet community of the country to which the ccTLD is assigned to oversee the administration and management of the ccTLD. The **Internet Assigned Numbers Authority (IANA)** is then responsible for the allocation.

In recent years, with the aim of expanding the range of options available to domain name registrants and increasing the flexibility and creativity of the DNS, ICANN has introduced a number of new gTLDs. These new gTLDs include a wide variety of types, such as `.travel`, `.io`, `.shop`, and `.book`. They are meant to reflect the diversity of online activities and to provide users with more descriptive and relevant domain names.

This expansion of gTLDs by ICANN represented an important development in the evolution of the DNS, and while it brought some benefits for users, businesses, and other stakeholders, it also created new security challenges [38].

2.2.4 Zone Files

A **zone** is a portion of the DNS managed by a single entity. A **zone file** describes all the **Resource Records (RRs)** related to a zone, and its format is defined in two IETF standards [28, 39]. DNS resource records are records that contain information about domain names and their corresponding resource (e.g., IP Address, Mail Server address, etc.).

There are several types of DNS records, including name server (NS) records, which specify the DNS nameservers that are responsible for a particular domain or subdomain, and address (A) records, which map a domain name to an IPv4 address. Other types of DNS records include IPv6 mapping (AAAA) records,

mail exchange (MX) records, which specify the mail servers for a domain, and Canonical Name (CNAME) records, which allow domain name label aliasing. DNS records are stored in zone files.

Zone files are files that contain all the DNS records for a particular domain or subdomain. Zones are used by DNS servers to keep and manage information regarding the domain names that they are responsible for.

In Table 2.1, we report an example zone file for the domain `example.org` with a non-exhaustive set of DNS records. The leftmost column in our zone zone file represents the hostname or the FQDN, the second column specifies the CLASS. Nowadays, bar some exceptions, the only used class is the Internet (IN) class. The third column is the Resource Record Type (RRTYPE). It is used to identify the kind of resource provided by the last column: the RDATA (i.e., the actual information served).

In our example zone file, we reported some commonly used DNS records:

- **SOA record**: specifies the primary DNS server for the domain and additional information about the zone file, such as the serial number, refresh interval, and expiration time. The **serial number** is used by other DNS servers to identify whether they have the most recent version of the zone file because it is incremented each time the file is updated. The **refresh interval** and **retry interval** specify how long a secondary DNS server should wait before checking for updates to the zone file and again, respectively, if it cannot connect to the primary server. The **minimum interval** is the default time-to-live (TTL) for records in the zone file, and the **expire interval** is the time after which a secondary server will stop responding to queries if it cannot reach the primary server.
- **A record**: used to map a domain name to an IPv4 address. In our case, the A record specifies that the domain `example.org.` is mapped to the IPv4 address 192.0.2.1.
- **AAAA record**: used to map a domain name to an IPv6 address.
- **CNAME record**: used to specify an alias for a certain label. In our case `redirect.example.org.` is an alias of `foo.com.`.
- **MX record**: used to specify the mail server for a domain. An MX record is composed by a number – the **priority**, and a label that specifies the mail server. In case of multiple MX records, as in our case, the Mail Transfer Agent (MTA) will prefer the server with lowest priority and uses the others in case of failure.
- **NS record**: specifies the authoritative name servers for a domain and determines which servers are responsible for answering queries about that

domain. NS records play a crucial role in the DNS ecosystem, enabling the delegation of responsibilities within the hierarchy of the domain name system. Consequently, in this thesis we will undertake an in-depth analysis of the configuration of NS records by network operators.

Having multiple records for the same hostname and RRTYPE is generally allowed in the DNS and is widely used for providing load balancing (e.g., multiple A records can help to balance the traffic among different web-servers). In case of NS records, multiple records are used both for load balancing and resilience. Before failing, well-configured DNS resolvers will try to exhaustively query all the available nameservers.

Zones may, and in general should, also contain other kinds of records. [2]

2.2.5 DNS Message Format

Before going in depth on how DNS resolution works, we need to understand some basic concepts related to the format of queries and responses provided by DNS servers. All DNS communications are carried in a single format known as a message. This message level format is divided into five sections (some of which can be empty):

- **HEADER section**: defines the sections that follow and also indicates if the message is either a query or a response, a standard query or any other type of opcode, and so on. This section is always present.
- **QUESTION section**: carries the question (i.e., what is the object queried, e.g., `utwente.nl` A record (?)).
- **ANSWER section**: contains Resource Records (RRs) that provide an answer to the question (e.g., `utwente.nl` 130.89.3.249)
- **AUTORITY section**: contains RRs that refer to an authoritative nameserver (e.g., `ns1.utwente.nl`).
- **ADDITIONAL section**: contains RRs which are related to the query, but are not strictly an answer to the question. An example of ADDITIONAL records can be the IP address of `ns1.utwente.nl`.

2.2.6 In the Life of a DNS Query

To better understand how various DNS components interact with each other, we need to immerse ourselves in the everyday life of a DNS query.

[2] A more extensive list of DNS records types can be found at `https://www.nslookup.io/learning/dns-record-types/`

In Figure 2.2, we present a simple example of a DNS query to a resolver with an empty cache. This assumption will show the full resolution process. Our journey consists of the following steps:

- **Step 1:** The user asks the browser to open a website, `utwente.nl`.
- **Step 2:** The browser does not know the address of the website requested by the user. In order to perform the resolution, the browser will ask the DNS resolver(s). In our example, we simplify the request by illustrating the case of a single resolver. What typically happens is that users rely on a local resolver (e.g. installed on the same machine) called a **stub resolver** for caching and performance reasons, which forwards queries to an external **recursive resolver**.
- **Step 3:** The first operation performed by our recursive resolver after receiving the query is to look in the local cache to see if the name has been queried recently and the answer is still valid (i.e., TTL not expired). In our example case, the response is negative, and the cache is completely empty, meaning we need to start from the root zone.
- **Step 4:** The root server receives a query for an A record for the domain `utwente.nl`. It does not know the answer but knows who is responsible for all `.nl` domains: SIDN. The root server also knows the nameservers' names of SIDN's authoritative nameservers.
- **Step 5:** The root server returns the NS records for the `.nl` domain to the resolver. Usually, multiple records are returned, but in our example case, we show the fictional case of a single record `ns1.dns.nl`. Note that these records provided are not `ANSWERS`, but are `AUTHORITATIVE` information, meaning they indicate to the resolver who to contact further. Alongside these NS records, the server may return `ADDITIONAL` records, containing the address of `ns1.dns.nl`, if known to the server.
- **Step 6:** The recursive resolver tries to resolve `ns1.dns.nl` using the `ADDITIONAL` record information or local cache. In case of failure, a new recursive resolution (i.e., steps 2-9) is performed for `ns1.dns.nl` before continuing. In our example case, our recursive resolver knows the address of `ns1.dns.nl` and, similarly to what happened in Step 5, will ask the authoritative nameserver `ns1.dns.nl` to return the A record for the domain `utwente.nl`. Also in this case, the nameserver `ns1.dns.nl` does not know the answer but knows who is responsible for all utwente.nl domains: University of Twente, it also knows the nameservers' names of UTwente's authoritative nameservers.

- **Step 7:** The nameserver returns the NS records for the `utwente.nl` domain to the resolver, specifically `ns1.utwente.nl`. In this case, for the sake of simplicity, our resolver already knows the IP address of `ns1.utwente.nl` because it got this information from the `ADDITIONAL` section in the referral from `.nl`

- **Step 8:** Our resolver then contacts the nameserver `ns1.utwente.nl` to request an A record for the domain `utwente.nl`. Finally, the nameserver has the answer and provides it to our resolver.

- **Step 9:** The address of `utwente.nl` is returned to the resolver.

- **Step 10:** The resolver then sends the address of the `utwente.nl` domain to the waiting browser.

While the DNS resolution process may seem long and require multiple round-trips to different servers, in the real world, resolvers speed up the process using caching. All information collected during the resolution process is stored in the local cache of the resolvers and used for subsequent queries until their Time-to-Live (TTL) value expires. This means that root servers are rarely reached for resolving the IP address of domains on popular top-level domains (TLDs),

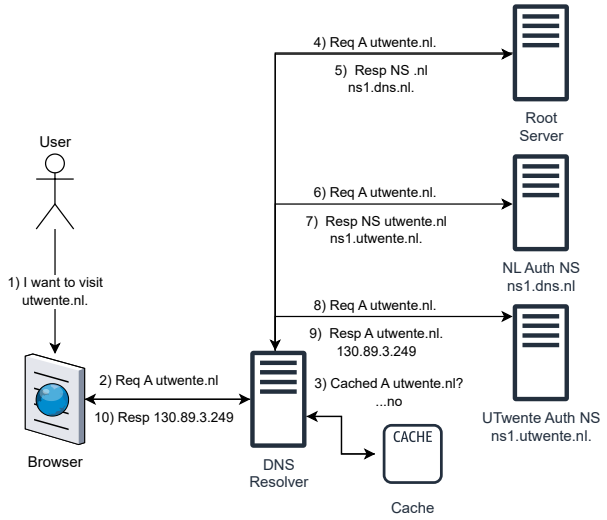


Figure 2.2: DNS resolution example: a fictional resolution of the domain `utwente.nl`. made from a resolver with an empty cache.

especially for large resolvers. These resolvers prefer to contact the known authoritative nameservers directly for a specific TLD. Similarly, popular queries are returned from cache.

An interesting aspect to note in the resolution described above is that the IP address of the nameserver, `ns1.dns.nl.`, is provided by the root servers. Although these servers are not authoritative for answering queries for `.nl.` names, they can provide the A/AAAA answers for related nameserver records, breaking the circular dependency that would otherwise be established. These records are known as `glue records` and can be either `in-bailiwick`, where they share part of the hierarchy with their ancestor for which they provide authoritative DNS, or `out-of-bailiwick` in the opposite scenario (e.g., if the address were `ns1.dns.com.`).

Resolvers also take on the responsibility of retrying in case of failure. If multiple authoritative nameservers are returned for the resolution of a specific domain, the resolver may try to contact all of them (also several times) before giving up and returning an error to the user. This process highlights the importance of having diverse deployments of DNS authoritative nameservers for better resilience. In case of partial failure, the resolvers will take care of contacting another nameserver.

We can also see from the figure the relevance of some elements in the chain, such as the root servers and TLD nameservers. In principle, they are queried for all domains with empty or disabled caches. This implies that their deployment should be robust and resilient against Denial of Service (DoS) attacks and misconfiguration.

2.2.7 The RRR Model: Registries, Registrars and Registrants

The DNS is not just a system where technical records play a role. As already defined in the introduction, the DNS became a multi-billion dollar industry thanks to the model where administrative responsibility for managing parts of the name space is delegated to various, often commercial parties. When we talk about the DNS ecosystem, we usually identify 3 main stakeholders. Those three stakeholders form the RRR model: (i) the registry, (ii) the registrar, (iii) the registrant.

A **registrant** is an individual or an organization wanting to register and possess a domain name. In order to accomplish this goal, registrants pay a registrar registration fee. This fee differs for different TLDs and registrars.

A **registrar** is an entity that accepts those payments and *creates* the domain in interaction with the registry. Domain names are usually sold on an annual basis, this implies that registrars are also responsible for the life-cycle manage-

ment of domain names, both in terms of resource management and recurring billing.

A **registry** operates the whole infrastructure of top-level domains and processes the requests for domain creation, renewal and deletion issued by registrars. There is one single registry for each TLD and those are designated to be unique by ICANN and IANA via a contract for gTLDs and by the countries for ccTLDs.

This model with three stakeholders brought a complexity into the creation and the management of domain names, due to the multitude of different parties interacting with each other. Usually registrants can perform actions on the domain name via the customer portal of the chosen registrar. The communication between registrars and registry, instead, takes place using the Extensible Provisioning Protocol (EPP) protocol.

2.2.8 EPP - the Extensible Provisioning Protocol

In this section, we will outline the workings of the EPP protocol. Readers should have a fundamental understanding of this protocol's basic principles since it plays a vital role in domain creation and management. Later on in this thesis, we will show that misconfigurations can occur if registrars or registrants do not fully follow how this protocol works in all of its processes. EPP is an XML text protocol defined in RFC5730 [40]. The primary goal of EPP is to permit multiple service providers to manipulate objects in a shared centralized object directory. Over the years, EPP became the standard Internet domain registration protocol between registrars and registries.

Listing 2.1: Create a new domain: example.org

```
<command>
 <create>
  <domain:create>
   <domain:name>example.org</domain:name>
   <domain:period unit="y">2</domain:period>
   <domain:ns>
    <domain:hostObj>a.iana-servers.net</domain:hostObj>
    <domain:hostObj>b.iana-servers.net</domain:hostObj>
   </domain:ns>
   <domain:registrant>registrantID</domain:registrant>
  </domain:create>
 </create>
</command>
```

The protocol defines and describes the interaction between these two parties via a standard set of atomic and idempotent commands. EPP defines three main object types: domains, contacts and hosts. EPP also defines *actions*, such as: check, info, create, update, delete, transfer and renew. *Domain* objects represent the domain itself, *contacts* are the contact information of the registrant, and finally *hosts* represent glue records. In the following, we provide a brief overview of how EPP interaction works between registrars and registries in terms of messages exchanged.

The XML shown in Listing 2.1 is related to the creation of a new domain `example.org`. The relevant fields are the duration of registration (in this case 2 years), the nameservers ([a,b].iana-servers.net) and the registrantID which refers to a registrant contact object inserted before in the database. In this case the nameservers are out-of-bailiwick, which, as explained before, means that the nameservers for a domain are not in the same domain. For an in-bailiwick domain, instead, we need a procedure for creating the corresponding glue record through EPP. This is performed with the creation of host objects.

In the case shown in Listing 2.2, A and AAAA glue records are created for the name `a.iana-servers.net`. According to the EPP protocol, the creation of host and domain resources are independent operations. A registrar typically creates glue records in the case of an in-bailiwick `NS` record.

Listing 2.2: Create a new glue record: a.iana-server.net

```
<command>
 <create>
  <host:create>
   <host:name>a.iana-servers.net</host:name>
   <host:addr ip="v4">199.43.135.53</host:addr>
   <host:addr ip="v6">2001:500:8f::53</host:addr>
  </host:create>
 </create>
</command>
```

The EPP specification [40] does not define who is responsible for cleaning up glue records if they are no longer required (e.g., in case of expired domains). A registrar could periodically check this, or a registry could check if glue records in its database are actually required by an *in-bailiwick* domain before publishing a zone on its name servers. When a domain expires, after a grace period, the registrar returns the responsibility of managing the domain to the registry, which should ensure that all related records are deleted.

The registry or the registrar has the responsibility of checking if a host record (glue) is required due to an `NS` record of the same domain pointing to

it. A registry could perform additional checks before serving an updated zone by removing unnecessary glue records. Equally, a registrar can enforce policies during the creation of these records, such as creating glue records only when `NS` records point to them and deleting them when these records are removed or updated. These overlapping responsibilities exist until the domain is registered. When a domain expires, after a grace period, the responsibility of managing the domain is returned to the registry, which should ensure that all related resources are deleted.

2.3 DDoS Attacks

Denial of Service (DoS) attacks are one of the earliest electronic system threats. Attackers aim to overload target machines with requests or traffic, resulting in the target machines being unable to provide service to users, hence the name **denial of service**.

2.3.1 DDoS Attacks Characteristics

DoS attacks can be of different classes. We can have **resource exhaustion attacks**, **volumetric attacks**, or **semantic attacks**. For example, the well-known TCP-SYN flood, is a clear example of a resource exhaustion attack aiming to saturate the victim's network state tables. To achieve this goal, the TCP-SYN flood attack involves the attacker sending a large number of TCP-SYN packets to the victim, making it believe that the attacker wants to initiate many TCP connections. This quickly exhausts the memory structure of the victim host, making new connections from legitimate hosts impossible.

Volumetric attacks, instead, are attacks aimed at saturating the target host's or upstream router's network bandwidth. Examples of volumetric attacks are: UDP flood attacks, ICMP floods attacks etc..

Finally, **semantic attacks** exploit specific weaknesses (e.g., in L7 protocols) to cause high CPU load or exploitation, making the service unavailable. A prime example of semantic attacks are application-aware attacks (e.g., initiating an exhaustive time-consuming search through a database made possible by the server not imposing proper limits).

Intuitively, an attack from a single source is easily detectable and can be blocked. To overcome this issue, attackers perform Distributed Denial of Service (DDoS) attacks, which are DoS attacks originating from multiple locations. Botnets of infected devices are often used to perform these attacks on a large scale. One notable botnet used for DDoS attacks was the Mirai botnet, which was used during the Dyn attack described in chapter 1.

2.3.2 Reflection and Amplification

Attackers may use multiple sources to conceal their tracks while performing a direct attack or use public large deployments with significant bandwidth to conduct volumetric attacks. To achieve their objective and increase the number of sources or bandwidth, attackers can employ both **reflection** and **amplification** attacks. These attacks serve to amplify the scale of the attack, resulting in a larger impact.

In **reflection attacks**, attackers send traffic to a service pretending to be the victim. The service will reply to what appears as a legitimate request from the victim, causing the victim to handle the unwanted traffic and impacting the provided service. **Amplification attacks** are often used in conjunction with reflection attacks. These attacks rely on the principle that some services provide larger responses compared to the request, and the amplification factor is calculated by dividing the response size by the request size. Amplification attacks are very effective. Reflection and amplification protocols are typically transmitted over UDP, though in some cases they can be performed using TCP [41]. DNS and NTP are two of the most widely used protocols in performing these attacks due to their high amplification factor.

Reflection and Amplification attacks rely on the ability to spoof source addresses in packets and transmit them globally on the internet. Network Ingress Filtering (BCP 38) [42] is an IETF best practice that recommends a filter at the border of each network to only allow packets with legitimate source addresses to enter the Internet. CAIDA monitors the adoption of source address validation through its Spoofer measurement platform [3], which provides a client software tool for various platforms (Windows, Linux, Mac) that allows users to test spoofing capabilities of networks and reports results to CAIDA for public aggregation. According to Spoofer data, approximately 18% of the 1503 measured IPv4 /24 prefixes are still vulnerable to IP spoofing. This highlights the fact that, despite 23 years since the publication of BCP 38, some operators have not yet implemented the best practice of ingress filtering.

2.3.3 Identifying DDoS Attacks

To comprehend the consequences of a DDoS attack, it is crucial to first identify its occurrence. Unfortunately, obtaining data on DDoS attacks can be challenging. The ability to detect a DDoS attack is dependent on the specific type of attack.

For instance, to deduce the behavior of attackers in reflected attack cases, complex honeypot reflectors that imitate commonly used sources are necessary.

[3] https://www.caida.org/projects/spoofer/

To identify spoofed attacks, access to a vast source of backscatter traffic is required, which can be monitored using a network telescope - a large infrastructure that observes a significant portion of unused IPv4 space, also known as a darknet. Finally, detecting unspoofed attacks necessitates the cooperation of victims and/or network providers, who do not typically share such data.

Several research project aim to detect DDoS attacks. The AmpPot Project monitors reflection and amplification attacks through different Honeypots, including DNS-based ones [43]. CAIDA monitors Randomly Spoofed Denial of Service Attacks (RS-DoS) [27] through the UCSD Network Telescope by discovering responses to spoofed queries in Internet Background Radiation passively observing a large fraction of unused IPv4 network space (currently a /9 and /10, ≈1/341 of the entire IPv4 space). We will leverage this last dataset in this thesis as source of events of DDoS attacks.

2.4 IP Anycast

IP Anycast, first described in RFC1546 [44], is a technique used to locate a service supported by multiple servers without specifying which exact server to use. Over time, IP Anycast has evolved to address its associated engineering challenges. Today, it is a fundamental mechanism for enhancing the resilience and performance of Internet services by providing more physical deployments and optimized routing. The widely adopted traditional method of Anycast leverages routing protocol preferences to choose the optimal route to one of the multiple servers with the same IP address. This technique is used for both IPv4 and IPv6 and involves announcing the same service IP address from different sites worldwide, allowing routing protocols to determine the best server for each client. The group of users or client IP prefixes routed to a specific anycast site is known as the catchment.

To better understand how anycast works, assume you want to visit a website and it happens to be hosted by a provider using anycast. When you enter the URL of the website into your browser window, your request is sent via the Internet to the web-server. Given that the server IP is anycasted, the network infrastructure will route your request to the nearest available anycast node. An anycast node is a host that share the same IP address with other servers physically located in diffenent locations. Let us pretend there are four anycast nodes in London, Amsterdam, Paris and New York. The anycast network will route requests to the node that is nearest (i.e., in terms of routing, and often geographically) to your device. As a result, if you are in Enschede, your request will be sent to the nearest anycast node in Amsterdam. If you are in Bristol, however, your request will be directed to the anycast node in London because

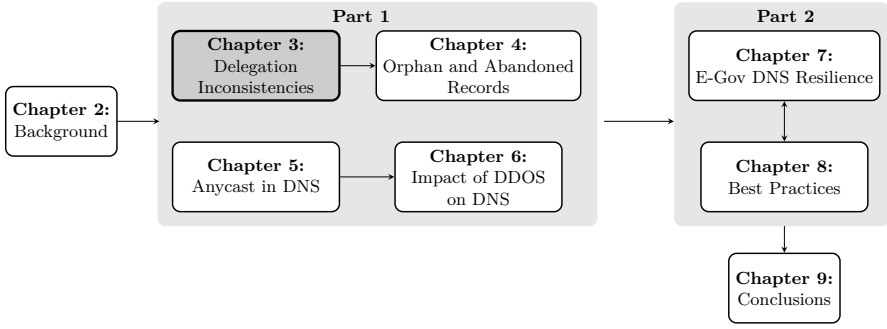
that is the nearest node to you. In this way anycast minimizes latency and offers quick responses. Additionally, if for example, the server in Amsterdam fails, the anycast network will automatically redirect your request to one of the other available nodes.

In the example discussed earlier, we made the assumption that the route selection process always chooses the replica that is geographically closest. However, in the real world, this process is more complex. The selection of a replica is determined by the decisions of routers using the **Border Gateway Protocol (BGP)**. This process can be influenced by a wide range of factors and decisions made by network operators, like community values and peering preferences. This complexity creates a challenging scenario for configuring well-deployed anycast networks [45].

Anycast has been proven to have several benefits in terms of enhancing availability, distributing load, and reducing access times. For these reasons anycast has been increasingly adopted by cloud providers, Content Delivery Networks (CDNs), DDoS protection services, and in particular DNS providers. In this scenario, the identification of anycast addresses and their sources is becoming crucial to accurately assess the Internet's resilience, as anycast is gaining popularity. However, the current IPv4 method of assigning the same unicast address to multiple hosts and using routing to implement anycast can make it difficult to identify anycast addresses due to the non-transparency of routing. We study how anycast has become pervasive in the DNS later on in this thesis.

CHAPTER 3

DNS delegation inconsistencies



In the previous chapter, we introduced the different entities responsible for handling the information in the DNS ecosystem. We saw that some information, such as delegations, has to be duplicated at different levels of the hierarchy. In this chapter, we will focus on the consistency of nameserver delegations among parent and child zones. The delegations are consistent if both the parent and the child zone nameserver records are kept in sync. To achieve this goal, the operators should ensure that this synchronization takes place. In this chapter, we show a consistent number of SLDs affected by nameserver delegation inconsistency (RQ-1). We will investigate the possible issues that arise in case of inconsistency and their impact on the resilience of the DNS ecosystem, showing cases where there is a risk for hijacks and reduced resilience (RQ-2). The study discussed in this chapter was performed in late 2019, published in an academic conference [29] and presented at RIPE80 winning the RACI grant award.

3.1 Introduction

One of the key mechanism that enables the DNS to be hierarchical and distributed is *delegation* [46]. Recall from the previous chapter that the DNS hierarchy is organized in parent and child *zones*—typically managed by different entities. Those entities share common information (NS records) among which are the authoritative name servers for a given domain. While RFC1034 [28] states that the NS records at both parent and child should be "consistent and remain so", there is evidence that this is not always the case [47].

In this chapter, we analyze this issue by *(i)* providing a broad characterization of inconsistencies in DNS delegations, and *(ii)* investigating and shedding light on their practical consequences. Specifically, we first evaluate if there are inconsistencies between parent and child sets of NS records (NSSet) for all active second-level domain names of three large DNS zones: `.com`, `.net`, and `.org` (section 3.4)—together comprising of more than 166M domain names (50% of the DNS namespace), as well as all top-level domains (TLDs) from the Root DNS zone [48]. We show that while 80% of these domain names exhibit consistency, 8% (i.e., 13 million domains) do not. These inconsistencies affect even large and popular organizations, including Twitter, Intel and AT&T. Overall we find that at least 50k `.com`, `.net`, and `.org` domains of the Alexa Top 1M list are affected. In chapter 7, we will show that this inconsistency also affects sensitive deployments, such as e-government domains.

We classify these inconsistencies into four categories (section 3.4): the cases (i) in which the parent and child NSSets are *disjoint* sets, (ii) the parent NSSet is a *subset* of the child NSSet, (iii) the parent NSSet is a *superset* of the child NSSet and (iv) the parent and child NSSet have a non-empty intersection but do not match (ii) or (iii). These inconsistencies are not without harm. Even in the case in which disjoint sets of NS records resolve to the same IP addresses, case (i) introduces fragility in the DNS infrastructure, since operators need to maintain different information at different levels of the DNS hierarchy, which are typically under separate administrative control. Case (ii) may lead to unresponsive name servers, while case (iii) points to a quite understandable error of modifying the child zone while forgetting the parent, but it offers a false sense of resilience and it results in improper load balancing among the name servers. This lower resilience can potentially have out of control effects in case of DDoS attacks, with unpredictable behaviour on end-user resolutions. Finally, case (iv), which we see happening in more than 10% of the cases in which parent and child have a non-empty intersection, suffers from all the aforementioned risks.

To understand the practical consequences of such inconsistencies, we emulate all four categories (section 3.5) by setting up a test domain name and issuing DNS queries from more than 15k vantage points. Our experiment highlights

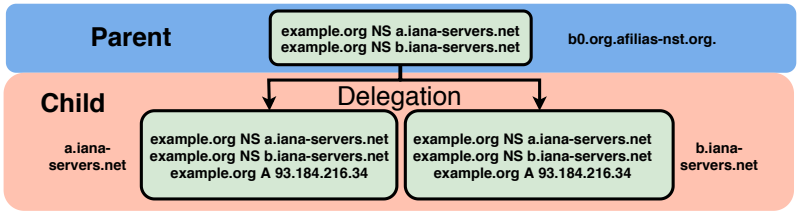


Figure 3.1: Domain name delegation: parent and child authoritative servers.

the consequences of delegation inconsistency on query load distribution in the wild. We then investigate how popular DNS resolvers from different vendors deal with such inconsistencies (section 3.6), and find that some resolvers do not comply with RFC specifications.

Finally, we conclude the chapter discussing our findings and offering recommendations for domain name operators to manage the inconsistencies we identified.

3.2 Background

Recall from chapter 2 that DNS uses a *hierarchical name space* [28], in which the root node is the *dot* (`.`). Zones under the root—the top-level domains such as `.org`— are referred to as *delegations* [46]. These delegations have second-level delegations of their own such as `example.org`. To create delegations for a *child* zone (such as `example.org`), DNS NS records [28] are added to the *parent* zone (`.org` in Figure 3.1). In this example, the NS records are `[a,b].iana-servers.net`, which, in practice, means that these records are the *authoritative* name servers for `example.org`, i.e., servers that have definitive information about the `example.org` zone.

RFC1034 states that the NSSet should be consistent between parent and child authoritative servers. This, however, is far from trivial. Parent and child zones and servers are almost always maintained by different organizations across administrative boundaries. The most common case is where the parent is a TLD. Delegation changes in the parent go through the so-called Registry-Registrar-Registrant (RRR) channel for almost all TLDs. In this model, the Registry operates the TLD, the Registrar sells domain names under the TLD and the Registrant is the domain holder. If the domain holder wants to change the delegation, they can make the change in their child zone, but need to file a request with the Registry through the Registrar. This process currently always happens via an out-of-band channel (not through the DNS) and in some cases

may even require forms on paper. Add to this that domain holders may not always be aware of this complexity and the requirement to keep parent and child in sync, and it is clear to see that keeping the DNS consistent is prone to human errors.

3.3 Related Work

The problem of Parent-Child consistency is addressed in RFC7477 [49], which introduces a method to automatically keep records in the parent in sync through a periodical polling of the child using SOA records and a new type of record (CSYNC). Unfortunately, RFC7477 lacks deployment, due to its complexity.

Pappas *et al.* [50] analyzed divergence between parent and child delegations on sample domains ($\sim$ 6M) from multiple zones and found inconsistencies in 21% of the DNS zones evaluated, in three different years. Kristoff [47] analysed delegations in `.edu` and finds that 25% of `.edu` delegations suffer some form of inconsistency. In his work, he considers 3 types of inconsistency: superset, subset and disjoint-set. Our work significantly expands on both studies by considering both the largest generic TLDs `.com`, `.net` and `.org` and the root zone of the DNS ($\sim$ 166 million domains, section 3.4) and evaluating implications for resolvers in the wild (section 3.5).

Liu *et al.* show that dangling delegation records referring to expired resources (e.g., cloud IP addresses or names) left in the parent or child pose a significant risk [51]. An attacker can obtain control of these records through the same cloud services by randomly registering new services, and in this way take control of the domain. Finally, Moura *et al.* [52] have looked into the consistency of time-to-live values [28] of parent and child NS records.

3.4 Parent and Child Delegation Consistency

RFC1034 [28] and RFC2181 [53] state that DNS NS records must be configured at both parent and child zones. In this section, we evaluate the consistency of `NS` records at parents and children in the wild considering all second-level domains (SLDs) under `.com`, `.net`, and `.org`, on 2019-10-16. We also evaluate the records in the Root DNS zone on 2019-10-30. We make use of OpenINTEL, a large-scale DNS measurement platform [25]. OpenINTEL collects daily active measurements of over 65% of the global DNS namespace every day. For each SLD, we extract the sets of `NS` records from the parent and child authoritative servers, respectively indicated as P and C.

Table 3.1 shows the results of our comparative analysis. The first row shows the total number of SLDs for each TLD zone on the date considered. For the

	.com SLD	.org SLD	.net SLD	Root TLD	.com Ratio	.org Ratio	.net Ratio
Total domains	142,302,090	9,998,488	13,181,091	1528			
Unresponsive	19,860,226	949,137	1,663,403	0	14.0%	9.5%	12.6%
$P = C$	111,077,299	8,291,257	10,443,314	1476	78.0%	82.9%	79.2%
$P \neq C$	11,364,565	758,094	1,074,374	52	8.0%	7.6%	8.2%
$P \cap C = \emptyset$	6,594,680	418,269	548,718	16	58.0%	55.2%	51.0%
$IP(P) = IP(C)$	3,046,075	216,130	245,936	16	48.2%	53.9%	46.7%
$IP(P) \neq IP(C)$	3,265,171	184,885	280,988	0	51.8%	46.1%	53.3%
$IP(P) \cap IP(C) = \emptyset$	1,415,838	83,720	137,913	0	43.3%	45.3%	49.1%
$IP(P) \cap IP(C) \neq \emptyset$	1,849,333	101,165	143,075	0	56.7%	54.7%	51.9%
$P \cap C \neq \emptyset$	4,769,885	339,825	525,656	36	42.0%	44.8%	49.0%
$P \subset C$	3,506,090	236,257	369,442	18	73.5%	69.5%	70.2%
$P \supset C$	681,082	64,161	98,345	10	14.3%	18.9%	18.7%
Rest	582,713	39,407	57,869	8	12.2%	11.6%	11.1%

Table 3.1: Parent (P) and Child (C) NSSet consistency results. "IP" refers to `A` records of the NSSet of P and C.

three zones, ∼80% of SLDs have a consistent set of `NS` records at both the parent and the child zones. However, ∼8% of SLDs (∼ 13M) do not. For comparison, consider that 13M is almost as many domain names as some of the largest country-code TLDs (Germany's `.de`, one of the largest, has 16M SLDs [54]). The remaining 12% of domains are unresponsive to our queries. This could happen for different reasons, i.e. misconfigurations, failure, etc., not addressed in this work. We even see that 52 TLDs in the Root zone have inconsistent NSSets. Out of these, 26 are country-code TLDs (ccTLDs). We notified these ccTLD operators, in order to resolve these non-conforming setups, since they can have an adverse effect, among others, on load balancing and resilience against DDoS attacks.

Inconsistent NSSets Classification

We classify inconsistent domain names into four categories: the cases in which (i) the parent and child NSSets are *disjoint*, (ii) the parent NSSet is a *subset* of the child NSSet, (iii) the parent NSSet is a *superset* of the child NSSet and (iv) the parent and child NSSet have a non-empty intersection but do not match (ii) or (iii).

For case (i), we observe that 51–58% of domains have completely *disjoint NSSets* ($P \cap C = \emptyset$). Depending on if resolvers are parent or child-centric, in this case resolvers will trust different `NS` records.

Given the surprising results for disjoint sets, we investigate the IP addresses of the NS records (IP(P, C, lines 4-7 in Table 3.1).[1] We discover that in half of the cases, domains have disjoint NSSets that point to the same addresses, i.e., there is an inconsistency of names but addresses match. In the other half, there is inconsistency also in addresses. Of these, $\sim 45\%$ have completely disjoint sets of IP addresses, for the remaining 55% there is some sort of overlap.

Disjoint sets may increase the risk of human error even in the case of name servers resolving to the same IP address, since operators would need to maintain redundant information in the parent and child, thus introducing fragility in the DNS ecosystem. Disjoint sets also may lead to lame delegations [46], i.e., pointing resolvers to servers that may no longer be authoritative for the domain name.

Finally disjoint sets can be related to another malpractice: *CNAME configured on the Apex* [55]. However, further analysis shows that only a negligible percentage of cases are related to this malpractice.

Considering partially matching SLDs ($P \cap C \neq \emptyset$), we observe that 69–73% belong to case (ii), where the parent NSSet is a *subset* of the child NSSet. This may be intentional, e.g. an operator may want to first update the child and observe traffic shifts, and then later update the parent. Alternatively, operators may forget to update the delegation at the parent after updating the child.

Case (iii) where the parent NSSet forms a *superset* of the child NSSet ($P \supset C$) occurs in 14-18% of cases. This situation may introduce latency in the resolution process due to unresponsive name servers. Finally, the *Rest* category is case (iv), where the NSSets form neither a superset nor a subset, yet they have a non-empty intersection. Between 11-12% of SLDs fall in this category, and are susceptible to the range of operational issue highlighted for the previous categories.

Note that the OpenINTEL platform performs the measurements choosing *one* of the child authoritative nameservers. To verify how often sibling name servers have different configurations (child-child delegation inconsistency), we executed a measurement on a random sample of $\sim 1\%$ of `.org` domains (10k domains). The measurement suggests that $\sim 2\%$ of total parent-child delegation inconsistency cases also have child-child delegation inconsistencies, meaning that our results give a lower bound for the problem of parent-child mismatches. In fact, the OpenINTEL resolver could randomly choose a server configured correctly, while the others are not.

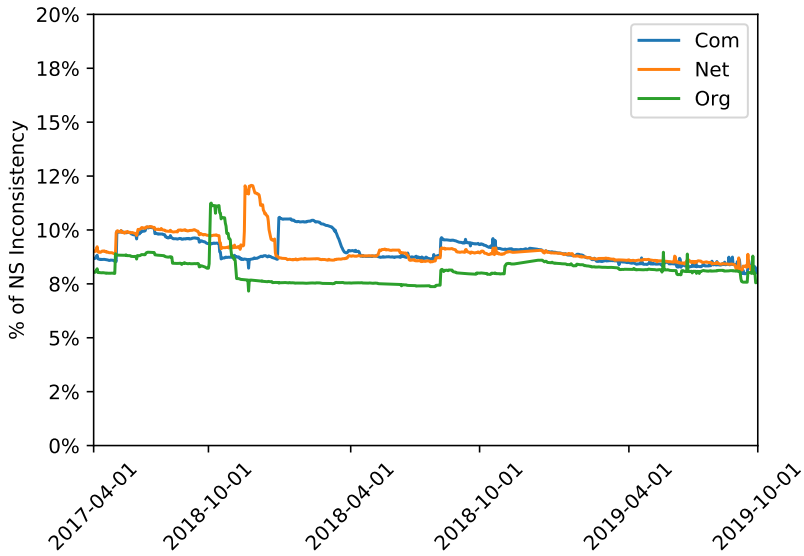


Figure 3.2: NS inconsistency ($P \neq C$) from 2017-04-01 until 2019-10-01

3.4.1 NS Inconsistency over Time

The results presented in Table 3.1 show NS inconsistency for a single day. It is also interesting to understand how this misconfiguration evolves over time. We analyzed NS inconsistency for the case $P \neq C$ over the two and a half year-period preceding the date of the analysis presented in Table 3.1. Figure 3.2 shows the results of this analysis. The fraction of domains affected by this misconfiguration remains similar over time. This result suggests that this NS inconsistency is a long-term misconfiguration in the DNS ecosystem.

3.5 Implications of NSSet Differences in the Wild

We observed that roughly 8% of studied domains have parent/child inconsistencies. In this section, we investigate the consequences of such inconsistencies, by emulating the four categories of NSSet mismatches. We configure parent and child authoritative servers in eight different configurations (Table 3.2), and

[1]This covers 96% of names with disjoint NSSets, the remaining 4% are indeterminate due to unresolvable names in the NSSets.

	Disjoint		**Subset**		**Superset**		**Rest**	
Experiment	Min-Off	Min-On	Min-Off	Min-On	Min-Off	Min-On	Min-Off	Min-On
Measurement ID	*23020789*	*23019715*	*23113087*	*23113622*	*23114128*	*23115432*	*23117852*	*23116481*
Frequency	600s							
Duration	2h							
Query	`A` $probeid-$timestamp.`marigliano.xyz` with 30 seconds TTL							
NSSet Parent	[ns1, ns3]		[ns1, ns3]		[ns1, ns2, ns3, ns4]		[ns1, ns2, ns3, ns4]	
NSSet Child	[ns2, ns4]		[ns1, ns2, ns3, ns4]		[ns2, ns4]		[ns2, ns4, ns5, ns6]	
TTL NS Parent	3600 s							
TTL NS Child	3600 s							
Date	20191003	20191003	20191025	20191025	20191025	20191026	20191027	20191027
Probes	9028	9031	8888	8883	8892	8879	8875	8875
VPs	15956	15950	15639	15657	15647	15611	15557	15586
Queries	190434	190333	184364	185706	186960	185015	182992	186472
Answers	178428	178416	169224	175200	175080	174804	174288	174504
From ns1, ns3	109661	175124	132179	169482	52233	83607	53944	84709
From ns2, ns4	65527	322	31753	1557	118835	86804	83100	85739
From ns5, ns6	N/A	N/A	N/A	N/A	N/A	N/A	31740	1545
fail	3240	2970	5292	4161	4012	4393	5504	2511

Table 3.2: Experiments to compare differents in Parent/Child NSSet

explore the consequences in terms of query load distribution. Our goal is to study these consequences in a controlled environment, where the authoritative name servers are in the same network. In the real world, the authoritative name servers are often distributed geographically and the query load can depend on external factors, e.g. nearest server, popularity of a domain in a certain region, etc.

