# Detecting malware and cyber attacks using ISP data

Andersen, Martin Fejrskov

# DETECTING MALWARE AND CYBER ATTACKS USING ISP DATA

BY
MARTIN FEJRSKOV ANDERSEN

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY
DENMARK

# Detecting malware and cyber attacks using ISP data

Ph.D. Thesis by

Martin Fejrskov Andersen

Cyber Security Group
Department of Electronic Systems
Aalborg University
Denmark

| | |
|---|---|
| Dissertation submitted: | May 2022 |
| PhD supervisor: | Professor Jens Myrup Pedersen<br>Aalborg University |
| Co-supervisor: | Assistant Professor Emmanouil Vasilomanolakis<br>Aalborg University |
| Company supervisor: | Per Olsen, Head of Security<br>Telenor A/S |
| PhD committee: | Professor Hans-Peter Schwefel (chairman)<br>Aalborg University, Denmark<br>Associate Professor Joao Gondim<br>University of Brasilia, Brazil<br>Professor Albert Cabellos<br>Universitat Politècnica de Catalunya, Spain |
| PhD Series: | Technical Faculty of IT and Design, Aalborg University |
| Department: | Department of Electronic Systems |

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-905-9

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

The cover image consists of two images from Pixabay [1] [2]

Printed in Denmark by Stibo Complete, 2022

# Abstract/Resumé

## English

Internet Service Providers (ISPs) can access their subscribers' IP traffic and various other data, such as information about the device type used, the approximate geographical location of a subscriber, and contact information for the subscriber. This data seems attractive to use for a number of use cases, such as malware and cyber attack detection, as a significant part of a country's Internet traffic is available, and as customers can be contacted for attack mitigation purposes. Although this is a use case with a desirable outcome for both subscribers, ISPs and the society as a whole, ISP data can also be used for much less honourable purposes. Therefore, the use of ISP data is subject to strict regulatory requirements in certain world regions such as the European Union.

In a collection of papers, this thesis expands the current state of the art by describing which data is technically and legally available to ISPs in the European Union, how the regulatory requirements on anonymization can be implemented, and by presenting a number of novel use cases for anonymized NetFlow and DNS data. The thesis presents a method that allows ISPs to use NetFlow data to estimate the amount of DNS traffic that is directed at 3rd party DNS resolvers compared to the amount of traffic towards the ISP's default DNS resolvers. The method is expanded in order to assess whether the 3rd party resolvers were chosen because they offer malware/parental filtering capabilities, or because they offer an uncensored service. A separate contribution presents a method to evaluate the impact of applying blacklists on an ISP's DNS resolvers. The method uses NetFlow records to assess which flows towards blacklisted domains and servers would never have been created if DNS based blocking was activated for all customers. This measure of impact is found to be an improvement to existing impact measurement methods, such as counting the number of blocked DNS requests. Another contribution presents a new botnet Command and Control (CnC) scheme that uses format-preserving encryption of IP addresses to provide an alternative to existing Fast Flux techniques. Although a DNS and NetFlow based detection method is also presented (and no botnets of this type are detected in the wild), such a CnC scheme could potentially raise the bar for defenders in the future. Finally, a method is presented that use NetFlow features to classify DNS resolvers in use as malicious or benign. Existing methods either rely on access to application layer data, which would not be legal to apply by ISPs in the EU, or rely on excessive Internet scanning, which may raise attacker awareness. Data from Telenor, a national ISP in Denmark, is used to validate and provide results for the theories and methods presented in the thesis.

The overall thesis statement claims that using the multitude of data types available from ISPs is advantageous for malware and cyber attack detection as compared to only using IP traffic data. In general, this claim is refuted, as the data technically and legally available to an ISP in the EU can be derived by deep inspection of the IP traffic. However, through the examples provided by the papers included in this thesis, it is shown that ISP data can be valuable when more specific use cases that are primarily applicable in an ISP context are considered, even despite anonymization being applied. While this conclusion is positive from a privacy point of view, it can still be debated whether the legislation provides the right balance between privacy and cyber security.

## Dansk

Internetudbydere har adgang til deres abonnenters IP-trafik og forskellige andre data, f.eks. oplysninger om den anvendte enhedstype, abonnentens omtrentlige geografiske placering og abonnentens kontaktoplysninger. Disse data kan være attraktive at bruge til en række formål, såsom detektering af malware og cyberangreb, da en væsentlig del af et lands internettrafik er tilgængelig, og da kunder kan kontaktes så konsekvenserne af et angreb kan afbødes. Selvom dette giver et ønskeligt resultat for både abonnenter, internetudbydere og samfundet som helhed, kan internetudbyderdata også bruges til langt mindre prisværdige formål. Derfor er brugen af data fra internetudbydere underlagt strenge lovgivningsmæssige krav i visse verdensregioner som f.eks. den Europæiske Union.

I en samling af artikler udvider denne afhandling den aktuelle state of the art ved at beskrive, hvilke data der er teknisk og juridisk tilgængelige for internetudbydere i den Europæiske Union, hvordan de lovgivningsmæssige krav til anonymisering kan implementeres, og ved at præsentere en række nye brugsformål til anonymiserede NetFlow- og DNS-data.

Afhandlingen præsenterer en metode, der gør det muligt for internetudbydere at bruge NetFlow-data til at estimere mængden af DNS-trafik rettet mod tredjeparts DNS-resolver til sammenligning med mængden af trafik

mod internetudbyderens standard DNS-resolvere. Metoden udvides til at vurdere, om tredjeparts-resolverne blev valgt, fordi de tilbyder malware-/forældre-filtreringsfunktioner, eller fordi de tilbyder en ucensureret service. Et andet bidrag præsenterer en metode til at evaluere virkningen af at anvende blacklists i en internetudbyders DNS-resolvere. Metoden bruger NetFlow-data til at vurdere, hvilke flows mod blacklistede domæner og servere, der aldrig ville være blevet oprettet, hvis DNS-blokeringen var aktiveret for alle kunder. Denne måling af indvirkning viser sig at være en forbedring af eksisterende målemetoder, såsom måling af antallet af blokerede DNS-forespørgsler. Et andet bidrag præsenterer en ny metode til botnet Command and Control (CnC), der bruger formatbevarende kryptering af IP-adresser som et alternativ til eksisterende Fast-Flux-teknikker. Selvom en DNS- og NetFlow-baseret detektionsmetode også præsenteres (og ingen eksisterende botnets af denne type opdages), kan metoden potentielt hæve barren for forsvarere i fremtiden. Derudover præsenteres en metode, der bruger NetFlow-features til klassificere DNS-resolver som ondsindede eller godartede. Eksisterende metoder er enten afhængige af adgang til applikationslagsdata, som ikke ville være lovlige at anvende af internetudbydere i EU, eller er afhængige af storstilet internetscanning, som kan advare angriberen. NetFlow- og DNS-data fra Telenor, en national internetudbyder i Danmark, bruges til at validere og levere resultater for de teorier og metoder, der præsenteres i afhandlingen.

Afhandlingens overordnede påstand er, at det til afsløring af malware og cyberangreb er fordelagtigt at bruge de mange datatyper, der er til rådighed for internetudbydere, fremfor kun at bruge IP trafikdata. Generelt tilbagevises denne påstand, da de data, der teknisk og juridisk er tilgængelige for en internetudbyder i EU, kan udledes fra en dybere inspektion af af IP-trafikken. Gennem eksempler præsenteret i afhandlingens artikler, viser det sig imidlertid, at internetudbyderdata kan være værdifulde selv om anonymisering anvendes, når mere specifikke brugsformål, der primært gælder i en internetudbyder-sammenhæng, overvejes. Selv om denne konklusion er positiv ud fra et privatlivssynspunkt, kan det stadig diskuteres, om lovgivningen skaber den rette balance mellem privatlivets fred og cybersikkerhed.

# Preface

This thesis presents the research output of my three-year Industrial PhD study project at Telenor A/S, a national Internet Service Provider in Denmark, and the Department of Electronic Systems at Aalborg University in Denmark. The project was launched with the overall goal to develop methods to use Internet and mobile telephony providers' knowledge of customers and data traffic to identify cyber-attacks and malware infections. The thesis is submitted as partial fulfilment of the requirements for obtaining the degree of Doctor of Philosophy (PhD) from Aalborg University.

The thesis is written in the format of a collection of papers, and therefore the bulk of the thesis consists of 5 scientific papers published in, or submitted to, peer-reviewed conferences. I am the main author and contributor to the papers, and statements from each co-author detailing their contribution to each paper have been approved by the The Technical Doctoral School of IT and Design prior to the submission of this thesis. These statements are also presented to the PhD committee as part of the assessment. The 5 papers considered part of the thesis are listed below in chronological order of writing.

**Paper A** : Martin Fejrskov, Jens Myrup Pedersen, Emmanouil Vasilomanolakis: "Cyber-security research by ISPs: A NetFlow and DNS Anonymization Policy", published and presented at the International Conference on Cyber Security And Protection Of Digital Services (Cyber Security) 2020 (Chapter 2 on page 5)

**Paper B** : Martin Fejrskov, Jens Myrup Pedersen, Emmanouil Vasilomanolakis: "Using NetFlow to measure the impact of deploying DNS-based blacklists", published and presented at the EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2021 (Chapter 5 on page 41)

**Paper C** : Martin Fejrskov, Jens Myrup Pedersen, Leon Böck, Emmanouil Vasilomanolakis: "An uneven game of hide and seek: Hiding botnet CnC by encrypting IPs in DNS records", published and presented at the IEEE Conference on Communications and Network Security (CNS) 2021 (Chapter 6 on page 63)

**Paper D** : Martin Fejrskov, Emmanouil Vasilomanolakis, Jens Myrup Pedersen: "A study on the use of 3rd party DNS resolvers for malware filtering or censorship circumvention", accepted at the International Conference on ICT Systems Security and Privacy Protection (IFIP SEC) 2022 (Chapter 7 on page 85)

**Paper E** : Martin Fejrskov, Jens Myrup Pedersen, Emmanouil Vasilomanolakis: "Detecting DNS hijacking by using NetFlow data", submitted for review to the International Workshop on Security (IWSEC) 2022 (Chapter 8 on page 103)

In addition to the papers listed above, the following article is co-authored during the PhD studies. As this article is of more legal than technical nature, it is not considered a part of the thesis, and is not included in print.