We emulate an operator that (i) has full control over its child authoritative name servers and (ii) uses the same zone file on all authoritative name servers (zones are synchronized). We place all child authoritative servers in the same network, thus, having similar latencies. We expect this to result in querying resolvers distributing queries evenly among child authoritatives [56].

As vantage points, we use RIPE Atlas [57], measuring each unique resolver as seen from their probes physically distributed around the world (3.3k ASes). Many Atlas probes have multiple recursive resolvers, so we treat each combination of probe and unique recursive resolver as a vantage point (VP), since potentially each represents a different perspective. We therefore see about 15k VPs from about 9k Atlas probes, with the exact number varying by experiment due to small changes in probe and resolver availability.

3.5.1 Disjoint Parent and Child NSSet

We have configured our test domain (`marigliano.xyz`) for the disjoint NSSet experiment as shown in Figure 3.3. For this experiment, we set the NSSet at the

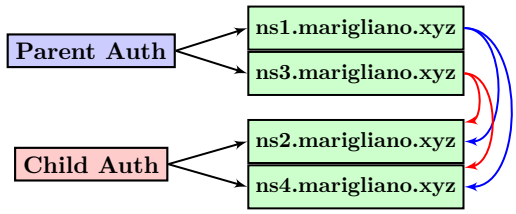


Figure 3.3: Disjoint NSSset Experiment for marigliano.xyz

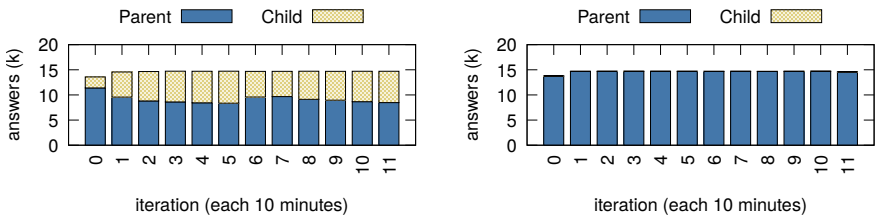
parent to [ns1, ns3].`marigliano.xyz`, while on the child authoritative servers, we set the NSSet to [ns2, ns4].`marigliano.xyz` (Table 3.2).

Zone files: we then configure the zone files of [ns1–ns4] to answer NS queries with [ns2,ns4], if explicitly asked, i.e., the same records pointed to by the child authoritative servers. By doing that, we are able to single out resolvers that are *parent-centric*, since they will only contact [ns1,ns3].

As vantage points, we use ∼9k Atlas probes, and configure them to send `A` queries through each of their resolvers for $probeid-$timestamp.`marigliano.xyz`, which encodes the unique Atlas probe ID and query timestamp, thus avoiding queries of multiple probes interfering with each other. We also set the TTL value of the record to 30 s, and probe every 600 s, so resolver caches are expected to be empty for each round of measurements [20].

Our goal is to determine, *indirectly*, which NS records were used to answer the queries. To do that, we configure [ns1,ns3] to answer our `A` queries with the IP `42.42.42.42`, and [ns2,ns4] with the IP `43.43.43.43`. We use this approach instead of inspecting the query log on the server-side to speed up parsing and to avoid duplicated detection.

Figure 3.4a shows the results of the experiment. In round 0 of the measurements, we have a warm-up phase of RIPE Atlas probes, where not all the probes participate. Furthermore, we expect resolvers to have a cold cache and to use the NSSet provided by the parent. As the figure shows, this is mostly the case although 253 unique resolver IPs (different probes can share the same resolver) do contact the child name servers. This can be either due to them sending explicit NS queries (and thus learning about [ns2,ns4]) or because some probes share upstream caches. In subsequent rounds, we expect more traffic to go to the child name servers [ns2, ns4]. This is because resolvers learn about the child delegation from the "authority section" included in the response to the `A` query to ns1 or ns3. According to RFC2181 resolvers may prefer this information over the delegation provided by the parent. Indeed, in rounds [1–11] we see traffic also going to the child name servers. However, not all traffic goes to



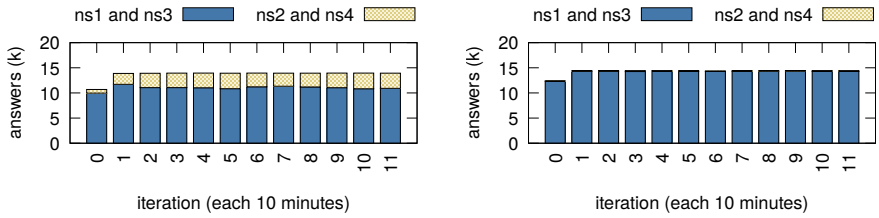
(a) Results for normal responses (b) Results with minimal responses

Figure 3.4: Disjoint NSSet experiments

servers in the child NSSet, because not all resolvers trust data from the "authority section" due to mitigations against the so-called Kaminsky attack [58]. A key takeaway of this experiment is that domain owners may mistakenly assume traffic to go to the name servers in the child NSSet if they change it, whereas for this change to be effective, they must also update the parent NSSet.

The situation is even worse in our second experiment. Here, we configure [ns1–ns4] to answer with *minimal responses*, which prevents these servers from including "extra" records in the authority and additional sections of DNS answers. This means we do not expect resolvers to learn about the existence of [ns2,ns4] at all, since they are no longer present in the "authority section" of responses to the `A` queries. Only if resolvers perform explicit `NS` queries will they learn about [ns2,ns4]. As Figure 3.4b shows, as expected, almost all resolvers exclusively send their queries to the name servers in the NSSet of the parent. Only about 40 vantage points receive data from the name servers in the child NSSet, indicating their resolvers likely performed explicit `NS` queries. Authoritative name servers are increasingly configured to return minimal responses to dampen the effect of DNS amplification attacks, especially for DNSSEC-signed domains [59]. A key takeaway from this experiment is with this configuration becoming more and more prevalent, it becomes even more important to keep parent and child NSSets correctly synchronized.

Real-world case: On 2019-10-30, we notified India's `.in`, given they had `ns[1-6].neustar.in` as NS records at the parent, and `[ns1-ns6].registry.in` at the child. However, altogether, both NSSets pointed to the same `A/AAAA` records and, as such, resolvers ended up reaching the same machines. After our notification, `.in` fixed this inconsistency on 2019-11-02 (we analyzed DNS OARC's root zone file repository [60]). Besides `.in`, 15 other internationalized ccTLDs run by India had the same issue with their NSset, and were also fixed.



(a) Results for normal responses (b) Results with minimal responses

Figure 3.5: Subset NS sets experiments

3.5.2 Parent NSSet is a Subset of Child

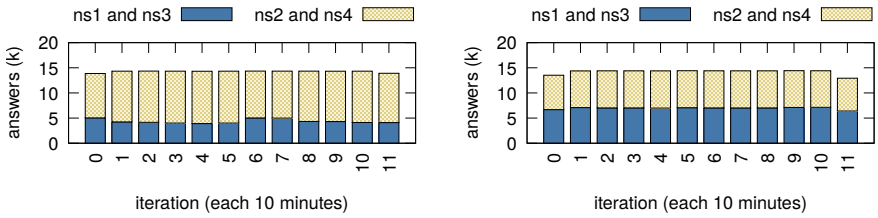
Recall from Table 3.1 that the majority (69-73%) of cases in which parent and child NSSets differ fall into the category where the child NSSets contains one or more additional `NS` records not present in the parent NSSet. A common reason to add additional `NS` records is to spread load over more name servers, and we assume this to be one of the reasons for this common misconfiguration.

We set up experiments to determine the consequences on query distribution if you have this setup. In other words: how many queries will eventually be answered by the extra `NS` record? We configure our test domain with [ns1, ns3] at the parent and [ns1, ns2, ns3, ns4] at the child. Like in the previous section, we configure [ns1, ns3] to give a different response to the `A` queries sent by the Atlas probes than [ns2, ns4], so we learn how many queries were answered by the name servers that are only in the child NSSet.

Figure 3.5a shows the results. Similarly to the results shown in subsection 3.5.1, most resolvers will use the `NS` records provided by the parent. Given that the child NSSet includes the NSSet at the parent, we see that the extra name servers receive only ∼24% of the queries. If in addition we configure the name servers to return minimal responses, we see that, just as in subsection 3.5.1 virtually no resolvers contact the extra name servers in the child NSSet (Figure 3.5b). A key takeaway from these two experiments is that the, perhaps, expected even load distribution domain owners are hoping to see will not occur if only the child NSSet is updated. This again underlines the importance of keeping parent and child in sync.

Real-world Case: `att.com`:

A real-world example that demonstrates that this type of misconfiguration also occurs for prominent domains is the case of `att.com`. We discovered that AT&T's main domain `att.com` had a parent NSSet containing `[ns1...ns3].attdns.com`, whereas the child had `[ns1...ns4].attdns.com`. We



(a) Results for normal responses (b) Results with minimal responses

Figure 3.6: Superset NS sets experiments

notified AT&T of this misconfiguration and on 2019-10-24 the issue was resolved when the fourth name server (`ns4.attdns.com`) was also added to the parent.

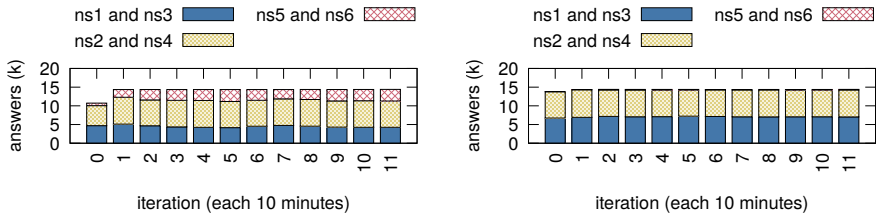
3.5.3 Parent NSSet is a Superset of Child

Roughly 14-18% of domain names that have different NSSets at parent and child have, one or more extra `NS` records at the *parent* ($P \supset C$ in Table 3.1). This could be due to operators forgetting to remove name servers that are no longer in use at the parent, but also the reverse case of the previous section in which a new name server is added at the parent but not added at the child.

To investigate the consequences of this for resolvers, we carry out experiments using Atlas VPs, setting four `NS` records at the parent ([ns1, ns2, ns3, ns4], as in Table 3.2) and only two at the child ([ns2, ns4]). Our goal is to identify the ratio of queries answered by the extra `NS` records at the parent.

Figure 3.6a shows the results for the experiment. As can be seen, the servers listed both in the parent and in the child ([ns2,ns4]) answer, on average, 68% of the queries. In case minimal responses are configured (Figure 3.6b), we see the queries being distributed evenly among the NS records in the parent. Consequently, having authoritative servers include an authority section in their answer to the `A` queries seems to cause *some* resolvers to prefer the child NSSet over the one in the parent. For example, Atlas VP (21448, 129.13.64.5) distributes queries only among ns2 and ns4, in the case of normal responses, instead it distributes queries among all name servers in case of minimal responses.

These measurements then confirm that including "authority data" in the authoritative server responses will cause *some* resolvers to prefer only the child authoritative servers.



(a) Results for normal responses (b) Results with minimal responses

Figure 3.7: Rest NS sets experiments

3.5.4 Mixed NSSets (Rest)

We have shown in Table 3.1 that in 11% of cases, the NSSet of the parent and child do not have a subset/superset relationship. Instead, some elements are present in both, but both parent and child have at least one NS that is not available in the other. To simulate this scenario, as shown in Table 3.2, we set four NS records at the parent: [*ns1*,ns2,*ns3*,ns4]. Then, at the child, we set [ns2,ns4,*ns5*,*ns6*], where the highlighted names show the ones not shared.

Figure 3.7a shows the experiment results. We see that [ns2,ns4], which are listed at both parent and child receive most queries. Then, records set only at the parent ([ns1,ns3]) are second to receive more queries. Finally, records set only at the child ([ns5,ns6]) receive the least amount of queries. In case of minimal responses (Figure 3.7b), the name servers only present at the child ([ns5,ns6]) receive virtually no traffic.

3.5.5 Discussion

Having inconsistent NSSets in parent and child authoritative servers impacts how queries are distributed among name servers, which plays an important role in DNS engineering. Overall, for all evaluated cases, queries will be unevenly distributed among authoritative servers – and the servers listed at the parent zone will receive more queries than then ones specified in the child.

3.6 Resolver Software Evaluation

The experiments carried out in section 3.4 evaluate DNS resolver behavior in the wild. Since we use RIPE Atlas, we do not know what resolver software is used, if probes use DNS forwarders, or what kind of cache policies they use. We, however, see the aggregated behavior among a large set of configurations.

	Bind	Unbound	Knot	PowerDNS	Windows-DNS
Ubuntu-18-04	9.11.3-1	1.6.7	2.1.1	4.1.1	N/A
Ubuntu-16.04	9.10.3-P4	1.5.8	1.0.0	4.0.0	N/A
CentOS 7	9.9.4	1.6.6	2.4.1	4.1.9	N/A
CentOS 6	9.8.2rc1	1.4.20	N/C	3.7.4	N/A
Source	9.14.0	1.9.0	N/C	4.1.9	N/A
Windows	N/C	N/C	N/C	N/C	2008r2, 2012, 2016, 2019

Table 3.3: O.S. and resolver versions evaluated (N/Available, N/Covered)

	(i) A Query	(ii) NS Query	(iii) A Query Then NS Query		(iv) NS Query Then A Query	
Query			First	Second	First	Second
Answer	C(A)	C(NS)	C(A)	C(NS)	C(NS)	C(A)
Cache	C(A); C(NS)	C(NS)	C(A); C(NS)	C(A); C(NS)	C(NS)	C(NS); C(A)
Minimal response enabled						
Answer	C(A)	C(NS)	C(A)	C(NS)	C(NS)	C(A)
Cache	C(A); P(NS)	C(NS)	C(A); P(NS)	C(A); C(NS)	C(NS)	C(NS); C(A)
Information provided by: C⇒ Child, P⇒ Parent						

Table 3.4: Expected Resolver Behavior

In this section, we focus on evaluating specific DNS resolver software instead, in a controlled environment, in order to understand how they behave towards DNS zones that are inconsistent with regards to their parent/child NSSet. Our goal is to identify which vendors conform to the standards. In particular, we pay attention as to whether resolvers follow RFC2181 [53], which specifies *how* resolvers should rank data in case of inconsistency: child authoritative data should be preferred.

We evaluate four popular DNS resolver implementations: *BIND* [61], *Unbound* [62], *Knot* [63], and *PowerDNS* [64]. We do this under popular Linux server distribution releases, using default packages and configurations. In addition, we evaluate resolvers shipped with various Windows server releases. Table 3.3 shows which vendors and versions we evaluate.

Experiments

We configure the authoritative name servers for our test domain (`marigliano.xyz`) as a *disjoint* NSSet, as in subsection 3.5.1. We configure the parent zone with [ns1,ns3].`marigliano.xyz`, and the child with [ns2,ns4].`marigliano.xyz`

Each experiment includes the four tests described in Table 3.4 (i–iv), in which we vary query types and query sequences. In (i), we ask the resolver for an `A` record of a subdomain in our test zone. In test (ii), we ask for the `NS`

record of the zone. In (iii) we send first an `A` query followed by an `NS` query, to understand if resolvers use non-authoritative cached `NS` information to answer to the following query violating §5.4.1 of RFC2181 [53]. In (iv) we invert this order to understand if authoritative records are overwritten by non-authoritative ones in the cache.

We dump the cache of the resolver after each query, and show which records are in cache and received by our client (we clear the cache after each query). Table 3.4 shows the expected NS usage by the resolvers, if they conform to the RFCs.

3.6.1 Results

We evaluate five resolver vendors and multiple versions. In total, we found that out of 22 resolvers/vendors evaluated, 13 conform to the RFCs. Next, we report the non-conforming resolver vendors/versions.

For experiment (i), in which we query for `A` records, we found that BIND packaged for Ubuntu did not conform to the standards: it caches only information from the parent and does not override it with information from the authoritative section provided by the child (which comes as additional section). This, in turn, could explain part of results of parent centricity observed in section 3.5.

For experiment (i) and (iii), if we compile the latest *BIND* from source it also does not behave as expected: it sends the parent an explicit `NS` query before performing the `A` query. This is not a bad behavior, i.e., it does not violate RFCs, instead it tries to retrieve more authoritative information. However, either if the name server information retrieved and used in the following query is the one provided by the child, *BIND* caches the data from the parent. This behavior of BIND could be one explanation of the small number of child-centric resolvers shown in section 3.5 with Minimal Responses.

For experiment (iii), *PowerDNS* packaged for `CentOS 6` and `Ubuntu Xenial`, and Windows (all) use the cached non-authoritative information to answer the *NS* query in the test, not conforming to RFC2181.

PowerDNS Notification

We reached out to the developers of *PowerDNS*, who have confirmed the behavior. They do not maintain older versions anymore and the fix will not be backported due to the low severity of the problem. Our suggestion to the package maintainers of the distributions is to update their packages to a newer version of the software.

3.7 Concluding Remarks

Given a domain name, its NSSet in the parent and child DNS zones should be consistent. This requirement is mandated both by RFC1034 [28] and common sense. In this chapter we show, across the `.com`, `.net` and `.org` zones (50% of the DNS namespace), that roughly 8% (13M) domains do not conform to this principle. We also show that DNS resolvers in the wild differ in behavior in returning information from the parent or child.

Inconsistencies in parent and child NSSets have consequences for the operation of the DNS, such as improper load balancing among the name servers, increased resolution latency and unresponsive name servers. We strongly advise operators to verify their zones and follow RFC1034. To automate this process, we advise zone operators to consider supporting CSYNC DNS records (RFC7477) or other automated consistency checks, so the synchronization can be done in an automated fashion.

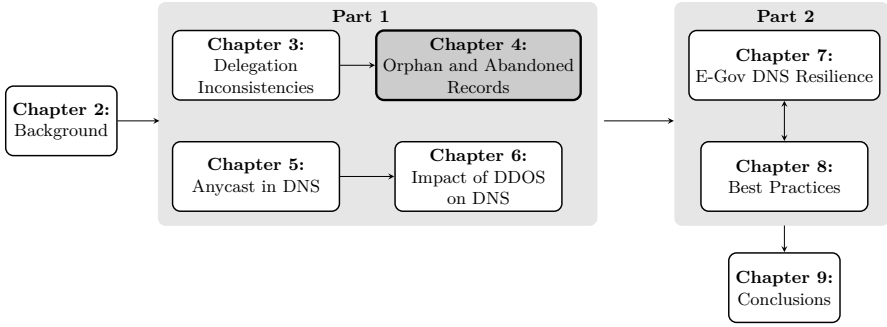
Finally, we also recommend that resolver vendors conform to the authoritative information ranking in RFC2181 (taking into account the recommendations to mitigate the Kaminsky attack as specified in RFC5452), and when possible, to *explicitly* ask for the child's NS records, similarly to what is done in DNSSEC, where signed records are only available at the child (section 3.6).

Concurrently with this research, the DNSOP Working Group (WG) initiated a new Internet draft, "Delegation Revalidation by DNS Resolvers", to address the issue of DNS delegation inconsistency [2]. At the request of one of the draft's authors, we made available a similar controlled and experimental setup as used in the research through the tool *SuperDNS.nl - DNS Misconfigurations in a Controlled Environment* [3], allowing developers to test their resolvers.

[2] `https://tools.ietf.org/html/draft-huque-dnsop-ns-revalidation`
[3] `https://superdns.nl/`

CHAPTER 4

Orphan and Abandoned Records



The existence of different entities responsible for the DNS infrastructure creates complexity in managing DNS information. In this complexity, misconfigurations and errors can occur, with an impact on the overall security and reachability of the DNS. In the previous chapter, we saw that if operators do not keep the different delegations synchronized, resilience problems can arise. In further investigation of the misconfigurations affecting DNS resilience, in this chapter we will focus on records left over due to mismanagement and lack of update of DNS zone files. Our results shows a widespread misconfiguration in the DNS ecosystem (RQ-1) with a severe impact on domain operations, increasing their hijacking risk (RQ-2). We notified the affected TLDs by providing an effective solution to mitigate the hijacking risk. The study discussed in this chapter was performed in early 2020, published in an academic conference [31] and presented at DNS-OARC 33.

4.1 Introduction

Due to the complexity of managing DNS information in the vast DNS ecosystem, misconfigurations and errors are prone to occur, affecting the overall security and reachability of the DNS. The administration of a domain involves multiple parties, including the registry, registrant, and registrar, and is subject to misconfigurations that can result in reachability and security issues. In the previous chapter, we studied this problem in depth to identify possible delegation misconfigurations and inconsistencies arising from this communication.

In this chapter, we will analyze a specific misconfiguration, defined as *orphan records* [30]. Orphan records are a leftover of a domain that has expired, and should have been removed by the registry or by the registrar together with the expired domain they belong to. Orphan records form a security risk as unwitting third party domains may still point to these orphan records in their delegation. An attacker can easily hijack domains referring to orphan records by re-registering the domain they belong to. By doing so, the attacker gains full control of the domain and can potentially redirect all traffic for malicious purposes. In this chapter we reproduce and extend an analysis of this phenomenon performed by Kalafut *et al.* [30] in 2010. Our goal is to understand: *A decade after the original analysis, what does the orphan records phenomenon look like?* Compared to [30], we characterize the *orphan records* phenomenon through a significantly larger dataset of ∼2k TLDs and over a wider time window of 25 months. We also introduce a related type of misconfiguration, which has not been considered before. We refer to it as *abandoned records*. *abandoned records* could be seen as predecessors of new orphan records.

The remainder of the chapter is organized as follows. In section 4.2 we provide background on the misconfigurations, on the reasons why these misconfigurations arise and on the previous analysis by Kalafut *et al.* [30]. We describe our dataset and analysis methodology in section 4.4. Section 4.5 presents our results with an analysis of the *orphan* and *abandoned records* phenomenon. Finally, we provide concluding remarks in section 4.8.

4.2 Background

Zone files and Glue Records

Recall from chapter 2 that a top-level domain (TLD) is a special type of zone that typically only has one task: to delegate second-level domains. This delegation uses `NS` records that identify the name server for a domain. If the `NS` record for a domain points to a record that is inside the domain (called *in-bailiwick*),

that name is included in the zone as a *glue record* to enable the resolution process to continue.

Consider for example:

```
example.com.  86400  IN  NS  ns1.example.com.
```

To resolve `example.com.`, we need to resolve `ns1.example.com.`, but this implies resolving the `example.com.` delegation. Defining the `A/AAAA` glue record for `ns1.example.com.` in the parent zone file breaks this circular dependency and allows the domain to be resolved. *Glue records* are usually the only `A/AAAA` records admitted in TLD *zone files*. A notable exception to this condition represented by the `.de` *zone* is explained in § 4.6.5.

Orphan and Abandoned Records

Glue records are supposed to be removed after a delegation is removed or changed. Earlier work indicates, however, that this does not always happen in practice [30]. In this chapter, we define an **orphan record** as a former glue record for which the related domain no longer exists in the zone (the delegation has been removed). We also define an **abandoned record** as a former glue record for which the related domain still exists in the zone but the delegation no longer requires that glue record. Under normal operation, abandoned records do not show up in the DNS resolution, as there is no longer a relation with the domain the record served. Abandoned records show up in the additional section only when they are referred to by a delegation of other domains of the zone. However, it is still questionable if they should remain in a TLD zone file at all. Finally, we define **junk records**, as the union of orphan and abandoned records.

4.3 Related Work

Kalafut *et al.* [30] characterized the problem of orphan records in terms of their spread, usage, lifetime and hosted resources, for a 31-day timeframe. The authors considered zone files for 6 TLDs as well as malware URL feeds. We reproduce part of their results, but for a significantly longer, 25-month timeframe, enabling long-term characterization of the junk records phenomenon. Moreover, as 10 years have passed since Kalafut's original study and we focus on a recent period, we can analyze how this problem has evolved over the past decade. Where possible, we run our analysis on the same zones as in [30], but also extend the analysis to other zones available to us. Liang *et al.* [65] proposed

example.com.	86400	IN	NS	ns.external.org.
ns1.example.com.	3600	IN	A	1.2.3.4
ns1.expired1.com.	3600	IN	A	3.2.5.4
ns1.expired2.com.	3600	IN	A	8.4.5.6
active.com.	86400	IN	NS	ns1.expired2.com.
good.com.	86400	IN	NS	ns1.good.com.
ns1.good.com.	3600	IN	A	1.2.3.5

Table 4.1: Example `.com` Zone File
■ Algorithm 1 (O) ■ Algorithm 2 (A)

TLD	Coverage	Start-Date	End-Date	TLD	Coverage	Start-Date	End-Date
info	98.3%	2017-04-01	2019-04-30	ca	94.3%	2017-04-01	2019-04-30
mobi	96.2%	2017-04-01	2019-04-30	fi	97.5%	2017-04-01	2019-04-30
asia	94.4%	2018-11-20	2019-04-30	aero	94.4%	2018-11-20	2019-04-30
org	99.9%	2017-04-01	2019-04-30	biz	93.8%	2018-11-20	2019-04-30
com	96.0%	2017-04-01	2019-04-30	name	94.4%	2018-11-20	2019-04-30
net	98.6%	2017-04-01	2019-04-30	nu	99.3%	2017-04-01	2019-04-30
us	99.4%	2018-11-19	2019-04-30	se	99.3%	2017-04-01	2019-04-30
ru	99.1%	2017-06-17	2019-04-30	CZDS	99.9%	2017-04-01	2019-04-30

Table 4.2: Overview of datasets used in this work

a method for keeping DNS records locked in the cache of open DNS resolvers after the domain expires. The authors defined these records as *ghosts* and prove that by performing queries against open resolvers and by crafting an ad-hoc response in the controlled authoritative nameserver, it was possible to refresh the *TTL* value for the record in the cache of the open resolver even if the domain no longer exists in the parent zone. While their work does not specifically focus on junk records, it indirectly refers to generic expired domains in the parent zone.

4.4 Methodology and Dataset

Methodology

We developed two algorithms to respectively identify orphan and abandoned records inside zone files. These algorithms rely on the principle that in the zone file the only `A` records available are glue records[1].

[1]So as to reproduce the work in [30], we did not consider `AAAA` RRs.

Algorithm 1 identifies orphan records and it is similar to the one described in [30]. The algorithm first collects all the domains in `A` records available inside the zone file. Then it trims the domains to the second level domain (SLD). Finally, it looks for SLDs that do not have any associated `NS` record.

Algorithm 2 identifies abandoned records. The algorithm collects the list of domains in the `A` records available in the zone file. Like Algorithm 1, it trims the domain to the SLD and looks for the SLDs for which the `NS` records do not point to the extracted `A` records.

Table 4.1 provides an example of records retrieved by the two algorithms. Algorithm 1 identifies `ns1.expired1.com.` and `ns1.expired2.com.` as orphans since no `NS` records exist for `expired1.com.` or `expired2.com.`. Algorithm 2 marks `ns1.example.com.` as abandoned (a delegation for `example.com.` exists, but points elsewhere).

Dataset

The TLDs we consider for this study are `.aero`, `.asia`, `.biz`, `.ca`, `.com`, `.fi`, `.info`, `.mobi`, `.name`, `.net`, `.nu`, `.org`, `.ru`, `.se` and `.us`. We also include 1184 new gTLDs introduced by ICANN, which we collectively refer to as "CZDS"[2]. We make use of the daily zone file collection of OpenINTEL. The combined zones cover a period of 25 months, from April 2017 to May 2019 (760 days). This set of zone files contains per day on average 3,283,404 unique `A` records, 199,249,769 unique domains, and 1,317,987 unique in-bailiwick domains. A zone file might occasionally not be collected (e.g., due to contract renewal processes). This happens at maximum for 5.6% of the measurement period (42 days for `.asia`). This means that we can consider our results a lower bound for the orphan and abandoned records problem. Table 4.2 lists the effective start and end dates for each TLD in our dataset, and the percentage of days covered in the range. We used Apache Spark [66], an open source cluster computing framework, to perform our analysis.

4.5 Characterizing Orphan and Abandoned

4.5.1 Orphan Record Distribution

Kalafut *et al.* [30] identified `.info` as the TLD with the highest percentage of orphan records in the period between 2009-04-01 and 2009-05-01. Our analysis shows that 10 years later, the number of orphan records in this TLD is still rising (Table 4.4). Of an average of 169,946 `A` records per day, in the studied

[2]The ICANN system that regulates access to the zones for these domains is called the Centralized Zone Data Service (CZDS).

period, an average 24.9% of these are orphan records, with a maximum of 36% and a minimum of 17.3%. Comparing these results to [30] we find an increase of 6.1% on average and of 17.2% as a maximum.

The `.mobi` TLD shows a similar trend, with an average of 6,855 `A` records per day. The mean percentage of orphan records is 22.7%, with a peak of 37.5%. Compared to [30], the number of orphan records tripled, with an increase of 12% in the total number of records, whereas the total number of records grew 1.71×. We note that the number of orphan records for `.mobi` has decreased towards the end of the period we analysed, but found no evidence that this is due to a targeted cleanup action.

TLD	$\frac{\#Gluerecords}{day}$	$\frac{\#Orphan}{day}$	Prev Orphan	Prev #Glue	$\frac{\#Abandoned}{day}$
.info	169,946	43,687	18.8%	139,126	70,180
.mobi	6,855	1,602	10.7%	4,062	2,972
.asia	6,122	1,140	7.5%	1,313	3,294
.org	364,568	21,929	3.7%	206,513	234,256
.com	1,873,668	0	0.4%	1,566,392	602,641
.net	303,387	0	0.2%	331,896	55,327
.us	26,042	1,869	3,931*	N/A	1,577
.ru	79,492	54	1,801*	N/A	2,998
.ca	23,537	980	1,368*	N/A	3,467
.fi	3,908	0	N/A	N/A	0
.aero	626	22	N/A	N/A	342
.biz	22,958	6	N/A	N/A	2,113
.name	1,820	50	N/A	N/A	42
.nu	2,184	0	N/A	N/A	0
.se	20,053	48	N/A	N/A	0
CZDS	387,897	17,144	N/A	N/A	84,805

*Kalafut *et al.* report the number of orphans instead of the percentage for these TLDs due to lack of access to the zone files [30].

Table 4.3: Orphan and abandoned records for each TLD over *2017-04-01 – 2019-04-30*.
The absolute numbers shown are daily averages.

A remarkable difference with [30] is that `.com` and `.net` no longer contain any orphan records. We discuss the case in § 4.6.3.

For `.asia` and `.org`, we find more records compared to [30] with an almost constant trend. For `.us`, `.ca` and `.ru` instead, we identify fewer records compared to [30], which already underestimated the number of records for these TLDs, as they did not have access to the respective zone files. This means that these TLDs improved their management of glue records.

4.5.2 Abandoned Record Distribution

The TLD with the highest percentage of abandoned records is `.org`, with a mean of 64.2% abandoned records. The `.info` TLD exhibits a lower percentage (41.8%). Considering the percentage of orphan records and abandoned records together, `.mobi`, `.info`, `.asia` and `.org` show a percentage of junk records around 65%, casting doubts on the management of these zones. Also `.com` and `.net` show a relevant number of abandoned records. For `.fi`, `.nu`, and `.se`, we do not find any abandoned records.

	Orphan Percentage			Abandoned Percentage			Sum
TLD	Min	Max	Mean	Min	Max	Mean	
.info	17.3%	36.0%	24.9%	36.5%	49.0%	41.8%	66.7%
.mobi	5.5%	37.5%	22.7%	36.0%	54.2%	44.0%	66.7%
.asia	17.8%	19.9%	18.6%	52.2%	55.5%	53.8%	72.4%
.org	4.4%	7.5%	6.0%	62.8%	65.5%	64.2%	70.2%
.com	0.0%	0.0%	0.0%	31.4%	33.0%	32.2%	32.2%
.net	0.0%	0.0%	0.0%	9.5%	19.6%	18.2%	18.2%
.us	6.8%	8.0%	7.2%	5.1%	8.5%	6.1%	13.3%
.ru	0.0%	0.1%	0.1%	2.0%	4.1%	3.8%	3.8%
.ca	3.9%	4.5%	4.2%	13.9%	15.7%	14.7%	18.9%
.fi	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
.aero	3.0%	4.3%	3.6%	53.5%	55.7%	54.7%	58.3%
.biz	0.0%	0.0%	0.0%	7.5%	12.5%	9.3%	9.3%
.name	2.2%	3.0%	2.7%	1.8%	2.8%	2.3%	5.0%
.nu	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
.se	0.2%	0.3%	0.2%	0.0%	0.0%	0.0%	0.2%
CZDS	0.5%	23.9%	5.0%	3.5%	31.5%	22.2%	27.2%

Table 4.4: Percentage of orphan and abandoned records for each TLD over *2017-04-01 – 2019-04-30*

4.5.3 IP Address and Domain Distribution

We now investigate how many domains and how many IP addresses are related to junk records. The distribution of the IP addresses related to the orphan and abandoned records shows 38% of records that point to a single IP address for orphans and 67% for abandoned records. Moreover, 88% of orphans and 92% of abandoned records refer to a single or to two IP addresses, with an average of 2.32 and 1.83 orphan and abandoned records, respectively, per IP. Compared to [30], the number of orphans per IP decreased (from an average of 3.2 orphans per IP in [30]). Since `.com` and `.net` dominated the number of orphans in [30], we assume that the cleaning of these zones is reflected in this decrease. There are also some peculiar cases. For example, in `.info`, 1,754 orphan records point to Cloudflare's public resolver `1.1.1.1`. This is the result of a misconfiguration of `NS` resolvers (circular dependency), which causes unreachability of the domain.

We also analyze domain distributions. For orphan records, in 94% of the cases, we find two orphan records for a single SLD. This result is consistent with the common configuration practice of DNS, in which administrators set up two authoritative nameservers, thus two `A` glue records for a domain. In 7% of the cases, we find one orphan record for a single SLD. For abandoned records, in 85% of the cases, we find two abandoned records for a single SLD, and in 21% of the cases we find one.

4.5.4 Lifetime of Records

Figure 4.1 shows the lifetime CDF of orphan and abandoned records. Lifetimes are the uninterrupted time segments during which we consider glue records to be orphaned or abandoned in our analysis. The plot contains only data for `.info`, `.mobi`, `.org`, `.ca`, `.se`, and CZDS. We do not include other TLDs since their zone collection started later in time in OpenINTEL. However, the shape of the CDF is similar across the different TLDs, with some exceptions that we explain later. The number of orphan records that lived at most 1 day is 19%, which is higher compared to the 12% found in [30]. Also, [30] indicates that only 2% of the orphan records last their entire measurement duration (31 days). In our case, 4% of orphan records survived for more than 760 days (the time frame of our analysis). The results for abandoned records are higher than for orphans: only 8% of records lived one day or less and 28% of records lived more than 760 days. Interestingly, when we look at individual TLDs, we find this difference between orphan and abandoned lifetime mainly present in `.org` and the new gTLDs, where abandoned records lived longer than orphan records.

The CDF in Figure 4.1 shows that ∼4% (21,640) of all the orphan records we find (541,002), persisted for more than 760 days (our observation period). These records were orphans during all the period of our analysis and represent

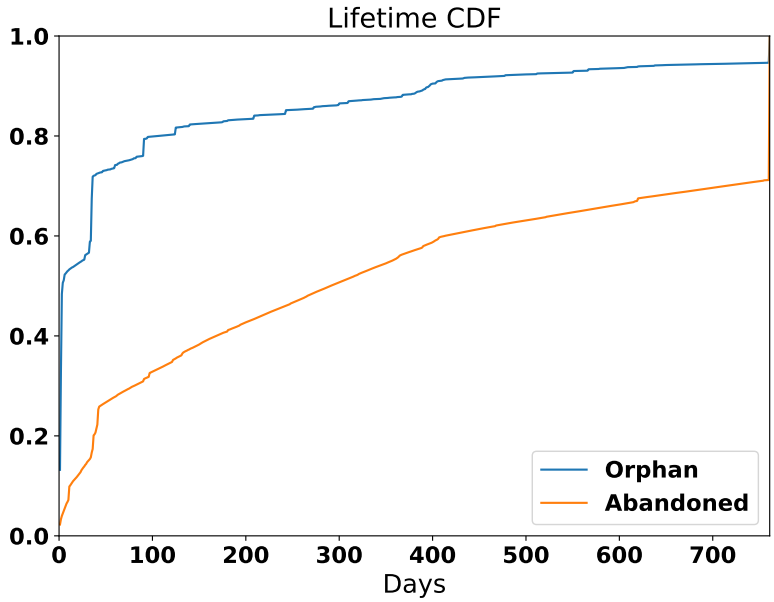


Figure 4.1: Lifetime of orphan and abandoned records

a significant fraction of the orphan records we observed daily (Table 4.3). In a similar way, for abandoned records, we discover 496,384 persistent records that survived more than 760 days. These results confirm that junk records are *a long-term misconfiguration*, which persistently affects the TLD zones.

4.5.5 NS References to Orphans

As we discussed in the introduction, `NS` records for other domains may refer to orphan records. This creates a serious vulnerability: an attacker can register the domain of the orphan record, thus redirecting all queries to a malicious authoritative name server under their control. By controlling the orphan resource records in a malicious authoritative name server, an attacker can divert traffic to any malicious destination (NS hijacking). This hijacking affects all domains that define as an authoritative name server the hijacked orphan record.

We find 39,683 NS records that refer to orphan records, either in the same zone, or in other zones. In Table 4.5, we show the reference matrix for each TLD. For typographical reasons we exclude empty and irrelevant columns. The

Orphan records											
Ref by	ru	org	asia	CZDS	se	name	mobi	us	ca	info	Total
aero	0	0	0	0	0	0	0	36	4	0	40
asia	0	0	26	23	0	0	0	0	0	11	60
biz	0	17	12	93	4	0	2	62	101	96	387
ca	0	16	0	5	0	0	0	2	10,320	24	10367
com	0	1,337	41	276	34	2	19	3,923	6,223	1,111	12966
CZDS	3	194	11	404	0	0	1	44	208	292	1157
fi	0	0	0	0	0	0	0	0	0	38	38
info	0	101	5	126	4	1	7	291	219	453	1207
mobi	0	8	4	99	0	0	30	7	149	5	302
name	0	0	0	28	2	68	0	0	9	0	107
net	0	277	10	139	19	0	3	1,566	874	136	3024
nu	0	14	0	0	18	0	0	0	1	0	33
org	0	288	6	133	14	2	3	4,695	1,038	160	6339
ru	43	26	0	3	0	0	24	0	0	14	110
se	0	0	0	0	110	0	0	0	0	2	112
us	0	33	5	22	0	0	0	3,287	67	20	3434
Total	46	2,311	120	1,351	205	73	89	13,913	19,213	2,362	39,683

Table 4.5: NS records pointing to orphan records

most referenced orphan records are in `.ca`, and are referenced ∼10k times in `.ca` and ∼6k times in `.com`. This matrix also helps us understand that the removal of an orphan record could have an impact on other domains in other TLDs, and for this reason removal should be analyzed carefully.

4.5.6 Orphan DNSSEC-signed Records

Another issue with orphan records concerns DNSSEC-signed zones featured by most TLDs. Normally, glue records are not signed, since the TLD is not authoritative for the domain (the name servers to which the domain is delegated are). If the domain is deregistered, however, the glue records are implicitly (and unintentionally) promoted by the registry to records that are part of the TLD zone, and will be DNSSEC-signed. This behaviour results in signing and providing warranty about the authenticity of junk records, increasing the zone file size and raising doubts about the legitimacy of these signatures.

4.6 Origin of Orphan and Abandoned Records

4.6.1 Relationship between Orphan and Abandoned Records

TLD	Dist(O)	Dist(O)*	Dist(A)	$A \implies O$	$O \implies A$
info	127,014	103,650	169,702	44,938	3,769
mobi	4395	4,037	6,575	2,047	81
asia	1679	1,679	3,857	475	27
org	82,231	66,814	369,330	52,559	5,763
CZDS	320,158	297,219	357,123	31,346	1,880
Total	535,477	473,399	906,587	131,365	11,520

*Orphan records with with *birth date* $\geq$ *2017-04-01*

Table 4.6: Relationship between (O)rphan and (A)bandoned

Orphan and abandoned records are both useless glue records left in the zone file of TLDs. This raises the question if an abandoned record can become an orphan record or vice-versa. An abandoned record that becomes an orphan could be a sign of poor management practices at the registry or registrant, who do not clean the zone file. Moreover, it could help us to infer records that will likely be orphan records after the expiration of the related domain. An orphan record that becomes an abandoned record indicates that someone registered the domain related to the orphan records. This could happen for legitimate reasons (e.g., people not being aware of the orphaned status of the domain), or for malicious purposes (e.g., to take control of the orphan record). Table 4.6 shows the results of our analysis.

Abandoned $\implies$ *Orphan* Our dataset shows a total of 535,477 distinct orphan records, of which 473,399 come into being in our window of analysis, i.e., records with *birth date* $\geq$ *2017-04-01*. We focus on this category to investigate if there is any relation with abandoned records. We find indeed that roughly 27.7% of orphan records were previously abandoned records. This strong correlation confirms that abandoned records are likely to become orphans at a later point in time.

Orphan $\implies$ *Abandoned* We do not find many records morphing from orphan to abandoned. In fact only 11,520 of 535,477 orphan records in our dataset became abandoned (2.1%). A likely explanation is that orphan records get registered again. This can occur without the registrant being aware of the

orphan status, meaning that unnecessary (junk) records related to a domain might be present in the zone without the registrant noticing it. However, if the registrant is aware of the orphan status of a record, then we might have witnessed a hijack (see § 4.5.5).

4.6.2 WHOIS Orphan and Abandoned

We performed a WHOIS information dump of orphan and abandoned records on *2019-12-03* using SpiderWho [67] for parallel lookup. We analyzed the WHOIS information to understand: (i) if the domains are in the WHOIS database (i.e. domain registered), (ii) if these misconfigurations belong to a specific registrar, or (iii) to a special administrative status of the domains (e.g., locked, expired, etc.).

Out of 54,421 domains belonging to orphan records, 29,418 (54%) have no associated WHOIS information at all in the WHOIS server of the related TLD, meaning they were potentially available for registration. Of the remaining 25,003 domains, 19,491 were registered through Namecheap, 2,201 domains were in the clientHold state, 294 are inactive, and 290 in pending delete state. Interestingly, when we tried to recreate an orphan through the Namecheap web interface – in order to understand if the registrar behaves in a bad way (i.e., not deleting orphan records) – we were unsuccessful.

For abandoned records, by definition, all related domains were registered. We found only few records not available in the WHOIS database (related to failure in querying or parsing). Of 165,492 domains, 34,334 are registered with GoDaddy and 20,512 with Namecheap. We tried to recreate an abandoned record through Namecheap and GoDaddy without success. We suspect that it is possible to create these records through API calls for managing the domains, which we did not verify as a premium account is required.

4.6.3 The Case of `.com` and `.net`

A difference in our results compared to Kalafut *et al.*, is that `.com` and `.net` no longer contain any orphan records. Our dataset does not allow us to pinpoint the point in time when these records disappeared as this occurred prior to *2017-04-01*. For this reason we use the archives of `.com` and `.net` zone data provided by DNS-OARC [68] in order to detect the date of occurrence. Figure 4.2 shows the temporal evolution of orphan records between May 2009 and December 2010. Differently from our data collections, DNS-OARC does not collect and publish zone files daily. Based on these data we can pinpoint the disappearance of orphan records between July 7 and July 15, 2010. Figure 4.2 also shows a net decrease of orphans already in March 2010. We can trace this back to [69]

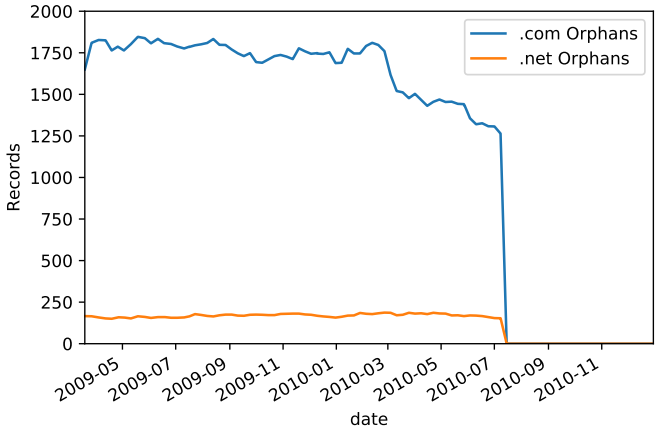


Figure 4.2: Orphans in `.com` and `.net` 2009 – 2010

in which Verisign explicitly announced that from March 1, 2010, glue records would no longer be promoted to authoritative status. However, they also stated that: "These records will not actually be removed: although they will not be returned when queried for directly, they will appear in the additional section of referrals that reference them". After July 2010, Verisign also started removing orphan records from the zone files that are made available through the zone file access program. Since these records are still returned in DNS referrals, this means that the zone file made available through the zone access program no longer exactly reflects the state of the registry database.

4.6.4 The Case of `.se`

In the case of .se, we find 48 orphan records. These records are all subdomains of: `org.se`, `pp.se`, `fhsk.se`, `d.se`, `g.se`, `ns.se`, `ac.se`, and `fh.se`. These domains reflect the former structure of the *.se* namespace (pre 2003), in which any registered domain was a subdomain of a registry-managed second-level domain [70]. The *.se* domain name structure was then liberalized. However, the *.se* operators confirmed to us that some old records are maintained in the zone file for legacy purposes and are banned for registration. Therefore, their presence does not pose any issues.

4.6.5 The Case of .de

Even though we do not analyze .de domains, their administration policy represents a special case that was not considered in [30]. DENIC permits users to publish and manage domains directly in the .de zone with the following three restrictions: (i) maximum 5 RRs; (ii) only A, AAAA and MX RRs; (iii) records are checked (for legitimacy) by the DENIC staff. This opportunity to directly manage subdelegations breaks the common operating model of TLDs and impacts the discovery of orphan and abandoned records. In particular, these A records can be false positives for the analysis conducted by Kalafut in [30]. However, given that .de is not considered in our analysis (due to lack of access to .de zone files) and .info, .mobi and .org (which are the most affected by the orphan and abandoned misconfiguration) do not allow this operational model of directly publishing and managing records in the zone, we are confident that this does not impact the conclusions and the main results of our analysis.

4.7 Ethical Considerations

We perform our study of orphan and abandoned records through the analysis of zone files provided by registry operators to the OpenINTEL project. The results related to orphan records could lead to potential NS hijacking. For this reason, and because of the contractual restrictions under which OpenINTEL gets access to most zone files, we publish only aggregated results and we do not refer to specific cases. Furthermore, we performed the WHOIS scan using a conservative approach and on a single day in order to not overload the WHOIS servers.

4.8 Concluding Remarks

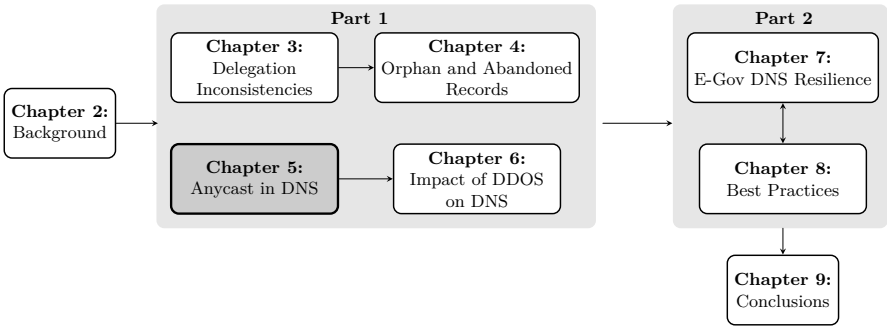
Our work was prompted by the work by Kalafut *et al.* [30] and aimed at evaluating the state of orphan misconfiguration a decade later. We discovered that for the .com and .net TLDs, the number of orphan records has fallen to zero, which means that operators have introduced mechanisms for cleaning their zone files. Unfortunately, these best practices are not adopted by all TLD registry operators. For some TLDs, the number of orphan records has increased over 10 years. Also, in the new gTLDs introduced after [30], this misconfiguration is widespread. We also discover and analyze another misconfiguration: the abandoned record. Our analysis shows that this misconfiguration is more broadly present than the orphan misconfiguration. Even if these records are not resolved, common sense would suggest they should be removed by registries or registrars,

as they potentially represent the initial stage of orphan creation. Our study also shows that the removal of these records from the zone file may not be a simple operation, since it can incur the risk of breaking other domains. We suggest all registry operators address this misconfiguration by at least making domains related to orphan records not available for registration or by considering to clean up their *zone* removing orphans. We reached Afilias, one of the most affected registries, to fix the problem in more than 200+ TLD zones under its care. Afilias planned to remove all problematic orphan glue records and also adjusted security settings to prevent the persistence of such records when names were deleted in the future[3]. Furthermore, Afilias notified registrars so that they could inform the few domain owners who were relying on orphan glue records to make necessary adjustments immediately.

[3] `https://circleid.com/posts/20200811-afilias-to-protect-tlds-against-potential-orphan-glue-exploits/`

CHAPTER 5

Anycast Adoption in the DNS Authoritative Infrastructure



In the previous chapters, we directed our attention to miscommunication between different operators, with potentially catastrophic consequences for DNS resilience. In this chapter, we will focus instead on the choices of single DNS operators in the wild to increase DNS resilience. One of the most effective techniques to increase DNS resilience is IP anycast. Performing an anycast census at scale, however, is a challenging task. In what follows, we will show a methodology to overcome this challenge. We will then use the data collected with this census to study anycast and classical resilience technique adoption in the global TLD and SLD space over time (RQ-3). Our results show a significant portion of the DNS relying on redundant infrastructure, with, however, worrying traits of organizational centralization. The study discussed in this chapter was performed in early 2021, published in two academic conferences [32, 33] and won the TMA 2021 best paper award.

5.1 Introduction

The traditional way of providing DNS resilience relies on explicit name server replication. An authoritative nameserver provides a set of name server replicas in response to a query (e.g., `ns1.foo.com, ns2.foo.com, ns3.foo.com`). If any such server fails — even silently — a requesting resolver can re-issue their request to a different replica. Distributing these name server replicas in disjoint networks insulates the overall service from the failure of any one network. So long as any one replica remains operational and reachable, name service can still be provided. Over time another mechanism has emerged for providing resilience at the network layer: IP anycast. In the IP anycast model, geographically diverse server replicas all use the *same* IP address by arranging for different networks to all announce the *same* network prefix. Employing this architecture for the DNS moves the decision for replica selection from an explicit choice made by the requesting party (typically a client or recursive resolver) to an implicit choice implemented by BGP and the ISP's routing policy.

In this chapter, we focus on the evolution of anycast for providing DNS nameservice. As first step, we present a methodology to efficiently perform an IP anycast census at scale. We then leverage the collected data to empirically characterize the adoption of anycast by nameservers supporting TLDs and SLDs (second level domains, *a.k.a.* registered domains), between 2017 and 2021. We show that anycast is now the dominant mode for providing DNS service — used by 97% of TLDs and 62% of SLDs in our dataset. We find that this adoption is not driven by the actions of individual domain owners, but is dominated by the engineering choices of a few large DNS infrastructure providers. The top 10 anycast-supporting DNS providers account for 92% of all domains with anycast name service. A single registrar, GoDaddy, accounts for the majority of anycast adoption in SLDs. To investigate the relationship between resilience and infrastructure diversity, we show that domains using anycast name service frequently exhibit lower diversity in their use of IP addresses and ASNs. As a result, anycast-based name service does not eliminate the resilience problem, but offers a different resilience risk profile. We conclude by reviewing different failure modes that shed light on how anycast changes the risk profile of a given deployment.

5.2 Related Work

The DNS ecosystem has been the subject of several studies focusing on diverse aspects of DNS resilience and robustness. Allman investigated the extent to which DNS administrators do not provide significant infrastructure diversity in hosting their domains [71]. Lame delegations also affect the robustness of

the DNS ecosystem. Akiwate *et al.* found that lame delegations are surprisingly common, even in popular domains [72]. The consolidation of the Internet and DNS ecosystem, which we also observe in anycast adoption, has also been studied. Kashaf *et al.* found a considerable concentration in the use of third-party services for authoritative DNS name service [15]. Moura *et al.* found an increasing consolidation of the recursive resolution infrastructure [73]. The impact of operator practices on DNS query performance has also been extensively studied [56, 74, 75, 76] culminating in recommendations for large DNS operators [77]. Our work focuses on anycast adoption in the authoritative nameserver infrastructure – an aspect that has not been investigated in depth. The use of anycast in DNS was first studied by Xun *et al.* [78] who used CHAOS queries to enumerate anycast instances, and estimate adoption of anycast in TLD authoritative nameservers. Their findings show, in 2013, between 56% to 72% of TLD authoritative nameservers adopted anycast. Our work expands their analysis by using anycast census data showing an increased adoption of anycast, in 2021, to 97% of TLDs. In 2015, Cicalese *et al.* [79] performed an anycast census using a methodology, called *iGreedy*, based on the Great-Circle Distance. Bian *et al.* [80] proposed a passive approach to anycast enumeration using public BGP data from route collectors. Recently, we proposed a methodology [33] to measure anycast using anycast vantage points. We summarize the working principles of this methodology in section 5.3. To make this study possible, our work uses our 2021 measurements and the 2017 measurements by Cicalese *et al.* (section 5.4). As such, our work leverages and builds on top of previous anycast enumeration studies.

5.3 Measuring Anycast at Scale

Identifying which addresses are anycasted and from where they are announced is a fundamental step to provide a more accurate assessment of the Internet's resilience. Unfortunately, the IPv4 approach to anycast relies on the principle of assigning the same unicast address to multiple hosts and leveraging routing to implement the anycast addressing. Due to the opacity of routing, identifying which address is anycast is not straightforward. Traditional methods of measuring anycast are based on the Great Circle Distance (GCD) technique. The basic idea behind GCD is that if two probes in the world reach a node with a latency that violates the speed of light, the node must be anycast. The GCD approach has proven to be effective and results in a tool for anycast measurement called iGreedy [79]. *iGreedy* uses the GCD for anycast detection, enumeration and geolocation. More specifically, the technique uses speed of light violations to infer distinct anycast replicas and then uses multiple observations and a sub-

sequent greedy algorithm to enumerate replicas. City-level geolocation relies on a maximum likelihood estimator. However, the GCD approach has some disadvantages; it requires a significant number (∼200) of geo-located probes in order to perform an accurate measurement, and has a relevant footprint in terms of generated traffic. RIPE Atlas offers a solution to this, however, this can be unsuitable for a long-term census of the entire Internet.

5.3.1 A new Approach to Anycast Measurements

We propose a new measurement and inference technique, MAnycast^2, which relies on an anycast testbed to efficiently detect anycast prefixes. MAnycast^2 relies on the principle of *using anycast to measure anycast*, which involves sending probes (ICMP echo requests) from multiple anycast vantage points to a target IP address and then checking which vantage points received the responses (ICMP Echo responses). The number of vantage points receiving responses reveals whether a target is unicast or anycast. The traffic of the ICMP echo-responses to the anycast IP should be routed back to a single node, if the target is unicast and on multiple nodes, in case the target is anycast. Compared to traditional anycast detection techniques, MAnycast^2 does not rely on latency measurements. Instead, it leverages the routing system to discriminate anycast deployments. We tested MAnycast^2 using two different anycast platforms – Tangled [81] and PEERING [82]. We were able to perform an anycast census of the entire ISI IPv4 hitlist [83] in ≈2.5 hours, with only 10 pings per target IP address, demonstrating the lower traffic footprint and the speed of our methodology. We validated the finding of MAnycast^2 with public ground truth of anycast networks (for example, DNS root servers), reaching operators for confirmations and using a GCD-based tool (iGreedy) with RIPE Atlas. Our comparison shows low error rates, especially when combined with the GCD approach as a second measurement stage. More details on MAnycast^2 can be found in [33]. In the following section, we will show how this combination works and the results collected by our anycast census.

5.4 Datasets and Limitations

In this section we describe the DNS and anycast datasets that we use for our study and we outline considerations in handling this data.

5.4.1 Datasets

Anycast Our work builds on two anycast IPv4 censuses. The first census was published in June 2017 by Cicalese *et al.*, who ran their *iGreedy* measurement

Date	Total SLDs	Responsive SLDs	Unresponsive SLDs
2017-06-01	189.6 M	164.4 M (87%)	25.2 M (13%)
2021-01-31	210.4 M	187.5 M (89%)	22.9 M (11%)

Table 5.1: Total SLDs and SLDs with responsive authoritative nameservers.

on PlanetLab and RIPE Atlas to create an anycast census [79]. The resulting dataset contained 5,486 distinct /24 anycast prefixes.[1]

Leveraging the methodology highlighted in section 5.3, we performed an anycast census in January 2021 using our MAnycast^2 tool [33]. [2] MAnycast^2 uses *iGreedy* (on RIPE Atlas VPs) to cross-validate detected anycast prefixes and perform enumeration and geolocation. MAnycast^2 used 20 distinct vantage points provided by SIDN Labs. The January 2021 anycast census dataset contained 9,999 distinct /24 anycast prefixes.

DNS We use DNS data provided by the OpenINTEL project, which measures ∼65% of the global DNS namespace by actively querying for the resource records of second-level domains (SLDs) under a sizeable number of top-level domains (TLDs) on the Internet [25]. OpenINTEL's *daily* measurement actively queries for, among others, the authoritative nameserver records (i.e., NS records), as well as the IPv4 addresses (i.e., A records) of the names encountered in NS records. While the OpenINTEL project regularly expands its coverage of the namespace and has steadily added TLDs over time, we include only TLDs that were already covered at the time of the first anycast census in June 2017. This set consists of 1053 TLDs. To account for missing data points on particular days (which is rare but could occur, e.g., due to incidental outages) we require each TLD to be in the dataset for 95% of all days between the anycast census dates. The TLDs we consider involve: the (legacy) generic TLDs `.com`, `.net` and `.org`; the new generic TLDs (ngTLD) such as `.tokyo`; and the country-code TLDs (ccTLD) `.at, .ca, .dk, .fi, .nl, .nu` and `.se`. The resulting DNS dataset accounts for ∼164 million domains in 2017 and ∼187 million in 2021.

To correlate a domain's anycast deployment with its popularity, we also use OpenINTEL measurement data for domain names of the top 1 million popularity lists for Alexa (2017-06-01 and 2021-01-31) and Cisco Umbrella (2021-01-31).

Metadata We use CAIDA's prefix-to-AS dataset [26] to map IP addresses of authoritative nameservers to their covering prefix and announcing AS num-

[1] iGreedy anycast census: `https://anycast.telecom-paristech.fr/dataset/`
[2] MAnycast^2 anycast census: `https://github.com/ut-dacs/Anycast-Census/`

ber(s), and CAIDA AS-to-organization data [84] to map AS numbers to organizations. Finally, we use Netacuity data to geolocate unicast IPv4 addresses.

5.4.2 Data Considerations

Our analysis involves a few assumptions and decisions that factor into our results. First, we consider only responsive SLDs (Table 5.1). Consequently, our results interact with active DNS infrastructure. We observed ∼12% unresponsive SLDs, which is consistent with the findings of Akiwate *et al.* [72].

Second, our analysis involves active DNS measurement data. Consequently, we consider nameservers learned from explicit NS queries, and A records that follow active resolution. We do not rely (directly) on the NS records and the A records (glue) in zone files. In other words, we consider only the records that are provided by the authoritative nameservers. In chapter 3 we have found inconsistencies between parent and child zones for up to 5–12% of observed SLDs. In the same chapter, we showed that whether DNS resolution follows parent or child records is resolver-dependent.

Finally, we are aware from the associated papers that both anycast inference methodologies we use can include classification errors [33, 79]. MAnycast^2 (combined with *iGreedy*) as well as *iGreedy* alone can result in false negatives (i.e., anycast deployments identified as unicast). However, both techniques deliver a conservative lower bound estimate of anycast deployment. Therefore, our anycast adoption analysis is a conservative lower bound estimation.

Our analysis relies on data collected on a daily (OpenINTEL) and quarterly basis (MAnycast^2). We developed the analysis code to be reused for reproducibility and continuous assessment of DNS anycast adoption. We publicly released the code for TLD adoption analysis (section 5.5) [85]. Cases that sporadically require additional measurements (e.g., traceroute in section 5.7) are analyzed manually.

5.5 Anycast Adoption by TLDs

Given their critical role in the DNS, we start by characterizing anycast adoption by top-level domains (TLDs). We used snapshots of the root zone from DNS-OARC [68] for our two time periods. Table 5.2 summarizes the number and kinds of TLDs in each period. The total number of TLDs in the root zone decreased slightly from 1533 TLDs in 2017 to 1505 in 2021: 60 TLDs from 2017 were no longer in the 2021 root zone (such as `.intel`, `.telefonica`), and 32 new TLDs were added between 2017 and 2021 (e.g., `.ss`, South Sudan's ccTLD, and `.amazon`). Between the two periods, though, most of the TLDs (1,473) were *delegated* in both root zone files (∩).

	2017-06-01	2021-01-31
TLDs	1,533	1,505
removed	–	60
added	–	32
∩	1473	
ccTLDs	247	
gTLDs	7	
ngTLDs	1219	

Table 5.2: Root zone TLD snapshots in 2017 and 2021. Our work analyzes TLDs present in both the 2017 and 2021 snapshots. We break down the TLDs analyzed as either legacy gTLDs, new gTLDs or ccTLDs.

	gTLD		ccTLD		new gTLD		Total	
	2017	2021	2017	2021	2017	2021	2017	2021
Unicast	1	1	79	34	25	15	105 (7.1%)	50 (3.4%)
Mixed	2	1	137	160	117	139	256 (17.4%)	300 (20.4%)
Anycast	4	5	31	53	1,077	1,065	1,112 (75.5%)	1,123 (76.2%)
Total	7	7	247	247	1,219	1,219	1,473	1,473

Table 5.3: Breakdown of TLDs with unicast, anycast, or mix of both anycast and unicast (mixed) authoritative nameservers in 2017 and 2021. Anycast adoption (including mixed) in 2021 reached $\sim 97\%$

We focus on this intersection of TLDs that are present in both zone files. Since there is significant variation in the types, history, management, and use of top-level domains [38, 86], we classify these TLDs into three categories: ccTLDs (e.g., `.jp` and `.de`), gTLDs ("original" gTLDs: `.com, .edu, .gov, .mil, .org, .net, .int`) and ngTLDs (`.tokyo, .xyz, .top`).

For each TLD, we extract its NS records and associated A records. For each A record, we label it as using anycast if it matches the anycast prefix datasets described in subsection 5.4.1, otherwise we label it as unicast. Since not all A records for a TLD may have the same label, we classify each TLD into three categories: those whose A records are all anycast, those whose A records are all unicast, and those with mixed usage (some, but not all, A records have anycast IPv4 addresses).

5.5.1 Increasing Adoption of Anycast

Table 5.3 shows the results of this TLD classification. Overall, the use of anycast for TLD authoritative nameservers, in whole or part, shows increased adoption between 2017 and 2021. In 2017, 1,368 TLDs (93%) used anycast (in whole or in part), and just 105 (7%) used unicast. In 2021, anycast adoption increased, with 1,423 TLDs (97%) using anycast while only 50 TLDs (3%) relied solely on unicast authoritative nameservers. For the ccTLDs, 45 of the 79 ccTLDs using unicast (57%) in 2017 moved to either mixed (23) or full (22) anycast infrastructure by 2021. For the ngTLDs, which already had widespread anycast adoption, 10 of the remaining 25 TLDs (40%) using unicast in 2017 moved to mixed or full anycast by 2021. For the original gTLDs, `.gov` moved from mixed to full anycast support, leaving `.mil` as the only original gTLD not using anycast.

A significant reason for the increase in ccTLDs using full anycast was the set of 18 ccTLDs in the mixed category in 2017, including `.cz, .io, .nl` and `.in`, that solely used anycast by 2021. For most of these (14 ccTLDs), the change was simply because they dropped the unicast nameserver. Perhaps 2017 marked a transition period where they balanced old and new infrastructure, and by 2021 those ccTLDs were committed to full anycast.

Not all changes increased anycast adoption. For instance, `.ki` (Kiribati) changed from full anycast to mixed infrastructure, and three ccTLDs (`.ve`, `.pa`, and `.cd`)[3] changed from mixed to unicast only. These changes reflect the choice of, and dependence on, underlying services. The `.pa` and `.ve` ccTLDs, for example, employed the Internet System Consortium (ISC) authoritative anycast service (`sns-pb.isc.org`), which shut down on January 31, 2020 [87].

5.5.2 Anycast Infrastructure Expansion

One motivation for a TLD to switch from unicast to mixed or full anycast for authoritative name service is to improve availability and resilience. A proxy metric, admittedly rough, for representing the implications of this change is the scale of authoritative nameserver infrastructure. For a TLD, the total infrastructure is the combined unicast IPv4 addresses and anycast sites across all of the A records attached to the NS records for the TLD.[4] We refer to this count as the number of *combined replicas* providing name service for the TLD. For instance, France's `.fr` in January 2021 had 4 NS records with 4 IP addresses (1 unicast and 3 anycast), and the anycast addresses were distributed across 68 sites. In this case, we consider `.fr` to have 69 combined replicas.

[3] Venezuela, Panama, and the Democratic Republic of Congo, respectively.

[4] We note that, due to the complexity of geolocating (and enumerating) anycast sites, we consider these results a lower-bound estimation.

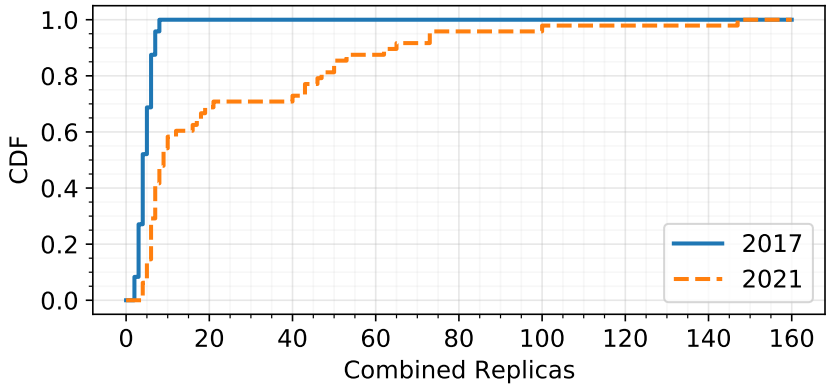


Figure 5.1: Marked increase in combined replicas for ccTLDs moving to Mixed or Anycast from Unicast

The combined replica metric is a rough proxy metric because, from the client point of view, there is a difference between a unicast address, which is globally reachable, and an anycast site, which is reachable only to the portion of users mapped to that site by BGP (i.e., anycast "fragments" the IP address space [74]). Still, it does reflect the infrastructure investment supporting name service and an upper bound on availability and resilience (discussed further in section 5.7).

We first focus on the 48 ccTLDs that were unicast only in 2017 and changed to mixed or full anycast by 2021. Figure 5.1 shows the CDFs of the number of combined replicas across these TLDs for both snapshots in time. In 2017 these ccTLDs had a median combined replica count of 4 (in this case, 4 NS records each with a single IPv4 address) and a 75%-ile of 6. After switching to anycast, the median number in 2021 increased to 9 replicas, and the 75%-ile to 43 — a significant increase from 2017 in terms of supporting infrastructure.

For the new gTLDs, most (1,025) used anycast in both 2017 and 2021, and only 10 switched from unicast to mixed or full anycast. We do not include a graph for these 10 new gTLDs, but to summarize they also experienced a significant increase in the scale of infrastructure: their median combined replicas increased from 4.5 to 34.

Moreover, Figure 5.2 shows that the scale of anycast infrastructure increased considerably between 2017 and 2021. The graph focuses on the 30 ccTLDs that used full anycast in both 2017 and 2020, and it shows the CDFs of the number of anycast sites across those TLDs. The median combined replicas increased

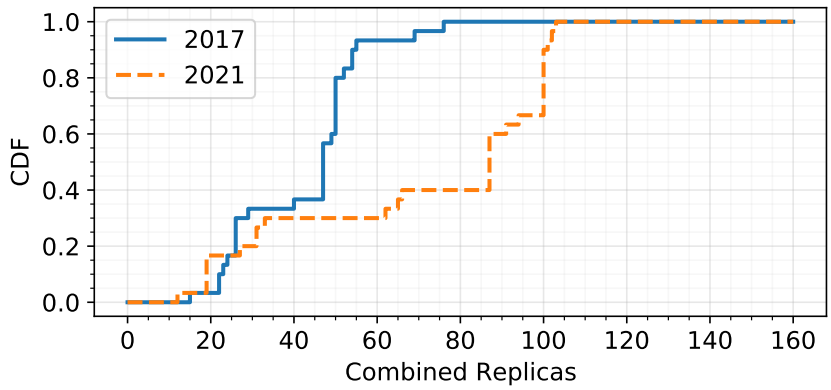


Figure 5.2: Significant growth in replica infrastructure for ccTLDs using Anycast in both periods

from 47 in 2017 to 87 in 2021. Anycast infrastructure is scaling considerably over time, and name service naturally benefits from this scaling.

5.5.3 Large Providers Drive Anycast Adoption

For gTLDs and ngTLDs, the top 10 providers for TLDs account for 88% of the use of anycast by TLDs in 2021, with Neustar (36%), Afilias (22%) and Verisign (10%) leading the list. For ccTLDs, the landscape is more fragmented, reflecting the more nuanced balance that ccTLDs make between using first and third-party infrastructure. The top 10 providers are responsible for 69% of anycast ccTLDs, and the biggest operator, PCH, manages 25% of the ccTLDs, followed by NetNod (9%) and RIPE (5%).

5.6 Anycast Adoption by SLDs

The TLD authoritative nameserver infrastructure has substantially adopted anycast. The next step in the resolution process is at the second-level domain (SLD). Have SLDs followed suit, making DNS resolution fully reliant on anycast authoritative nameservers? To identify anycast authoritative infrastructure in SLDs, we correlate the 2017 and 2021 anycast census datasets with the authoritative infrastructure measurements provided by OpenINTEL. We extract the NS records of the SLDs and all of their related A records from OpenINTEL.

Type	Anycast	Unicast	Mixed	Total
		2017		
#SLD	74.2M (45.1%)	84M (51.1%)	6.2M (3.8%)	164.4M
#NS(IP)	10,700	899,028	N/A	909,728
		2021		
#SLD	106.4M (56.8%)	70.9M (37.8%)	10.2M (5.4%)	187.5M
#NS(IP) 18,179	756,459	N/A	774,638	

Table 5.4: Anycast in DNS: 2017 and 2021 adoption. Number of SLDs relying on unicast infrastructure decreased between 2017 and 2021 by 13.3%. Overall anycast adoption (Mixed+Anycast) reached 62.2%

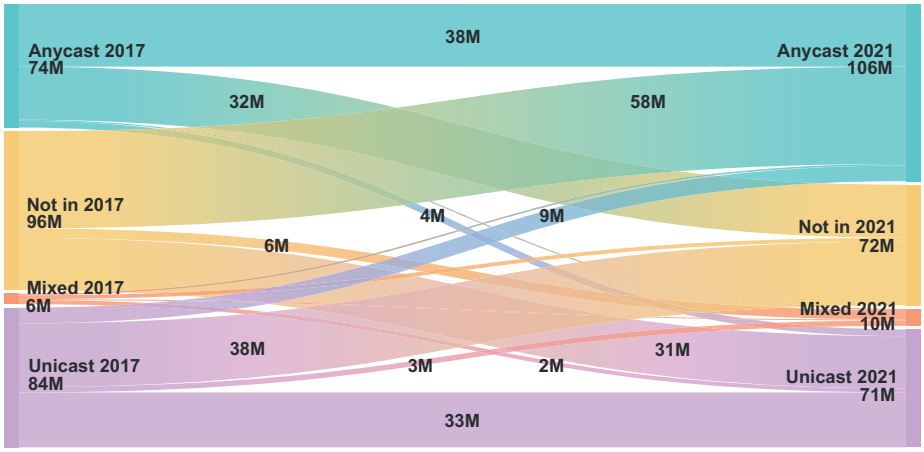


Figure 5.3: Evolution of anycast authoritative deployment between 2017 and 2021.

Based upon the IPv4 addresses in their associated A records, for each domain we use the anycast census datasets to classify it as unicast, anycast, or a mix of the two.

Table 5.4 summarizes the results of our classification for the 2017 and 2021 snapshots. As with TLDs, SLD infrastructure is also increasingly relying upon anycast infrastructure for authoritative name service. In 2017, 51.1% of domains relied on unicast infrastructure, 45.1% domains relied solely on anycast, and 3.8% of domains relied on a mix of the two. By 2021, the domains relying upon unicast dropped by 13.3%, the domains relying upon anycast increased by 11.7%, and the domains on mixed infrastructure increased by 1.6%.

Org	SLD	%	Org	SLD	%
GoDaddy	44,145,357	54.53%	Google	3,433,523	4.24%
CloudFlare	6,955,596	8.59%	Uniregistry	2,376,567	2.94%
1&1 IONOS	4,808,600	5.94%	Akamai	1,451,470	1.79%
DynDNS	3,883,403	4.80%	Amazon	1,068,653	1.32%
VeriSign	3,878,585	4.79%	One.com	1,016,796	1.26%

Table 5.5: Top 10 anycast organizations 2017, responsible for 90% of the anycast adoption. GoDaddy was market leader.

Finally, looking at the IPv4 addresses of the authoritative servers, only 2.3% of them rely on anycast infrastructure. These results suggest that anycast is used by only a few companies, yet half of the domains in the DNS rely on these companies (subsection 5.6.1).

To visualize the evolution of anycast adoption between 2017 and 2021, Figure 5.3 is a Sankey diagram[5] showing how domains changed categories between the two snapshots. For example, of the 84M domains relying on unicast in 2017, 33M of them still relied on unicast in 2021, 3.2M relied on a mix of unicast and anycast, 9.4M relied solely on anycast, etc. Given that anycast adoption is more prevalent in 2021, the diagram provides more detail on the sources of those domains. For instance, new domains are more likely to rely on anycast: nearly twice as many new domains in 2021 that did not exist in 2017 use anycast rather than unicast.

In contrast, the majority of SLDs that are no longer responsive in 2021 were using unicast infrastructure in 2017. Moreover, 9.4M domains shifted from a unicast infrastructure to full anycast, 3.2M from unicast to mixed anycast, and 4M shifting from anycast back to unicast. Examining these last 4M SLDs more closely, they are primarily domains with GoDaddy, Dyn, CloudFlare, and 1&1 moving to other minor registrars/infrastructures.

5.6.1 A Concentrated Set of Providers Drives Anycast Adoption

The increasing adoption of anycast among TLDs is tied to the deployment of anycast by a concentrated set of providers (subsection 5.5.3), and we similarly look at providers to explain increased anycast adoption among SLDs. Using IP→AS→organization mappings (subsection 5.4.1), we identified the top 10 anycast organizations both in 2017 (Table 5.5) and in 2021 (Table 5.6). These top 10 anycast organizations are responsible for 90% of anycast adoption in 2017

[5]Interactive Visual: `https://public.flourish.studio/visualisation/5568561/`

Org	SLD	%	Org	SLD	%
GoDaddy	52,681,291	44.11%	1&1 IONOS	6,033,089	5.05%
Cloudflare	15,252,317	12.77%	NSONE	3,160,888	2.65%
Google	11,014,408	9.22%	Amazon	2,949,373	2.47%
NeuStar	7,968,959	6.67%	NetActuate	1,902,258	1.59%
Zenlayer	6,800,764	5.69%	Tencent	1,781,520	1.49%

Table 5.6: Top 10 anycast organizations 2021, responsible for 92% of the anycast adoption. GoDaddy's market share slightly decrease, Cloudflare increased.

and ∼92% in 2021. These results confirm that adoption is primarily driven by large DNS providers.

Looking at individual companies, GoDaddy unsurprisingly is the largest company by far, in terms of SLDs hosted, that operates anycast services for their authoritative nameservers. In 2017, GoDaddy accounted for more than half (∼55%) of the anycast SLDs. In 2021 the percentage decreased (∼44%), but the absolute numbers increased. GoDaddy is the largest registrar in the world, and therefore their infrastructure choices as registrar (and, by default, DNS hosting provider) heavily influence the DNS ecosystem.

Next is Cloudflare, where anycast adoption for SLDs increased from ∼9% in 2017 to ∼13% in 2021. In contrast to GoDaddy, Cloudflare's core business is not as a registrar (even if it recently started a registrar service), but to offer CDN and DDoS protection services to their customers. As a result, customers likely choose Cloudflare for better performance, resilience, and availability of their Web services. But since Cloudflare adopted anycast, its customers benefit from it as well. In short, technical and business decisions of the company drive anycast adoption.

Among the other top 10 providers is a mixture of popular Web site building and hosting (1&1) and cloud providers, which operate DNS hosting themselves (e.g., Route53 and Cloud DNS) or with third parties (e.g., other DNS registrars).