**Paper F** : Leon Böck, Martin Fejrskov, Katerina Demetzou, Shankar Karuppayah, Max Mühlhäuser, Emmanouil Vasilomanolakis: "Processing of Botnet Tracking Data under the GDPR", published in the Journal of Computer Law & Security Review 2022 [3]

The PhD project is fully funded by Telenor A/S and Innovation Fund Denmark. I would therefore like to thank both organizations for giving me the opportunity and support to pursue a PhD study. In particular, I would like to thank Peter Nødbak, my former manager at Telenor, for initiating the project, and Per Olsen, Head of Security at Telenor, and Brian Jørgensen, Department Manager at Telenor, for their role as company supervisors. At Aalborg University, thanks goes to the Cyber Security Group including my supervisors Professor Jens Myrup Pedersen and Assistant Professor Emmanouil Vasilomanolakis, my fellow PhD student Kaspar David Hageman and the rest of the Cyber Security Group.

Martin Fejrskov Andersen

Martin Fejrskov Andersen<br>
2nd of May, 2022

# Contents

# CHAPTER 1 Introduction

There is a growing cyber threat from both criminal actors and nation states that targets both public authorities and private companies [4]. This calls for an increased security posture throughout the society, and a majority of businesses expect to increase investments in cyber security in 2022 [5]. This does, however, not only apply to authorities and business. The majority of consumers are also willing to pay Internet Service Providers (ISPs) to provide increased cyber security protection [6] [7].

It varies how and if ISPs seize this opportunity, and the efforts fall into three main categories. Some products inspect or modify Internet traffic at end user devices, such as traditional anti-virus products for personal computers. Other products take advantage of the ISP's ownership of the network infrastructure and processes the same Internet traffic on equipment deployed in the ISP's network, such as firewalls or Distributed Denial of Service (DDoS) protection systems. A third category of products does not process the consumers' Internet traffic at all, such as cyber attack insurance plans, backup services, mobile device management, or similar. Although the first and third categories of products can be offered by any company, the second option can only be provided by ISPs. As the second option could also represent an easier adoption path for a consumer (as no apps need to be installed etc.), this provides ISPs with a unique value proposition.

ISPs have access to the traffic to all customers, and ISPs know which IP addresses are assigned to which customers. This knowledge could be used to identify attacks and warn the customers, even if they do not subscribe to any security products. Being able to deploy such a detection and warning mechanism is also a value proposition that is unique to ISPs.

Having an ISP inspect the traffic of all customers to provide a broad threat overview and early warning of unprotected subscribers does, in spite of the good intentions, come with a significant privacy threat. The European Union ePrivacy Directive therefore clearly forbids ISPs to inspect data that is not relevant to process in order to deliver a specific product [8]. As the payload of Internet packets do not need to be inspected when delivering an Internet access product, it is not legal to inspect the payload for malware detection purposes either.

A tempting solution to this problem is to bundle all Internet access products with a security product that requires each customer's traffic to be inspected so that malware can be detected and blocked. Introducing such a product into all existing subscriptions as an opt-out solution is, however, not legal in the European Union either, due to the Net Neutrality Regulation [9]. On the other hand, an opt-in solution is unlikely to have sufficient adoption to provide the desired overview of threats to (nearly) all customers.

Network-wide malware detection at an ISP in the EU will therefore need to operate on the set of data that is permitted by the ePrivacy Directive without additional opt-ins. At the first glance, this seems to limit the malware detection options at ISPs. However, as ISPs are in the unique position that they have access to other sources of data than just the IP traffic, the limitation imposed by the ePrivacy Directive may be offset by opportunities derived from the availability of these data sources. Examples of such data sources include information about which customers are assigned which IP address, which Radio Access Technology is used by a mobile phone, which cell a phone is attached to, which DNS lookups are performed by that customer, and which device brand and model is used by a subscriber. This leads to the following thesis statement:

*Compared to only using IP traffic data, it is advantageous to use data from ISPs to identify cyber attacks against customers and to identify customers infected with malware.*

## 1.1 Thesis outline

This thesis consists of 10 chapters, of which 5 contain papers that are published or in review. Paper A and Paper D focus on anonymization and data quantification rather than on malware. Paper B, Paper C and Paper E narrow the focus to different aspects of detection of malware and cyber attacks. The remaining chapters provide background information, discussion etc. as outlined below. Paper A is mostly qualitative of nature, but the proposed method is applied in the remaining papers. These papers primarily use a quantitative approach to evaluate proposed methods and prototypes.

Given the thesis statement presented above, it is natural to start out by asking the following initial question: Which kind of data is available to an ISP, and under which conditions can it be used? The contribution of Paper A in **Chapter 2** is to answer this question by providing an overview of the data available from both a technical and legal perspective. The paper concludes that anonymized DNS data from the ISP's own resolvers and NetFlow data from the ISP's routers are the only data sources that are both available and relevant for malware detection. By considering related works on privacy, the paper proposes an anonymization policy for the two data sources.

**Chapter 3** follows up on the conclusion of Chapter 2 by surveying the state of the art of using DNS or NetFlow data for malware detection purposes. Both of these data types have individually been the subject of many academic papers, and they are also actively used in commercial products. However, no commercially available products and only a few academic papers rely on the combined feature set of DNS *and* NetFlow data. Neither academic papers or commercial products address the anonymization requirement. Using a combination of anonymized DNS and NetFlow data to detect malicious traffic therefore represents a relatively unexplored niche from both an academic and commercial perspective.

Combining and making conclusions based on NetFlow and DNS data is a more complex process than it seems to be at the first glance. To address this, **Chapter 4** provides a brief introduction to this topic, as well as an overview of the properties of the data used for this thesis.

When performing the survey in Chapter 3 it was clear that a common security measure is to let DNS resolvers block responses containing domain names and IP addresses that are known to host malicious content. The impact of such blacklists are often measured by the number of blocked DNS responses. As not all DNS queries are followed by a connection towards the IP address in the response, measuring the number of blocked DNS responses could overestimate the impact significantly. The contribution of Paper B in **Chapter 5** is to use NetFlow records to measure the impact instead. The method also makes it possible to assess the amount of web vs. non-web traffic being blocked, how many flows are created towards blacklisted IP addresses that host multiple (benign and/or malicious) domains, how many flows are created towards spam hosts vs. phishing/malware/botnet hosts, etc. The method is applied to data from Telenor DK to show that an approach that includes both DNS and NetFlow data represents an improvement to existing impact measurement methods.

It is a common assumption that when an application receives a DNS response containing a particular IP address, it will create a connection to this specific address (if a connection is created at all). It is, however, interesting to consider what could happen, if this assumption is intentionally violated by a malicious actor. For example, what if a bot-master does not want to divulge the IP address in clear-text in a public directory? Such a botnet scheme is the contribution of Paper C in **Chapter 6**, where the main idea is to encrypt the IP address using semantic- and format-preserving encryption, create an DNS A record containing the encrypted IP address, and let the client application perform the decryption before making a connection to the CnC infrastructure. As the scheme does not divulge the clear-text IP address in DNS records, and as the use of Fast Flux (FF) is not necessary, use of the proposed scheme raises the bar for the defender, as the scheme makes it possible for a botnet to avoid existing FF detection mechanisms (by not using FF), and make it impossible for a defender to create IP address based blacklists derived from DNS data. Paper C also contributes with a DNS and NetFlow based detection method that is validated using an emulated bot deployed in Telenor DK's network.

Applying blacklists on ISP resolvers as described in Chapter 5 will only increase the security posture of a user if the user uses the ISP's resolvers. Furthermore, the detection method outlined in Chapter 6 requires that the ISP has access to DNS application layer logs. A natural question is therefore if sufficiently many users are using the ISP's resolvers for the DNS logs to be considered representative for the entire user population. The answer to this question is one of the contributions of Paper D in **Chapter 7**. In the paper, a method is presented that uses anonymized and sampled NetFlow records to estimate how many DNS responses are returned from 3rd party DNS resolvers compared to the default ISP owned resolvers. Both UDP, TCP, DNS-over-TLS and DNS-over-HTTPS based queries are considered. The method is applied to data from Telenor DK to show that less than 10% of the total DNS traffic is from 3rd party resolvers. The second contribution of Paper D is to outline the *reasons* users can have for not choosing ISP owned resolvers. Two reasons are explored in depth, namely the availability of malware and/or parental filtering on the resolver, and the desire to circumvent censorship performed by the ISP's DNS resolvers. Applying the method to data from Telenor DK suggests that 3rd party resolvers are not to a great extent chosen because of their malware filtering capabilities, but they are chosen in order to circumvent censorship.

As elaborated above, users have the freedom to choose which DNS resolver should be used. However, in some cases malware will make this decision on behalf of the unwitting user and select a malicious resolver. When using a malicious resolver, DNS based security measures such as DNSSEC or DNS-over-TLS can no longer be trusted, and a large number of redirection attacks become easier. In Paper E in **Chapter 8** it is assumed that a user will never actively choose to use a resolver that is not being advertised on the web as being a public resolver. The contribution of the paper is a Random Forest based method that uses NetFlow features to classify the resolvers actually being used as either well-known or malicious. This approach has a number of advantages to existing methods, such as avoiding excessive Internet-wide scanning, not relying on features controllable by a malicious actor, and that access to the application layer data (such as the domain names or IP addresses in the responses) is not needed.

A discussion of selected topics that relate to more than a single paper, such as the applicability of the anonymization policy proposed in Paper A and the business value of the research field from an ISP perspective, is the topic of **Chapter 9**. **Chapter 10** concludes on the thesis statement and provides perspectives on the future use of ISP data for malware detection.

CHAPTER 2

# Paper A

## Cyber-security research by ISPs: A NetFlow and DNS Anonymization Policy

**Main author:**
Martin Fejrskov
*Technology, IP Network and Core*
*Telenor A/S*
Aalborg, Denmark
mfea@telenor.dk

**Co-authors:**

Jens Myrup Pedersen
*Cyber Security Network*
*Aalborg University*
Aalborg, Denmark
jens@es.aau.dk

Emmanouil Vasilomanolakis
*Cyber Security Network*
*Aalborg University*
Copenhagen, Denmark
emv@es.aau.dk

**Presented at:**
*International Conference on Cyber Security and Protection of Digital Services (Cyber Security) 2020*

Funded by Telenor A/S and Innovation Fund Denmark.

# Abstract

*Internet Service Providers (ISPs) have an economic and operational interest in detecting malicious network activity relating to their subscribers. However, it is unclear what kind of traffic data an ISP has available for cyber-security research, and under which legal conditions it can be used. This paper gives an overview of the challenges posed by legislation and of the data sources available to a European ISP. DNS and NetFlow logs are identified as relevant data sources and the state of the art in anonymization and fingerprinting techniques is discussed. Based on legislation, data availability and privacy considerations, a practically applicable anonymization policy is presented.*

**Keywords:** *ISP · privacy · DNS · NetFlow · IPFIX · cyber-security · anonymization*

## 2.1 Introduction

Research in cyber-security is highly dependent on the availability of real-life traffic traces for a number of different purposes. When collecting these traffic traces, researchers and practitioners should consider aspects like legal requirements and the privacy risk involved. However, these topics may not be within the researchers' area of knowledge. This can have a number of undesirable consequences like increased project lead time, legal problems when sharing project data, or spending time on research that is irrelevant because it cannot be applied in practice. The purpose of this paper is to help researchers and practitioners avoid some of these pitfalls when collecting data at an ISP level.

To protect the privacy of the subscribers, European ISP legislation forbids the use of certain data, and sets anonymization requirements on data usage, requirements that also apply to positive use cases like cyber-security research. However, the legislation does not present which specific anonymization techniques must be used for specific data sources. Some studies of anonymization techniques and privacy risks focus broadly rather than on giving practical guidelines for specific use cases. Other studies investigate how specific data sources can present a privacy problem in different use cases, but do not consider if the data is already unavailable from a legal perspective or how to mitigate the privacy risk.

In this paper, we firstly identify the ISP data sources legally and technically available for research. Furthermore, we present a practically applicable and privacy-preserving anonymization policy, for NetFlow and DNS logs, that complies with the relevant ISP legislation. This allows researchers and developers to start with a focus on implementation rather than legislation when creating solutions targeted for ISP deployments.

This paper is organised in seven sections. Section 2.2 gives an introduction to the relevant legislation and anonymization requirements, and Section 2.3 on page 9 provides an overview of the data sources often technically available to an ISP. Having limited the relevant scope to two data sources, Section 2.4 on page 12 presents related work on anonymization techniques and on subscriber fingerprinting based on anonymized DNS and NetFlow logs. Sections 2.5 on page 14 and 2.6 on page 20, build upon the knowledge derived from all previous sections to propose and discuss concrete anonymization policies for individual fields in NetFlow and DNS logs, thus providing the primary contribution of the paper. Lastly, Section 2.7 on page 23 summarizes and concludes the paper.

## 2.2 Legislation

To identify the legal opportunities and challenges, an overview of relevant legislation is needed, which will be the topic of this section.

### 2.2.1 ePrivacy Directive

The ePrivacy Directive [8] from 2002 and the related national implementations, regulate among other things how ISPs are allowed to handle data related to the subscribers data traffic. The 2009 update of the ePrivacy Directive does not contain any changes relevant to this paper.

Although the General Data Protection Regulation (GDPR) [10] is newer than the ePrivacy Directive, the latter is considered *lex specialis* to the GDPR. This means that the ePrivacy Directive overrides the GDPR in any situation that is specifically described in the ePrivacy Directive. Furthermore, as the ePrivacy Directive specifically regulates ISPs and their handling of subscriber data traffic, the GDPR is considered out of the scope of this paper.

Articles 5, 6 and 9 in the ePrivacy Directive set the following limitations relevant to this paper on processing a subscribers traffic or location data:

- Data *already* being processed for the purpose of transmission must be made anonymous before additional processing.
- Data *not* being processed for the purpose of transmission or as part of a value added service cannot be processed.
- Data can be processed for a specific value added service but only if consent is available.

In the context of this paper, "processing" means any form of storage, manipulation, forwarding etc. of customer IP traffic, location data etc. [10] In addition, "processing for the purpose of transmission" refers to processing needed to transfer IP packets (routing, switching), performing DNS lookups (caching, recursing), authenticating the subscribers, routing packets to the correct cell tower and similar operations [8].

As it is practically impossible to have all subscribers sign up to a value added service relating to cyber-security research (and thereby providing consent), using anonymized data is the only viable strategy.

### 2.2.2 Opinion on Anonymization Techniques

Various anonymization and pseudonymization techniques and their relation to the legal framework are described in "Opinion 05/2014 on Anonymization Techniques" [11]. "Opinion" documents contain the elaboration of a specific directive or regulation, and are considered recommendations, not legislation. This specific opinion is written to elaborate on the anonymization requirements in the Data Protection Directive, a predecessor to the GDPR, but is still applicable and relevant.

Both the Opinion and recital 26 in the GDPR make a clear distinction between pseudonymization and anonymization, and makes it explicit that a requirement from the ePrivacy Directive to anonymize certain data is not fulfilled by the use of pseudonymization.

Two main anonymization techniques are described:

- **Randomization** including noise addition and permutation techniques "alter the veracity of the data in order to remove the strong link between the data and the individual". As an example, an IP address (A) in a specific data record can be substituted with a random IP address (B), and the same IP address (A) in another data record can be substituted with a different random IP address (C).

- **Generalization** including aggregation (k-anonymity), L-diversity and T-closeness techniques "generalize, or dilute, the attributes of data subjects by modifying the respective scale or order of magnitude". As an example, the IP addresses in all data records can be replaced by a smaller IP prefix.

The differential privacy technique is also described, but as this technique requires the original data to be retained, this technique is not compliant with the anonymization requirement of the ePrivacy Directive.

The Opinion concludes that in most cases it is not possible to give minimum recommendations for parameters to use as each data set needs to be considered on a case-by-case basis.

### 2.2.3 Summary

The ePrivacy Directive mandates that only data already being processed by the ISP for the purpose of transmission can be used for cyber-security research, and the data can only be retained in an anonymized form. The Opinion on Anonymization Techniques details which anonymization techniques are considered compliant. The specific data sources to be used for cyber-security research must therefore be determined before further anonymization considerations can be made.

## 2.3 Data sources

The restrictions posed by legislation depends on the type of data that is to be processed. In this section, the data sources available to Telenor Denmark will be described as a representative example of data sources being generally available to an ISP. Table 2.1 on the next page summarizes the data sources, their content and their usage restrictions based on the presentation of legislation in section 2.2 on page 7. This will provide an overview of which data sources are both legally *and* technically available for cyber-security research, which can help researchers determine if their research can be applied legally in practice.

The data sources that require anonymization are described in more detail in the following sections. Note that all data sources containing personal identifiable information like IP addresses require anonymization or consent to be used. Omitted are those that are not relevant in relation to Internet cyber-security research, thus excluding for example the SMS/MMS service and non-Internet based telephony services. Section 2.3.4 on page 12 summarizes and discusses possible use cases for the available logs.

### 2.3.1 Identity of the subscriber

**IP assignment log** Assigning an IP address to a subscriber is handled by different components depending on the access type (DSL/fiber/coax/mobile). Each component can, however, create an accounting log entry containing the subscriber identity (DSL-number or IMSI) and the assigned/revoked IP address. In a Telenor Denmark context, the DSL-number is a 4-6 digit broadband customer identifier that (despite the name) enumerates both coax, fiber and DSL customers.

**CGNAT log** If mobile subscribers are assigned private IP addresses, Carrier Grade Network Address Translation (CGNAT) functionality is used. CGNAT can operate just like regular Network Address Translation (NAT) except

| **Name** | **Usage restriction** | **Contents** |
|---|---|---|
| IP assignment log | Anonymized | IP address, IMSI/IMEI/DSL-number |
| CGNAT log | Anonymized | Private/public IP address, port block |
| Customer database | Contract/consent | Person name, geographical address, IMSI/DSL-number |
| Modem/router at customer | Contract/consent | Attached device name, MAC and IP |
| EPDG CDR log | Anonymized | IP address, IMSI, RAT type (WiFi) |
| Cell database | None | Geographical address, gain/height/tilt etc. |
| Mobility event log | Anonymized | IMSI/IMEI, RAT type, cell ID |
| NetFlow log | Anonymized | TCP/UDP/IP session information |
| DNS log | Anonymized | Source IP address and port, queried domain name and response |
| Layer 3-7 DPI | Contract/consent | IP address, malware type |
| PGW application log | Contract/consent | IMSI/IMEI, IP address, layer 7 specific information |
| PGW flow log | Contract/consent | IMSI/IMEI, TCP/UDP/IP session and layer 7 application enumeration |

**Table 2.1:** *ISP data sources relevant for cyber-security research*

that the NAT is performed at the ISP premises rather than at the customer premises. Multiple customers thereby share the same public IP. The Telenor Denmark CGNAT device reserves a range of 64 ports (a "port block") to each private IP address. Upon assigning/revoking this port block, a CGNAT log entry is created containing private IP address, public IP address and port block. Notice that a log entry is not created for each TCP/UDP session, it is only created for each port block allocation. The use of NAT logs can be relevant when distinguishing between different mobile subscribers sharing the same IP address.