In contrast to providers using anycast, unicast deployment is less concentrated: the top 10 accounts for only 63% of the total unicast SLDs. In terms of types of companies, the top 10 unicast DNS providers, both in 2017 and 2021, are almost all Chinese providers with two notable exceptions of Amazon and OVH. For Amazon, nearly 6.6 million SLDs were hosted on non-anycast services (primarily third-party EC2 instances). OVH, a popular European hosting provider, offers optional anycast service for DNS nameservers, and customers must pay a premium of 1.21 Euro per year for anycast. Nearly all SLDs using OVH's authoritative infrastructure use unicast: we measured 4,156,201 domains using OVH's unicast infrastructure, and just 130,951 domains using its premium

Source	Anycast	Mixed	Unicast	Total
		2017		
.com	53.3M (49.2%)	3.9M (3.6%)	51.2M (47.2%)	108.3M
.net	5.7M (45.1%)	0.5M (3.7%)	6.5M (51.2%)	12.6M
.org	4.9M (53.4%)	0.3M (3.2%)	4.0M (43.4%)	9.3M
ngTLDs	7.5M (34.7%)	1.2M (5.4%)	13.0M (60.0%)	21.7M
ccTLDs	2.6M (21.8%)	0.5M (3.7%)	9.0M (74.5%)	12.1M
.se	614K (43.3%)	146K (10.3%)	660K (46.5%)	1,421K
.nl	256K (4.8%)	70K (1.3%)	5,014K (93.9%)	5,400K
Alexa	337K (35.3%)	33K (3.4%)	584K (61.3%)	953K
Umbrella	N/A	N/A	N/A	N/A
		2021		
.com	76.7M (58.7%)	8.2M (6.3%)	45.7M (35.0%)	130.7M
.net	6.3M (54.3%)	0.6M (4.9%)	4.7M (40.8%)	11.6M
.org	6.0M (64.0%)	0.2M (1.6%)	3.2M (34.4%)	9.4M
ngTLDs	12.4M (55.4%)	1.0M (4.5%)	9.0M (40.1%)	22.4M
ccTLDs	4.9M (37.3%)	0.2M (1.5%)	8.0M (61.2%)	13.1M
.se	810K (57.2%)	13K (0.9%)	594K (41.9%)	1,416K
.nl	1,277K (22.2%)	36K (0.6%)	4,446K (77.2%)	5,760K
Alexa	423K (51.6%)	14K (1.7%)	383K (46.7%)	820K
Umbrella	157K (61.1%)	13K (4.9%)	87K (33.9%)	256K

Table 5.7: Anycast in DNS: 2017-2021 adoption per TLD. The specific case of the Netherlands and Sweden shows how two similar countries can have a completely different anycast adoption for authoritative nameservers due to registrar choices.

anycast infrastructure. We speculate that offering anycast as an optional paid service results in low anycast adoption for OVH customers.

5.6.2 Role of Registrars in Anycast Adoption

The GoDaddy example shows that popular registrars play a fundamental role in anycast adoption. Popular registrars like GoDaddy generally operate across the entire gTLD market, resulting in a roughly similar degree of anycast adoption from the new gTLDs (55.4%) to .com, .net, and .org (64%).

The ccTLD perspective looks quite different, with a much lower overall adoption of 37.3% of SLDs. One example particularly stands out, where the anycast adoption in .se (Sweden) is notably high while the anycast adoption in .nl (Netherlands) is comparatively low. The adoption of anycast in .se is related

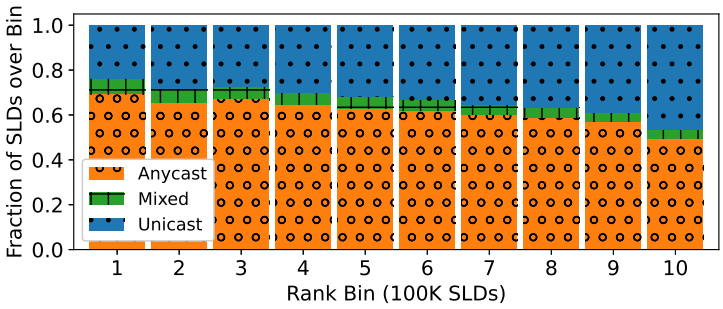


Figure 5.4: Anycast adoption correlated with domain popularity as ranked by Cisco Umbrella.

primarily to the implementation by Loopia AB, the largest registrar in Sweden (confirmed by the Swedish registry, IIS). The largest registrar in the Netherlands, TransIP B.V., has yet to adopt anycast. As we will see later on in this thesis (subsection 6.6.1), this choice not to adopt anycast can have catastrophic consequences if an operator comes under attack. More generally, `.nl` domains are spread across different small local registrars, which usually do not want to implement a global anycast infrastructure (due to related costs and complexity) or to pay for third-party service. These factors accentuate the low adoption of anycast for `.nl` domains.

In short, as expected, large registrars play a fundamental role in anycast adoption, with GoDaddy for gTLDs and Loopia for ccTLDs serving as notable examples.

5.6.3 Anycast Adoption and Domain Popularity

Popular domains by necessity are scalable, reliable, and available, and anycast is an increasingly popular mechanism to support those goals. As a result, either due to the extensive infrastructure that the domains deploy themselves, or by relying upon large-scale third-party infrastructure, we expect domain popularity to correlate with the use of anycast for the domain's authoritative name service.

To validate this expectation, we check which SLDs on the Cisco Umbrella list on January 31, 2021, relied upon anycast infrastructure for their nameservers.[6] In particular, we group sets of 100,000 ranked SLDs together into bins. We then calculate the percentage of SLDs in each bin that rely upon anycast using the

[6]We used Umbrella instead of Alexa since Umbrella ranks domains based on the DNS query load received by the Cisco Umbrella OpenDNS service. As a result, this measure correlates better with popularity in DNS resolution.

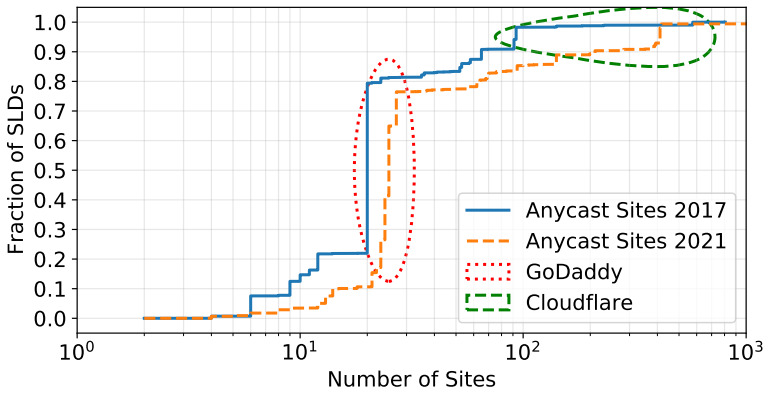


Figure 5.5: Distribution of number of anycast authoritative nameserver sites per SLD. In 2021, average number of sites slightly increased.

2021 anycast census dataset. For each ranked bin, Figure 5.4 shows the fraction of SLDs that use unicast, anycast, or a mix of the two for authoritative name service. The results clearly confirm the expected correlation that more popular domains rely upon anycast. Indeed, among the 100,000 highest ranked domains, more than 76% of the SLDs rely upon anycast in whole or part.

Since the use of anycast for authoritative name service often depends on the underlying provider, we also examine the providers behind the most popular domains. Looking at the top 10,000 SLDs of the Umbrella list, there are many different anycast providers (e.g., Google, Facebook, Microsoft, Apple, etc.) together with classic CDN and DDoS protection providers such as Cloudflare, Amazon, Akamai, etc. Looking at the entire top 1M Umbrella list, Cloudflare (and partially Amazon) lead the anycast adoption market. GoDaddy accounts for 10% of the SLDs using anycast, and those SLDs tend to be less popular. While the extensive DNS scan showed GoDaddy to be the most popular anycast provider due to its DNS market share (subsection 5.6.1), the bulk of its customer base has SLDs that are in the long tail of the popularity distribution.

5.6.4 Anycast Infrastructure Expanding

The anycast datasets include the number of anycast sites for each anycast IPv4 address. As in subsection 5.5.2 for TLDs, we can use this information to examine how the scale of the underlying anycast infrastructure has changed over time for SLDs.

Figure 5.5 shows CDFs of the number of anycast sites supporting authoritative name services across all SLDs. The graph focuses on just the SLDs fully using anycast, and for each SLD we sum the number of anycast sites across all IPs associated with the SLD. Since GoDaddy is the most common provider of anycast for SLDs (subsection 5.6.1), it determines the largest mode of the distribution. In 2017, GoDaddy used 20 distinct anycast sites, making this scale the most common anycast deployment for a domain. By 2021 GoDaddy had expanded its infrastructure slightly to 25–30 anycast sites.

In contrast, Cloudflare significantly expanded its anycast infrastructure between 2017 and 2021. In 2017 SLDs relying upon Cloudflare were supported by ∼90 anycast sites, and by 2021 the number of sites increased to 130.

5.7 Implications of Anycast Adoption for DNS Resilience Risk Profiles

The goal of the redundancy mechanism in the DNS is resilience against failure. In traditional unicast DNS, the recommended best practice is to maintain at least two authoritative nameservers (IP diversity) in different network segments (routed prefix diversity), and ideally in different networks (AS diversity) and geographic regions (geographic diversity) [21]. This investment in diversity provides resilience against failures of individual servers, subnets, entire networks, or connectivity in a specific region.

With unicast, the settings explicitly manifest this diversity. If one server fails, the client resolver can (and must) be responsible for re-issuing the query to a different authoritative nameserver. However, it can be operationally costly and complex to arrange for the subnet diversity recommended for unicast DNS deployments. We speculate that the complexity of arranging topological and geographical diversity for authoritative DNS is a major driving force behind the introduction of large anycast services where domain registrants can outsource their authoritative DNS service provisioning.

With anycast, service diversity is not explicit in the DNS settings, but manifests in the routing system. That is, in case of a link failure of a single authoritative nameserver, the Internet routing processes re-route packets to a different authoritative nameserver replica with the same IP address. Importantly, the effect of anycast adoption on resilience depends on deployment parameters as well as failure conditions. For example, if a server fails, the client can go to another server. If the IPs are diverse, then a failure of one network can be tolerated by a client-side retry. If the IPs are routed by different ASes then the same mechanism tolerates ISP-level failures.

Anycast hides at least some of the replica choice decision from the client. If all NS entries point to the same IP, then resolution relies entirely on anycast and if a server or subnet fails silently (i.e., there is no route withdrawal) then everyone routed to that advertiser is effectively black-holed. If the domain is both using anycast and returns multiple replicas, then the client can still tolerate failure via retry so long as those distinct replicas are not all a) the same IP (in case of server-level failure), b) on the same network (network-level failure), or on the same AS (AS-level failure). Potentially, anycast re-routing could interrupt TCP sessions, which are usually used by DNS resolvers when responses exceed 512 bytes and EDNS is not supported. However, we advocate that the risk and the possible impact is small, given the short-lived TCP sessions and the in-protocol reliability mechanisms.

Notably, most SLDs rely on two authoritative server IPs both for anycast and unicast deployments (Figure 5.6), likely because registries and registrars usually require two distinct NS records. However, the number of distinct IPs for anycast deployments peaks at 8, and 90% have 4 or fewer. In contrast, 10% of unicast deployments have 12 distinct IP addresses. The AS-level diversity has a similar contrast: the vast majority of anycasted SLDs are anycasted entirely from a single AS or sibling ASes that belong to the same company (e.g., Neustar), while 40% of unicast SLDs use two or more ASNs (Figure 5.7). Concentration of services within a single AS/company is a natural market force, but comes with its own risks. Manifestations of these risks [14] has motivated the non-significant number of mixed deployments, that attempt to optimize multiple dimensions of DNS resilience [88].

We also found two interesting anycast cases of this "single point of failure", with many domains routed behind the same prefix: 1&1 IONOS SE, responsible for ∼6.8 million domains and Loopia AB. 1&1 IONOS SE announces their anycast network from a single /22 block; our anycast geolocation data (based on the iGreedy second-stage measurement of [33]) indicates that all their anycast sites are in the same location for all the different IPs. Using traceroute from all the nodes of the NLNOG RING network troubleshooting platform [89], we discovered that the four /24 networks composing the 1&1 IONOS SE /22 block are routed behind the same last hop. Similarly, Loopia AB serves ∼500K domains via anycast from a single /24 block. This means that Loopia relies as its only resilience mechanism uniquely on anycast, with all the consequences related to the possible silent failing of one of the instances (e.g., DNS unreachability for part of the users). The 5% of nameservers pointing to the same IP (Figure 5.6) is another interesting case of a single point of failure. Even if registrars require two nameservers, operators effectively provide lower diversity by pointing the nameservers to the same IP. We find this behavior also in ∼8% and ∼3% Alexa and Umbrella domains, shifted towards the tail of the lists.

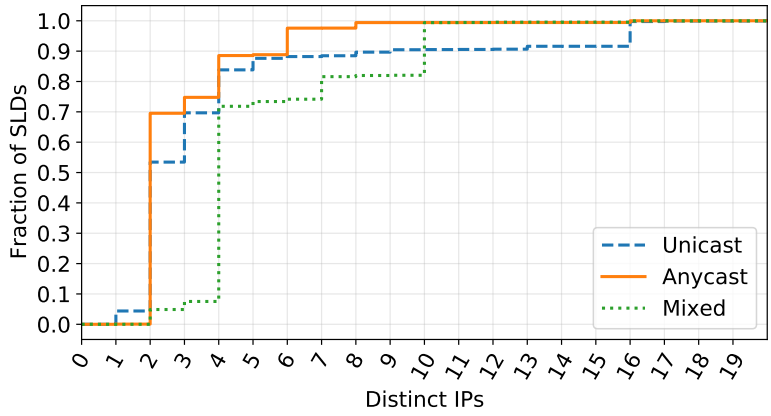


Figure 5.6: Number of IPs corresponding to SLD authoritative nameservers. Anycast authoritative nameserver deployments tend to have fewer IPs, since anycast provides diversity via the routing system.

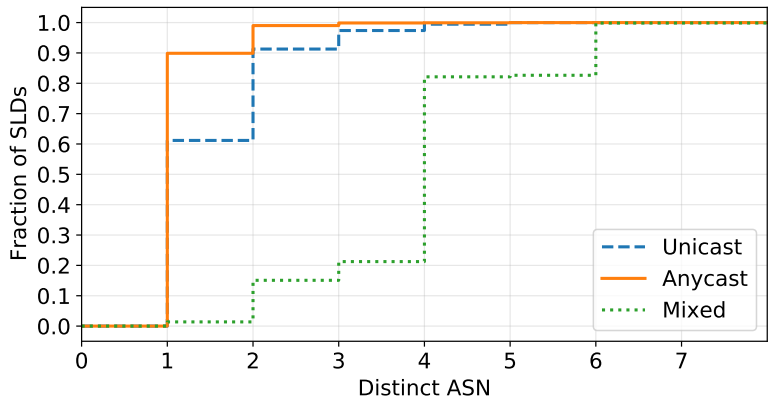


Figure 5.7: AS Diversity for SLDs with anycast, unicast and mixed authoritative nameservers. Anycast deployments are usually concentrated in a single ASN.

Placing servers in different geographical locations reduces latency and improves resilience against disasters. Given that anycast deployments are often globally distributed, we expect and observe higher country diversity (Figure 5.8) for these deployments. For unicast infrastructure, ∼75% of domains rely on authoritative servers in the same country, ∼20% spread over two countries and

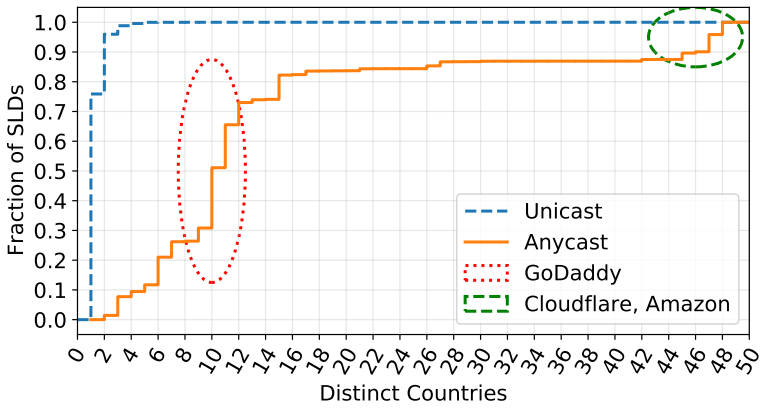


Figure 5.8: Distribution of SLD authoritative nameservers across countries. Anycast deployments are more globally distributed.

only ∼5% spread over more than two. In contrast, anycast authoritative services are hosted in 6–12 countries on average (depending on whether GoDaddy is the provider). Another ∼14% are hosted in more than 42 distinct countries (Cloudflare and Amazon deployments). To conclude, anycast can suffer from administrative or business failures (e.g., global misconfiguration, attacks, etc) for centralized deployments (i.e., single companies), but, at the same time, it helps to increase geographical availability and resilience of the DNS ecosystem.

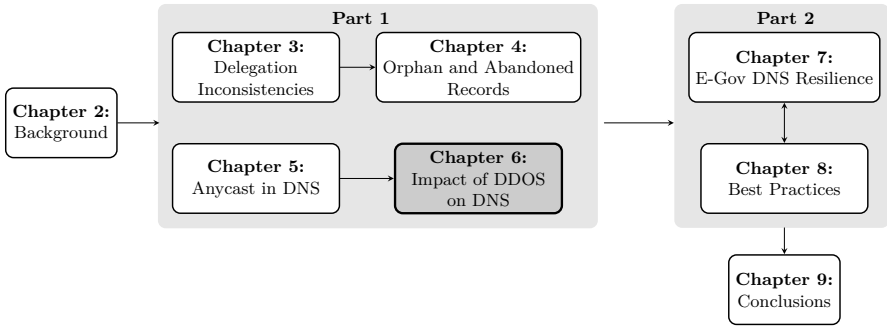
5.8 Concluding Remarks

We have characterized anycast adoption in authoritative DNS infrastructure for TLDs and SLDs. We found high adoption of anycast as a resilience mechanism, reaching 97% for TLDs and 62% for SLDs. This adoption is driven mostly by engineering choices of a few very large DNS infrastructure providers. In our data set, one provider (GoDaddy) was responsible for the majority of anycast adoption in SLDs. Finally, we examined the relationship of anycast deployments to other traditional metrics of infrastructure diversity. Our findings show that anycast adoption changes the DNS service availability risk profile but does not eliminate all such risks. In fact, anycast can hide certain types of availability failures and limit recovery options. A mixed deployment that includes traditional unicast redundancy as well as anycast options mitigates this risk, but increases cost and complexity.

To conclude, we remark that at the time of writing, there is no publicly available IPv6 anycast census. Hence, IPv6 anycast DNS infrastructure is out-of-scope for this work. As future work, we plan to expand our previous efforts [33] to also support IPv6 anycast measurement, and to compare IPv4 and IPv6 anycast authoritative nameserver deployments. Given that IPv6 is also popular among large companies, we expect that they often implement and offer anycast for both IP stacks.

CHAPTER 6

Impact of DDoS attacks on DNS infrastructure



In the previous chapter, we established the extent of the adoption of resilience techniques and best practices such as the use of anycast and topological redundancy in nameserver infrastructure. To gauge the effectiveness of established best practices in improving the resilience of DNS to attacks, it is crucial to examine the impact of DDoS attacks on the performance of DNS services. In this chapter, we combine two existing data sets – DDoS activity inferred from a sizeable darknet and contemporaneous DNS measurement data – over a 17-month period. Our analysis reveals that millions of domains – up to 5% of the DNS namespace – experienced a DDoS attack during the observation period. While most attacks did not severely impact DNS performance, in some instances, we observed 100-fold increases in DNS resolution time or complete unavailability (RQ5). Our measurements also captured a significant attack against a major provider in the Netherlands (TransIP) and attacks against Russian infrastructure during the ongoing Russo-Ukrainian war that began in 2022

The study presented in this chapter was carried out in early 2022 and was published in an academic conference [34].

6.1 Introduction

As discussed in chapter 1, Distributed Denial of Service attacks are one of the most critical threats on the modern-day Internet. They are cheap, effective, and keep growing in intensity [16, 90, 91]. DDoS attacks that impact the Domain Name System (DNS) are of particular concern, since DNS serves as a support infrastructure for most applications, content distribution platforms, and many security services [92]. Recall our statement at the start of this thesis: *If you can stop the DNS, you can effectively stop most Internet communication.*

The persistent DDoS problem triggers questions regarding how pervasive DDoS attacks against critical infrastructure actually are, and what impact they have. Attackers generally know they are launching an attack (although not necessarily how successful it is), and a victim of a successful attack is generally aware of it due to service impairment, but may not want to publicize that fact. But independent study of DDoS attacks at scale is a long-standing challenge. A third party has to contend with discerning an attack from myriad root causes of service impairment on the global Internet. Heavily capitalized players can put significant resources into monitoring millions of IP addresses in network traffic [93], but these approaches are difficult to scale, and not within reach of academic research efforts.

In this chapter we develop a scalable method to map DDoS attacks targeting or affecting DNS infrastructure. We use two unique macroscopic data sets to develop this mapping: the UCSD Network Telescope, which collects backscatter traffic from ongoing DDoS attacks against IPv4 address space, and the OpenINTEL measurement project, which performs daily DNS queries of over 65% of registered domains, allowing detection of substantial changes in DNS query latency or reachability to authoritative nameservers over time. Resolution times experienced by OpenINTEL during attacks indicate their impact on the DNS; network telescope traffic allows partial inferences of attack timing and intensity.

We expand on the following contributions:

1. We synthesized two data sets that capture global IPv4 behavior to discover evidence of attacks against tens of millions of domains ($\approx 5\%$ of the DNS namespace), although often with negligible performance impact.

2. We discovered attacks against DNS providers that impaired performance and reachability for millions of domains.

3. Our data confirms the effectiveness of the use of anycast and diversity in nameserver deployment in providing resilience against DDoS attacks.

4. We document corroborating evidence of politically motivated attacks on Russian infrastructure.

5. We analyze the limitations of our data sets to infer effectiveness of attacks, and propose approaches to overcome them in our pursuit of more accurate characterization of the DDoS ecosystem.

This chapter illustrates the value of combining longitudinal datasets in extracting cybersecurity-related insights into Internet evolution, in this case regarding the observable harms of DDoS attacks to performance and availability of critical services.

6.2 Background

6.2.1 DDoS Attacks

Distributed Denial of Service (DDoS) attacks are a notorious type of cyberattack. While conceptually simple, DDoS attacks can be highly effective in disrupting networks and denying users access to online services. Attackers are known to misuse core Internet infrastructure to bring about attacks, as well as target it. With society ever-increasingly relying on the Internet as its communications fabric, the persistent threat that DDoS poses to Internet stability and reliability is nothing short of grievous.

In chapter 2 we show how, by and large, attacks can be classified as *resource exhaustion*, *volumetric* and *semantic*. The prior involve using sheer host resources and network traffic volumes (e.g., TCP state exhaustion, high packet rate and/or byte magnitude). The latter involves abusing specific weaknesses (e.g., in L7 protocols) without relying on a high rates per se. We can further classify the attacks into distinct categories, namely: *unspoofed*, *reflected* and *randomly spoofed*. *Unspoofed* attacks involve sending network traffic directly from the attacking infrastructure (e.g., IoT botnet) to the victim host, without application of source IP address spoofing. *Reflected* (or indirect) attacks involve specific source IP address spoofing, to dupe so-called reflectors (e.g., open DNS resolvers) to send traffic to the victim host, in response to requests purportedly coming from the victim host. Finally, *randomly spoofed* attacks involve randomly (and often uniformly) spoofing the source IP address, in an attempt to conceal the attacking infrastructure.

Obtaining data on DDoS attacks is non-trivial. Inferring attacker behavior in the case of reflected attacks requires complex honeypot reflectors to mimic frequently used sources. Detection of spoofed attacks requires access to a large source of backscatter traffic, i.e., a large darknet. Finally, detection of unspoofed attacks requires the collaboration of victims and/or network providers, who are not generally sharing such data. The challenges with data access limit the ability of researchers to characterize the evolution of DDoS attacks on the Internet.

The two longitudinal data sets available to us allow a focus on randomly and uniformly spoofed attacks. Sizeable attacks of this type will use many spoofed IP addresses and thus appear as sourced from a wide range of networks, captured by both of our data sets.

6.2.2 DNS and IP Anycast

The DNS was already designed with reliability in mind. For example, RFC1034 [28] requires every zone the be available on at least two authoritative nameservers. RFC2182 [21] further recognizes that diversity in terms of topological and geographical placement of redundant servers increases reliability. Ironically, the number of root server IP addresses is capped at thirteen. In the early 2000s, however, operators of DNS root servers started distributing replicas of these servers around the world, for which they rely on IP anycast. We discussed in chapter 2 how IP anycast leverages the border gateway protocol to allow multiple server instances to use the same IP address. In the previous chapter, we showed the challenges of anycast deployment due to the requirement of specific knowledge and routing resources. While it is a great way to add resilience for critical infrastructure, arguably it may be superfluous for others. Finally, the DNS comes with caching mechanisms for performance, and to reduce the likelihood of resolution failure in case of intermittent connectivity issues. The strong rise in use of content delivery networks, however, reduces the effectiveness of caching, as CDNs typically configure lower cache lifetimes (i.e., time-to-live values) to aid with DNS-based load-balancing.

6.3 Related Work

DDoS Attack Detection Several studies focused on inferring DDoS attacks on the Internet. Moore *et al.* introduced the method to detect randomly spoofed denial of service attacks (RSDoS) leveraged in this chapter [94]. In their study, they explain that by monitoring a large address space, they can infer denial of service activity from the backscatter traffic observed in the Internet background radiation (IBR). Furthermore, they define threshold values to more accurately extract signals of attacks and eliminate sources of noise in the data. Leveraging this approach, Jonker *et al.* provided a macroscopic characterization of DDoS attacks on the Internet and investigated factors influencing migration to DDoS Protection Services (DPSs) [16]. Whereas Jonker *et al.* focused on hosting infrastructure, in this chapter we quantify and characterize the impact of attacks against DNS authoritative infrastructure.

Fachkha *et al.* designed an attack detection methodology using network telescopes by examining received DNS DDoS amplification attack traffic [95]. Other researchers have used honeypots to detect DDoS attacks.

Kramer *et al.* [43] developed AmpPot, a series of fake amplifier instances designed to monitor DDoS amplification and reflection attacks. Bailey *et al.* [96] designed an analogous system that used a two-tier approach with lightweight honeypots to monitor suspicious activity and high-end honeypots for behavioral analysis.

Attacks against DNS Infrastructure DNS infrastructure represents a frequent target of DDoS attacks. Moura *et al.* evaluated the impact of large DDoS attacks against the DNS root server infrastructure in November 2015 [19]. They investigated how different services respond to stress and the performance impact of policies and mechanisms deployed to handle the attack. Their analysis demonstrated the efficacy of anycast as a resilience technique against DDoS attacks. Our study corroborates their findings, showing that deployment of anycast for DNS nameserver infrastructure remains the best protection against DDoS attacks.

In 2018, Moura *et al.* studied the benefit of DNS caching for DNS services severely impacted by DDoS attacks [20]. Their controlled experiments showed that the presence of caching allowed almost all end users to tolerate attacks causing up to 50% of packet loss on the DNS infrastructure.

Several studies focused on a high-impact attack against DNS provider Dyn in October 2016 [14, 97, 98], characterizing its impact, effects on the global Internet ecosystem, and DNS customer behavior after the attack. Abhishta *et al.* further investigated Dyn DNS customer behavior before and after this attack, showing that customers that relied heavily on a single company for their authoritative nameservers switched to using other servers after successful attacks [88].

Four years after the Dyn attack, Kashaf *et al.* investigated third-party dependencies of modern web services. They showed that 89% of the Alexa Top-100K websites still critically depend on a third-party DNS, CDN or CA provider, despite the fact that such an exclusive dependency was why the Dyn attacks had such far-reaching effects on users [15].

DNS Resilience Focusing on DNS resilience, Allman analyzed the structural DNS robustness of the DNS authoritative ecosystem over 9 years, showing adoption of different resilience techniques by DNS operators [71]. In the previous chapter we expanded on this topic by providing an extensive characterization of the adoption of anycast in DNS authoritative infrastructure, showing a massive adoption for half of the domains measured. Akiwate *et al.* characterized the

#Attacks	#IPs	#/24 Prefixes	#ASes
4,039,485	1,022,102	404,076	25,821

Table 6.1: RSDoS Dataset: November 2020-March 2022

prevalence of lame delegations on the DNS ecosystem and their negative impact on resilience and performance [72]

6.4 Dataset

We join two primary, long-standing datasets for this study. To get indicators of DDoS attacks against IPv4 address space, we use inferences made from UCSD Network Telescope (UCSD-NT) data. To study which DNS authoritative nameservers exhibit performance degradations, we use contemporaneous DNS measurement data from the OpenINTEL project. We also use several ancillary datasets to support our analysis.

6.4.1 DDoS Attacks Inferred from Internet Background Radiation (IBR)

The UCSD-NT announces and captures traffic destined to two globally routed networks – a /9 and /10 address block, covering approximately 1/341 of the total IPv4 address space. The collected traffic is referred to as *Internet Background Radiation (IBR)*, a significant component of which is *backscatter*, including packets that are sent in response to randomly spoofed DDoS attacks. CAIDA curates the raw data to create a *Randomly and Uniformly Spoofed Denial of Service (RSDoS) attacks* feed that consists of a 5-minute tumbling window of aggregated statistics of response packets sent by victims of RSDoS attacks [27]. We use this data feed to establish a lower bound of DDoS attacks against specific IP addresses. In addition to a timestamp of each 5-minute window, this data set includes several fields that we use to characterize attacks: the number of /16 subnets in the telescope that receive packets from the inferred victim in the 5-minute window; protocol, first observed port, and number of unique ports targeted; and peak observed packet rate during the window. The first port indicates which service was under attack in single-port attacks. The RSDoS data contains 4,039,485 inferred attacks for November 2020 to March 2022 (Table 6.1).

6.4.2 OpenINTEL - Active DNS Measurements

OpenINTEL is a large-scale measurement platform that performs daily querying of all domain names registered under many top-level domains (TLDs), including all gTLDs participating in ICANN's Centralized Zone Data Service (CZDS) platform, legacy gTLDs (e.g., `.com`, `.net`, `.org`) and several ccTLDs (e.g., `.at`, `.nl` and `.ru`) [25]. OpenINTEL also measures domain names included in various Top lists. OpenINTEL performs several queries, including NS queries, for each domain name, and stores the round trip time (RTT) to complete the query, along with response status codes (e.g., NOERROR, SERVFAIL, TIMEOUT). OpenINTEL triggers explicit NS queries to deal with parent-child inconsistency, and prefers the authoritative answer as discussed in chapter 3. Explicit NS queries trigger a direct response from the targeted authoritative nameservers, providing us the effective RTT to reach them. The query process uses DNS resolver software *Unbound* [62] to randomly select an authoritative nameserver for the first query for every registered domain (i.e., excluding caching[1]). This *agnostic* resolver behavior captures the actual resilience mechanisms implemented by DNS operators, but also prevents us from identifying which specific authoritative nameserver responded to each query.

6.4.3 Anycast Census and Additional Datasets

We use quarterly census snapshots of anycast deployment taken from January 2021 until January 2022 using Manycast[2]. We identify DNS anycast deployments with the same approach as described in chapter 5. We also leverage CAIDA's prefix-to-AS dataset [26] to map IP addresses to the AS number(s), and CAIDA's AS-to-organization [84] to map AS numbers to organizations. Finally, we use the open resolver scans of Yazdani *et al.* [99] to filter out incidental IPs of open resolvers showing up in the DNS authoritative infrastructure.

6.5 Methodology

Our methodology consists of four steps:

1. Create an aggregated dataset of the OpenINTEL data;
2. Map IP addresses under attack to nameservers under attack;
3. Extract the list of domains associated with those nameservers;

[1] Additional queries may leverage cached NS or other records, providing a successful resolution of domains under attack reducing the visibility on the real impact of attacks.

4. Use the RTT data to infer performance impairment for queries to those domains.

Our analysis interval is the 17-month period from November 1, 2020 to March 31, 2022, which lines up with the anycast census data (§6.4.3).

6.5.1 Extrapolating DNS Performance Metrics

OpenINTEL does not record which authoritative nameserver provided the answer to a query for a specific domain, so we aggregated performance metrics for all IPv4 nameserver IP addresses in common for one or more domains, which we define as its *NSSet*. This aggregation allows us to estimate the impact of nameserver deployment scenarios on resolution performance. Each NSSet contains the IP addresses of the authoritative nameservers as well as their corresponding autonomous system number (ASN), prefix, and country code. For each NSSet, we collect, in a 5-minute interval (i.e., the same granularity as the RSDoS attack dataset), the number of domains resolved by OpenINTEL, and the average, minimum, and maximum RTT observed for that interval, and number of errors (e.g., Timeout, SERVFAIL, etc.). This data allows us to define the following metric for the impact of an attack on the RTT of queries for a domain, and thus the impact on end users:

$$Impact\_on\_RTT = \frac{Average\ RTT\ (5\,min)}{Average\ RTT\ (Day\ Before)} \tag{6.1}$$

Significant RTT increases above a baseline are indicative of either an attack causing network congestion or other path impairments. By joining the OpenINTEL data with RSDoS data, we can correlate RTT changes with inferred RSDoS attacks. We evaluated using different time-window metrics as a baseline (e.g., Average RTT (Week/Month Before)) finding similar results. We decided to stick with the previous daily metric to minimize errors due to infrastructural changes in the DNS hosting architecture. While averaging RTT may mask outliers, it provided us a stable metric to identify the impact of DDoS attacks.

6.5.2 Joining Datasets

Figure 6.1 shows how we join the RSDoS-compiled IPs under attack with the list of nameservers successfully queried on the day before the attack. This step maps IPs under attack to nameservers under attack. We join the resulting dataset with the list of domain names those nameservers hosted, as observed on the day before the attack. This step yields the list of domains under attack. By using the list of nameservers on the previous day, we minimize the chance of missing a nameserver that is unreachable due to an attack. We assume daily changes in

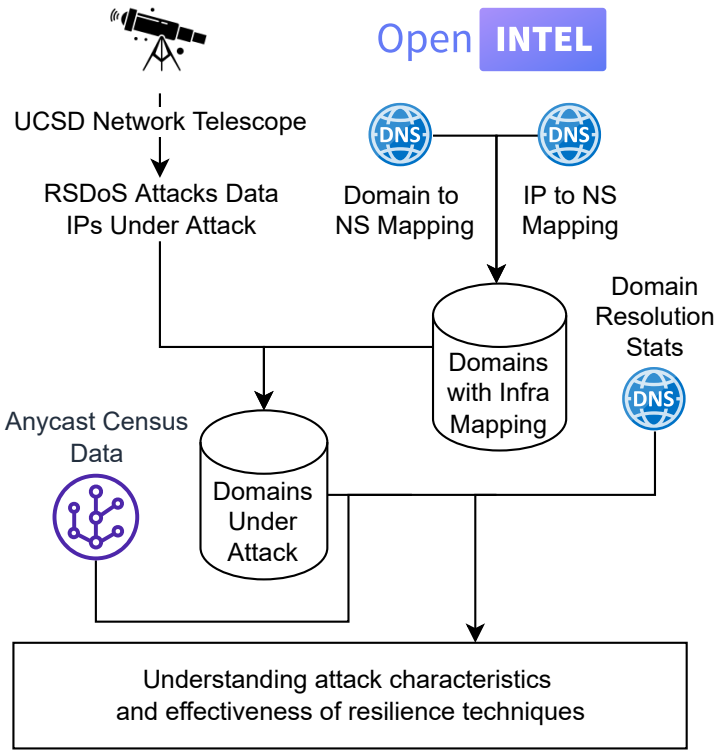


Figure 6.1: Data analysis pipeline: The RSDoS feed joined with the OpenINTEL measurement provides information on the impact on DNS infrastructure during DDoS attacks

nameserver infrastructure will not significantly affect our analysis. We join the list of domains under attack with our RTT data for NSSets. We use additional metadata (subsection 6.4.3) to characterize performance during various attack windows.

6.5.3 Limitations

The following limitations of our data sets constrain our inference capability. First, OpenINTEL's *agnostic* DNS resolution (§6.4.2) means that we cannot know which authoritative nameserver responded to a query. The random selection of authoritative nameserver means that eventually it should query each

nameserver, but it also restricts our ability to discern behavior of (and thus the impact of attacks on) different nameservers for the same domain. While this limitation does not allow us to pinpoint the behavior of single nameservers affected by the attack, it enables inference of how a typical end user would experience DNS resolution. Therefore, we can estimate a realistic worst case scenario of end-user experience in resolving a domain with an empty cache. This empty-cache configuration implies that the TTL value for a specific domain will not impact the resolution performance of OpenINTEL.

Second, OpenINTEL resolves domains using both IPv4 and IPv6 addresses, but the RSDoS data includes only IPv4 addresses. During an attack on IPv4 DNS infrastructure, separate parallel IPv6 infrastructure might be operational, limiting the impact of an attack. On the other hand, often IPv4 and IPv6 services share the same infrastructure and even server [100], in which case our inferences would hold.

Third, the network telescope detects only a specific kind of attack, which uses randomly spoofed IP addresses to launch a volumetric attack. During multi-vector attacks, we have limited visibility of overall attack duration and intensity. We also have no visibility into reflected and unspoofed attacks. As a relevant data point, Jonker *et al.* [16] compared two data sets of inferred attacks over two years, finding 60% of attacks as randomly spoofed (observed in RSDoS data), and 40% as reflected attacks (observed in the AmpPot data).

Finally, the single vantage point from which OpenINTEL queries in a highly complex Internet topology limits the precision of our visibility of the performance impact of attacks, especially in case of anycast deployments where catchment can mask ongoing attacks in specific geographic regions.