**EPDG CDR log** In order to use Voice-over-WiFi service the mobile phone must create an IpSec tunnel towards the Evolved Packet Data Gateway (EPDG). The EPDG can create a log line containing the IMSI and the source IP address of the IpSec tunnel. This log is known as a Call Data Record (CDR), despite the fact that it is not the phone call, but the tunnel establishment that is logged. This provides two interesting pieces of information: First the fact that a phone is attached to WiFi rather than being completely offline. Second, it shows which broadband subscription the mobile phone is connecting from. This can be used to distinguish between an infected broadband subscriber and an infected mobile subscriber using a broadband subscriber′s WiFi.

### 2.3.2 Mobile location

**Cell Database** Information about the geographical location, frequency, antenna gain/height/tilt, topography etc. of all cells is available in a central database. This can be used to estimate the coverage area of a specific cell.

**Mobility event log** Phone mobility on 4G is handled by the Mobility Management Entity (MME) component, and this component can create a log line for each mobility event containing the subscriber identity (IMSI/IMEI), the destination cell identity (a 5-6 digit number) and the destination Radio Access Technology (RAT, 2G/3G/4G).

### 2.3.3 Internet activity

**NetFlow log** The routers of the backbone network can emit NetFlow/IPFIX records. Most ISPs have equipment capable of doing this, but the specific implementation varies. ISPs may emit NetFlow logs from all routers or no routers, and may use varying levels of sampling/aggregation.

**DNS log** Most subscribers (both mobile and broadband) use the ISP's DNS resolvers for name resolution. A log entry can, depending on the logging method, contain the client source IP/port, the query and the response. The authoritative DNS servers are considered less relevant for the topic of this paper, as traffic from ISP subscribers will in most cases be visible at the DNS resolvers as well.

### 2.3.4 Summary

This section outlines the different data sources available to Telenor Denmark as an example of a typical ISP, and identifies if consent or anonymization is required for data usage. Specifically for cyber-security research, the point of focus is the Internet activity (described by DNS and NetFlow logs) rather than the location or the subscriber identity. The DNS and NetFlow logs must be anonymized before use, and this is the topic of the rest of the paper.

## 2.4 Related work

The previous sections argue that of the data sources technically and legally available to an ISP, NetFlow and DNS logs are the most interesting to cyber-security research. Having limited the scope, it is now relevant to identify existing, related work on DNS and NetFlow anonymization and fingerprinting. First, we provide a few notes on terminology and a general overview of related work. Afterwards, we discuss relevant papers in NetFlow and DNS respectively.

### 2.4.1 Terminology and overview

Many papers describe topics relating to anonymization, privacy and fingerprinting, so in order to discern which papers are the most relevant, an introductory note on terminology and preconditions is needed:

- Aggregation vs. generalization: Some papers use the terms generalization and aggregation interchangeably or with different definitions. For this paper, the terminology applied in RFC6235 will be used [12], and only generalization approaches are considered to preserve utility.

- Anonymization vs. pseudonymization: A brief look at existing literature, including taxonomy papers, shows that the distinction between anonymization and pseudonymization required by legislation is not often used, as typically the term "anonymization" is used for pseudonymization techniques as well.

- Anonymization must be applied before data analysis: Some techniques such as (k,j)-obfuscation [13] are based on a statistical analysis of the

| | Related Work | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Aspect** | 11 | 12,14 | 15 | 16,17,18,19 | 20 | 21 | 22 | 23 |
| NetFlow or DNS | | X | X | X | X | X | | X |
| Anon. techniques | X | X | X | X | | | X | |
| Protocol fields | | X | | | | X | X | X |
| Privacy risk | X | X | X | | X | X | | |
| Legislation | X | | X | | | | | |

**Table 2.2:** *Notable related work and aspects in focus*

entire data set to be obfuscated, thus requiring all data to be stored in a non-anonymized form prior to release, which is not in line with legal requirements.

The goal of this paper is to provide a *DNS and NetFlow* anonymization policy stating which *anonymization technique* should be applied for individual *protocol fields*, while taking the *privacy risk* and ISP related *legislation* into consideration when focusing on the cyber-security research use case. Related works cover some but not all of these aspects, as illustrated by Table 2.2.

### 2.4.2 NetFlow

A good introduction to the topic of passive internet measurement in general, including many aspects ranging from a legal overview to lessons learned on various practical deployment work is written by the authors of [15]. One of the lessons learned is that considering legislative aspects is a time consuming and complicated process, a problem that this paper attempts to address.

RFC 6235 provides a thorough walk-through of anonymization and pseudonymization options for the individual fields of the IPFIX protocol [12]. The paper categorizes various anonymization techniques into different classes, however, only the classes named "generalization" (such as truncation) or "set substitution" (such as noise addition) can be considered anonymization rather than pseudonymization techniques [11]. The paper does not provide any specific suggestions such as the length to be used for IP address truncation or on how much the precision of a timestamp should be degraded.

A comprehensive survey of anonymization techniques and 25 tools is written by the authors of [14]. The paper also discusses the relevance of anonymizing different fields in the different protocol layers in a network packet capture. The paper concludes with a number of statements like "The port number should not be anonymized as it will have a big impact on the usefulness of a network capture and cannot be directly used for identification" [14] and "Currently, in an environment without completely trusted parties, it is not recommended to share complete anonymized datasets. The current protection against re-identification is still inadequate." (due to the large amount of context available in complete datasets) [14]

### 2.4.3 DNS

Two papers show that it is possible to perform user fingerprinting based on the domain name part of DNS logs [16, 17]. However, no suggestions on how to anonymize the DNS logs in data storage / mining environments are provided.

The authors of [19] describe the best privacy practices for DNS operators. Authenticity and confidentiality mechanisms like DNSSEC and DNS-over-TLS are described, but the section detailing how to protect data at rest focuses mainly on data minimization, IP address anonymization and TCP/TLS related features.

The implications of using only requests for the top *n* most popular host names for identity fingerprinting, as well as using only requests for anything but the *n* most popular host names is discussed by the authors of [16]. This is relevant in the context of cyber-security research as this idea can be used for data minimization, thus decreasing the privacy risk.

Bloom filters [24] rely on hash functions to store domain names in an irreversible way. While this provides good privacy, it also reduces the utility of the stored data, as data can then only be used to search for *already known* malware related domain names. This excludes for example domain names created by a Domain Generation Algorithm. While this can be sufficient from an operational perspective, it is less interesting to a cyber-security researcher, and therefore Bloom filters will not be considered further in this paper.

### 2.4.4 Summary

Related work does not provide a concrete answer on how an anonymization policy could be implemented, but does provide some good directions. The most specific input to an anonymization policy is provided by RFC 6235, which suggests specific techniques like truncation, but not directions on the truncation length. Based on these directions, section 2.5 and 2.6 on page 20 will provide a suggestion for a legally compliant anonymization policy suitable for ISP cyber-security research.

## 2.5 NetFlow anonymization policy

Based on the directions offered by legislation and related work on anonymization of NetFlow described in the previous sections, this section will provide a suggestion for a legally compliant anonymization policy suitable for ISP

| Field | Technique | Specifics |
|---|---|---|
| Bytes/packets | Precision degradation | NetFlow 1:n sampling |
| Start/end time | Reverse truncation | Remove AM/PM info |
| IP addr.(no NAT) | Truncation | Truncate to /24 prefix |
| IP addr.(CGNAT) | None | - |
| IP addr.(Infrastr.) | None | - |
| IP addr.(external) | Truncation | Truncate to /16 prefix |
| IP protocol | Binning | TCP+UDP+ICMP/"other" |
| ICMP type+code | None | - |
| Port (no NAT) | None | - |
| Port (CGNAT) | Truncation | Truncate to /2 prefix |
| Port (Infrastr.) | None | - |
| Port (external) | None | - |
| TCP flags | None | - |

**Table 2.3:** *NetFlow anonymization policy assuming 64 port block based CGNAT*

cyber-security research. The section describes the choice of protocol field-/features, elaborates on the choice of anonymization technique for the individual fields, and concludes by providing the pseudo-code implementing such a policy.

### 2.5.1 Choice of features

The IPFIX features most typically used for cyber-security research [25] are listed in Table 2.3 along with the suggested anonymization policy. ICMP type/code and TCP flags are also added to table. The following paragraphs describe the considerations for each field noted in the table.

### 2.5.2 Feature anonymization details

**Total bytes and packets** The total count of bytes and packets in a TCP/UDP session can be used for user profiling and for attacks against other anonymization techniques [12]. Moreover, it can under some circumstances be used as part of an algorithm to determine which web sites are visited [23].

The discussion may, however, be less important in practice, as NetFlows are typically sampled 1:n when collected by an ISP for performance reasons. The sampling also automatically provides a precision degradation of packet and byte counts, which is considered a valid method of anonymization for that field [12]. From a performance perspective, network equipment vendors consider $n <= 512$ a very low sample rate. This order of magnitude for sampling seems likely to be sufficient for anonymization purposes although to the best of our knowledge, no research has been conducted on quantifying this.

**IP addresses** The authors of [18] conclude that if any other type of IP address anonymization technique than truncation is applied, re-identification of a host in NetFlow traffic is possible when active fingerprinting techniques are applied. If IP address truncation is applied, other fields may still be able to identify the host, though.

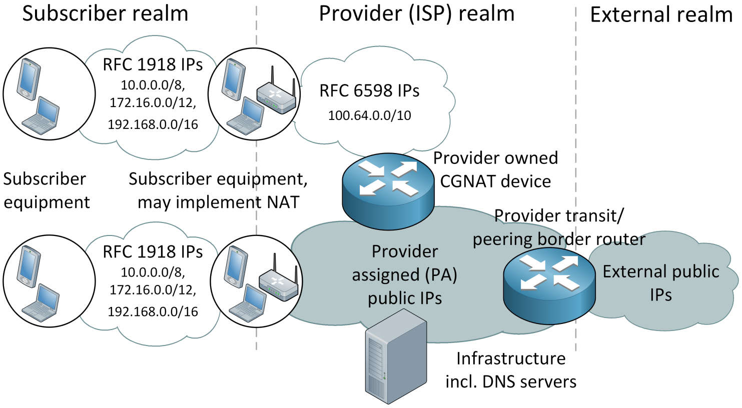

**Fig. 2.1:** *Classification of IP addresses*

From the perspective of an ISP capturing NetFlow at the border routers, 4 different categories of IP addresses are relevant to describe separately, as illustrated in Figure 2.1: Some Provider Assigned (PA) IP addresses are allocated directly to subscriber equipment (typically broadband routers at customer premises), some PA IP addresses are allocated for use on the outside of the CGNAT device and some PA IP addresses are allocated to ISP infrastructure, including the DNS servers, content caches, routers etc. IP addresses outside the ISP/provider realm (more specifically IP addresses not announced through Border Gateway Protocol (BGP) by the ISP outside the ISPs Autonomous System (AS)) will simply be referred to as "external" IP addresses.