Reactive Measurement

To mitigate these limitations, we have built a reactive measurement platform that iteratively targets the full list of authoritative nameservers when resolving a domain name. Every time an RSDoS feed reports an attack, our platform joins the list of IPs under attack with the list of nameservers provided by OpenINTEL and the registered domain names that delegate authority to said servers. For every attack, we trigger probes of 50 related domain names every 5 minutes during the attack and in the 24 hours after (to characterize the post-attack baseline behavior). We choose this threshold to avoid additional burden on infrastructure already overloaded by attacks. Moreover, we spread our 50 measurements over the entire 5-minute window. We launched these measurements operationally in January 2022, so could not use them for our longitudinal analysis, but we did leverage them to study the impact of attacks against Russian infrastructure (subsection 6.6.2). Although our current infrastructures probes

Target Nameserver		A	B	C
December 2020 Attack	Observed Packer Rate (PPM)	21.8K	3.8K	2.9K
	Inferred Traffic Volume	1.4 Gbps	247 Mbps	188 Mbps
	Attacker IP Count	5.79M	1.57M	1.33M
March 2021 Attack	Observed Packer Rate (PPM)	125K	123K	13K
	Inferred Traffic Volume	8 Gbps	7.8 Gbps	845 Mbps
	Attacker IP Count	7M	6.19M	823K

Table 6.2: Attack metrics for two DDoS attacks on TransIP. The first attack targeted nameserver A more intensely; the second targeted all three similarly.

from a single vantage point in the Netherlands, we are in the process of acquiring additional vantage points to increase visibility of how attacks affect performance and availability in different geographic regions (e.g., due to anycast catchment).

We built our analysis pipeline using Kafka [101], Spark Structured Streaming [66], Apache Flume, and Grafana to display results. We use this pipeline to trigger reactive measurements with a maximum delay of 10 minutes after the start of an attack. In the future we can use this platform to perform near real-time characterization of DDoS attacks on DNS infrastructure.

6.6 Results: Case Studies

6.6.1 Large European Hosting Provider

Our first case study exemplifies how DDoS attacks can impact large providers, severely degrading DNS performance for end users. We focus on two attacks against TransIP, a large European DNS and hosting provider. Both attacks were reported [102, 103] and acknowledged by TransIP [104]. At the time of the two attacks (December 2020 and March 2021) TransIP was responsible for $\approx$ 8% of `.nl` domains, potentially affecting millions of users. By joining the two data sets, we infer that these attacks potentially affected $\approx$ 776K domain names, two-thirds of which ($\approx$510K) were `.nl` domains. At the time of the attacks (and still at the time of writing this thesis), TransIP used three unicast

IPs for nameservers for the domain names they hosted, all of which appeared as RSDoS attack targets (A, B, and C in Table 6.2).

In December 2020, the network telescope data shows evidence of RSDoS attack activity from 2020-11-30 at 22:00 to 2020-12-01 12:30 (UTC). We estimate an attack rate of 124Kpps (21.8K packets per minute at the telescope[2]) against nameserver A. Nameservers B and C seem to have experienced lower-intensity attacks (Table 6.2). The lower intensity inferred for B and C suggests low impact on overall DNS operations, but OpenINTEL measured a 10× increase in DNS resolution time, indicating significant impairment (recall that OpenINTEL randomizes which NS to query for each registered domain, thus over 770K domains it is overwhelmingly likely to send a similar number of queries to all three nameservers). The performance impairment ended on 2020-12-01 at 08:00, 8 hours after the RSDoS-inferred end of the attack (Figure 6.2). One explanation for this behavior is that the attackers moved to a different kind of DDoS attack not observable by the network telescope. Another explanation is the need for human intervention to restore DNS service quality.

[2]21.8kppm × 341 / 60s = 124K pps.

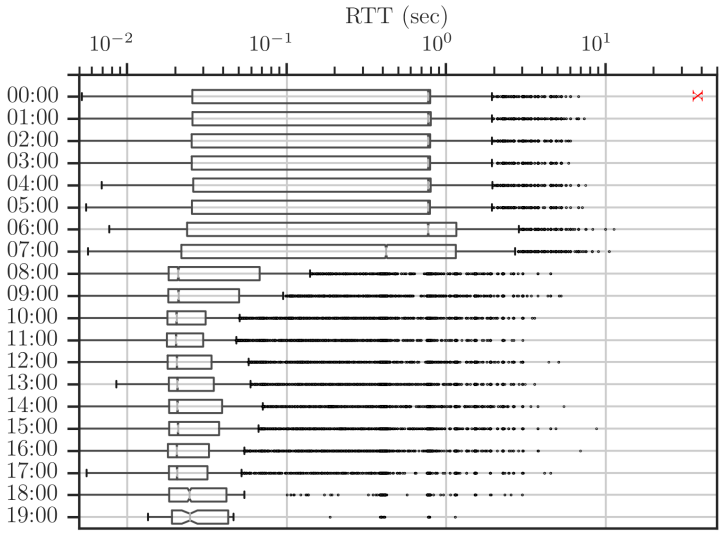


Figure 6.2: RTT variations in DDoS attacks on TransIP. The attack hours are marked with a red cross. Effects of the December attack persisted for hours after the RSDoS-inferred end of the attack.

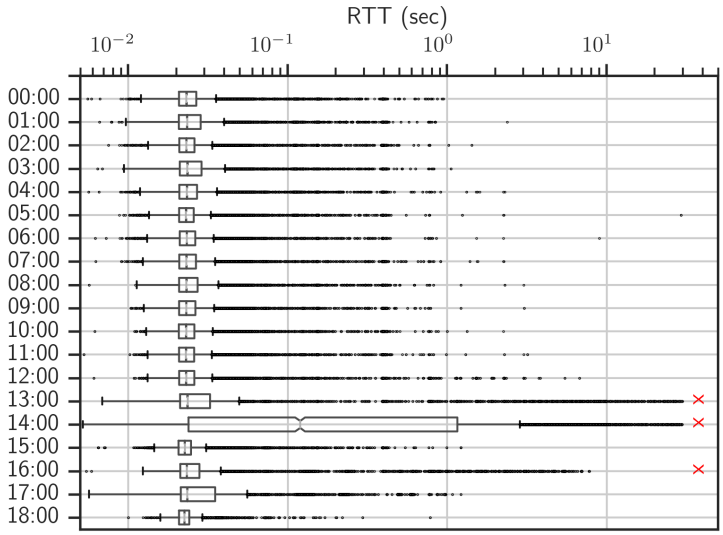


Figure 6.3: RTT variations in DDoS attacks on TransIP. The attack hours are marked with a red cross. The March attack induced larger RTT impairments.

In their report, TransIP stated that the March 2021 attack was more intense [104]. Consistently, the telescope observed a peak packet rate 6× greater than for the December 2020 attack and, as shown in Figure 6.4, ≈ 20% of OpenINTEL queries timed out and failed to resolve (compared to a negligible fraction in December). The March attack more likely impacted end users because it induced complete unreachability of domain names. Nevertheless, differently from the December attack, the time frame in which we observed an impact matched the interval inferred through the telescope data (Figure 6.3). This difference might be associated with TransIP's reported use of a DDoS protection/scrubbing mechanism [104]. Since OpenINTEL observed no evidence of nameserver changes during the attack, we speculate that the scrubbing service might have been deployed at the IP level.

Resilience Techniques Adopted by TransIP

This case study shows that even with traffic scrubbing, DDoS attacks can affect resolution for hundreds of thousands of domains. More strategic deployment of DNS infrastructure would have improved its resilience. TransIP served the registered domains using three unicast authoritative nameservers, on three differ-

ent subnets, in two separate geographic locations (Amsterdam and Eindhoven), behind a single ASN. While hosting these nameservers behind different subnets increased resilience, the lack of anycast deployment left these domains dependent on three physical servers and (at most three) network links. Moreover, hosting these domains within a single ASN means they relied exclusively on a single company's infrastructure. Using a more diversified infrastructure, by using anycast and/or third-party hosting providers, would have further mitigated the effects of these attacks. Finally, our analysis shows that ≈ 27% (203,217) of the domains hosted by TransIP relied on third-party hosting for their web content. These domains likely felt the December attack less, i.e., simply experienced slower DNS resolution but during the March attack they likely became entirely unreachable due to DNS resolution failures, despite having a third party operating their web site.

6.6.2 Attacks on Russian Assets in 2022

The TransIP case illustrates a type of attack against commercial infrastructure, whereas in this second example we discuss attacks targeting infrastructure hosting specific domains and likely motivated by political reasons. Specifically, we focus on several attacks targeting Russian government web sites in March 2022, shortly after Russia's invasion of Ukraine.

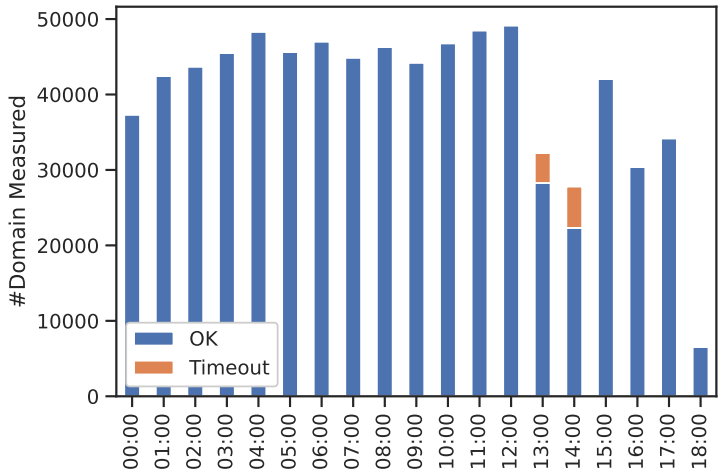


Figure 6.4: Timeout errors during the March 2021 attack on TransIP reached 20% of observed domains, leading to resolution failures for end users

Russian Ministry of Defense

The first attack targeted `mil.ru`, the domain of the Ministry of Defense of the Russian Federation. Three nameservers on the same /24 subnet were authoritative for both the international and the Cyrillic IDN name of mil.ru and for several subdomains.

These three nameservers were under attack for 8 consecutive days, March 11-18, according to RSDoS inferences from the network telescope data. The telescope detected a modest-intensity attack, although newspapers reported a severe attack on the network infrastructure of mil.ru and other government web sites [105, 106]. Newspapers also reported the geo-fencing of mil.ru in response to the attack, allowing connections only from within the Russian state. OpenINTEL completely failed to resolve mil.ru (and the related Cyrillic domain) for most of the attack period (from March 12 to 16, inclusive), whereas our reactive measurements (§6.5.3) found the domain unresolvable for the entire duration of the attack, with none of the three nameservers responsive.

RDZ Railways

Another interesting case study is related to RDZ railways. RSDoS data indicates an attack from 15:30 to 20:45 on March 8, 2022. Our reactive measurement system launched queries to resolve the domain in the 24 hours following the start of the attack, and found the domain became intermittently responsive at 06:00 on the next day. We also found evidence that this attack was co-coordinated via a Telegram channel named *IT ARMY of Ukraine* (Figure 6.5). A message

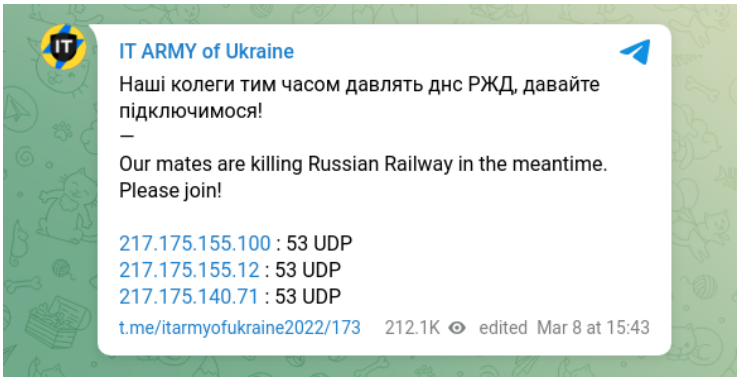


Figure 6.5: Telegram coordination of the DDoS attack. We manually inspected the messages to find evidence of correlation.

on the channel at 15:43 provided the IP address of the 3 RDZ railway DNS nameservers, asking for assistance to crowdsource an attack on port 53/UDP, 12 minutes after RSDoS-inferred start time of the first attack.

Resilience Techniques Adopted by Russian Infrastructures

The attack on `mil.ru` is a textbook illustration of poor resilience in DNS infrastructure. The three nameservers were unicast, hosted behind the same ASN/-company, and even on the same /24 subnet. This lack of resilience contributed to the attack's success, apparently forcing the Russian government to geofence the entire network to protect DNS service. Moreover, being hosted on the same /24, the three nameservers (and other services on the same subnet, including the `mil.ru` web site) likely shared upstream network resources. This network bottleneck implies that a single intense volumetric attack targeting a service on the network can affect all services hosted on the network. The RDZ railways domain had a slightly more resilient deployment. The three nameservers were hosted on two separate /24 subnets, but still used unicast and a single ASN. However, as in the mil.ru case, the attacker targeted all three nameservers, and simple prefix diversity was not sufficient to withstand the attack.

6.7 Longitudinal Attacks Analysis

Although the case studies illustrate the value of joining these two data sets together to corroborate known attacks, our ultimate goal is to use this method to identify and track the prevalence and scope of unreported attacks against global DNS infrastructure in the wild. We used data from November 2020 to March 2022 to identify all RSDoS-inferred attacks against DNS infrastructure, either directly targeting nameserver IPs or targeting /24s that host nameservers.

ASN	#Attacks	Company	ASN	#Attacks	Company
15169	7,324	Google	16509	1,564	Amazon
46606	2,841	Unified Layer	8068	1,240	Microsoft
13335	2,428	Cloudflare	54113	1,054	Fastly
16276	2,192	OVH	199608	894	Birbir
24940	2,172	Hetzner	48678	562	Pendc

Table 6.3: Top 10 ASNs attacked from November 2020 to March 2022. Large DNS hosting companies and cloud providers are the most frequent targets, usually with negligible effects.

Year	Month	#DNS Attacks	#Other Attacks	Total Attacks	DNS IPs	Other IPs	Total (Unique) IPs
2020	11	2,550 (1.63%)	156,884 (98.37%)	159,434	798 (1.64%)	47,839 (98.36%)	48,637
	12	3,876 (1.08%)	356,042 (98.92%)	359,918	1,070 (0.94%)	113,354 (99.06%)	114,424
2021	1	2,927 (1.68%)	171,089 (98.32%)	174,016	930 (1.43%)	63,971 (98.57%)	64,901
	2	2,873 (1.98%)	141,949 (98.02%)	144,822	827 (1.52%)	53,461 (98.48%)	54,288
	3	3,294 (1.18%)	276,503 (98.82%)	279,797	929 (0.52%)	177,514 (99.48%)	178,443
	4	3,522 (2.12%)	162,361 (97.88%)	165,883	802 (1.36%)	58,077 (98.64%)	58,879
	5	3,973 (1.99%)	195,540 (98.01%)	199,513	880 (1.19%)	72,899 (98.81%)	73,779
	6	2,244 (0.98%)	227,874 (99.02%)	230,118	821 (0.96%)	84,294 (99.04%)	85,115
	7	2,245 (0.66%)	335,948 (99.34%)	338,193	967 (0.91%)	105,917 (99.09%)	106,884
	8	4,473 (1.53%)	288,369 (98.47%)	292,842	1,055 (1.14%)	91,517 (98.86%)	92,572
	9	2,577 (1.05%)	242,713 (98.95%)	245,290	780 (1.12%)	68,561 (98.88%)	69,341
	10	1,968 (0.86%)	226,124 (99.14%)	228,092	624 (1.25%)	49,310 (98.75%)	49,934
	11	2,662 (0.94%)	281,907 (99.06%)	284,569	835 (1.06%)	77,942 (98.94%)	78,777
	12	2,984 (1.35%)	218,070 (98.65%)	221,054	706 (1.04%)	67,422 (98.96%)	68,128
2022	1	2,028 (0.86%)	232,999 (99.14%)	235,027	705 (1.23%)	56,616 (98.77%)	57,321
	2	1,368 (0.57%)	238,407 (99.43%)	239,775	572 (0.88%)	64,201 (99.12%)	64,773
	3	3,294 (1.37%)	237,848 (98.63%)	241,142	669 (0.94%)	70,778 (99.06%)	71,447
Total		48,858 (1.21%)	3,990,627 (98.79%)	4,039,485	8,864 (0.87%)	1,013,238 (99.13%)	1,022,102

Table 6.4: Monthly attack activity summary. Attacks toward IPs used as DNS nameservers constituted $\approx 1-2\%$ of the total attacks.

6.7.1 Overview of Attacks in 2020-2022

Table 6.4 shows that attacks on the DNS infrastructure are between the 0.57% and 2.12% of total attacks detected by the telescope, spanning $\approx 1-2\%$ of the total affected IPs. Although this is a small percentage of the total number of attacks, the IP addresses may be nameservers that host millions of domains. We focused on attacks directly targeting nameserver IPs (rather than the containing subnet or announced prefix).

Figure 6.6 shows the monthly counts of *potentially affected* domains, i.e., one of its nameservers was under attack. On average, 10-100 domains were potentially affected by attacks, although much larger numbers are common. We also estimated the attack's intensity and the handling capacity of the target infrastructure. We identified 8 peaks of potentially 10 million domains affected – a series of attacks trying to target around $\sim 4\%$ of the global DNS infrastructure measured by OpenINTEL. These specific attacks did not substantially harm the performance and operation of these large providers.

We analyzed which companies received the most attacks during our measurement window, finding spikes against Cloudflare and Google DNS infrastructure (Table 6.3). We analyzed the target IPs for these attacks (Table 6.5) and found they related to Google's public DNS service (8.8.8.8 and 8.8.4.4) and Cloudflare's Quad1 (1.1.1.1). We see these open resolver nameserver IP addresses in our data due to misconfigured domains pointing their NS records at these IPs. We filtered out such attacks toward open resolvers, since they are not used for authoritative DNS resolution.

IP	#Attacks	Type
8.8.4.4	2,803	Google DNS
REDACTED	2,566	Unified Layer
8.8.8.8	2,298	Google DNS
1.1.1.1	1,118	CloudFlare DNS
204.79.197.200	668	Bing
194.67.7.1	481	Beeline RU
13.107.21.200	438	Bing
REDACTED	400	Company NAS
REDACTED	346	Private IP
23.227.38.32	273	Cloudflare

Table 6.5: Top 10 IPs attacked. The presence of open resolver IP address (8.8.4.4, 8.8.8.8, 1.1.1.1) on this list implies that misconfigured domains use them as authoritative NS resolvers. Attacks on such heavily provisioned anycast targets are likely ineffective.

We noticed many low-impact attacks against a shared IP address hosted on Unified Layer. After manual inspection, we discovered the IP address has been hosting the web site of an American Youtuber. VirusTotal [107] suggests that the address may have been used in the past for malicious activities. We also found evidence of several attacks against a Russian DNS provider, Beeline, during March 2022. Beeline provided DNS hosting for several Russian banking web sites such as Sberbank, Russian Agricultural Bank, and Eurasian Development Bank. Finally, we found targeted IPs related to Bing and Cloudflare and two private IPs likely related to misconfigured servers leveraged for attacks, belonging to private companies which we omit from our report.

Key Takeaway: *Approximately 0.5-2% of RS-DoS attacks observed by the network telescope reached, and perhaps targeted, DNS infrastructure. Some of these attacks hit deployments of 10 Million domain names, with negligible performance impact. Frequent targets included open resolvers, large DNS providers, and hosting companies.*

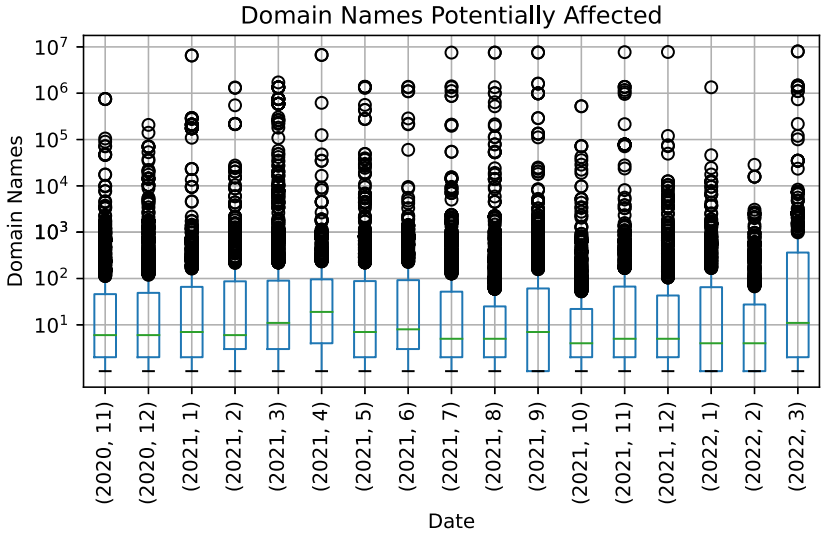


Figure 6.6: Registered domains potentially affected by attacks, by month. Some attacks hit deployments serving more than 10 million domain names.

6.7.2 Targeted Services (Ports)

In our analysis of protocol and port usage by the attacks, we found that 80.7% of attacks to DNS authoritative infrastructure targeted a single port and protocol (Figure 6.7). Almost 90.4% of these attacks used TCP (mainly TCP SYNs), 8.4% targeted UDP ports, and 1.2% used ICMP. Historically, DNS was a service provided via UDP. But in the last decade, the introduction of DNSSEC and its need for larger responses led to expanded support for DNS-over-TCP. This fact

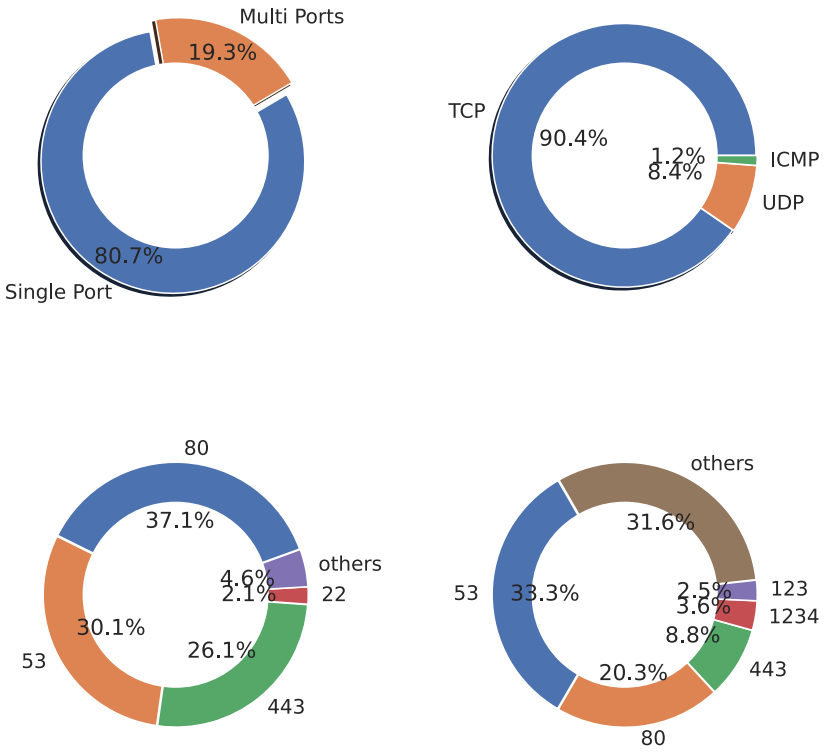


Figure 6.7: Distribution of protocol and destination ports used by attacks. Most attacks targeted a single port, usually via TCP. The most targeted port was 80 (HTTP), followed by 53 (DNS).

and the popularity of TCP SYN flood attacks increased the prevalence of TCP-based attacks. The port distribution varies: 37% of TCP-based attacks targeted port 80 (HTTP), and 30% targeted port 53 (DNS); the other significantly used port was 443 (HTTPS). One explanation for the use of HTTP(s) ports against DNS nameserver IPs is the awareness that sometimes the same IP address hosts both DNS and web services. UDP-based attacks exhibited a more varied port distribution, but one-third of UDP attacks targeted port 53.

What stands out is that *the majority of attacks do not target port 53*. This suggests that DNS itself may not be the primary target of these attacks although without knowing an attacker's motivations, we cannot be certain of this. Regardless of whether DNS is the target, if the goal of the attacks is to flood the link of the target or to exhaust system resources of the target host, they may still have an impact on DNS resolution. We discuss this further when we consider *successful attacks* in section 6.7.3.

Key Takeaway: *Most attacks towards DNS authoritative nameserver IPs targeted a single port, usually via TCP. The most attacked port was 80 (HTTP), followed by 53 (DNS).*

6.7.3 Performance Impact of Attacks

To assess the performance impact of attacks on DNS infrastructure, we computed a longitudinal 5-minute performance metric based on OpenINTEL's RTT measurements for each NSSet deployment (described in §6.4.3). To reduce possible sources of noise, we considered only NSSets with at least five domains measured during the attack. Using this constraint, we inferred 12,691 distinct events of attacks to distinct NSSets in the window where OpenINTEL actively measured domains for which the targeted nameserver(s) were authoritative.

Complete Failure in Resolution of Domain Names

In 99% of these 12,691 attacks, authoritative nameservers continued to provide the answer. However, in 1% of cases we saw domains fail to resolve, with timeout (92%) or SERVFAIL (8%) errors. This result shows that despite most attacks not harming operations, some caused end user failure in resolution (e.g., the TransIP examples discussed in §6.6.1).

Figure 6.8 shows the failure rate as a function of the number of domains resolved by OpenINTEL (y-axis); the dot's color represents the number of domains (order of magnitude) hosted by the NSSet under attack. Most domains that failed to resolve belonged to small infrastructures. Some attacks induced resolution failures (timeout errors) on large infrastructures hosting $>$ 10k domains. The most effective attack in this size range causing failed resolution

for 100% of domains belonged to nic.ru, a Russian registrar. They offer secondary nameservers as a service; those nameservers were attack targets during March 2022.

Most effective attacks occurred against smaller deployments (100-10k domains). A Spanish ISP (Euskaltel) responsible for 1,405 domains failed to respond to 83% of queries for its domains during the attack. 99% of domains that failed resolution in this way used unicast nameserver infrastructure.

The impact on end-users in cases of complete resolution failure depends on several factors, mainly related to caching policy. A popular domain (i.e., queried frequently, available in most caches) with a high TTL value may be less affected than a less popular one.

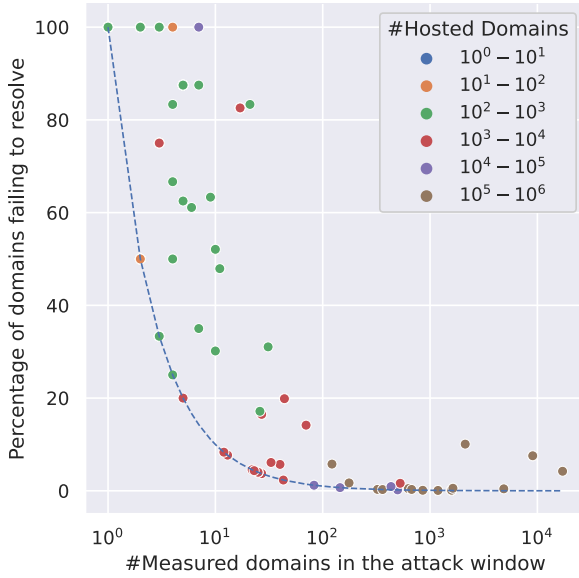


Figure 6.8: Percentage of measured domains failing to resolve, colored by number of hosted domain names. The base curve represents a single failure per attack window, everything above this line represents NSsets that experienced failures for multiple domains. Attacks with higher induced failure rates cause complete unreachability for end users.

We also consider the targeted port for successful attacks and ask: *are successful attacks more likely to specifically target DNS service ports?* Recall from section 6.7.2 that the majority of attacks do not target port 53. When we look at the port distribution of successful attacks, we see that the port distribution looks different: 49% of attacks target port 53 (DNS), 31% target port 80 (HTTP) and 11% target port 443 (HTTPS). While the fraction of attacks that are successful (i.e., they lead to resolution failures) is small, the difference in port distribution suggests that successful attacks are more likely to be specifically targeting the DNS. We speculate that this result is related to application-aware attacks, where attackers try to overload both the network and the application (i.e., DNS authoritative software). Nevertheless, there are also other types of attacks that lead to a breakdown in name resolution, and there may be parallel attacks going on that we cannot observe through the network telescope.

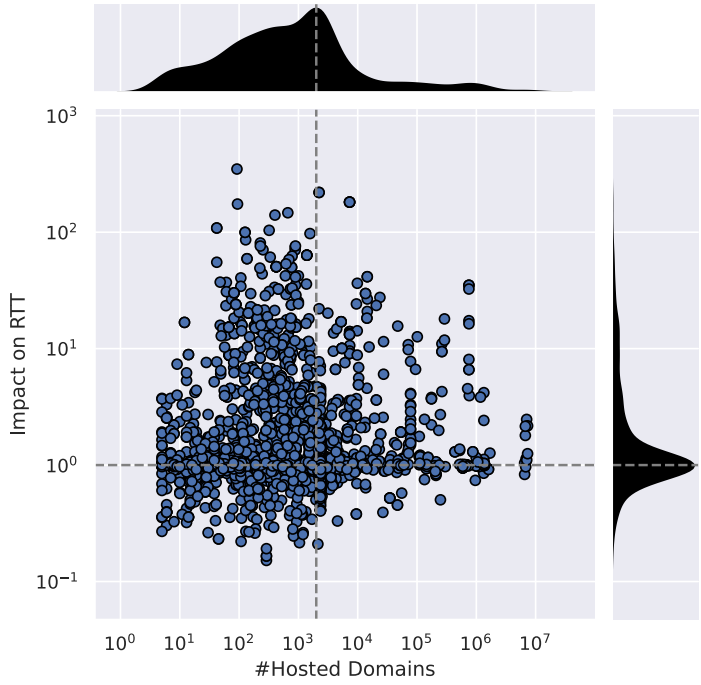


Figure 6.9: RTT impact vs. number of hosted domains. Most attacks had negligible impact on DNS resolution performance. However, some attacks led to peaks of 100-fold increase in resolution time, likely affecting end users.

Resolution Performance Impairments

Figure 6.9 shows the consequences of DDoS attacks in terms of RTT increase on different hosting sizes of NSSets. Most attacks did not cause observable impairments, but ≈ 5% of them (585) induced a 10-fold increase in RTT on 616 NSSets. In one-third (198) of these 585 attacks mentioned before, we see RTT peaking at more than 100-fold the baseline RTT. These high-impact attacks concentrated mainly on small-medium size infrastructure, hosting between 100-10k domain names. We also saw evidence of attacks on very large infrastructures (10M domain names); these manifested a smaller increase of 2-3 times the original RTT.

Table 6.6 shows the most affected companies we investigated, by ASN and associated NS name. NForce B.V, a Dutch hosting provider was the most affected, followed by another Dutch company Co-Co NL. The third one Nordisk Media Utveckling is a Swedish company responsible for registration of popular and trademark-protected domains. We also found some general VPS/cloud providers (e.g., Hetzner, My Lock, DigiHosting, Linode, ITandTEL) and large DNS hosting provider GoDaddy. Interestingly, we also found an attack against Apple Russia's ASN on Jan 21, 2022, well before Russia's February invasion of Ukraine and related attacks.

Key Takeaway: *Most attacks were ineffective, but some attacks had a critical impact, causing complete failure or dramatically increased latency of resolution.*

Impact on RTT	Company
348×	NForce B.V.
219×	Co-Co NL
181×	NMU Group
174×	Hetzner
146×	My Lock De
140×	DigiHosting NL
100×	Apple Russia
76×	GoDaddy
75×	Linode
74×	ITandTEL

Table 6.6: The most affected companies in terms of RTT increase. The vast majority are small to medium size DNS hosting providers.

6.7.4 Attack Inferred Intensity Correlation

Correlating attack intensity inferred by the network telescope with impact on DNS infrastructure is non-trivial. In some cases the network telescope may observe only one low-intensity vector of a high-impact multi-vector attack. Or vice-versa, high-intensity attacks targeting large and redundant infrastructures may have little impact. Although the case studies we examined were clearly observable in the network telescope, overall we did not see a strong correlation using Pearson's coefficient between RSDoS impact metrics and observable impacts on DNS resolution performance (Figure 6.10).

We also saw high bandwidth (high packets/min) attacks on DNS infrastructure that continued to operate well. Attacks with low intensity as inferred by the telescope sometimes matched higher spikes in RTT of domain resolution.

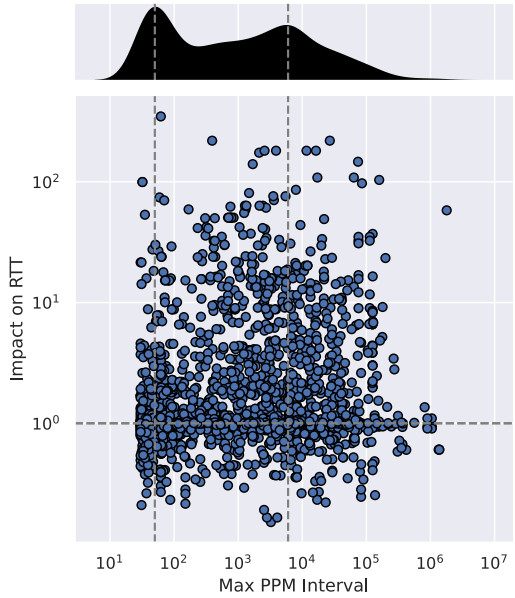


Figure 6.10: We found low correlation between RTT Increase and Attack Intensity, implying that infrastructure handling capacity and deployment of resilience techniques play a fundamental role in withstanding DDoS attacks. Telescope data serves as a useful signal of ongoing attacks, and as an indicator of where to perform additional measurements, without providing exhaustive information on attack intensity.

We speculate there may be two underlying causes: deployment of resilience techniques, which mitigates the performance degradation induced by attacks; and multi-vector attacks not fully detected by the telescope. We also found no correlation between the RSDoS-reported number of attackers and DNS resolution performance impact or failure. We saw a bimodal distribution centered around 50 PPM (inferred to be 17K PPM after interpolating from the telescope address space to the entire IPv4 space) and 6000 PPM (inferred 2M PPM).

Key Takeaway: *Telescope data reveals signaling of ongoing attacks but does not enable prediction of performance impact.*

6.7.5 Attack Duration Correlation

Figure 6.11 correlates attack impacts on resolution time and duration. The attack durations show a bi-modal distribution with modes at 15 minutes and 1 hour. High-impact attacks characterize these two intervals of the distribution, whereas attacks with longer duration have decreasing impact. Attacks may

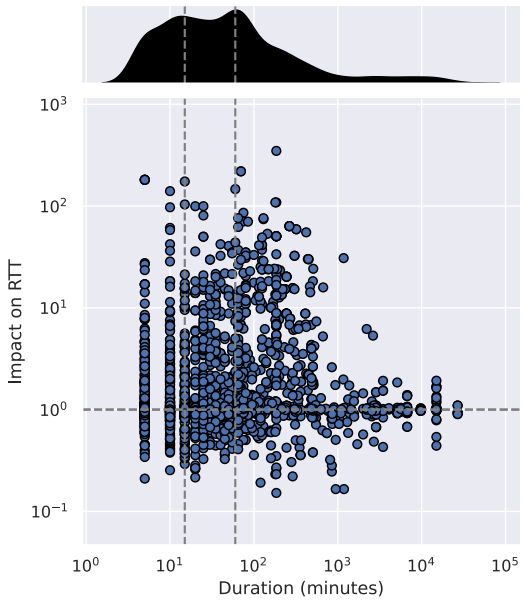


Figure 6.11: Correlation between RTT Increase and Attack Duration. Attacks are generally short lived, but the longer the attack lasts, the more likely RTT increases will impair performance for end users.

be short duration for several reasons, including that the attack succeeds and impedes responses that serve as backscatter signal, or that part of the attack is not visible to the network telescope. Long-term ineffective attacks could just represent background Internet noise. The only exception that we found was an attack against a German cloud provider, Contabo, which lasted 19 hours with a 30× increase in resolution RTT.

Key Takeaway: *Attacks with impacts on DNS are generally short-lived with an average duration between 15-60 minutes.*

6.7.6 Resilience Technique Efficacy

The impact of a DDoS attack is strongly related to the resilience techniques deployed. We analyzed several DNS resilience techniques to identify their possible effects on attack mitigation.

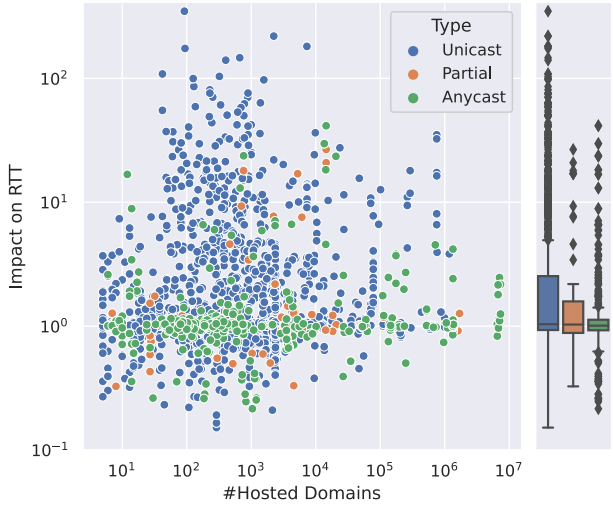


Figure 6.12: Efficacy of anycast as a resilience technique. The impact of attacks on RTT increase for unicast hosted domains is generally higher. No DNS infrastructure experiencing 100-fold RTT increase was using anycast.

Anycast vs. DDoS

Anycast deployments tend to suffer less under attack, (i.e., RTT increase 1-1.5 – Figure 6.12). Partial anycast deployments (i.e., anycast deployed only on a subset of authoritative nameservers) show attacks having a small impact on the infrastructure. Most effective attacks are related to nameservers hosted on unicast infrastructure (§6.7.3). In most cases of resolution failure, the domains relied on a unicast deployment. This result lends further support for the best practice of using anycast as a resilience technique against DDoS attacks.

AS Diversity

We did not find a clear link between AS diversity and effectiveness against DDoS attacks, but it seems to make more of a difference for larger ASNs (∼1M domains), as shown in Figure 6.13. The graph shows the single behavior of a

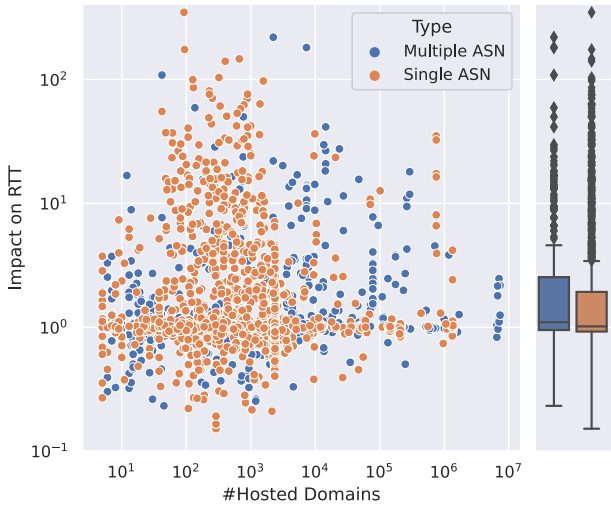


Figure 6.13: AS Diversity efficacy as a resilience technique. NSsets that host a higher number of domains are more likely to have multi-AS deployments, but these alone do not provide a significant level of protection compared to single-AS deployments.

multi-variable system in terms of resilience technique (Anycast, AS Diversity, Prefix Diversity), which generally combine to reduce the impact of attacks. However, in cases of complete failure of reachability of domains, most of those domains (81%) relied on a single ASN deployment (§6.7.3).