Whether PA and external IP addresses should be subjected to the same truncation length is discussed in [20] based on a "risk vs. utility" analysis. Choosing the "sweet spot" with the most utility preserved, this would be equivalent to truncating PA IP addresses to their /24 prefix and external IP addresses to their /16 prefix.

Truncation of prefixes is considered an implementation of k-anonymity [11]. Extensions to k-anonymity like l-diversity and t-closeness require an analysis of the data distribution before anonymization. These extensions are therefore not immediately implementable in practice.

**Timestamps** Several papers discuss host fingerprinting based on ICMP Timestamp Requests/Replies and the TCP Timestamp option, as these contain a timestamp that originates from the host. However, neither DNS or NetFlow logs contain a host originated timestamp, as only the timestamps from the NetFlow/DNS log capture devices are logged. This timestamp can be used in an injection attack to identify a host in a traffic trace with pseudonymized (permuted) IP addresses [21].

In the case where only one subscriber in a truncated IP prefix is actively generating traffic at a specific period of time (e.g., during nighttime), this subscriber does not benefit from IP address truncation. To preserve anonymity, the precision of the timestamp could be reduced to for example an hour or a minute. These approaches are typically infeasible for research, as the order of events is not preserved. An approach not described by the authors of [12] or other known sources is to simply remove AM/PM information from the timestamp. This approach has the advantage compared to traditional precision reduction that the order of all interrelated NetFlow/DNS events that do not cross the AM/PM time boundary is preserved.

**IP protocol** The IP header protocol field is not considered privacy sensitive by any known papers, the authors of [14] even omit the discussion of the field entirely. RFC 6235 suggests using the binning technique such that 4 bins are used: TCP, UDP, ICMP and "all other protocols", an approach which seems suitable for cyber-security research as well.

**ICMP type/code** ICMP messages and their payload are widely used for OS fingerprinting by tools such as Nmap. The methods typically involve differentiating using TTL or some other IP field, however a specific method creates ICMP requests using illegal combinations of type and code values, and the ICMP response code can then in some cases reveal the OS family [26]. To anonymize this, the code field could be omitted from the logs. However, doing so comes with a significant drawback, as it will obviously also hide any malware using the technique for OS detection.

Note that when using NetFlow logs from an ISP, the OS family revealed will typically not reveal the end users' operating system. Instead, it will reveal either the OS family of the subscribers' modem/router/firewall or the CGNAT device deployed by the ISP. Therefore, the reasonable compromise for cyber-security research seems to be not to anonymize this field.

**Ports** The authors of [22] conclude that anonymizing ports or IP addresses, as opposed to anonymizing other individual fields, have the biggest impact on the utility of the data. However, the risk and the risk/utility trade-off is not discussed in the paper. Not much research was found that quantifies the risk of host fingerprinting based on port numbers when IP addresses are truncated in practice. The authors of [21] provide a short note describing that injected flow patterns are no longer recognizable under certain anonymization policies. However, they do not describe a systematic approach or conclusion for this. This is likely caused by the fact that much attention has already been given to properly randomizing TCP port numbers to avoid Denial-Of-Service and Man-In-The-Middle attacks [27]. The authors of [14] conclude that the port number should not be anonymized as it will have a big impact on the usefulness of a network capture.

**TCP flags** TCP flags can be used for OS fingerprinting using a technique similar to the one described for fingerprinting using ICMP type and codes: Specific flags in a request can trigger an OS-specific flag combination in the response. Analyzing TCP flags is key in detecting malware employing DDoS SYN attacks and other attack types.

As with the ICMP type and code, the OS family revealed by TCP flags will typically not reveal the type of CGNAT device deployed by the ISP. Therefore, the reasonable compromise seems to be not to anonymize this field.

**NAT** Most ISPs implement CGNAT for at least a subset of their subscribers, so that one IP address contains traffic from more than one subscriber. Many port allocation schemes exist, and it is beyond the scope of this paper to describe all. However, from an anonymization perspective, two different consequences of introducing CGNAT can be relevant: *decreasing the truncation length of the IP address* and *increasing the truncation length of the port*.

In a CGNAT scheme where a single RFC6598 IP address is shared by for example 32 subscribers (5 bits) by random port assignment, the PA IP address truncation length could be reduced from a /24 (256 addresses,8 bits) to a /29 (4 addresses,32+5-8=29 prefix) to preserve utility as the expected amount of hosts grouped will then be the same.

In a scheme where the port allocation is not random, but based on a range of ports being reserved for a particular host, or where initially randomly assigned ports are heavily reused for the same subscriber, the port information must be truncated using the same methodology as the IP addresses. For example, if a port block of 64 ports (6 bits) are reserved for each user, and anonymization equivalent to a /24 IP prefix (256 addresses, 8 bits) is desired, the port number must be reduced to 16-6-8=2 bits. However, the PA IP address truncation can then be reduced to a 32+16-6-8=34 prefix, effectively

**Listing 2.1:** NetFlow anonymization policy.

```
1  def anontimestamp(timestamp t)
2      if t.hour >= 12:
3          t.hour = t.hour-12
4      return t
5
6  def anonipport(int32 ip, int16 port)
7      if ip in listOfSubscriberAssignedPrefixes:
8          ip = ip & 0xFFFFFF00
9      else if ip in listOfCGNatPrefixes:
10         port = port & 0xC000
11     else if ip in listOfInfrastructPrefixes:
12         // do nothing
13     else: //external
14         ip = ip & 0xFFFF0000
15     return ip,port
16
17 starttime = anontimestamp(starttime)
18 endtime = anontimestamp(endtime)
19 srcip,srcport = anonipport(srcip,srcport)
20 dstip,dstport = anonipport(dstip,dstport)
21
22 if protocol != (ICMP or TCP or UDP):
23  protocol = 0
```

making the anonymization of the PA IP address unneeded.

### 2.5.3 Pseudo-code: a NetFlow anonymization policy

The pseudo-code listed in Listing 2.1 implements the anonymization policy summarized in Table 2.3 on page 15 assuming sampling by the NetFlow emitter. Lines 2-3 remove AM/PM information, lines 8 and 14 truncate IP addresses to /8 and /16 prefixes, line 10 truncates the port number to a /2 prefix for customers with NAT (assuming 64 port range based CGNAT). It is noteworthy that the implementation can be made with basic operations. This allows a high level of performance, which is required for ISP deployments. Searching for an IP address in a list of prefixes (lines 7, 9 and 11) should also be implemented effectively. This is considered trivial, assuming a small amount of non-overlapping prefixes is used, and therefore it is omitted for readability. Finally, lines 22-23 implement binning of protocol information into 4 different bins.

| Field | Anonymization tech. | Specifics |
|---|---|---|
| Timestamp | Reverse truncation | As NetFlow |
| Client IP address | As Netflow | As NetFlow |
| Client TCP/UDP port | As Netflow | As NetFlow |
| DNS header | None | - |
| DNS response TTL | Binning | 5 predefined bins |

**Table 2.4:** *Content independent DNS anonymization policy*

## 2.6 DNS anonymization policy

Similar to the previous section but focusing on DNS rather than NetFlow, this section will provide a suggestion for a DNS anonymization policy, and provide the pseudo-code implementing such a policy.

### 2.6.1 Choice of features

The choice of features for DNS based cyber-security research is very diverse [28]. Moreover, whereas NetFlow is only a format for logging passively collected flow properties, DNS is a service used (and potentially attacked) by subscribers. This calls for a more full-featured approach to logging than focusing on a few specific fields. One example is that a protocol violation could be made intentionally by a client to attack the DNS service, and this can only be discovered if the specific field containing the violation is logged.

A DNS packet typically consists of a header section and a query section, and response packets also include one or more sections containing the answer to the query. The answer sections can contain a number of different resource records (RRs). The content, typically an IP address or domain name, and the interpretation of the query and answer RRs depend on flags in the header as well as on which specific type of information is queried.

The increased field diversity and inter-dependency makes DNS log anonymization more complicated than NetFlow log anonymization: Table 2.4 lists a number of fields for which an anonymization policy can be directly described, whereas Table 2.5 on the next page lists the type dependent anonymization techniques. The lines of the tables are elaborated in the following.

### 2.6.2 Feature anonymization details

**Timestamp, Client IP address and TCP/UDP port** For these fields, the anonymization policy also used for the similar fields NetFlow packets is chosen. To the best of our knowledge, no research is made that indicates that

| **Opcode** | **Class** | **Type** | **Domain anon. technique** |
|---|---|---|---|
| Not Query | - | - | None |
| Query | Not IN | - | None |
| Query | IN | Common types | Minimization |
| Query | IN | Uncommon types | None |

**Table 2.5:** *Content based DNS anonymization policy*

DNS and NetFlow logs should be subject to different anonymization requirements relating to these fields.

**DNS header** The DNS header consists of a number of identifiers, response codes and flags. Many of these are needed to parse the non-header components, and no fields contain directly personal identifying information. The randomness of the Message ID has, like the randomness of the TCP/UDP source port number, been subject to scrutiny to prevent Man-in-the-Middle attacks, so this field is expected to be properly randomized to not represent a privacy risk.

**TTL** The TTL value found in the answer sections of a DNS packet could, together with the timestamp, be used to determine that two clients requested the same RR, as these would have the same TTL. Nevertheless, it is unknown whether this can be practically exploited for subscriber fingerprinting. The bins $[0, 1), [1, 100), [100, 300), [300, 900), [900, \infty)$ are found to be relevant for cyber-security research [29], and therefore this technique is chosen.

**Uncommon opcodes, classes and types** Request and response messages containing an Opcode of any other value than "query" (such as "status" or "update"), query messages of any other class than IN (such as Chaos and Hesiod) and IN class query messages of any other type than the 15 most common types (see below) are represented by the first two lines and the last lines in Table 2.5. A smaller data sample collected at Telenor Denmark suggests that traffic in these three categories represent misconfigured equipment, malformed packets and spurious requests with an empty response. This type of traffic does not seem to be the result of human Internet usage behavior and is therefore not likely to represent any privacy risk. However, as mentioned initially, the traffic may represent an attack initiated by malware, and therefore the data is still relevant to retain.

**Common types** On Telenor Denmark's resolvers, the 15 most common query types in the IN class are A, AAAA, A6, CNAME, PTR, MX, TXT, SRV, NAPTR, NS, SOA, DS, RRSIG, DNSKEY and NSEC3. Resource records of these types typically consist of a QNAME component (the name queried) and an RDATA component (the response to the queried name). Either of these components can contain an IP address, a domain name or a string of text containing either of the two, such as SRV or TXT records. It is clear that any anonymization policy applied to an RR must be applied to both the QNAME and RDATA components, as one component can typically be derived from the other by issuing a new DNS request, thus breaking the anonymization.

**Domain name** As described in Section 2.4 on page 12, the queried domain name can be used to fingerprint subscribers, and the only known anonymization strategy is data minimization. The authors of [16] suggest two minimization strategies: Omitting the most or least popular hostnames. From a cyber-security research perspective, omitting the least popular hostnames severely decreases data utility. As an example, botnets based on Domain Generation Algorithms (DGAs) are likely to be rendered undetectable.

The authors of [16] argue that omitting the most popular hostnames would have only a limited effect on fingerprinting risk, though the effect increases when the 500-1000 most popular hostnames are omitted. However, it is questionable if this result applies on an ISP network in 2020. The paper analyses data from approximately 3600 users on a campus network in 2010, where removing the 1000 most popular hostnames is equivalent to removing 51,2% of all queries. Nevertheless, on Telenor Denmark's network having around 1.7 million subscribers in 2020, the same percentage of queries relates only to 15 domains and associated subdomains[1]. This suggests that significant data minimization (removing $> 50\%$) could decrease the fingerprinting risk. If the omitted domain names represent domains that are less interesting from a cyber-security research perspective, the utility of the data can be preserved while decreasing the fingerprinting risk.

OS fingerprinting can be avoided using the same technique, by simply adding known OS-specific domain names and IP addresses to the list of omitted domains. This includes for example captive portal detection mechanisms (such as resolving "connectivitycheck.gstatic.com"), proxy detection (resolving the "wpad" hostname), etc.

### 2.6.3 Pseudo-code: a DNS anonymization policy

The pseudo-code listed in Listing 2.2 on the facing page implements the anonymization policy summarized in Tables 2.5 on the previous page and 2.4 on page 20. The anonymization functions for timestamps and client IP address/port (lines 1 and 2) can be found in Listing 2.1 on page 19. Line

[1] Specifically: apple.com, facebook.com, akadns.net, google.com, googleapis.com, snapchat.com, akamaiedge.net, fbcdn.net, icloud.com, apple-dns.net, doubleclick.net, gstatic.com, netflix.com, microsoft.com and googlevideo.com.

**Listing 2.2:** DNS anonymization policy.

```
 1  timestamp = anontimestamp(timestamp)
 2  ip,port = anonipport(ip,port)
 3
 4  if header.opcode == Query:
 5   if query.class==IN and query.type in commonTypes:
 6     if any in commonDomainList in query.name:
 7      query.name = ""
 8
 9   foreach rr in answerSectionsOfPayload:
10    if rr.class==IN and rr.type in commonTypes:
11     rr.ttl  = integerBinning(listOfIntervals)
12     if any in commonDomainList in rr.name:
13      rr.name = ""
14      rr.data = ""
15     if any in commonDomainList in rr.data:
16      rr.name = ""
17      rr.data = ""
```

11 represents the binning of the TTL value, but the implementation of the function itself is left out for brevity. Lines 5-7 clear the queried domain name if it matches or is a sub-domain of the domain names listed in commonDomainList. Lines 12-17 perform the same operation on the Answer RRs, which includes searching for the domain name in both the question (rr.name) and response (rr.data) part of the RR. For brevity, the Answer payload section is considered to also include the Additional and Authoritative sections.

The DNS anonymization pseudo-code is clearly more computationally heavy than the NetFlow anonymization pseudo-code due to the use of string operations. This is to some extent mitigated by the list of common domains being short.

## 2.7 Conclusion

It has previously been unclear what traffic data an ISP has available for cybersecurity research, and under which legal conditions it can be used. This paper attempts to address this by presenting relevant legislation and data sources, and by presenting an anonymization policy for the relevant data.

The EU ePrivacy Directive puts strict requirements on which data can be used by ISPs. Only data that is already used for the purpose of transmission can be used for other purposes, and then only when anonymized. If use of other data and/or use of data in a non-anonymized form is desired, an explicit consent from the subscriber is required. We present the relevant data sources available to a typical ISP, using Telenor Denmark as example, and argue that *DNS and NetFlow* data are identified as relevant to cybersecurity research and as technically and legally available data sources under the condition that the data is anonymized before further processing. We elaborate by proposing anonymization policies (in the form of pseudo-code) for DNS and NetFlow log data.

The proposed anonymization policies make use of various techniques for generalization, such as truncation of IP addresses, precision degradation of timestamps, data minimization on collected DNS logs etc. as mandated by legislation and suggested and inferred by best practices and related work. The pseudo-code implements the anonymization in a computationally inexpensive way such that application at ISP-scale traffic rates is possible. The anonymization policies and related pseudo-code are considered the primary contribution of this paper, giving researchers and developers a concrete and technically focused starting point when creating solutions targeted for deployment in ISPs.

CHAPTER 3

# State of the art

This chapter contains a survey of the state of the art on DNS and NetFlow based malware detection in both commercial offerings and academic papers. This serves as an general introduction to existing research and current product capabilities, but also as a means to identify unexplored topics for the thesis.

## 3.1 Commercial

Some of the commercially available products treat DNS as a high-value source of information, some just threat it like any other user application. However, each product typically has one primary focus, and to identify the state of the art within NetFlow and DNS analysis, it is therefore only interesting to identify market leaders in these two areas and describe their products capabilities. This will be the topic of the next two sections.

### 3.1.1 NetFlow

Many commercially available solutions for detecting and blocking malicious network traffic use a mix of different methods to do so: Traditional firewalls only rely on IP addresses and ports (OSI layer 3-4) information, whereas Next-Generation firewalls allow or block specific (OSI layer 5-7) applications. Intrusion Detection and Prevention Systems (IDPS) do not necessarily have application awareness, but rely on signatures or fingerprints for detection of known threats. These solutions are typically deployed in-line in the traffic stream, so that all packet layers are available for analysis, and so that any malicious traffic can be blocked. Therefore, these products rarely focus on ingesting NetFlow data, and are therefore of less interest in to this thesis.

Conversely, Network Behaviour Analysis (NBA) or Network Behaviour Anomaly Detection (NBAD) products traditionally focus on detecting anomalies by measuring volumes from NetFlow data, and comparing those to expected threshold values, to detect unknown threats. NBA vendors have lately started to employ machine learning techniques rather than simple preset thresholds, and some also look at the entire network packet. Therefore, this product category is interesting to elaborate further.

An NBA vendor overview is available from general, open sources like Wikipedia [30], but more detailed vendor/product analyses are provided by companies like Gartner [31]. To find the market leader(s) relevant to this thesis, the following criteria are used to shortlist the available products:

- Products must focus on detecting malicious activities, rather than focus on network device capacity/bandwidth and performance monitoring (an NPMD system like Paessler PRTG or SolarWinds).
- Products must not exclusively focus on DDoS protection, thus excluding for example NetScout (formerly Arbor Networks).
- Products must focus on behavioural analysis of network traffic (an NBA system), rather than signature matching of network traffic (an IDPS system like Snort and Suricata)
- Products must clearly state support for NetFlow input. This excludes many popular NBA vendors, for example Darktrace, Vectra and Corelight (including Zeek/Bro).
- Products must focus on NetFlow rather than focus on parsing of logs in general (such as a SIEM system like SolarWinds SEM, IBM QRadar and Splunk).
- Products must use NetFlow as a primary basis for behavioural analysis. This excludes ExtraHop Reveal(x), Plixer Scrutinizer, LogRhythm Netmon, Palo Alto Cortex XDR and HPE-Aruba Introspect, as these primarily use NetFlow for simple or secondary/supporting purposes.

Products satisfying these criteria are: Cisco StealthWatch [32], McAfee Network Threat Behaviour Analysis [33], ManageEngine NetFlow Analyzer [34], Bitdefender NTSA (previously known as Redsocks MTD) [35], Riverbed Flowtraq [36], Solana Networks SmartFlow [37] and FlowMon Anomaly Detection System [38]. The three first products provide a good use case overview. The remaining four provide only outdated or no real use case overview, making it hard to asses the products based on publicly available material. However, as a curiosity, the capabilities of the company Redsocks (later Bitdefender) may be described indirectly by the state of the art of the academic research in section 3.2 on page 30, as their former product manager has also been active in the academic community [39] [40].