24 Prefix Diversity

Nameservers hosted on a single /24 prefix are likely hosted on the same network infrastructures (i.e., L2 switch, upstream router, etc). Figure 6.14 shows that using a single /24 prefix is generally the worst approach for deploying nameservers. Using two or more prefixes contributes significantly to resilience. §6.7.3 showed that 60% of NSsets that experienced failures were NSsets that relied on a single prefix. For most domains in this subset that experienced a complete failure in resolution, where 100% of queries failed to resolve, 30% of

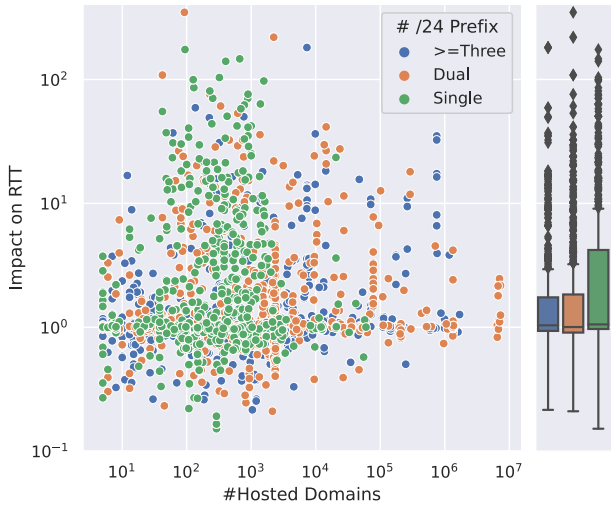


Figure 6.14: /24 Prefix Diversity efficacy as a resilience technique. Our data set shows that using a single unicast prefix to serve DNS infrastructure was likely the worst decision in terms of resilience. See Figure 6.12 for evidence these were mostly unicast prefixes.

their NSsets were deployed on two prefixes and just 10% of their NSsets were served by three or more prefixes.

Key Takeaway: *Anycast deployments suffer less from attacks, indicating increased DNS infrastructure resilience. Hosting nameservers across multiple prefixes or multiple ASNs also appears to provide increased resilience to devastating attacks.*

6.8 Ethical Considerations

One ethical concern in studying the impact of volumetric attacks on critical infrastructure is to avoid measurements that induce additional burden to infrastructure under attack. For this reason, analyses mostly relied on existing operational data collections that have little to no impact on the attacked infrastructure or end users, e.g., RSDoS (based on passive traffic observations), OpenINTEL (a lightweight probing architecture), Prefix2AS (leveraging RouteViews BGP collection infrastructure). We also developed a reactive measurement system to measure DNS infrastructure inferred to be under attack. To avoid causing harm by performing these measurements, we limited our query rate to 50 domains every 5 minutes for each IP under attack. Moreover, the system distributes these 50 queries evenly across the 5-minute interval, sending approximately one query every 6 seconds, a negligible load on nameserver infrastructure.

Another ethical concern is public exposure of IP addresses that are the target of successful attacks. One might argue that exposing these IP addresses might increase the chance of future attacks against them. To overcome this concern, we decided not to reveal IP addresses in this work but did mention the associated companies. The only exception we made relates to already public information on the Internet (e.g., newspapers, tech reports).

6.9 Concluding Remarks

Calls for adoption of techniques to support resilience of DNS infrastructure began decades ago, starting with classical topological redundancy as described in RFC2182 [21], and more recently with anycast techniques [108]. Our results, including the lack of correlation between inferred attack intensity and actual impact on DNS users, provide evidence to support the relative effectiveness of such techniques at a macroscopic level. Well-provisioned and strategically designed DNS nameserver infrastructure can withstand severe attacks with negligible impairments to end users. On the other hand, even small attacks can pose risks to organisations that neglect to architect resilience into their critical DNS infrastructure. Our analysis corroborates the importance and refines

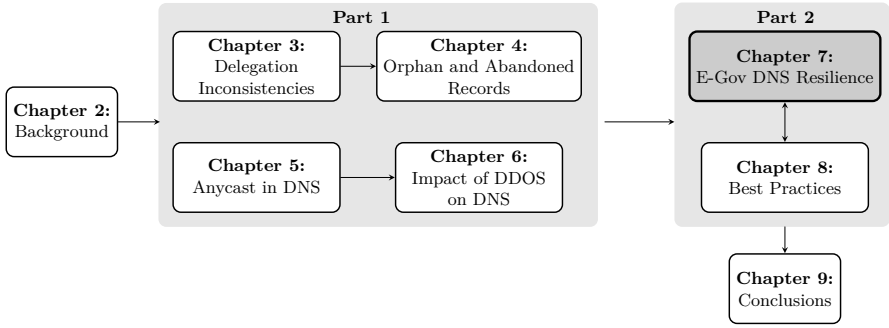
the prioritization of several actionable recommendations for DNS infrastructure providers:

- Distributed anycast deployment is generally the most effective method to operationally mitigate the effects of DDoS attacks on end users (§6.7.6).
- Classical DNS resilience strategies, i.e., prefix and AS diversity, also contribute to resilience, although in our data set these techniques appear to provide fewer benefits to resilience relative to anycast.
- It is sensible for operators of small nameserver deployments to rely on multiple third-party large infrastructures to provide backup resilience.

With regard to the overarching goal of achieving a collective understanding of the DDoS landscape for network operators, policy makers, and researchers, we have demonstrated the importance of continuous monitoring of the global DNS infrastructure, including correlating macroscopic feeds of attack inferences with active measurements that capture evidence of impaired user experience.

CHAPTER 7

e-Government DNS Resilience



In previous chapters, we examined the configuration of DNS structures, emphasizing the critical importance of implementing multiple levels of redundancy within the system. We proved that this redundancy enables the DNS infrastructure to withstand stress events, including DDoS attacks. Although best practices for this have been established by the operator community, adopting all of them requires expertise and resources.

In this chapter, we examine the adoption of these resilience techniques in setups that are of critical importance to society, such as government services. We analyze the DNS structuring of e-government domain names used by four countries (The Netherlands, Sweden, Switzerland, and the United States) and show the adoption of best practices (RQ-4), inter-country differences, and provide recommendations to improve DNS robustness.

This study was conducted in early 2022 and was published in an academic conference[35].

7.1 Introduction

Governments increasingly use digital avenues for communication with citizens and residents, further solidifying the Internet as core communications fabric of modern societies. Electronic government (e-government) refers to the set of services governments offer online to their citizens and residents [109]. E-government has the potential to save costs and provide faster service, easing access to people with disabilities or mobility challenges. The COVID-19 pandemic has exacerbated the payoffs of investments in e-government, by allowing parts of government services to operate normally despite all restriction measures [109, 110, 111], such as lockdowns and social distancing.

E-government depends on the Internet, which in turn relies on the Domain Name System (DNS) [28, 39] as one of its core services. Every web page visit or e-mail that is sent requires DNS resolution. As shown before, if part of the DNS fails – be it by accident or intentionally as a result of malicious action – domains can become unreachable.

As such, given that e-government also depends strongly on the DNS, proper operation of the DNS is vital to keeping e-government services accessible. E-government DNS structuring should therefore be resilient against (partial) failure to avoid service interruption. As we saw in the previous chapters, increasing resilience is not easy. The DNS is prone to many types of configuration errors, which can lead to service unreachability. While best operational practices exist to help increase DNS resilience, some techniques require expert knowledge, operations, and resources – all of which can complicate adopting said practices.

While there have been various studies analyzing DNS infrastructure (e.g., [15, 19]), few focused specifically on the DNS infrastructure resilience of e-government domains (e.g., [112]). In this chapter, we quantitatively study and evaluate the infrastructure of e-government DNS, for both web and e-mail services, with regard to DNS and IP-based redundancy. Our goal is to approximate what could happen if these services were to suffer stress events, such as DDoS attacks. We compare the e-government DNS infrastructure of multiple countries, leveraging the access to the list of e-government domains that we have. We study three countries in continental Europe (the Netherlands, Sweden, and Switzerland) as well as the United States in North America. We obtain the lists of e-government domain names for these countries and use active measurements to evaluate DNS configuration and structuring (section 7.4).

We show that 80% of `.gov` domain names carry the risk of relying on a *single* DNS provider (section 7.5). For each of the three continental European countries, roughly 40% of the respective domains do so. This risk can be easily remediated by adding additional DNS providers to e-government domain names. Moreover, the vast majority of domains of all countries are concentrated in a

handful of providers. The top five providers for each of the four countries are almost exclusively from specifically the country in question.

We also evaluate e-government domains IP-anycast based replication and DNS caching. We show that for most of the continental European countries we studied, e-government domains are not replicated with IP anycast, whereas `.gov` domains are. The prior should also employ IP anycast for their DNS to increase their resilience. With regards to caching, we show that much of the e-government domain DNS infrastructure is configured to not leverage most of the caching features, by setting very low time-to-live (TTL) values for the authoritative name servers. Lastly, we analyze the DNS configuration of e-government domains for incoming mail exchange and investigate which providers domains rely on to this end. We show that Microsoft tops the list for all four countries (section 7.7).

Overall, we find notable differences among the four evaluated countries with regards to replication of DNS infrastructure. Swiss e-government domains are lagging behind the other three countries in the use of several best practices.

7.2 Background

In this chapter, we analyze the deployment and structuring of authoritative DNS servers in e-government. *Authoritative DNS servers* (ADNS, for brevity hereafter), are those that know the contents of a DNS zone from memory [46].

ADNS have a central role in DNS resolution. Without them, all domains under their respective zones would become unreachable. To mitigate this risk, DNS operators can deploy multiple techniques to increase the redundancy and resilience of ADNSes.

Recall our example DNS zone in chapter 2. We saw that each DNS zone can use multiple ADNS servers [39] – an example of which is `wikipedia.org`, which uses `ns[1-3].wikimedia.org` as ADNSes. Using IP anycast can further replicate ADNS instances by announcing their respective IP addresses from multiple locations globally. The last level of replication in DNS is having multiple servers on each anycast location – all behind a load balancer.

On the resolver side, *caching* is the most important technique to offer a safety net for unavailable ADNS [20, 52]. Whenever a resolver queries an ADNS, it keeps the responses in a memory cache. ADNS operators specify the maximum caching time using the time-to-live (TTL) value of the response [28]. Caching not only protects users from ADNS DDoS, but also improves response times by having cache hits. However, the safety net holds only as long as records are cached.

The DNS is used for more than IP address resolution. Enabling domain-destined (incoming) e-mail relies on a specific DNS record: the Mail Exchanger (MX) record. MX records are provided in a label format (e.g., google.com has smtp.google.com as MX label), which must be resolved by the sending Mail Transfer Agent (MTA) to determine the location (IP address) of the receiving MTA. As e-mail service can be provisioned by a third party, the authoritative DNS infrastructures for MX label and MX address resolution are not necessarily the same.

7.3 Related work

DNS and e-government: The closest research work to ours is by Houser et al. [112], who also investigate e-government DNS. They look into web domains while we also look into e-mail DNS infrastructure. They cover government domains of 193 countries and use, like us, active measurements to measure ADNS infrastructure. Our approach, however, differs in several ways: first, the input domains: we obtain a list of e-government domain names either publicly (`.gov` and `.se`) or privately (`.nl` and `.ch`) – so we have a complete view of these zones. Houser et al., however, focus on inferring domain names using a combined set of methods. While their coverage is larger in terms of TLDs, it may miss e-government domains. Ours, in turn, cover only four countries but with a complete view of their domains. They analyze zone inconsistencies and delegation errors. We focus on e-government DNS structuring from a stress event angle.

DNS resilience: standardization efforts and ample research exists for DNS resilience and redundancy. RFC9199 [108] summarizes six considerations for large ADNS operators – these include using IP anycast on every single ADNS [19, 56], keeping in mind that optimizing routing is then more important than adding extra locations on anycast networks [74]. They also include advice to consider long TTL values to leverage the benefits of caching by resolvers [20, 52].

DNS consolidation and centralization: Allman previously studied ADNS replication and shared infrastructure for `.com` and `.net` [71]. Similar to us, he recommends using multiple ADNS providers and to deploy topologically diverse ADNSes. Allman found that 28% of the second-level domains do not meet the multiple networks requirement for ADNS diversity. Another study investigated the now-defunct top 100k Alexa [113] domains and their ADNS infrastructure [15], showing that 89% of the domains rely on managed DNS providers, and 28% use a single DNS provider. The authors also showed how 3 DNS providers host 40% of the 100k Alexa websites. Centralization on the resolver market has also been quantified from the Netherlands `.nl` ccTLD [73].

TLD	Netherlands .nl	Sweden .se	Switzerland .ch	United States .gov
E-government domains	1,309	615	3,971	7,972
SLD	602	614	3,971	7,972
FQDN	707	0	0	0

Table 7.1: Datasets for web domains (2022-06-08)

The authors found that one-third of the queries to `.nl` domain names originate from five large cloud/content providers.

7.4 Datasets and Measurements

7.4.1 Datasets

Table 7.1 shows the web domain datasets used in this chapter. We obtained e-government domain names from the Netherlands, Sweden, Switzerland, and the United States. By and large, the e-government domains we consider are second-level domains (SLD) such as `cdc.gov`. Thanks to our collaboration with the Netherlands' National Cyber Security Center (NCSC-NL) in this research, we were also able to obtain fully-qualified domain names (FQDN) associated with e-identity services in the Netherlands. These are typically fully-qualified domain names such as `login.town.example.nl`. We treat these as a different category when we compare them to other countries. We obtained Sweden's e-government domain list from the Swedish Internet Infrastructure Foundation (IIS), which operates `.se`. The Swiss e-government domains were provided to us by SWITCH, the `.ch` registry. For the United States' e-government domains, we analyzed a fairly complete list of `.gov` names, provided in a public dataset[1]. *E-mail* E-government domains may also be used for e-mail (e.g., `info@nsf.gov`), which involves MX record resolution (see section 7.2), and may involve an external e-mail provider. We evaluate external dependencies for e-government domains. For the Netherlands, we obtained a specific list of domains that are used for e-mail. For the other countries, we target obtained e-government names with MX queries.

7.4.2 Measurements

We instrumented our own DNS measurements for this study. We carried out these measurements from a single vantage point on 2022-06-07. This vantage

[1] `https://home.dotgov.gov/data/`

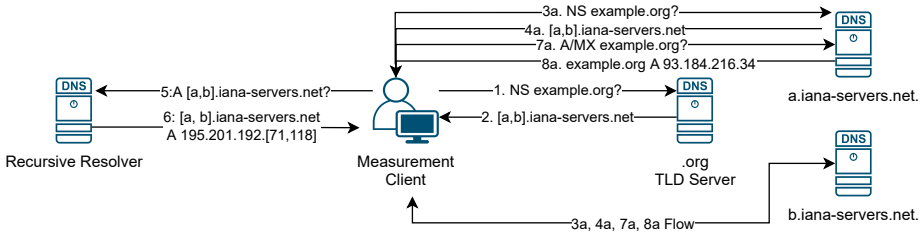


Figure 7.1: Measurement steps explained: Our software first queries the TLD server (PARENT), followed by all the CHILD servers.

point is provisioned on the Netherlands National Research and Education Network. Our measurement sequence is as follows (Figure 7.1). For each domain d, we measure:

1. The respective set of ADNS servers, NS_d, from both parent and child authoritative servers, as defined by d's NS records in the DNS (see $1-2-3-4$ in Figure 7.1)

2. For each $ns_d \in NS_d$, we then query for the IPv4 and IPv6 address records, as defined by ns_d's `A` and `AAAA` records in the DNS (see $5-6$ in Figure 7.1)

3. We then target each ns_d IP address measured with two queries related to d as follows (see also $7-8$ in Figure 7.1):

   (a) An `A` query to determine if ns_d is not misconfigured for the respective d (e.g., lame delegation)

   (b) An `MX` query to obtain the mail exchanger records (MX records) of d, MX_d[2]

4. We then measure the ADNS records, NS_{mx_d}, of each $mx_d \in MX_d$, query for the IP addresses of all $ns_{mx_d} \in NS_{mx_d}$, and use the measured IP addresses to perform a lame delegation check on ns_{mx_d} for mx_d

For each ADNS IP address learned in the previous step, we run additional measurements to determine if they are IP anycast. To perform this task, we utilize iGreedy [79]. As described in chapter 5, iGreedy detects anycast prefixes using the Great Circle Distance methodology (GCD). Running round-trip-time (RTT) measurements from geographically distributed VPs makes it possible to detect anycast instances by using speed of light constraint violations. We use 500

[2]For Netherlands' e-government names we only target those known to use e-mail.

RIPE Atlas probes as globally distributed vantage points for the measurements. The probes we select are all a minimum distance of 100km apart.

In our measurement approach, we opted to use iGreedy instead of the method described in chapter 5 to measure IP anycast. This was due to the smaller number of IP addresses of authoritative nameservers to measure and the absence of IPv6 support in the latter approach.

Regarding geolocation, the iGreedy measurement mechanics offer the means to determine this for anycast IP addresses. For addresses classified as unicast we rely on IP2Location for geolocation.

7.4.3 Limitations

Our study has several limitations. First, we perform DNS resolution from a single vantage point in the Netherlands, which may introduce bias if the targeted servers filter or change the responses based on the requesting IP. Second, our anycast census leverages ICMP reachability of the targeted IPs, which as discussed in [79, 114] can lead to a lower-bound estimation of anycast deployment. Finally, our analysis is scoped to only four TLDs, for which we were able to obtain lists of e-government domains.

7.5 Single Dependencies

In this section, we focus on e-government domains' singular dependency on providers and infrastructure. Given that DNS is highly distributed, we analyze the dependency of individual elements of the infrastructure as possible cause of unreachability of the e-government domains in case of failure.

7.5.1 ADNS Providers Dependency

We start by analyzing the number of DNS providers. For each e-government domain, d, we first measure the ADNS servers (subsection 7.4.2). In the case of `example.nl` (Figure 7.2), that would be two ADNS *server* names: `a.example.nl` and `b.example.com`. Then we resolve the ADNS server's IP addresses, using

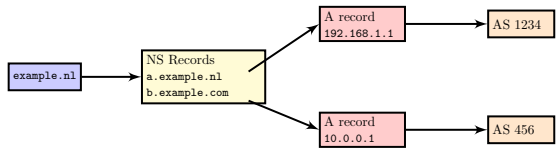


Figure 7.2: Relationship between domain names and DNS records

	NL	SE	CH	GOV
E-government domains	1,309	615	3,971	7,972
SLD	602	614	3,971	7,972
Responsive	601	609	3,546	7,911
single provider (v4/v6)	268/331	249/254	1,531/1,923	6,564/4,455
multi-provider (v4/v6)	333/266	360/254	2,013/344	1,306/578

Table 7.2: Responsive domains (2022-06-08)

A and AAAA type queries [8]. For example, in Figure 7.2, `a.example.nl` has 192.168.1.1 as IP address. For each IP address, we then look up its Autonomous System (AS) [115] number. Then we compute the number of unique ASes for d (for IPv4 and IPv6, separately). In our example, `example.nl` has two ADNS providers: AS1234 and AS456.

Table 7.2 shows the number of analyzed e-government domains. The number of responsive domains is slightly smaller than the number of *actual* e-

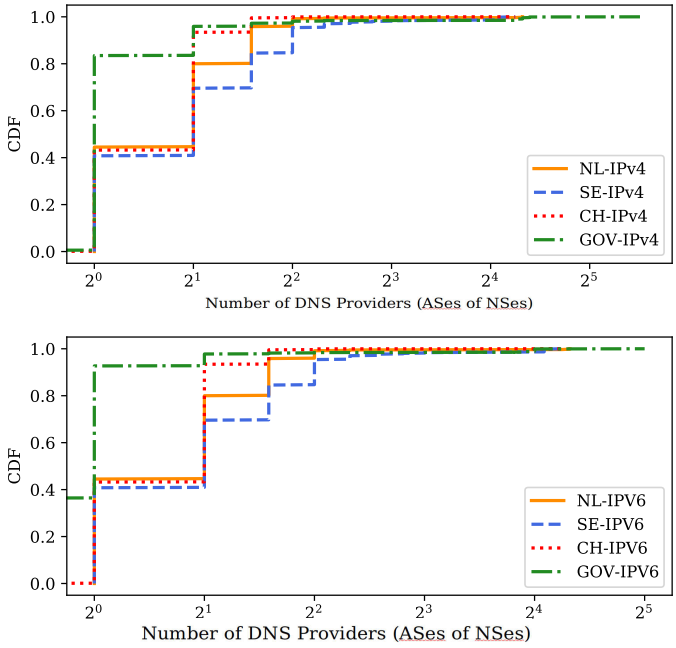


Figure 7.3: Number of ASes (DNS providers) for e-government domains

government SLDs. There are multiple reasons for this. Some domains have *delegation problems*: some list wrong ADNS servers (servers that are not responsive or are not authoritative for the domain name). For example, `daviscountyutah.gov` lists `168.180.200.18` as IP address of one of its ADNS (`dc-dns.daviscountyutah.gov.`). However, this IP does not respond to DNS queries. Other domains, such as `hudsoncountynj.gov`, although listed as a `.gov` domain name, have already been removed from the zone, so they do not exist.

For each responsive domain in Table 7.2, we compute the number of ADNS providers (as measured by their AS numbers). Figure 7.3 shows the CDF of DNS providers for the responsive e-government domains. For the ccTLDs (`.nl`, `.se`, and `.ch`), we notice that roughly 40% of the e-government domains have a single ADNS provider. For `.gov`, however, the majority of domains (80%+) have a single ADNS provider, although mostly hosted on well-provisioned and resilient providers.

ADNS Provider Consolidation and Centralization

Next, we focus on the side-effects of using third-party DNS providers: shared infrastructure. For each country, we compute the number of e-government domains that each ADNS provider has. Figure 7.4 shows the results. We can see that regardless of the zone, a handful of DNS providers exclusively operate the majority of domains. Table 7.3 shows, per DNS zone, the top ADNS providers and the number of e-government domains hosted.

We also see that *local* DNS providers provide service to most of the domains, i.e., DNS providers from the country for which we looked at e-government do-

	NL		SE	
	ASN	e-government	ASN	e-government
#1	20857 - Transip (NL)	112	39570 - Loopia (SE)	47
#2	48635 - CLDIN (NL)	39	1257 - Tele2 (SE)	23
#3	12315 - QSP (NL)	28	8068 - Microsoft (US)	21
#4	29311 - Solvinity (NL)	8	1729 - Telia (SE)	21
#5	48037 - SSC-ICT (NL)	8	3301 - Telia (SE)	19
	CH		GOV	
	ASN	e-government	ASN	e-government
#1	29222 - Infomaniak (CH)	278	44273 - GoDaddy (US)	1,215
#2	3303 - Swisscomm (CH)	115	13335 -Cloudflare (US)	909
#3	35206 - Novatrend (CH)	100	16509 - Amazon (US)	676
#4	9108 - Abraxas (CH)	97	21342 - Akamai (US)	334
#5	21069 - Metanet (CH)	91	16552 - Tiggee (US)	316

Table 7.3: Top ADNS provider concentration for single ADNS e-government domains (IPv4)

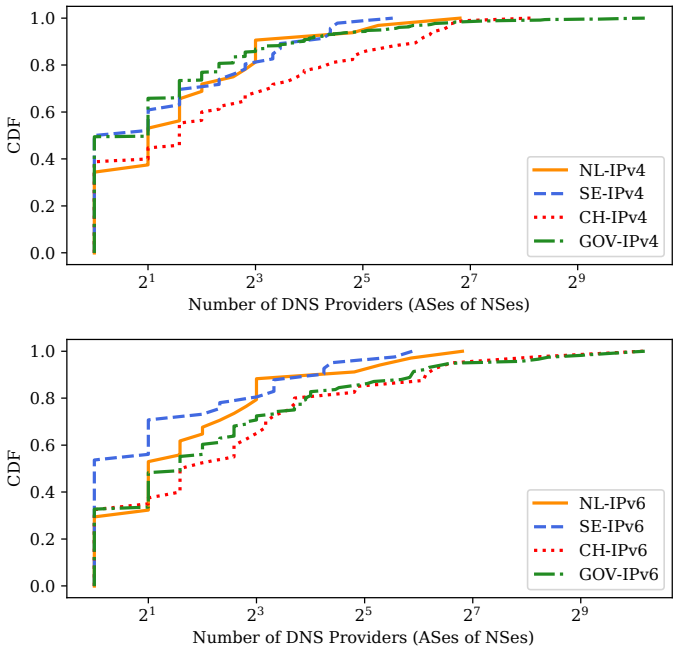


Figure 7.4: ADNS provider domains concentration

mains. The exception to this is Microsoft, which shows up as #3 for Sweden (we manually verified these domains and they use Microsoft Azure DNS servers, such as `ns3-02.azure-dns.org`).

These results show that although there has been a growing consolidation and centralization of DNS infrastructure over the last years in the hands of large US-based companies [15, 73], this has not been the case for the continental European e-government domains we study.

Implications: although relatively rare, large DNS providers can have (partial) failures – as in the case of Dyn and AWS [116]. In case of a massive DDoS attack on the provider, the associated e-government domains may experience serious reachability issues (clients will not be able to resolve them). While having a single DNS provider may simplify the configuration, it places the reliability of domains in the hands of a single provider. Given that DNS providers share infrastructure among all their DNS zones, even attacks on other domains can bring down e-government domains, due to collateral damage. As such, as also pointed out by Allman [71], it is better to use multiple ADNS providers and

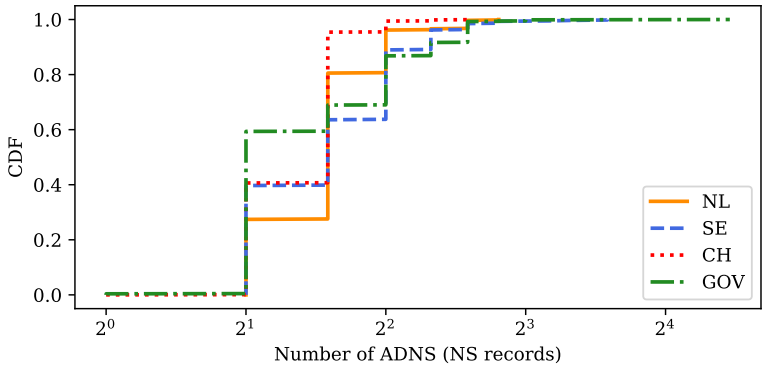


Figure 7.5: Number of ADNS servers (NS records) for e-government domains

even operate one of them in-house. Our contribution is to measure this for e-government domains from multiple countries – and show that more `.gov` e-government domains depend on a single ADNS provider than the other ccTLDs, which share similar rates.

7.5.2 ADNS Servers Dependency

To avoid single points of failure, the original DNS RFC (RFC1034 [28]) requires that domain names have at least two ADNS servers, as in Figure 7.2. We evaluate this requirement in terms of namespace – i.e., , as two different ADNS names (NS records in Figure 7.2) but also as different network prefixes. If two ADNS servers share the same unicast prefix, they are announced from the same location and, therefore, share the same infrastructure and are not topologically diverse.

We start with the ADNS server name analysis. Figure 7.5 shows the CDF of the domain names and their respective number of NS records. We see that the vast majority of e-government domains have at least two ADNS servers, conforming to RFC1034. We found one `.ch` domain that had only a single NS record at the child delegation (section 7.4), but two at the parent `.ch` delegation. This is caused by parent/child inconsistency (chapter 3).

We also identify 37 `.gov` domains with a single ADNS name in their child delegation. Out of these, 32 had more than one NS in their parent `.gov` authoritative server – but 5 did not. The `.gov` requirements stipulate that their domains

	NL	SE	CH	GOV
Responsive	601	609	3,546	7,911
Single prefix (v4/v6)	125/341	127/203	1,078/1,748	1,241/885
Anycast (v4/v6)	125/125	77/77	87/81	4,425/3,643

Table 7.4: E-government domains, prefixes and anycast usage

must have two ADNS servers[3]. However, these six domains[4] do not conform to this requirement. As such, these 37 violate what RFC1034 stipulates, and six violate `.gov` policy. We notified the `.gov` registry and registrar of this.

While most e-government domains have at least two ADNS servers (two different NS records), we now determine if this redundancy is also found on their associated IP prefixes. For each IP address, we retrieve their BGP prefix using CAIDA Prefix-to-AS mapping [26] and compute the number of prefixes each e-government domain has. Figure 7.6 shows the results. We see that Switzerland's `.ch` e-government domains lead the number of domains with a single BGP prefix (also shown in Table 7.4) – roughly one-third of its e-government domains have ADNS servers on the same network prefix. For IPv6, it is even worse: roughly 40% of the domains do not support DNS over IPv6, and another 40% are served from a single prefix.

Implications: RFC1034 states that ADNS servers for the same DNS zone should be placed in topologically distinct networks. We have seen that many e-government domains, for all zones, depend on ADNS servers located in the same topological location. This creates an unnecessary risk in case of failures or attacks. As such, we recommend that these operators configure ADNS servers in other distinct networks. Note that simply having different prefixes does not guarantee topological diversity [71], but having the same unicast prefix implies lack of topological diversity.

7.5.3 TLD Dependency

Next, we investigate what top-level domains (TLD) the ADNS servers of e-government domains depend upon. For example, in Figure 7.2, we see that `example.nl`'s NS records end in `.nl` and `.gov`, so it depends on two TLDs. While TLD failures are unlikely (just as large cloud provider failures), it is important to

[3]See `.gov` requirements at: `https://home.dotgov.gov/help/#what-are-the-name-server-requirements-for-gov-domains`

[4]these six are: `theftaz.gov`, `ncrealid.gov`, `bardstownky.gov`, `sjcpa.gov`, `cityofdelafieldwi.gov.`, `villageofpewaukeewi.gov`

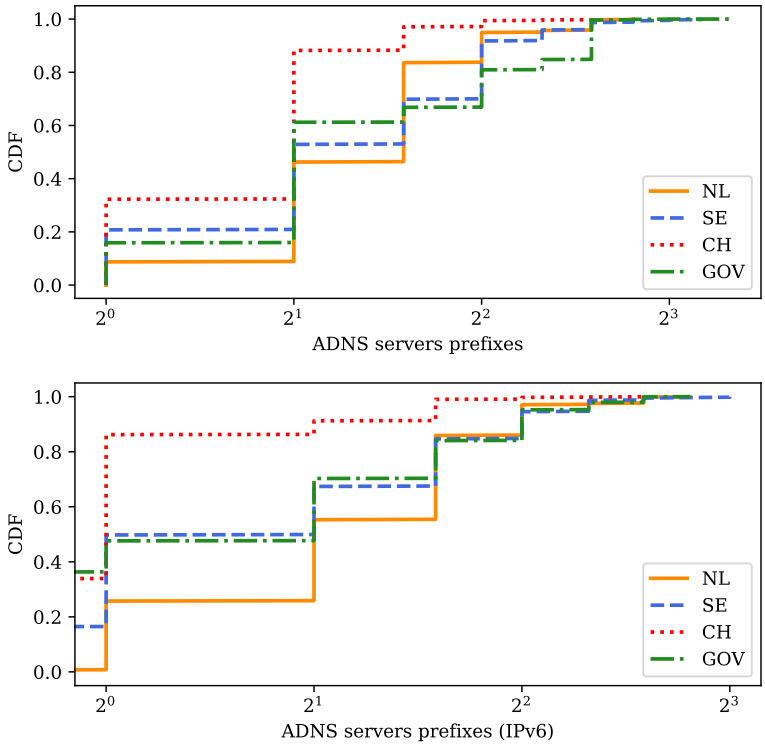


Figure 7.6: Number of distinct BGP prefixes that announce ADNS IP addresses

prepare for them and avoid that a potential TLD failure leads to e-government domain unreachability.

For each e-government domain, we compute the number of TLDs on which they depend, by analyzing the names of its ADNS servers. We then generate the CDFs per country, shown in Figure 7.7. We see that Swiss e-government domains are heavily concentrated in one TLD. The United States' `.gov` e-government ADNS servers are also heavily concentrated, followed by Sweden and the Netherlands.

Table 7.5 shows the top 5 TLDs for each country. To calculate this, we first generate a list of all ADNS servers for e-government domains per country. Then, we extract their TLDs and count and rank them. We see that cultural affinity seems to play a role in these results. The three countries from continental Europe use mostly their own countries' ccTLD, followed by either `.net` or `.com`

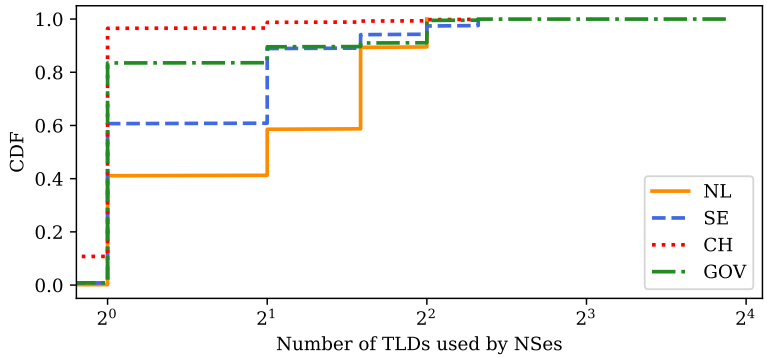


Figure 7.7: Number of TLDs used by e-government domains ADNS

	NL	SE	CH	GOV
#1	170 (.nl)	483 (.se)	609 (.ch)	2,507 (.com)
#2	69 (.net)	100 (.net)	190 (.com)	1,541 (.net)
#3	26 (.com)	82 (.com)	150 (.net)	894 (.gov)
#4	12 (.eu)	14 (.info)	19 (.org)	485 (.org)
#5	4 (.be)	8 (.org)	12 (.de)	302 (.us)

Table 7.5: Most used TLD by e-government ADNS servers.

(which are present in all 4 countries' e-government domains). The US's .gov most relies on .com domains, given it is a TLD operated in the US (as are all 5 in the .gov list) and where most cloud providers register DNS names.

Implications: e-government domains could benefit from extra resilience by having ADNS servers with FQDNs under a more diverse set of TLDs. This can protect such domains from failures in TLDs. Although this may be unlikely, these extra measures do not add much extra complexity and provide extra resilience. To illustrate this in practice, consider the domain digid.nl, which provides Dutch citizens with e-identity services to access their e-government services. This domain uses .com, .nl, .org and .eu as TLDs in its ADNS servers.

7.6 Anycast and Caching in E-government

In the previous section we focused on analyzing e-government domain dependencies on various parts of the Internet infrastructure. In this section, we focus on

if and how e-government domains rely on two particular techniques to improve resilience: the use of IP anycast and then of DNS caching.

7.6.1 Anycast Adoption

IP anycast is one of the cornerstones of DNS resilience. As such, DNS operators should deploy anycast to have more robust ADNS services [108]. For this reason, we quantify anycast adoption among e-government domains.

Figure 7.8 shows the CDF of e-government domains with regards anycast adoption. We see major differences between the countries under study. Around 58% of `.gov` domains have one or more anycast ADNS servers, whereas very few Swiss e-government domains do. The Netherlands and Sweden score in between; approximately 15–20% of their e-government domains have at least one ADNS that uses anycast. The reason for this, we believe, has to do with the ADNS providers. `.gov` is mostly served by large ADNS providers (Table 7.3) whereas the ccTLDs are mostly served by local companies, which may not deploy anycast or may charge an additional fee for this service.

Implications: IP anycast is widely deployed to improve DNS resilience. We see that most of the continental European e-government domains under consideration do not support anycast, while the majority of the US `.gov` domains do. We hope that the European e-government domains will in future also start using anycast.

7.6.2 Caching

DNS resolvers heavily deploy caching of DNS responses to improve response times to clients. It is by far the most efficient method to cut response times [52] and it can even suppress the effects of DDoS attacks [20], as clients can still resolve domain names thanks to cache hits, even when ADNS servers are unreachable.

While DNS caching is performed by resolvers, it is ADNS that controls how long records should stay in DNS resolver caches – by setting a time-to-live (TTL) value on each DNS record under their DNS zones. While TTL values range from 0 s to years, in practice most records fall between 10 min and 24 h [52]. It is suggested to configure ADNS NS records to have a TTL of at least a few hours [52, 108].

We next analyze the TTLs of both ADNS NS records and their respective IP address records (e.g., A and AAAA). Figure 7.9 shows the results for the zones we evaluate. For NS records, we see that most records fall between 1 and 24 h and many are equal to 1 h, which is considered *short* for an NS record [52, 108]. For A/AAAA records, we see that most fall within an interval of up to one hour.

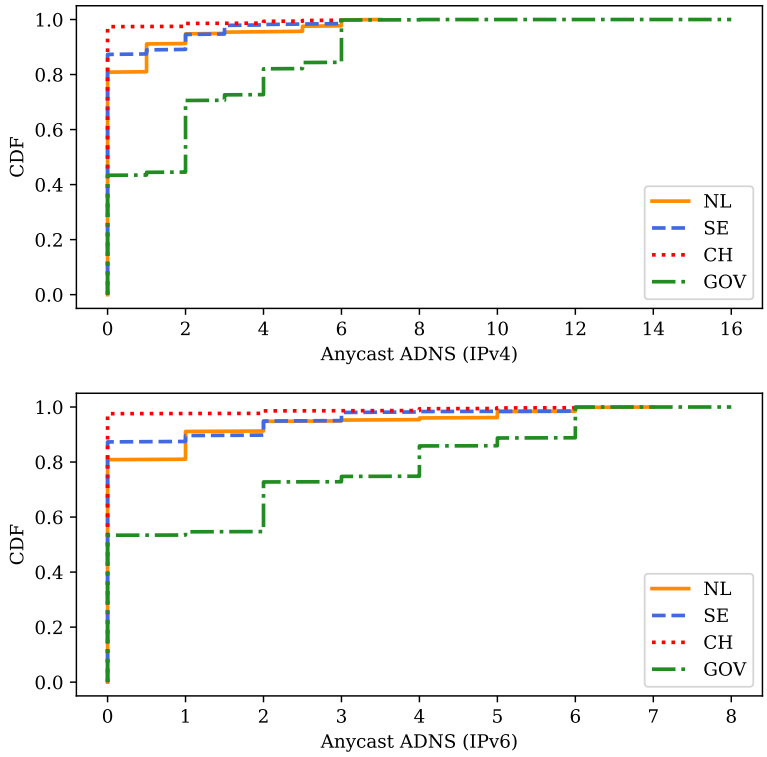


Figure 7.8: Anycast Adoption by e-government domains

Table 7.6 shows the TTL distributions. We see that the median TTL for Sweden and Switzerland is rather low (1 h) for the NS records. For A/AAAA records, we see that most domains have a 1 h TTL, which is considered reasonable.

Implications: Caching is the last line of defense for e-government domains: if all ADNS servers for a domain are down, a client may still be able to resolve and reach the e-government website if its resolver has the domain in question in cache. We see that the NS TTLs from Sweden and Switzerland (1h median) may be too low. The Netherlands and US have a 3 h median. These zones could benefit from longer caching if their TTLs are increased. However, such a configuration change must take into account if any DDoS protection services used by operators do not depend on DNS redirection, which benefits from shorter TTLs.

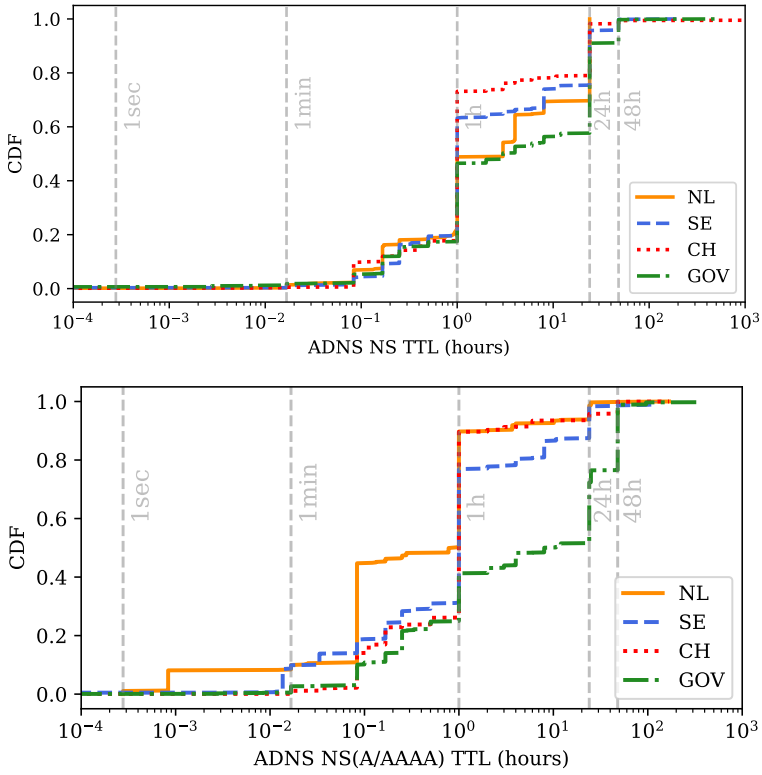


Figure 7.9: TTLs usage in ADNS

Table 7.6: TTL Distribution (s)

TTL	NL	SE	CH	GOV
NS TTL				
1st quartile	3,600	3,600	3,600	3,600
Median	10,800	3,600	3,600	10,800
3rd quartile	86,400	40,001	10,800	86,400
A/AAAA TTL				
1st quartile	300	900	1,800	3,600
Median	3,047	3,600	3,600	28,800
3rd quartile	3,600	3,600	3,600	90,000

TLD	Mail Domains	Same zone	Mixed	Out-of-zone
.gov	5,797	733 (12.6%)	121 (2.1%)	4,943 (85.3%)
.ch	2,126	302 (14.2%)	10 (0.5%)	1,841 (85.3%)
.se	544	113 (20.8%)	5 (0.9%)	426 (78.3%)
.nl	508	102 (20.1%)	5 (1.0%)	401 (78.9%)

Table 7.7: #Domains relying on same zone, mixed or out-of-zone DNS infrastructure for MX label resolution.

7.7 External Mail Dependencies

Recall from section 7.2 that labels in MX records must be resolved to determine the location of the receiving mail server, which can involve "external" ADNS infrastructure. As DNS infrastructure involved in MX resolution should also be resilient, we evaluate dependencies for the obtained e-government domains.

Table 7.7 shows the dependencies. Out-of-zone means that MX label resolution involves external ADNS infrastructure. Same zone means that the MX label is in the same zone as the e-government domain (e.g., `mx-west.fbi.gov` is in the zone of `fbi.gov`). Mixed means a combination, which can occur in case an e-government domain defines multiple MX records. We observe that e-government domains heavily rely on external ADNS infrastructure. The smallest percentage of same zone MX labels is seen for `.gov` (12.6%). We also see a small number of cases where two records are combined.

For e-government domains relying on third-party mail providers, we further investigate the mail provider, identified by SLD. As shown in Table 7.8,

	Domains			Domains	
MX Provider	#.nl	%.nl	MX Provider	#.se	%.se
outlook.com	164	39.0%	outlook.com	205	37.5%
ezorg.nl	46	11.0%	mailanyone.net	69	12.6%
ssonet.nl	17	4.0%	mx25.net	52	9.5%
barracudanetworks.com	13	3.1%	staysecuregroup.com	38	6.9%
minvenj.nl	12	2.9%	staysecuregroup.net	38	6.9%
MX Provider	#.ch	%.ch	MX Provider	#.gov	%.gov
outlook.com	425	22.1%	outlook.com	2,243	41.4%
infomaniak.ch	129	6.7%	google.com	532	9.8%
abxsec.com	120	6.2%	barracudanetworks.com	495	9.1%
tophost.ch	90	4.7%	pphosted.com	161	3.0%
ag.ch	78	4.1%	mimecast.com	157	2.9%

Table 7.8: Top 5 third-party e-mail providers per country

Microsoft Outlook prominently services e-government domains. We also observe several in-country mail providers for `.nl` and `.ch`. For example, for `.nl`, `ssonet.nl` is a large IT provider of the Netherlands Government.

Considering anycast of MX ADNS infrastructure, we observe that a significant percentage (87.5%) of third-party mail providers use anycast for their ADNS servers. We find a comparable percentage for `.se` and `.ch`. For `.gov` names, we see lower (62% of 5,944 FQDNs) anycast adoption.

To study the resilience of out-of-zone dependencies in terms of *network diversity*, we perform a case study for `.nl` providers. Among the MX labels for `.nl` e-government names, we identify 330 unique FQDN (i.e., MX labels). Our measurement data for these labels shows at least two NS records and two v4 ADNS servers for all labels, but only two-thirds with at least two v6 ADNS servers. Of the resolved ADNS infrastructure addresses, 66% are hosted in a single ASN for IPv4, and 72% for IPv6. All the v4 authoritative nameservers responded and only 2% of the v6 authoritative nameservers did not. *Implications:* Third-party e-mail providers on which e-government names depend offer, for the most, resilient ADNS infrastructure, hardening the additional resolution step for MX labels.

7.8 Discussion and Recommendations

A robust, redundant, and properly configured DNS is crucial for e-government services to be delivered to citizens and residents. We compare four countries with regards to their e-government DNS structuring. Our results show that there is plenty of room for improvement, which we cover next.

First and foremost, we show that there is much dependency on single DNS providers, for all countries under study (subsection 7.5.1). The e-government domains should add at least a second DNS provider, which could protect them against failure and attacks that can occur within individual providers (an event seen multiple times in the past [14, 104]). Secondly, we observe that many e-government domains have ADNS infrastructure in the same networks (subsection 7.5.2) – violating recommendations from the original DNS RFCs. We recommend e-government domains to adhere to these recommendations. Third, we found that for the evaluated countries in continental Europe, DNS service is largely provided by local providers, and not by the large US-based cloud and DNS providers (subsection 7.5.1). We can only speculate that this may be due to historical reasons – the large US-based cloud services are relatively new compared to most e-government domains. For e-government e-mail, however, it is completely different: Microsoft dominates the e-government market in all

countries – which could be due to the usage of Outlook's cloud-based e-mail services.

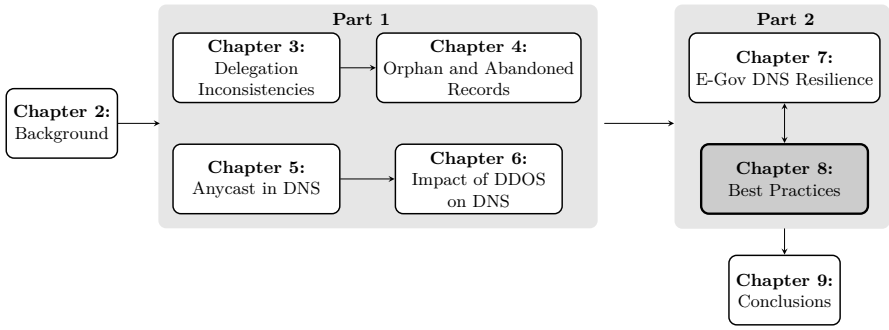
Our final recommendation is for operators to carefully set the TTL values of their DNS records, so they can leverage the benefits of caching in DNS during stress events section 7.6. This requires only a single parameter change. Similarly, we also recommend that countries deploy more IP anycast on their ADNS servers. We show that despite having the highest GDP per capita, Switzerland lags behind in terms of anycast adoption. We will present our findings to the respective countries' TLD operators.

7.9 Concluding Remarks

E-government has become an essential part of public administrations. As a core Internet protocol, the DNS underpins reachability of e-government services. In this chapter we evaluated DNS structuring for e-government services (web and e-mail) for four countries. Our results show that many e-government domains are not following the current recommendations for operation of large DNS providers, regardless of country. While e-government domains may operate without hiccups, it is not free of risks, as a motivated attacker could stress specific DNS infrastructures to compromise the reachability of many e-government domains. We hope our findings prompt the responsible operators to improve the redundancy and resilience of e-government DNS.

CHAPTER 8

Best Practices for Critical Services



In earlier chapters, we analyzed the deployment of DNS resilience mechanisms in various environments and evaluated their ability to withstand DDoS attacks. This chapter presents best practices for configuring authoritative DNS servers. By summarizing the findings of our research and previous scientific literature, we provide practical, concrete recommendations to operators when setting up these servers (RQ-3). These best practices have been suggested as advice for the National Cyber Security Centre of the Netherlands to enhance the resilience of the Dutch e-government domains.

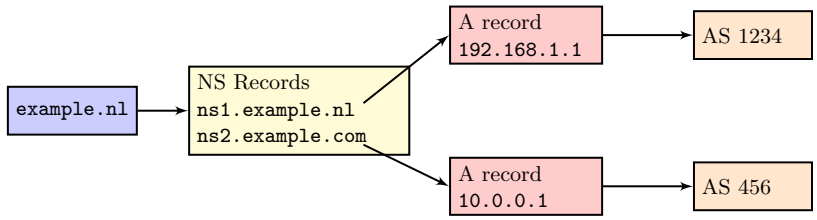


Figure 8.1: Example to illustate best practices for an example domain name (`example.nl`)

8.1 Introduction

Collaboration with industry, specifically DNS and network operators, is a crucial aspect of our research efforts. It allows us to validate our findings on misconfigurations and vulnerabilities in the DNS ecosystem. Through communication with stakeholders, we gain insights into resilience mechanisms and technical decisions that balance resilience, performance, and cost. One of the main objectives of this thesis is to transfer the knowledge we obtained to network operators, to aid them in improving DNS resilience through informed strategies, policies, and operations.

With this in mind, we have developed a set of best practices for configuring authoritative DNS servers for network operators, drawing on both our research and previous scientific studies. Our best practices primarily concern the dependability and availability of authoritative servers. We did not focus on best practices related to integrity, such as the use of DNSSEC [117] or best practices to reduce latency between clients and authoritative servers (performance) as we consider those out of the scope for this thesis.

We divide the best practices into three categories: *critical*, *recommended*, and *unmeasurable*, which we will explain in the following sections. For readers seeking a general understanding of the rationales behind our best practices and their classification, we advise reading subsections 8.2.1, 8.3.1, and 8.4.1. For those interested in technical details and measurability, we recommend reading subsections 8.2.2, 8.3.2, and 8.4.2.

8.2 Critical Best Practices

8.2.1 Rationale

Critical (section 8.2) 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

Metric	Description/Reference	Value
nNSes	Number of NS records for a zone/[39]	>=2
nIP(NSv4)	Number of Unique IP addresses for NSes (IPv4) [39]	>=2
nIP(NSv6)	Number of Unique IP addresses for NSes (IPv6) [39]	>=2
ResponsiveNSesV4	All authoritative servers are responsive for the domain [118]	`True`
ResponsiveNSesV6	All authoritative servers are responsive for the domain [118]	`True`
nPrefixes(NSv4)	Number of unique BGP prefixes for NSes (IPv4)[21])	>=2
nPrefixes(NSv6)	Number of unique BGP prefixes for NSes (IPv6) [21])	>=2
nAses(NSv4)	Number of unique ASes for NSes (IPv4) [71]	>=2
nASes(NSv6)	Number of unique ASes for NSes (IPv6) [71]	>=2
nGeoDiverseNSes	Number of NS distinct geographical locations [21]	>=2

Table 8.1: Critical Best Practice Metrics

basket". Single points-of-failure cause total unreachability of domain names when they fail. Due to the significant impact these best practices have on increasing the resilience of the DNS ecosystem, we have identified them as critical. These practices have been expressed in several RFCs [21, 39, 118] and have been confirmed to be of fundamental relevance by previous academic studies ([71], [72], and chapter 6).

Table 8.1 summarizes these critical best practices for authoritative DNS server operators. These practices primarily concentrate on the topological and geographical diversity and availability of nameserver deployments, encouraging operators to meet a minimum standard of two diverse online deployments to avoid SPoF.

8.2.2 Operational Details

In this section, we expand on each individual metric and practice. For that, we use the example shown in Figure 8.1, for a sample DNS zone: `example.nl`.

nNSes: Number of NS Records for a Zone

Description: each domain name is required to have at least two authoritative DNS servers [39], 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 8.1, this is shown by having two NS records: `ns1.example.nl` and `ns2.example.com`. Each NS record, in turn, may be operated by a different organization and using IP anycast, which provides extra redundancy (section 8.3.2).

Reference: this best practice has been proposed in the original DNS standard [39], 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`

nIP(NSv4): Number of unique IP Addresses for all NSes

Description: This metric consists of determining how many unique IPv4 addresses *host* the authoritative DNS servers. In our Figure 8.1, that would be the number of unique IPv4 addresses associated with both NS records (`ns1.example.nl` and `ns2.example.com`). Note that a single domain may have multiple NS records (fulfilling the requirements in section 8.2.2). 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 section 8.2.2, if analyzed alone, could provide a false sense of security.

Reference: This best practice is documented in RFC2182 [21].

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.

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

Same as in section 8.2.2, except it measures AAAA records (IPv6) instead of A records (IPv4).

ResponsiveNSesV4: All Authoritative Servers are Responsive for the Domain

Description: In our example domain in Figure 8.1, 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 not be authoritative for the zone in question (referred to as lame delegation [118]), and

ultimately may not be 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` server would *refuse* to answer the question, indicating the NL server is not authoritative for `.jp`.

Reference: Lame delegations are defined in RFC1713 [118] and evaluated in [72].

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

1. Get the IP addresses of all NS records.
2. For each address, send an SOA, A or NS query about the domain name in question. If the response is OK (RCODE=0 [28]), then the server is properly configured. If not, then the server has an issue.

ResponsiveNSesV6: All Authoritative Servers are Responsive for the Domain

Same as section 8.2.2, except for IPv6 addresses.

nPrefixes(NSv4): Number of unique BGP Prefixes for NSes (IPv4)

Description: IP addresses are announced on the Internet in blocks called "BGP prefixes" [115]. These prefix announcements contain information that help routers determine where address space can be reached. For example, suppose a telecom company announces an IP address block 192.168.0.0/24 (which covers 256 /32 addresses). This announcement is received by neighboring routers, which propagate it further. For resilience, 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 8.1, we see two addresses that *likely* belong to two different prefixes.

Reference: This falls in the category of having dissimilar infrastructure for authoritative servers. This is defined in [21].

How to measure it: For that, operators should analyze BGP prefix announcements in public sources of BGP data, such as RIPE RIS [119] and RouteViews [120]. To measure it, the steps to follow are:

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

nPrefixes(NSv6): Number of unique BGP Prefixes for NSes (IPv4)

Same as section 8.2.2, except for IPv6.

nASes(NSv4): Number of unique ASes for NSes (IPv4)

Description: As discussed in section 8.2.2, IP addresses are announced in BGP using prefixes. This announcement also contains what *Autonomous Systems*(ASes) are in the path to the prefix, and its *origin* AS. Ultimately, it is the origin AS that *hosts* the IP addresses in questions. To improve resilience, it is recommended to have the IP addresses of your authoritative server in more than one AS, to avoid single points of failure if something goes wrong with a particular AS.

Reference: This falls in category of having dissimilar infrastructure for authoritative servers. This is defined in [21].

How to measure it: To measure this metric, operators should analyze BGP route announcements from public sources, such as RIPE RIS [119] and RouteViews [120].

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.

nASes(NSv6): Number of unique ASes for NSes (IPv4)

Same as section 8.2.2, except for IPv6 addresses.

nGeoDiverseNSes: Number of NS distinct Geographical Locations

Description: Authoritative nameservers should be placed in different geographical locations in order to avoid that a physical disaster in a location (e.g., a fire) can affect all of the servers. To improve resilience, it is recommended to put the nameservers in different cities.

Reference: RFC2182 [21] states that secondary servers should be at geographically distant locations.

How to measure it: For that, operators should analyze IP geolocation databases such as Maxmind [121] or run active measurements to identify locations using RIPE Atlas [57].

Metric	Description/Ref.	Value
nTLDs	Use more than one TLD for NS records [71]	>=2
NS TTL	TTL values of NS records [20, 52, 108]	>=3600s
A(NS) TTL	TTL values for A (NS) records [20, 52, 108]	>=1800s
AAAA(NS) TTL	TTL values for AAAA (NS) records [20, 52, 108]	>=1800s
nAnycastIPv4	Number of Anycast Auth Servers IPv4 [19]	>=1
nAnycastIPv6	Number of Anycast Auth Servers IPv6 [19]	>=1

Table 8.2: Recommended Best Practices Metrics

8.3 Recommended Best Practices

8.3.1 Rationale

Recommended best practices refer to practices that improve the dependability of the DNS, but they are not so critical that not following them would lead to a SPoF (i.e., total unreachability). This category of best practices includes those focusing on optimal TTL configuration [108], diversity of TLDs [71], and adoption of anycast (chapter 5). While these practices are not critical, previous scientific literature has demonstrated their effectiveness in overcoming DDoS attacks. In particular, long TTL values allow customers to leverage cached values for a longer time in case of unreachability of a part of the resolution chain [20, 52]. Anycast adoption has proven to be one of the key strategies to absorb and limit the effects of DDoS attacks on a global scale, as shown in chapter 6 and the DDoS attack of November 2015 against the root nameserver infrastructure [19].

Table 8.2 summarizes these *recommended* best practices. As previously stated, these practices place emphasis on TTL configuration of NS and glue records, IP anycast adoption for both IPv4 and IPv6 deployments, and finally TLD diversity for nameserver records.

8.3.2 Operational Details

In this section we expand on these recommended best practices.

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 8.1, we see that the two NS records use different TLDs: `.com` and `.nl`. This means that if one of these two TLDs would become unreachable, the `example.nl` zone could still be reachable

via the other 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 long been known by the community, and is also documented by [71].

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).

NS TTL value

Description: DNS records, such as the NS records in Figure 8.1, always have a time-to-live field (TTL), which tells DNS resolvers the maximum time the DNS responses may be kept in the DNS cache of the servers. DNS caches, are the cornerstone of DNS performance [20, 52, 108]: having a cached response drastically reduces the response time to clients. Moreover, in case of DDoS attacks, having *longer* TTLs (say minimum one 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 *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 followed when choosing TTLs:

- For general DNS zone owners, [52] 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 save costs.
- Users of DNS-based load balancing or DDoS-protection 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: The role of caching in DDoS attacks in DNS has been the goal of several studies [20, 52, 108].

How to measure it: To measure the TTL value of a record, one must obtain an authoritative answer by asking the authoritative servers *directly*, 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 8.1 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 (chapter 3), given that these zones are typically managed by different organizations. However, most resolvers in the wild tend to follow the *child* authoritative server TTL [52]. For this reason, we will consider only the child TTL value.

A(NS) TTL

Description: in section 8.3.2, we analyze the TTL of NS records for a given domain. These NS records, in turn, need to have A and/or AAAA records to be reachable – these are the IP addresses that are used to route packets. In Figure 8.1, that refers to the TTL value of the A record (192.168.1.1).

The TTLs for A/AAAA records should be shorter than or equal to the TTL for the corresponding NS records for in-bailiwick authoritative DNS servers, since [52] 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 TTLs for A and AAAA records may still be desired if DDoS-mitigation services are required.

Reference: The role of caching in DDoS attacks on the DNS has been investigated in several studies [20, 52, 108].

How to measure it: To measure the TTL value of a record, one must obtain an authoritative answer by asking the authoritative server *directly* and bypass 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. For this reason, operators should measure them according to their setup at parent level.

AAAA(NS) TTL

Same as section 8.3.2, except for AAAA (IPv6) records.

nAnycastIPv4: Number of Anycast-based Authoritative Servers

Description: IP anycast consists of announcing the same IP prefixes from multiple locations [44]. Anycast is widely used in the DNS [122], 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 onto a different anycast site. Clients are mapped to nearby sites (nearby in BGP terms, and not necessarily geographical distance [74]). This distribution is not necessarily uniform, some sites may see far more clients than others.

In case of DDoS attacks against an authoritative server, some sites may experience the attack differently [19]: some sites may remain up while others go down. That behavior has been observed in the Root DNS events of November 2015 [19]. As such, operators can, on-the-fly, reconfigure their authoritative anycast DNS to try to steer DDoS traffic to one or a few sites, while others may remain up.

Reference: IP anycast is documented in [44]. Its DNS usage in [122]. Its relation to DDoS in [19] and chapter 6. How to measure anycast in the wild is documented in chapter 5.

How to measure it: Using iGreedy [79] or the MAnycast^2 approach outlined in chapter 5.

nAnycastIPv6: Number of Anycast-based Authoritative Servers

Same as section 8.3.2, except for AAAA (IPv6) records.

8.4 Unmeasurable Best Practices

8.4.1 Rationale

The last category of best practices, which we consider not measurable (section 8.3), includes physical and link layer practices that cannot be measured using traditional Internet Measurements (layer 3 and above). Given their importance we list them here, although we did not assess them, because 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, unmeasurable. Although these practices are relevant for the resilience of authoritative DNS servers, we consider them out of scope for this thesis. Operators seeking to implement critical (especially nation-state, government, and military) deployments should however carefully consider these aspects.

Metric	Description/Reference	Value
nPhysicalLocations	Number of unique physical locations hosting the authoritative name servers	>= 2
nPhysicalLinks	Number of distinct physical links connecting authoritative name servers	>= 2
nPhysicalServers	Number of unique bare-metal servers	>= 2
KeyServersOnClients	Place anycast sites or authoritative servers at key clients	NA

Table 8.3: Unmeasurable Best Practices Metrics

Table 8.3 summarizes these recommendations. We focused on the physical resilience of lines, locations, and servers and on the availability of those services closer to users in order to continuously provide service in case of disconnection from the global Internet.

8.4.2 Operational Details

In the following, we expand on each individual metric and practice.

nPhysicalLocations

Authoritative servers – either virtual or bare metal, should be placed in *distinct* physical locations, to avoid that any location-related failures (attacks, power outages, etc.) affects the authoritative DNS servers altogether. Consider the worst-case scenario, in which three authoritative servers are hosted in different IP address blocks, using different upstream providers, but all are physically hosted in the same, single datacenter: no matter how much redundancy is added, this setup still has a single point-of-failure, which is a single location. As such, we recommend operators to use multiple physical locations to host their services.

nPhysicalLinks

Similar to the number of physical servers, there must be multiple links that connect authoritative 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.

nPhysicalServers

The last metric concerns the number of physical servers hosting the authoritative 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.

KeyServersOnClients

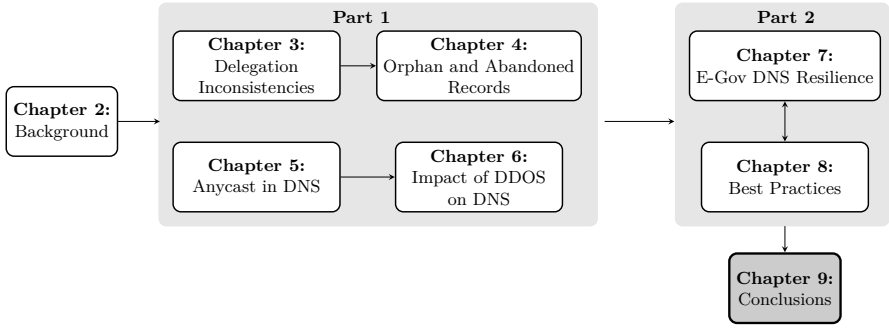
For specific services, such as, for example, e-government identity providers (e.g., DigiD, SPID, etc.), it may be worth to add *anycast sites* of authoritative servers in key client networks – for example, the networks of major ISPs and where most clients come from. Depending on the type of attack, this 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 an IXP. Clients will still be able to resolve the domain if they have access to servers on their ISP's network. This practice only improves resilience in cases the client's network is not able to reach the networks of the authoritative servers.

8.5 Concluding Remarks

In this chapter, we have identified and summarized best practices aimed at improving the dependability and the availability of authoritative DNS servers, thus improving the resilience of the DNS. We hope that by defining these best practices and conducting research that quantifies them, we can contribute to a more robust and secure DNS ecosystem. We also hope to encourage collaboration between researchers and industry to address challenges and develop further best practices for critical Internet infrastructure. While following these practices may incur financial costs, it is important to keep in mind that they can represent a significant advantage in terms of resilience. For this reason, we advise operators to implement these best practices.

CHAPTER 9

Conclusions



The results of our research are summarized in this final chapter of the thesis. We begin by highlighting the main conclusions, followed by a detailed analysis of the conclusions as they relate to the research questions posed in chapter 1. Finally, we conclude the chapter by discussing potential future research areas that extend and build on the work presented in this thesis.

9.1 Main Conclusions

In chapter 1 we set the main goal for this thesis. The goal was:

> ***Goal:*** *to perform a comprehensive real-world measurement of resilience mechanisms of DNS authoritative infrastructure and to assess their mitigative effects in the face of DDoS attacks.*

The complexity of the DNS exposes it to various types of configuration and deployment errors that can threaten its overall resilience. Conducting a comprehensive real-world measurement of the DNS ecosystem proved to be challenging due to the complexity and independence of operators' choices. These results underlined that the resilience of the DNS ecosystem has multiple facets.

Looking at these facets in more detail, first, we defined the various errors and choices that can impact DNS resilience, focusing on both the interactions between different elements of the hierarchy and the choices of individual operators. In chapters 3 and 4, we found that miscommunication between stakeholders in the DNS ecosystem, such as registries and registrars, leads to delegation inconsistencies within the hierarchy. We conclude that these inconsistencies pose a relevant threat to the DNS infrastructure and increase the attack surface, affecting the resilience properties (e.g., enabling lame delegations). To address this issue, we proposed operational and configurational best practices for zone management for operators to avoid the reduction of resilience, with virtually zero implementation costs.

In the second step of our research, we examined techniques for increasing DNS resilience. In chapter 5, we conducted a comprehensive real-world measurement of the adoption of resilience mechanisms in DNS authoritative infrastructure. The results revealed that while half of the SLDs and almost all TLDs adopt anycast and other well-defined resilience techniques, this adoption is primarily driven by a few large DNS providers. Furthermore, we concluded that a combination of traditional methods and anycast may be the best approach for improving resilience, as confirmed in chapter 6 where we prove its efficacy in overcoming DDoS attacks. As we already observed, a handful of large operators drive adoption of resilience techniques. Our intuition is that this is due to the increased costs and complexity of these solutions, which are easier to manage for large operators. These factors led to a lower adoption of these techniques by smaller operators and, as consequence to a centralization of the DNS ecosystem, reducing users' choices, as observed in chapters 5 and 7.

The main conclusion of this thesis is:
While traditional resilience techniques, such as network diversity, have been available for decades and provide some level of protection, we found anycast to be a powerful differentiator that really pays off in overcoming the challenges

posed by the increasing threat of DDoS attacks against vital internet infrastructure. While these requirements may come with additional costs, particulary for smaller operators, it is worthwhile to consider using multiple third-party providers to increase overall infrastructure resilience with careful planning and training. Alternatively, incentives should be provided to reduce the barriers to anycast adoption for smaller operators, allowing them to provide more reliable services at a lower cost.

9.2 Revisiting the Research Questions

In this section, we will delve deeper into the results of this thesis by revisiting each of the research questions outlined in chapter 1. Our first research question was as follows:

> ***RQ 1:*** *What types of inconsistency exist between different entities in the DNS and to what extent do they occur in practice?*

We explored this research question in chapters 3 and 4, specifically focusing on the problem of delegation inconsistency among parent and child levels of the hierarchy. We found that this issue is widely present, affecting over 8% of the DNS ecosystem for the major gTLDs `.com`, `.net`, and `.org`. We even discovered 52 TLDs with inconsistent delegations towards the root level of the DNS hierarchy. We categorized these inconsistencies into four types: (i) disjoint parent and child NSSets, (ii) child NSSet is a subset of parent NSSet, (iii) parent NSSet is a subset of child NSSet, and (iv) non-empty intersection but no match between parent and child NSSets. Our results also showed that the percentage of domains affected by this misconfiguration remains consistent over time. We concluded that this NS inconsistency is a widespread, long-term issue in the DNS ecosystem.

Furthermore, we also found that inconsistency in the DNS ecosystem is not limited to delegation issues. In fact, inconsistency among DNS data is not happening only in the DNS ecosystem itself. Registrars and registries exchange information for updating the zone files via an out-of-bound protocol called EPP. In chapter 4, we discovered that incorrect implementation of EPP protocols and poor zone file management by registries can lead to two types of misconfigurations called orphan and abandoned records. These records represent leftover resources in DNS zones that can compromise the stability and security of the overall DNS ecosystem. We found evidence of these issues dating back 12 years, indicating that it is a longstanding problem. We concluded that these misconfigurations represent a significant problem that needs to be addressed in order to ensure the stability and security of the overall DNS ecosystem.

These longstanding misconfigurations we discovered in our research led us to our second research question:

RQ 2: *What is the harm of these inconsistencies?*

To better understand the harm caused by these misconfigurations, we delved deeper into the potential threats they pose to the DNS ecosystem. As shown before, in chapter 3, we divided the parent-child inconsistency problem into four different categories and found that each category has different potential levels of risk. For example, leaving dangling records in the parent zone can expose users to the risk of nameserver hijacking and lame delegation, a similar threat can also arise from orphan records as discussed in chapter 4.

To further investigate the potential harm caused by these misconfigurations, we also needed to understand the actual effects on the process of DNS resolution. In chapter 3, we studied how resolvers treat these misconfigured domains, and in particular, which resolution path they follow. Our findings, both from controlled experiments and a study performed in the wild with RIPE Atlas, revealed a wide range of behaviors that is dependent on the specific resolver software and configuration.

RFC2181 [53] specifies how resolvers should rank data in case of inconsistency, but we found several instances where the specification was not respected, both in the wild and in our controlled experiment. This led us to the conclusion that the indeterminate nature of resolver behavior in terms of preferred resolution path in case of inconsistencies should prompt operators to fix these issues at the source.

While fixing resolvers to comply with RFC2181 is still a good and viable solution, ensuring their updated deployment at scale may be challenging. For this reason, we argue that operators should pay careful attention to configuring their zones to avoid inconsistencies altogether, using tools for configuration validation (e.g., Zonemaster[1]) and mechanisms for automatic synchronization (e.g., . CSYNC records).

In this context, as outlined in chapter 4, we made efforts towards fixing the orphan misconfiguration in collaboration with Afilias, thereby reducing the risks for users and network operators. We also actively participated in discussions on the *Delegation Revalidation by DNS Resolvers* Internet draft, which aims to resolve the parent-child issue discussed in chapter 3 at the resolver level. To further assist operators, we have deployed SuperDNS.nl[2], a tool for verifying resolver behavior in a controlled environment.

[1] `https://zonemaster.net`
[2] `https://superdns.nl`

As demonstrated, the identified inconsistencies in the DNS ecosystem are relatively easy to fix and should not require extensive training in zone management for operators. This can help to maintain the overall resilience of the DNS.

In contrast, however, expanding and improving DNS resilience may involve additional costs and training. In this thesis, we sought to identify the best strategies for achieving this goal of improving DNS resilience, prompting us to investigate the research question that follows.

> ***RQ 3:*** *Which are the key strategies to enable DNS resilience and to what extent are they deployed?*

To identify strategies for enabling DNS resilience, we looked at related RFCs, specifically RFC2182 [21]. From the early days of the DNS ecosystem, adopters emphasized the importance of network, geographical, and administrative diversity in the deployment of DNS authoritative nameservers. Over the years, a new and highly effective technique was introduced: IP Anycast [44]. In this thesis, we condensed these key strategies into a set of parameters: prefix diversity, AS diversity, geo-diversity, and IP Anycast adoption.

Previous literature has addressed the first three parameters with a limited scope in terms of scale and time [15, 71, 78]. To fill this gap, in chapter 5, we conducted a large-scale analysis in the wild for approximately 60% of the entire SLD DNS namespace, covering all four key strategies. At first glance, the results obtained were encouraging for the stability and resilience of the DNS ecosystem. We found that 97% of TLDs and 62% of SLDs were adopting anycast as a resilience strategy, and most of them had implemented at least one of the other resilience properties. However, upon further analysis, we found that this adoption was not driven by individual users, but rather by a few large DNS infrastructure providers. We concluded that large providers are dominating and centralizing the market with potential negative consequences for DNS administrative diversity.

We also investigated the combination of the three classical key strategies for DNS resilience and IP Anycast. Our findings showed that anycast usage in the wild correlated with a lower adoption of classical strategies. We found evidence of poorly configured providers relying only on IP Anycast as a resilience technique. We discovered a case where one provider relied on a single anycast prefix for their entire authoritative nameserver set. In the event of routing misconfigurations or large-scale attacks, anycast may not be able to effectively keep the system online. To better understand those issue, we also performed an in-depth investigation of the different failure modes that could occur due to attacks or misconfigurations. In conclusion, despite the high levels of adoption of IP Anycast and its utility, operators should still consider a mixed approach when

implementing key DNS resilience strategies for the DNS ecosystem, integrating classical strategies and the more novel IP Anycast.

In addition, to perform the study described in chapter 5, we developed a methodology to identify anycast prefixes in the wild. Recognizing the value of this data for network operators and researchers, we decided to make it publicly available. For this reason, we started releasing, in open access, a quarterly anycast census of the entire IPv4 space online[3]. To help network operators, we included IP Anycast deployment and network diversity best practices in our collection of best practices discussed in chapter 8.

During our analysis of the deployment of resilience techniques, we wondered if the type of service hosted on a specific domain name could lead to a different adoption of these strategies. As previously mentioned, we found that the decision to adopt these strategies is mostly made by providers rather than users. If this was also the case for setups that are of critical importance to society brought us to our fourth research question:

> ***RQ 4:*** *How do societally critical setups, such as government services, adopt resilience techniques?*

Our analysis revealed a concerning landscape in terms of DNS resilience strategy adoption by critical setups, such as e-government domains. We found evidence that those domains are managed by a limited set of DNS, web, and mail hosting providers. These providers showed a low adoption of anycast for their nameservers and were affected by several misconfigurations. Worryingly, although these domains represent a valuable resource for citizens and society, it seems that their management was not perceived as a priority.

We urge network providers and administrators of these domains to fix the misconfigurations we identified and to better adopt other resilience mechanisms, starting with relying on multiple providers.

The adoption of resilience techniques in the wild is just one aspect of their impact on the DNS ecosystem. To gain a more complete understanding, we also needed to investigate the effects of these techniques when challenged by DDoS attacks. Therefore, we asked:

> ***RQ 5:*** *To what extent do resilience mechanisms mitigate the effect of DDoS attacks?*

Unfortunately, obtaining information on DDoS attacks is not an easy task. Operators may be unwilling to share insights on DDoS attacks against their infrastructure to prevent damage to the reputation of their companies or to avoid giving attackers information on their efficacy or detection abilities. However, some types of DDoS attacks can be detected, such as Randomly Spoofed

[3] `https://github.com/ut-dacs/Anycast-Census`

DDoS attacks. We focused on this specific type of attack using data from the UCSD Network Telescope to assess the impact of DDoS attacks on the DNS authoritative nameserver ecosystem in chapter 6.

We showed how poor adoption of resilience mechanisms can lead to catastrophic consequences in the event of DDoS attacks. Of particular relevance is the case of the Russian railways and ministry of defence. Both of these had poorly deployed infrastructure and suffered extreme consequences during DDoS attacks related to the Russo-Ukrainian conflict. We also found that there is no direct correlation between the inferred power of the attack from the network telescope and the effectiveness of the attacks. Our conclusion was that the effectiveness of attacks is mainly dependent on the resilience and the size of the deployment of the authoritative nameservers, rather than on the attack power itself.

In our more in-depth analysis of resilience techniques, we found that anycast deployments are less affected by attacks and that hosting nameservers across multiple prefixes or multiple ASNs also provides increased resilience against devastating attacks. On the other hand, even small attacks can pose a risk to infrastructure that neglects to architect resilience into their critical DNS infrastructure. Our general conclusion from this research question is that while DDoS attacks are extremely frequent, adopting best practices and resilience techniques remains one of the best options to overcome them.

9.3 Directions for Future Research

In the final section of this chapter, we will focus on future research directions. The three pillars that we have defined as future steps are: (i) Observability of policy-making frameworks for the DNS ecosystem, specifically KINDNS, (ii) Real-time monitoring of DNS authoritative infrastructure under DDoS attacks, and (iii) Improving measurability of anycast at scale.

9.3.1 KINDNS Observability

The *Knowledge-Sharing and Instantiating Norms for DNS and Naming Security (KINDNS)* initiative proposed by ICANN aims to codify best practices into global norms for improved security. In a recent study [35], we identify challenges related to the measurability of ICANN's proposed practices. While some overlap with practices discussed extensively in this thesis in chapters 5, 7, and 8, several were found to be either not measurable or to have strong ethical implications in their measurability. Additionally, the KINDNS program currently only encourages self-assessment by operators, unlike the similar MANRS program for routing security which also includes third-party independent verification. The

KINDNS conversation is ongoing and stakeholders are still debating the contents of the proposed set of practices. However, we argue that from a scientific standpoint, an independent assessment of the adoption of KINDNS best practices is necessary to understand their impact on the health and stability of the DNS ecosystem. This assessment will allow policy-makers to understand the impact of defined best-practices in order to improve future frameworks.