The most thorough documentation of methods and capabilities is provided by Cisco Stealthwatch [41] [42], and this is also the only vendor that in any detail describes how machine learning is used in the product [43],

| Classification | Description/example | Typical ground truth base |
|---|---|---|
| Bad endpoint | Flows from/toward bogon IPs (for example IPs not registered at RIPE etc.) or IP/ports determined to be malicious or suspicious | "Threat intelligence" or "reputation list" supplied by the vendor or a third party. |
| Policy violation | Flows detected that violate company policies, or attempts to contact hosts or ports that are known not to be assigned inside the company | Automated import of rules from company firewall |
| Protocol deviation | Flows that violate or deviate from protocol rules for example TCP flows starting with an ACK, illegal TOS field value or suspicious ICMP type/code | Rules predefined in software |
| Excessive traffic amount | Flows with an abnormal amount of bytes or packets (individually or in total), or an abnormal amount of flows | Manual or automatically learned thresholds based on host groups and statistical methods |
| Flow length | Flows that are very long (in time) that may or may not contain actual traffic | Manual or automatically learned thresholds based on host groups |
| Systematic flow creation | A collection of flows from/toward a host during shorter or longer periods of time that indicate for example a port scan, brute force password guess, DDoS attack or beacon | Rules predefined in software, statistical methods |
| Combination | Combinations of methods listed above, for example a scan of many hosts and communication with one of them may indicate a worm in action | |
| Reputation calculation | The product continously maintains a reputation score for each endpoint based on observed events | The other categories mentioned in this table |

**Table 3.1:** *Netflow based detection categories*

whereas the other products only mention using machine learning as part of the marketing material.

Based on the documentation from various vendors, malicious behaviour is detected using several methods as summarized in table 3.1. Notice that this analysis is completely based on publicly available product manuals and/or promotional material, not actual product usage or technical implementation documentation.

### 3.1.2 DNS

In order to find the state-of-the art in DNS analysis tools, three different tool categories must be surveyed: The DNS resolver service itself (either on-premises or cloud based), SIEM tools operating on DNS logs originating from the DNS resolver, and IDS/NBA tools operating on the DNS traffic towards/from the DNS resolver. Unlike NetFlow NBA tools, no publicly available overview or comparison of DNS analytics tools that focus solely on detecting malicious activity could be found. To find the market leader(s) relevant to this thesis, the following criteria are used to shortlist the available products:

- Products that focus on protecting the DNS resolver itself from being the victim of attacks (like DDoS, software exploits, cache poisoning etc.) are considered out of scope, whereas detecting attacks that use DNS resolvers as the weapon (for example reflection attacks) is in scope.
- Products that focus only on authoritative servers are out of scope.
- Products must put a strong focus on detection of malicious activity. This excludes for example Bind, Djbdns and Unbound.
- Products must implement and focus on more features than domain blocking based on threat intelligence or content category feeds. This excludes the vast majority of cloud DNS providers like ns1, Quad9, Cloudflare, Google, Neustar/Verisign, Watchguard DNSWatch, Constellix, Verigio Proxywall, ThreatSTOP DnsDefence, Microsoft DNS analytics and EonScope DNSSense, but of course also threat intelligence providers like Deteque/Spamhaus, Surbl and malwaredomainlist.com.
- Products must focus on DNS as the primary area, rather than be a smaller feature as part of another product (for example an IDS or SIEM product, or LogRythm NetMon)

The remaining products fall roughly into three different categories: Some products like those available from NexusGuard, CSIS, EfficientIP, BlueCat and Netsurion document some advanced features, but not at any depth or breadth to be considered market leading. A larger feature set documented

| Classification | Description/example | Typical ground truth base |
|---|---|---|
| Bad endpoint | Request/response where the domain or IP is known to be malicious | Threat Intelligence (including RPZ) feeds |
| Protocol deviation | Requests that deviate from the protocol rules, for example setting invalid opcodes or never completing a TCP transaction | Predefined rules in software |
| Suspicious endpoint | Requests for young/unknown/dyndns domains | Whois, Pagerank, geoip, TTL, threat Intelligence Feed) |
| Tunnelling | Requests encode a message in requested domain name, answer encodes message in txt records. Some well-known antivirus companies even use this intentionally [44] | Lexical analysis |
| Traffic amount | Abnormally amount of requests for the same domain or related subdomains | Manual or automatically learned thresholds, statistical methods, lexical analysis |
| Enumeration attempts | Requests for typical host names in preparation for an attack on the company behind the related domain | |
| Fast flux / Domain flux | Requests for auto-generated domain names / IPs, some resolving to CnC infrastructure | Threat intelligence, statistical methods and lexical analysis |
| Reputation calculation | The product continuously maintains a reputation score for each endpoint based on observed events | The other categories mentioned in this table |

**Table 3.2:** *DNS based detection categories*

to a larger extent can be found at PowerDNS [45], PaloAlto DNS Security Service [46], Akamai ThreatAvert (formerly Nominum) [47], Plixer FlowPro Defender [44], AlphaSOC Analytics Engine [48] and F5 DNS Security [49].

The market leaders, from both a feature and documentation perspective seems to be Cisco Umbrella (formerly OpenDNS) [50] that publish for example how DGA algorithms are investigated [51], Infoblox Advanced DNS protection [52] that publish for example entire books on DNS security [53] and HP Arcsight DNS Malware Analytics (formerly Damballa) [54] that publish for example detailed tech reports about scalability [55].

As with the Netflow product survey, only publicly available product manuals and/or promotional material is used. Commercial products for detecting malicious behaviour based on DNS traffic use some or all of the methods summarized in table 3.2.

### 3.1.3 Discussion

It is clear that commercial solutions exist with great potential for detecting malicious traffic based on both NetFlow and DNS. However, some caveats exist: None of the products described above document the ability to anonymize or operate on pre-anonymized NetFlow data. In particular, this presents the problem of being unable to distinguish between two different subscribers using CGNAT, and the ability to correlate the cleartext IP addresses from threat intelligence feeds (including black/white lists) with the anonymized IP addresses when a subscribers IP is listed in threat intelligence or otherwise considered a bad host.

Interestingly, none of the products in the survey appear able to perform behavioural analysis on the combination of DNS and NetFlow features. Analysis is performed on DNS data to provide a reputation score, which is then added to the with the reputation score derived from NetFlow data. And NetFlow information is augmented with information from the relevant PTR record to identify the related domain. However, no public documentation could be found that shows that the products combine DNS and NetFlow data such as described in Chapter 4 on page 35 in this thesis.

## 3.2 Academic

Many different surveys and state of knowledge articles are available in the academic literature for both NetFlow and DNS based detection systems. This thesis will for brevity reference these rather than present a new survey. A good and contemporary introduction to the area of intrusion detection systems in general and an overview of the various techniques in use (Neural networks, univariate models, finite state machines, Hidden Markov Model, k-means etc.) and typically used training data sets is provided by Khraisat et al. [56].

### 3.2.1 NetFlow

Detecting malicious activity based on NetFlow has been the topic of extensive research, and several, very different surveys of the state of knowledge are available. A larger part focus specifically on botnet CnC infrastructure and associated application layer based detection, however a few focus on network/transport layer detection. Garcia et al. outline a wide range of botnet detection methods along with a list of desired properties of the detection methods and the evaluation thereof [57]. Next, an in-depth survey of 10 papers is performed, highlighting the assumed bot, botnet, temporal, and protocol behaviour in the various papers, providing an equivalent to Table 3.1 on page 27. Feature selection is often emphasized as an important parameter and Ferreira et al. provide an overview of the IPFIX feature selection by 71 papers [25]. Some papers address how to protect the DNS server against volume based attacks as detected solely through NetFlow information, but as previously argued, protection of the DNS server itself is out of scope of this survey.

### 3.2.2 DNS

Detecting malicious activity based on DNS has, like NetFlow, also been the topic of extensive research, and several different surveys of the state of knowledge are also available. Zhauniarovich et al. cover topics like how to enrich DNS data with for example GeoIP, different ground truth bases and algorithm performance evaluation metrics [58]. Alieyan et al. focus mostly on giving a more in-depth summary of the various detection methods [59]. The survey by Torabi et al. does not focus on papers but on specific systems created for passively detecting various DNS related anomalies as those referenced in 3.2 on page 29 [60]. A key finding is that most systems are not near real-time, mostly due to the application of supervised machine learning, and the paper therefore presents a new system with near real-time capabilities. Most recently, the survey by Singh et al. focus on using DNS data to detect botnets [28].

### 3.2.3 Combining DNS and Netflow

The surveys mentioned above reveal that detection methods typically focus either on the NetFlow perspective or the DNS perspective. Few include both Netflow and DNS based approaches, like Stevanovic et al., that presents the prevalent methods and strategies for detecting botnets using machine learning algorithms, and summarizes the methods and strategies used by 20 papers [61]. The survey by Lashkari et al. proposes a taxonomy framework and lists papers based on the used traffic features (NetFlow as well as DNS related), dataset features and other properties [62]. Most surveys use "flow anomaly based", "signature based" and "DNS based" as the primary classifiers. This makes sense from the perspective that papers typically have a relative narrow focus, on a specific botnet, some specific machine learning algorithms or similar.

Abnormal amounts of DNS traffic is detected through analysis of NetFlow packets by Huistra [63]. Similarly Grill et al. detects DGA based malware simply by looking at the ratio of flows towards the DNS resolver vs. flows towards any other host [64]. As only general DNS protocol knowledge is used, and no layer 7 information, these approaches should still be considered NetFlow-only approaches.

However, some papers do use both NetFlow and application-layer DNS features:

- Hananto et al. derive source IP address entropy from NetFlow logs and use DNS logs to measure the ration of NXDOMAIN responses [65]. It is, however, unclear if these methods are used independently, or if DNS and NetFlow data are correlated before analysis.

- A supervised machine learning model on TLS, DNS and HTTP features to detect encrypted malware in TLS flows is used by Anderson [66]. The main focus is on determining whether a TLS flow is malicious or not, therefore the paper also assumes knowledge about the TLS and HTTP layers not available in regular NetFlow packets.

- Fuzzy pattern recognition is applied by Wang et al. in a two-stage approach to network flows and DNS data, arguing that any botnet CnC activity will start with a DNS phase (finding the IP of the CnC infrastructure) followed by a network flow phase (exchanging information with the CnC infrastructure) [67]. The pattern recognition is then applied on the flow inter-arrival time in either of these phases. The specific domain queried is not used is not used as a part of the analysis, however, only the information about whether the query was successful or not.