9.3.2 Real-time Monitoring of DNS infrastructure under DDoS

In chapter 6 we showed the significance of collecting data on performance degradation during DDoS attacks on DNS authoritative infrastructure. Our study relied on a series of cases where OpenINTEL happened to measure while the nameservers were under attack. This scenario suggests a natural future direction. Using these macroscopic measurement data sources to trigger active measurements of critical infrastructure under attack can lead to additional insights into resilience and failure modes of different components. For example, measuring all nameservers for a given domain upon evidence of an attack will provide a more effective indication on whether and how end users experience resolution failure.

Measurement from multiple vantage points will also improve fidelity of inferences in the face of increasing anycast deployment. These techniques would overcome OpenINTEL's limitations of using the default rather than NS-exhaustive resolution process, and doing so from a single vantage point. The tradeoff is operational cost and measurement overhead. We have prototyped such a reactive measurement platform, and plan to use it to demonstrate the feasibility of real-time characterization of DDoS attacks against the global DNS infrastructure. Our future plans include the development of dashboards and tools for network operators to share the performance data collected. This will assist the community in better comprehending the impact and significance of ongoing DDoS attacks and making informed decisions on the best strategies to overcome them.

9.3.3 Improving Anycast Measurability at Scale

In chapter 5, we presented a novel approach for conducting a large-scale anycast census. As we collected the data, we recognized their uniqueness and significance for the academic and operator community. Future efforts will aim to enhance this methodology by addressing challenges identified during data collection. The most difficult challenge in successfully deploying our technique is the false negative classification of anycast IPs caused by routing policies. Our current solution is to use the GCD approach from iGreedy as second stage meas-

urement to eliminate these false-positives. However, to effectively eliminate this type of error, we must address and investigate the differences in connectivity levels at different vantage points of the anycast measurement framework that we use for our measurements.

Futhermore, our measurements also collect latency data. When ping responses arrive on the same VP that sent them, we can use the RTT to calculate distance. Given a larger testbed, this opens the door to geolocating anycast instances using the GCD approach [79]. This hybrid approach would detect anycast prefixes using anycast VPs and geolocate them using latency-based triangulation.

Bibliography

[1] B. Rymes, "Names," *Journal of Linguistic Anthropology*, vol. 9, no. 1/2, pp. 163–166, 1999.

[2] P. Mockapetris, "Domain names: Concepts and facilities," RFC 882, IETF, Nov. 1983.

[3] T. Böttger, G. Ibrahim, and B. Vallis, "How the Internet Reacted to Covid-19: A Perspective from Facebook's Edge Network," in *Proceedings of the ACM Internet Measurement Conference*, IMC '20, 2020.

[4] A. Lutu, D. Perino, M. Bagnulo, E. Frias-Martinez, and J. Khangosstar, "A Characterization of the COVID-19 Pandemic Impact on a Mobile Network Operator Traffic," in *Proceedings of the ACM Internet Measurement Conference*, IMC '20, 2020.

[5] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The Lockdown Effect: Implications of the COVID-19 Pandemic on Internet Traffic," in *Proceedings of the ACM Internet Measurement Conference*, IMC '20, 2020.

[6] A. Ukani, A. Mirian, and A. C. Snoeren, "Locked-in during Lock-down: Undergraduate Life on the Internet in a Pandemic," in *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, 2021.

[7] CAIDA, "DZDB." `https://catalog.caida.org/dataset/dzdb`.

[8] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS Extensions to Support IP Version 6," RFC 3596, IETF, Oct. 2003.

[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC 4033, IETF, Mar. 2005.

[10] D. Black, K. McCloghrie, and J. Schoenwaelder, "Uniform Resource Identifier (URI) Scheme for the Simple Network Management Protocol (SNMP)," RFC 4088, IETF, June 2005.

[11] D. Crocker, T. Hansen, and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures," RFC 6376, IETF, Sept. 2011.

[12] P. Hoffman and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA," RFC 6698, IETF, Aug. 2012.

[13] W. Lehr, D. Clark, and S. Bauer, "Changing Markets for Domain Names: Technical, Economic, and Policy Challenges," *SSRN Electronic Journal*, Jan 2020.

[14] Scott Hilton, "Dyn Analysis Summary Of Friday October 21 Attack." `https://web.archive.org/web/20190225060705/https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/`, 10 2016.

[15] A. Kashaf, V. Sekar, and Y. Agarwal, "Analyzing Third Party Service Dependencies in Modern Web Services: Have We Learned from the Mirai-Dyn Incident?," in *Proceedings of the ACM Internet Measurement Conference*, IMC '20, 2020.

[16] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem," in *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, 2017.

[17] J. Cardoso de Santanna, R. van Rijswijk, R. Hofstede, A. Sperotto, M. Wierbosch, L. Zambenedetti Granville, and A. Pras, "Booters - an analysis of DDoS-as-a-Service attacks," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, (United States), 2015.

[18] P. Lutscher, N. Weidmann, M. Roberts, M. Jonker, A. King, and A. Dainotti, "At Home and Abroad: The Use of Denial-of-service Attacks during Elections in Nondemocratic Regimes," *Journal of Conflict Resolution*, vol. 64, pp. 373–401, July 2019.

[19] 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*, IMC '16, 2016.

[20] 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*, IMC '18, 2018.

[21] R. Elz, R. Bush, S. Bradner, and M. Patton, "Selection and Operation of Secondary DNS Servers," RFC 2182, IETF, July 1997.

[22] R. Sommese, M. Jonker, and K. Claffy, "Observable KINDNS: Validating DNS Hygiene," in *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, 2022.

[23] B. Du, C. Testart, R. Fontugne, G. Akiwate, A. C. Snoeren, and k. claffy, "Mind Your MANRS: Measuring the MANRS Ecosystem," in *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, 2022.

[24] D. Clark, "The EU NIS-2 proposal and the DNS." `https://catalog.caida.org/paper/2022_eu_nis_2_proposal`. Accessed: 2022-12-2.

[25] R. van Rijswijk, M. Jonker, A. Sperotto, and A. Pras, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE journal on selected areas in communications*, vol. 34, pp. 1877–1888, June 2016.

[26] CAIDA, "Routeviews Prefix to AS mappings Dataset for IPv4 and IPv6," 2020.

[27] CAIDA, "UCSD Network Telescope Daily Randomly and Uniformly Spoofed Denial-of-Service (RSDoS) Attack Metadata," 2022.

[28] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034, IETF, Nov. 1987.

[29] R. Sommese, G. C. M. Moura, M. Jonker, R. van Rijswijk Deij, A. Dainotti, K. C. Claffy, and A. Sperotto, "When Parents and Children Disagree: Diving into DNS Delegation Inconsistency," in *Passive and Active Measurement*, Springer, 2020.

[30] A. J. Kalafut, M. Gupta, C. A. Cole, L. Chen, and N. E. Myers, "An empirical study of orphan DNS servers in the Internet," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, 2010.

[31] R. Sommese, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy, and A. Sperotto, "The Forgotten Side of DNS: Orphan and Abandoned Records," Proceedings of the 5th IEEE European Symposium on Security and Privacy Workshops, WTMC '20, (United States), 2020.

[32] R. Sommese, G. Akiwate, M. Jonker, G. Moura, M. Davids, R. van Rijswijk - Deij, G. Voelker, S. Savage, K. Claffy, and A. Sperotto, "Characterization of Anycast Adoption in the DNS Authoritative Infrastructure," in *5th Network Traffic Measurement and Analysis Conference, TMA 2021*, IFIP, 2021.

[33] R. Sommese, L. Bertholdo, G. Akiwate, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy, and A. Sperotto, "MAnycast2: Using Anycast to Measure Anycast," in *Proceedings of the ACM Internet Measurement Conference*, IMC '20, 2020.

[34] R. Sommese, K. Claffy, R. van Rijswijk-Deij, A. Chattopadhyay, A. Dainotti, A. Sperotto, and M. Jonker, "Investigating the Impact of DDoS Attacks on DNS Infrastructure," in *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, 2022.

[35] R. Sommese, M. Jonker, J. van der Ham, and G. C. M. Moura, "Assessing e-Government DNS Resilience," in *Proceedings of the 2022 International Conference on Network and Service Management (CNSM 2022)*, 2022.

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

[37] P. Faltstrom, P. Hoffman, and A. Costello, "Internationalizing Domain Names in Applications (IDNA)," RFC 3490, IETF, Mar. 2003.

[38] T. Halvorson, M. F. Der, I. Foster, S. Savage, L. K. Saul, and G. M. Voelker, "From .Academy to .Zone: An Analysis of the New TLD Land Rush," in *Proceedings of the Internet Measurement Conference*, IMC '15, 2015.

[39] P. Mockapetris, "Domain names - implementation and specification," RFC 1035, IETF, Nov. 1987.

[40] S. Hollenbeck, "Extensible Provisioning Protocol (EPP)," RFC 5730, IETF, Aug. 2009.

[41] K. Bock, A. Alaraj, Y. Fax, K. S. Hurley, E. Wustrow, and D. Levin, "Weaponizing Middleboxes for TCP Reflected Amplification," in *USENIX Security Symposium*, 2021.

[42] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827, IETF, May 2000.

[43] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "AmpPot: Monitoring and Defending Against Amplification DDoS Attacks," in *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions, and Defenses*, RAID 2015, pp. 615–636, 2015.

[44] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," RFC 1546, IETF, Nov. 1993.

[45] R. de Oliveira Schmidt, J. Heidemann, and J. Kuipers, *Anycast Latency: How Many Sites Are Enough?* No. ISI-TR-2016-708 in Information Sciences Institute technical report, United States: University of Southern California, May 2016.

[46] P. Hoffman, A. Sullivan, and K. Fujiwara, "DNS Terminology," RFC 8499, IETF, Jan. 2019.

[47] J. Kristoff, "DNS inconsistency." `https://blog.apnic.net/2018/08/29/dns-inconsistency/`, 2018.

[48] ICCAN, "Root Zone File," Dec. 2022. `http://www.internic.net/domain/root.zone`.

[49] W. Hardaker, "Child-to-Parent Synchronization in DNS," RFC 7477, IETF, Mar. 2015.

[50] V. Pappas, D. Wessels, D. Massey, S. Lu, A. Terzis, and L. Zhang, "Impact of configuration errors on DNS robustness," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 3, pp. 275–290, 2009.

[51] D. Liu, S. Hao, and H. Wang, "All Your DNS Records Point to Us: Understanding the Security Threats of Dangling DNS Records," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, 2016.

[52] 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*, 2019.

[53] R. Elz and R. Bush, "Clarifications to the DNS Specification," RFC 2181, IETF, July 1997.

[54] DENIC AG, "Statistics of .de Domains," 2019.

[55] C. Almond, "CNAME at the APEX of a zone." `https://www.isc.org/blogs/cname-at-the-apex-of-a-zone/`.

[56] M. Müller, G. C. M. Moura, R. de O. Schmidt, and J. Heidemann, "Recursives in the Wild: Engineering Authoritative DNS Servers," in *Proceedings of the ACM Internet Measurement Conference*, IMC '17, 2017.

[57] RIPE NCC Staff, "RIPE Atlas: A Global Internet Measurement Network," *Internet Protocol Journal (IPJ)*, vol. 18, pp. 2–26, Sep 2015.

[58] A. Hubert and R. v. Mook, "Measures for Making DNS More Resilient against Forged Answers," RFC 5452, IETF, Jan. 2009.

[59] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks: a comprehensive measurement study," in *Proceedings of the 2014 ACM Conference on Internet Measurement Conference*, IMC '14, 2014.

[60] DNS OARC, "Root Zone Archive." `https://www.dns-oarc.net/oarc/data/zfr/root`, Jan. 2020.

[61] Internet Systems Consortium, "BIND: Berkeley Internet Name Domain." `https://www.isc.org/bind/`.

[62] NLNet Labs, "Unbound DNS resolver," 2020.

[63] CZ.NIC, "Knot Resolver." `https://www.knot-resolver.cz`.

[64] PowerDNS, "PowerDNS Recursor." `https://www.powerdns.com/recursor.html`.

[65] J. Jiang, J. Liang, K. Li, J. Li, H. Duan, and J. Wu, "Ghost domain names: Revoked yet still resolvable," 2012.

[66] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache Spark: A Unified Engine for Big Data Processing," *Commun. ACM*, vol. 59, pp. 56–65, oct 2016.

[67] I. Foster, "SpiderWho," Dec. 2019.

[68] DNS-OARC, "Zone File Repository." `https://www.dns-oarc.net/oarc/data/zfr`.

[69] M. Larson, "[dns-operations] Upcoming DNS behavior changes to .com/.net/.edu name servers," Jan. 2010.

[70] A. Nyman, "Stoppa domänskojarna!."

[71] M. Allman, "Comments on DNS Robustness," in *Proceedings of the Internet Measurement Conference*, IMC '18.

[72] 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*, IMC '20, 2020.

[73] 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*, IMC '20, 2020.

[74] 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 Workshop*, Springer, 2017.

[75] G. C. M. Moura, J. Heidemann, W. Hardaker, P. Charnsethikul, J. Bulten, J. M. Ceron, and C. Hesselman, "Old but Gold: Prospecting TCP to Engineer and Live Monitor DNS Anycast," 2022.

[76] K. Schomp and R. Al-Dalky, "Partitioning the Internet Using Anycast Catchments," *SIGCOMM Comput. Commun. Rev.*, vol. 50, pp. 3–9, Oct. 2020.

[77] G. Moura, W. Hardaker, J. Heidemann, and M. Davids, "Considerations for Large Authoritative DNS Server Operators," RFC 9199, IETF, Mar. 2022.

[78] X. Fan, J. Heidemann, and R. Govindan, "Evaluating anycast in the domain name system," in *2013 Proceedings IEEE INFOCOM*, 2013.

[79] D. Cicalese and D. Rossi, "A Longitudinal Study of IP Anycast," *SIGCOMM Comput. Commun. Rev.*, vol. 48, pp. 10–18, Apr. 2018.

[80] R. Bian, S. Hao, H. Wang, A. Dhamdere, A. Dainotti, and C. Cotton, "Towards Passive Analysis of Anycast in Global Routing: Unintended Impact of Remote Peering," *SIGCOMM Comput. Commun. Rev.*, vol. 49, pp. 18–25, Nov. 2019.

[81] L. M. Bertholdo, J. M. Ceron, W. B. d. Vries, R. d. O. Schmidt, L. Z. Granville, R. v. Rijswijk-Deij, and A. Pras, "TANGLED: A Cooperative Anycast Testbed," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021.

[82] B. Schlinker, T. Arnold, I. Cunha, and E. Katz-Bassett, "PEERING: Virtualizing BGP at the Edge for Research," in *Proceedings of ACM CoNEXT '19*, 2019.

[83] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: Understanding Internet Reliability Through Adaptive Probing," in *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 255–266, Aug 2013.

[84] CAIDA, "Inferred AS to Organization Mapping Dataset." `https://www.caida.org/data/as-organizations/`, 2017.

[85] "Root Analysis–Source Code." `https://github.com/gmmoura/tma2021`.

[86] M. Korczynski, M. Wullink, S. Tajalizadehkhoob, G. C. M. Moura, A. Noroozian, D. Bagley, and C. Hesselman, "Cybercrime After the Sunrise: A Statistical Analysis of DNS Abuse in New GTLDs," in *Proceedings of the Asia Conference on Computer and Communications Security*, 2018.

[87] ISC, "Secondary Name Services." `https://www.isc.org/sns-pb/`, 01 2020.

[88] A. Abhishta, R. Van Rijswijk-Deij, and L. J. Nieuwenhuis, "Measuring the impact of a successful DDoS attack on the customer behaviour of managed DNS service providers," *Computer Communication Review*, vol. 48, no. 5, pp. 70–76, 2018.

[89] Job Snijders, "NLNOG RING." `https://ring.nlnog.net/`.

[90] M. Jonker, A. Pras, A. Dainotti, and A. Sperotto, "A First Joint Look at DoS Attacks and BGP Blackholing in the Wild," in *Proceedings of the Internet Measurement Conference 2018*, IMC '18, 2018.

[91] Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG), "M3AAWG State of the Union: DDOS Attacks," 2020.

[92] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the Adoption of DDoS Protection Services," in *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, 2016.

[93] Panda Security, "Network Attacks: DoS and DDoS Attacks," 2020.

[94] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Trans. Comput. Syst.*, vol. 24, no. 2, pp. 115–139, 2006.

[95] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting Internet DNS Amplification DDoS Activities," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, 2014.

[96] M. Bailey, E. Cooke, D. Watson, F. Jahanian, and N. Provos, "A hybrid honeypot architecture for scalable network monitoring," *Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep. CSE-TR-499-04*, 2004.

[97] S. Greenstein, "The Aftermath of the Dyn DDOS Attack," *IEEE Micro*, vol. 39, no. 4, pp. 66–68, 2019.

[98] Y. Liu, Z. Wang, and N. Li, "Characterizing the impact of DDoS attack on inter-domain routing system: a case study of the Dyn cyberattack," *Advances in Computer Science Research*, vol. 80, no. Csece, pp. 79–82, 2018.

[99] R. Yazdani, R. van Rijswijk-Deij, M. Jonker, and A. Sperotto, "A Matter of Degree: Characterizing the Amplification Power of Open DNS Resolvers," in *Proceedings of Passive and Active Measurement: 23rd International Conference, PAM 2022*, 2022.

[100] R. Beverly and A. Berger, "Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure Via Active Fingerprinting," in *International Conference on Passive and Active Measurement*, pp. 149–161, 2015.

[101] J. Kreps, "Kafka : a Distributed Messaging System for Log Processing," 2011.

[102] D. van Monsjou, "Internetproviders zijn getroffen door ddos-aanvallen - update 2 [Dutch]," Mar 2021.

[103] T. Hofmans, "Providers zijn maandag opnieuw getroffen door ddos-aanvallen [Dutch]," Dec 2020.

[104] T. NOC, "DDoS Post Mortem," Mar 2021.

[105] B. Barret, "DDoS Attempts Hit Russia as Ukraine Conflict Intensifies," Feb 2022.

[106] N. Rubenking, "I Went to a Russian Website and All I Got Was This Lousy Teapot," Feb 2022.

[107] VirusTotal, "VirusTotal," 2022.

[108] G. Moura, W. Hardaker, J. Heidemann, and M. Davids, "Considerations for Large Authoritative DNS Server Operators," RFC 9199, IETF, Mar. 2022.

[109] M. Gotze, S. Matic, C. Iordanou, G. Smaragdakis, and N. Laoutaris, "Measuring Web Cookies in Governmental Websites," in *14th ACM Web Science Conference 2022*, 2022.

[110] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "A Year in Lockdown: How the Waves of COVID-19 Impact Internet Traffic," *Commun. ACM*, vol. 64, Jun 2021.

[111] O. for Economic Co-operation and Development, "Responding to COVID-19: The rules of good governance apply now more than ever." `https://www.oecd.org/governance/public-governance-responses-to-covid19/`, June 2022.

[112] R. Houser, S. Hao, C. Cotton, and H. Wang, "A Comprehensive, Longitudinal Study of Government DNS Deployment at Global Scale," in *Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, IEEE/IFIP, 2022.

[113] Alexa, "Alexa: Keyword Research, Competitive Analysis & Website Ranking," Feb. 2022.

[114] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, "Census and Survey of the Visible Internet," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ACM, 2008.

[115] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, IETF, Jan. 2006.

[116] C. Williams, "Bezos DDoS'd: Amazon Web Services' DNS systems knackered by hours-long cyber-attack." `https://www.theregister.co.uk/2019/10/22/aws_dns_ddos/`, Oct. 2019.

[117] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Protocol Modifications for the DNS Security Extensions," RFC 4035, IETF, Mar. 2005.

[118] A. Romao, "Tools for DNS debugging," RFC 1713, IETF, Nov. 1994.

[119] Reseaux IP Europeens Network Coordination Centre (RIPE NCC), "Routing Information Service (RIS)," 2021.

[120] University of Oregon , "Route Views Project," 2021.

[121] Maxmind, "Maxmind," 2012.

[122] D. McPherson, D. Oran, D. Thaler, and E. Osterweil, "Architectural Considerations of IP Anycast," RFC 7094, IETF, Jan. 2014.

[123] G. Akiwate, R. Sommese, M. Jonker, Z. Durumeric, K. Claffy, G. M. Voelker, and S. Savage, "Retroactive Identification of Targeted DNS Infrastructure Hijacking," in *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, 2022.

[124] O. van der Toorn, R. van Rijswijk-Deij, R. Sommese, A. Sperotto, and M. Jonker, "Saving Brian's Privacy: The Perils of Privacy Exposure through Reverse DNS," in *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, 2022.

[125] L. Zembruzki, R. Sommese, L. Z. Granville, A. Selle Jacobs, M. Jonker, and G. C. M. Moura, "Hosting Industry Centralization and Consolidation," in *Proceedings of NOMS IEEE/IFIP Network Operations and Management Symposium*, NOMS '22, 2022.

[126] A. Affinito, R. Sommese, G. Akiwate, S. Savage, K. Claffy, G. Voelker, A. Botta, and M. Jonker, "Domain Name Lifetimes: Baseline and Threats," in *Proceeding of 5th Network Traffic Measurement and Analysis Conference*, TMA '22, IFIP, 2022.

[127] S. Miano, R. Doriguzzi-Corin, F. Risso, D. Siracusa, and R. Sommese, "Introducing SmartNICs in Server-Based Data Plane Processing: The DDoS Mitigation Use Case," *IEEE Access*, vol. 7, pp. 107161–107170, 2019.

[128] G. B. Fioccola, R. Sommese, I. Tufano, R. Canonico, and G. Ventre, "Polluino: An efficient cloud-based management of IoT devices for air quality monitoring," in *Proceeding of 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow*, RTSI '16, IEEE, 2016.

Acknowledgements

Someone may think that the preceding pages of this thesis were the most difficult to write. Let me tell you the truth – the ones that follow were the hardest part to write. Firstly, I would like to express my heartfelt acknowledgments and deepest apologies to those who I may have forgotten to mention, but who have been a part of this amazing four-year journey. Thank you.

Anna, I still vividly remember the day we first met. As I was searching for a PhD position around Europe, your warm welcome left a lasting impression on me. Thank you for your guidance and support throughout this long PhD journey. Your commitment to continuously helping me, despite having a family to take care of during the challenges of the pandemic, has been truly inspiring. I am grateful for your kindness, patience, and expertise, which have enabled me to grow both as a researcher and as a person. Thank you for being an exceptional supervisor, for our lengthy and inspiring conversations, both technical and non-technical, and for making this journey a great life experience. I could not have written this thesis without your technical and scientific support by my side.

Roland, I cannot thank you enough for your invaluable support and technical guidance. Your deep knowledge of DNS and Internet protocols really boosted my research. I will always cherish the time spent in your office, where you shared your expertise and helped me develop innovative research ideas. I feel privileged to have had you as a mentor these past years and beyond. Additionally, I would like to express my gratitude for the great project you created - OpenINTEL. Without this extremely precious source of data, this thesis would not have been possible. On a personal note, thank you for all your advice on exploring the Canary Islands, which I truly enjoyed for my astrophotography hobby.

During this PhD journey, I met a truly special person - Mattijs. You have been a friend, a colleague, a mentor, and in the end a daily supervisor for me. Your constant source of support and your willingness to share your vast knowledge and expertise in both technical and academic areas has been invaluable, and I have learned so much from you. Many of the ideas and future directions of this thesis are undoubtedly thanks to your input. Thank you for being like a big brother in research to me and for helping me to grow both as a researcher and as a person. Travelling together for work has created many memorable ex-

periences and I am grateful for the opportunity to have shared those moments with you. Thank you once again for everything.

KC, you are like a superhero of Internet research to me. If someone had told me during my bachelor studies in Italy, when I was doing a thesis on Internet measurement, that one day I would have the opportunity to work with you, I would have advised them to see a good doctor because their mind had gone nuts. KC, you have been an invaluable guidance to me, and despite your extremely busy research life and my inexperience, you took me under your wing and provided me with invaluable lessons. I am grateful for the opportunity you gave me to visit CAIDA in the summer of 2022 for three months. I will always remember our dinners at your place with the other "kids", Ben and Emiline. Finally, thank you for being part of my graduation committee.

Thank you also to Ben and Emiline, the "kids", for making my time in San Diego so memorable. I am grateful for the laughs, the adventures, and the full experience of American life. Our trip to Yosemite with Audrey will always hold a special place in my heart. Thank you for making my experience in the USA something to remember for the rest of my life.

Geoff and Stefan, my words for KC also hold true for you. Thank you for granting me the incredible opportunity to collaborate with you. Our conversations during research collaboration inspired many of the findings in this thesis. Thank you for the collaboration with Gautam on the exciting topic of DNS Hijacking. Finally, Geoff, thank you for being part of my committee.

Gautam, we have been peers working together over these years. We started with some overlapping in our fields of research, and instead of competing, we learned the invaluable lesson that collaboration and sharing in research is the key to success. Our collaboration has led to many great results, both in my thesis and in yours. Thank you for your support and friendship. I hope our collaboration will be the seed for great future works.

Alberto, you taught me that research is not merely a playground for exploration, but that there are also times when a rigorous and precise approach is necessary. Many of the soundest results and key takeaways in this thesis would not have been possible without your expert guidance. Thank you.

Aiko, thanks for the contribution you gave to this amazing reality of DACS group over the past years. During the time we managed to work together, I always appreciated your wisdom and I am grateful for your presence as part of my committee.

Christian and Ana-Lucia, thanks for beign part of my committee. I hope this occasion will be the seed of future collaborations togheter.

To all colleagues at DACS, thank you so much for all the support during these years and for making our work and workplace something to enjoy every

day. In particular, a special thank you to my former and current office mates, Leandro, Joao, and Bernt and to our wonderful secretaries, Nicole and Jeanette.

Thanks to all the bachelor and master students I supervised during my PhD. I hope that I was able to transmit to you as much knowledge and inspiration as you provided me over these years.

This journey was not only made up of colleagues but also of an amazing group of friends that I would like to thank. Over the years, I have thought several times about the order in which to write this part. I have decided to do it in the best way I can: in *random chronological order*.

Elena, you were one of the first people I met in Enschede. I could never have imagined that the two crazy months we spent together during your internship here would have such a profound impact on my experiences in this city. Every time we see each other, you bring me an indefinite amount of joy. Thank you for traveling here for my defense and for re-teaching me how to write formally in Italian. Apparently, I had completely forgotten how to do so.

Federico, thanks for being one of my paranymph today. I could not have chosen a better experienced paranymph in my life. Your friendship means a lot to me, and you showed me that life is not only about science but also about fun...and also about science while having fun. Thanks for all the legendary nights we spent together with Luigi il saggio, Iris, Sara, Davide, and Ani.

Guido and Michela, while you also belong to the gang, I want to give you a special thank you for making me the uncle of that awesome kid named Alessandro. Seeing his birth and growth over these years has been a truly heartwarming experience.

To the Avocado House: Martina, Lorenzo, and Federica. Thank you for being such great friends and for all the cheerful moments during the difficult time of the pandemic. We spent time cooking, playing, and enjoying each other's company. Thank you for all the crazy night in Enschede we had together with the Chief Nun Matteo, Giulio, and Stefano.

Antonia, my other pengui...ehm sorry, paranymph. When we met four years ago in Paris, neither of us could have expected the turns that both of our lives would take since that moment. Arranging your visiting PhD period in Enschede was one of the greatest decisions of our lives. We became inseparable friends (someone may say a bit "bulli"). We grew together, both in research and personal life, and always supported each other. Thanks for being such an amazing friend to me. I am extremely happy that you are joining our group as a postdoctoral researcher, so we can again enjoy endless coffee breaks in your office and chat about our lives. Thanks again for being my penguin today. I promise to try not to faint during my defense.

Speaking of your return, now you can finally take a closer look at our adopted "daughters" Ida and Sara. Jokes aside, I would also like to express my heartfelt

gratitude to them for all the time we have spent together, for their loyal long-lasting friendship, and for the *incrìvel aventuras* we had together so far and that are awaiting us in the future.

Benedetta, thanks for being that friend who is always there when needed, always there supporting me in both the brightest and darkest moments of my life. It is difficult to express in words just how wholesome and invaluable your friendship is to me. Thanks also for the amazing cover design of this thesis. I am delighted that these pages are adorned with your designer touch.

Rob, bedankt voor jouw vriendschap....ok I should stop trying to write in Dutch. Thanks for being such a great friend, for all the memories we made together in P-NUT, and for teaching me that even a Dutch person can have a bit of Neapolitan spirit at heart. For that, I am looking forward to the pizza store that we will open soon together.

To P-NUT, the most amazing PhD association in the world. Thank you to all the friends who joined in all the social and association activities. Being part of the board, for these four years, was a true privilege, and co-organizing events creating unforgettable memories was one of the best parts of my PhD life.

Thanks to all the beautiful souls who crossed paths with me during these four years and made it an unforgettable experience: Abhishta, Alice, Ana, Andrea, Baver, Daniele, Fardad, Federico O., Francesco, Giammarco, Kennet, Jacopo, Ionna, Lorenza, Lorenzino, Leonardo, Letizia, Lara, Marta, Maria, Valentina and many others.

Grazie a tutti i miei familiari e amici per essermi sempre stati vicini nonostante la distanza.

Ai miei genitori, mamma e papà: grazie per il più grande dono che mi avete dato, quello di far parte di questo mondo. Vi ringrazio per tutto l'amore e il supporto che mi avete dato e che continuate a darmi ogni giorno. So che non è facile vivere lontani e vedersi raramente, ma la vostra presenza e il vostro amore costante sono la roccia su cui continuo a costruire il mio futuro. Grazie ancora!

Enza, Nonna Enza....Nonnì. Sto scrivendo questi ringraziamenti in volo verso l'America a 10000 metri, magari da lassù mi senti di più così. Nonnì so quanto ci tenevi ad esserci e a vedere il tuo nipote acquisito diventare dottore, ma Crono è stato tiranno con noi e non ci ha voluto concedere questi ultimi sei mesi. Grazie per tutto l'amore che mi hai donato in questi anni e per avermi dimostrato che il bene va oltre ai legami di sangue. Quest'intera tesi è dedicata a te. Ti porterò per sempre nel mio cuore.

Raffaele

About the Author

Raffaele was born in Ottaviano (Italy) on December 25th, 1994. He received a Bachelor of Science (B.Sc.) degree (*cum laude*) in Computer Engineering from the University of Naples Federico II, Italy in 2016 and a Master of Science (M.Sc.) degree (*cum laude*) in Computer Engineering from Politecnico of Turin in 2018. During his master's degree, he worked on programmable dataplanes, which led to a publication on using SmartNICs as a DDoS mitigation technology.

From 2018 to 2023, he pursued his Ph.D. at the University of Twente, the Netherlands, where his doctoral research focused on DNS resilience. His research involved analyzing and characterizing DDoS attacks against DNS, investigating DNS misconfigurations and vulnerabilities, and quantifying existing as well as devising new countermeasures to mitigate attacks against DNS infrastructures.

In the summer of 2022, Raffaele was a visiting Ph.D. candidate at the Center for Applied Internet Data Analysis (CAIDA) in San Diego. During his time there, he began working on assessing KINDNS DNS hygiene best practices.

Raffaele can be reached via his private e-mail address: **raffysommy@gmail.com**

List of Publications

Below is a list of peer-reviewed academic publications that Raffaele has authored or co-authored in reverse chronological order:

- R. Sommese, M. Jonker, J. van der Ham, G.C.M. Moura. *"Assessing e-Government DNS Resilience"*, in Proceedings of the International Conference on Network and Service Management, 2022 [35].

- R. Sommese, K. Claffy, R. van Rijswijk-Deij, A. Chattopadhyay, A. Dainotti, A. Sperotto and M. Jonker. *"Investigating the Impact of DDoS Attacks on DNS Infrastructure"*, in Proceedings of the ACM Internet Measurement Conference, 2022 [34].

- G. Akiwate, R. Sommese, M. Jonker, Z. Durumeric, K. Claffy, G.M. Voelker, and S. Savage. *"Retroactive Identification of Targeted DNS Infrastructure Hijacking"*, in Proceedings of the ACM Internet Measurement Conference, 2022 [123].

- O. van der Toorn, R. van Rijswijk-Deij, R. Sommese, A. Sperotto, and M. Jonker. *"Saving Brian's Privacy: The Perils of Privacy Exposure through Reverse DNS"*, in Proceedings of the ACM Internet Measurement Conference, 2022 [124].

- L. Zembruzki, R. Sommese, L. Zambenedetti Granville, A. Selle Jacobs,M. Jonker, and G.C.M. Moura. *"Hosting Industry Centralization and Consolidation"*. In Proceedings of NOMS IEEE/IFIP Network Operations and Management Symposium, 2022[125]

- A. Affinito, R. Sommese, G. Akiwate, S. Savage, K. Claffy, G.M. Voelker, A. Botta, and M. Jonker. *"Domain Name Lifetimes: Baseline and Threats"* in Proceedings of the Network Traffic Measurement and Analysis Conference, 2022 [126].

- R. Sommese, G. Akiwate, M. Jonker, G.C.M. 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"* in Proceedings of the Network Traffic Measurement and Analysis Conference, 2021 [32].

- R. Sommese, L.M. Bertholdo, G. Akiwate, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy and A. Sperotto. *"Manycast2 – Using Anycast to Measure Anycast"*, in Proceedings of ACM Internet Measurement Conference, 2020 [33].

- 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 ACM Internet Measurement Conference, 2020 [72].

- R. Sommese, M. Jonker, R. van Rijswijk-Deij, A. Dainotti, K. Claffy and A. Sperotto. *"The Forgotten Side of DNS: Orphan and Abandoned Records"* in Proceedings of the Workshop on Traffic Measurements for Cybersecurity, 2020 [31].

- R. Sommese, G.C.M. 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 Proceedings of the Passive and Active Measurement Conference, 2020 [29].

- S. Miano, R. Doriguzzi-Corin, F. Risso, D. Siracusa, and R. Sommese. *"Introducing SmartNICs in Server-Based Data Plane Processing: The DDoS Mitigation Use Case"*, In IEEE Access Journal, 2019 [127]

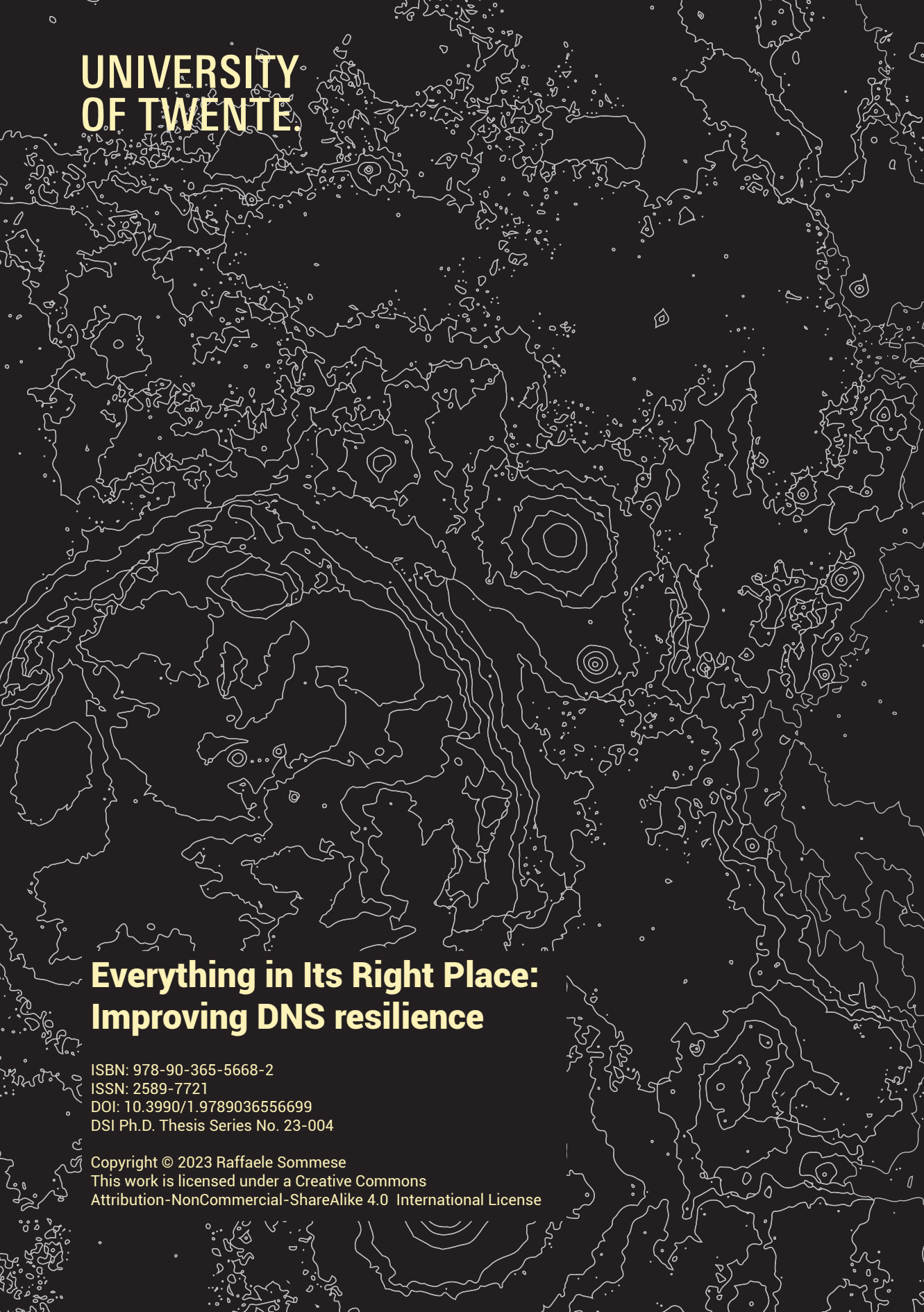
- G.B. Fioccola, R. Sommese, I. Tufano, R. Canonico, and G. Ventre. *"Polluino: An efficient cloud-based management of IoT devices for air quality monitoring"*, In Proceedings of International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow, 2016 [128].

Notable Awards

Raffaele's academic work has been recognised internationally. Below is a list of notable awards in reverse chronological order:

- **TMA 2021 Best Paper Award** – for his paper *"Characterization of Anycast Adoption in the DNS Authoritative Infrastructure"* [32].

- **RIPE Academic Cooperation Initiative (RACI) 2020 award** – as selected participant for his paper *"When Parents and Children Disagree: Diving into DNS Delegation Inconsistency"* (`https://www.ripe.net/participate/ripe/raci/alumni`)



UNIVERSITY OF TWENTE.

Everything in Its Right Place: Improving DNS resilience

ISBN: 978-90-365-5668-2
ISSN: 2589-7721
DOI: 10.3990/1.9789036556699
DSI Ph.D. Thesis Series No. 23-004