- The thesis by Janbeglou introduces the concept of unnamed traffic [68], but focuses on traffic classification rather than malware detection.

- Hageman et al. complements the work by Janbeglou and attempts to identify the origin of all unnamed traffic [69]. It is concluded that unnamed traffic is ubiquitous, and that it cannot be assumed to be malicious.

It is therefore an interesting research gap that only Janbeglou and Hageman (and possibly Hananto) seem to be the only authors that consider combining DNS and NetFlow data to the extent described in Chapter 4 on page 35.

## 3.3 Summary

Market leading, commercially available products use a large variety of techniques to identify malicious activities in a broad threat landscape, including some advanced behavioural techniques involving machine learning. Academic research papers typically focus on comparing, optimizing or improving specific detection techniques, but do not provide tools that cover the broad threat landscape. Given the strong focus on machine learning techniques in the commercial products, it seems inevitable that academic research will be absorbed into the market leading commercially available products over time. From the operational perspective of an ISP, a commercial solution should therefore be chosen to detect malicious traffic.

However, the survey of both the commercial products and the academic research has shown that a behavioural analysis of the combination of NetFlow and DNS features is a relatively unexplored area that may be interesting from both an operational and an academic perspective.

CHAPTER 4

# NetFlow and DNS data

As outlined in the previous chapter, combining features from DNS and NetFlow data is a relatively unexplored approach. Combining DNS data and NetFlow records could at first glance be seen as a trivial task. A task that can be completed by comparing timestamps, client IPs, and the server IP address found in the NetFlow record to the IP address found in the A record in the DNS response packet. However, various mechanisms, such as the use of DNS prefetching and response caching in client applications, sampling of flows in routers, the topological location of the data observation point etc. makes the correlation of DNS and NetFlow data much more complicated, thus affecting the trustworthiness of the conclusions that can be made on the data. As this is a reoccurring topic in this thesis, the topic deserves an introduction from a more general perspective than the use-case specific descriptions in the individual papers, and this will therefore be the topic of this chapter.

One approach to systematically describe the challenges in correlating NetFlow and DNS data is to first consider the set of all IP flows within and to/from a specific network under consideration, for example the IP network owned by an ISP or a company. These IP flows can be described in terms of whether they are observed or not and in terms of whether the IP flows carry DNS data or not, which is elaborated in Section 4.1. The purpose of this classification is to clearly define which data is considered for correlation. As elaborated in Section 4.2, this makes it easier to assess which conclusions can and cannot be made from the correlated data.

A flow can be represented in several NetFlow records, as NetFlow records are emitted periodically during the lifetime of a flow. The process of merging NetFlow records into flows is described in more details in Paper B. For the purpose of the more conceptual discussion in this chapter, only the flows and not the individual records are interesting.

## 4.1 Flow type and observability

The IP flows within a network can be considered members of different subsets as illustrated in the model in Figure 4.1. An key property highlighted by the model is that only a subset of the flows are observable in either DNS logs or NetFlow records. The main sets are as follows :

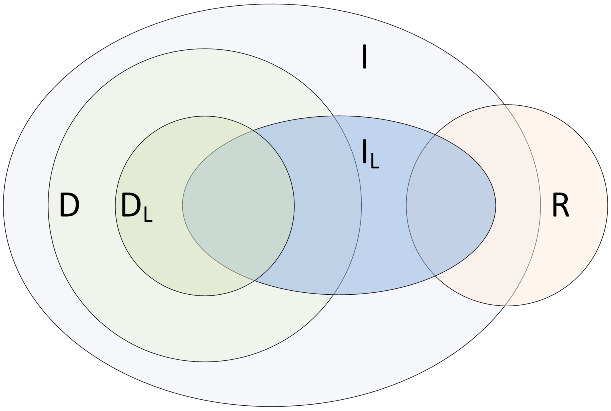

**Fig. 4.1:** *Classification of IP flows*

- $I$ is the set of all IP flows with source and/or destination at some endpoint inside the network under consideration.
- $I_L$ is the subset of $I$ that are logged in NetFlow records. If Netflow logs are not collected, $I_L = \emptyset$.
- $D$ contains the subset of the IP flows that carry DNS traffic.
- $D_L$ is the subset of $D$ that is captured in DNS application layer logs. If no such logs are captured, $D_L = \emptyset$.
- $R$ contains the flows to/from IP addresses that can be found in a preceding DNS query (elaborated further in Section 4.2).

The sets $D$ and $D_L$ could be considered to only include either traffic between clients and resolvers, or traffic between resolvers and authoritatives servers, depending on what is relevant for a particular analysis. The model reveals the existence of a number of subsets that must be considered as well, notably:

- $I \setminus I_L$ (all flows in $I$ that are not in $I_L$) represents ip flows that are not represented in NetFlow records, for example due to NetFlow sampling, or because the flow is not routed through any equipment that emits NetFlow records.
- $D \setminus D_L$ contains DNS traffic not being logged at application level. This could represent application layer logging based on sampling, or it could represent traffic towards third-party resolvers outside of the network, for which no application layer data is available.

- $I \setminus D$ contains flows that are not DNS traffic. It is worth noting that this set includes non-DNS traffic towards DNS servers.

Using this model, both the network itself as well as the collected data can be characterized. As an example, clients in a company network can be prevented from using anything but the company's own resolvers, on which all entries are logged at the application layer. In this case, $D = D_L$. Furthermore, the company may employ non-sampled NetFlow collection on all of their routers, such that $I = I_L$. This obviously reduces the number of subsets, and therefore also reduces the number of special cases that must be considered before concluding anything on the data.

The data and network used for this thesis are not that simple, and are instead characterized as follows. In this thesis, only DNS data between clients and resolvers are considered, thus defining $D$. Application layer DNS data, $D_L$, is only available from Telenor's own resolvers, not from 3rd party resolvers, and therefore $D \neq D_L$. As *all* DNS traffic towards Telenor's resolvers is available in application layer logs, the set $D \setminus D_L$ therefore only represents traffic towards 3rd party DNS resolvers. NetFlow records are only available from Telenor DKs border routers in sampled form, so only some of the traffic entering or exiting the network is logged, and therefore $I \neq I_L$. DNS traffic from customers to Telenor DK's resolvers are not represented in NetFlow records, and therefore $D_L \cap I_L = \emptyset$.

As the two examples above illustrate, different data sets can have very different properties. These properties can be important to clarify before using a data set to ensure that the results obtained by using the data are valid.

As a useful side effect, the model can make it easier to clarify terminology. In this thesis, DNS data refers to the application layer log data from the flows in $D_L$, and NetFlow data refers to the records derived from the set $I_L$.

## 4.2 Correlation of DNS and NetFlow data

The description of the model presented Figure 4.1 is not complete, as a key component is missing: The relation between the IP address found in the DNS response packet and the flow(s) created towards that IP address. For this purpose, the set $R$ is defined to contain the flows to/from IPs found in a preceding DNS query. In other words, $R$ contains flows where the source or destination IP of the flow is directly or indirectly referred to in the DNS application layer payload of a flow in $D_L$, and where the other address of the flow is the same as the source address of the DNS flow. By definition it is true that if $D_L = \emptyset$ (no application layer logs are available) then $R = \emptyset$. Extending with $R$ adds several notable intersections and complements, including:

- $R \cap I$ contains the flows that are created in order to connect to the IP

| For each... | map it to... | within a... |
|---|---|---|
| DNS response | next succeeding NetFlow flow | fixed interval |
| | | TTL interval |
| | all succeeding NetFlow flows | fixed interval |
| | | TTL interval |
| NetFlow flow | first preceeding DNS response | fixed interval |
| | | TTL interval |
| | all preceeding DNS responses | fixed interval |
| | | TTL interval |

**Table 4.1:** *Overview of the different methods to correlate DNS and NetFlow data.*

address returned in a preceding DNS response. These flows can be associated with a DNS query, and are denoted named flows.

- $R \cap I_L$ contains the named flows that are observed in NetFlow records.
- $I \setminus R$ contains flows for which no related DNS log entry exists. This is referred to as unnamed traffic. Examples include flows related to applications that do not use the DNS infrastructure to establish connections, flows that are not nameable because the DNS query is found in $D$ but not in $D_L$ (e.g. DNS traffic towards 3rd party resolvers), and flows for which the related DNS query was issued before initiating DNS logging.
- $R \setminus I$ represents potential, but non-existent flows. The purpose of this set is to illustrate that not all DNS response records are used by clients to create new connections. This can be caused for example by DNS responses that contain several, different response records of which only one is used, or by DNS prefetching by browsers.

The attempt of adding the set $R$ to the model reveals a number of caveats that must be considered when correlating DNS and NetFlow data: It cannot be expected that a flow can be found that correlates to all the IP addresses in a DNS record, either because such a flow is never created ($R \setminus I$) or because the flow may not be observed in NetFlow records ($R \cap I \setminus I_L$). Conversely, it cannot be expected that a DNS record can be found that relates to each flow either, both because not all DNS records are observed ($D \neq D_L$), and because not all flows are preceded by DNS requests ($I \neq R$).

The specific method to use for correlating DNS and NetFlow data will therefore necessarily depend on the properties of the collected data and the use case, and two main approaches exist. Either, a number of NetFlow flows are identified that map to each DNS response, or a number of DNS responses are identified that map to each NetFlow flow. As outlined above, these two approaches should be expected to yield different results. For each approach, it should also be considered whether a one-to-one or one-to-many mapping is interesting (in case there is a match at all). Furthermore, the maximum time interval between the DNS response and flow start could either be a fixed time interval or a variable time interval determined by the TTL value in the DNS response. These methods can be listed as in Table 4.1.

The data used for this thesis includes an unsampled DNS log and a sampled NetFlow log. Furthermore, the TTL values are binned for anonymization purposes. Therefore, the 5th method listed in Table 4.1 is the main approach used in this thesis. Other works focusing on named/unnamed flows use the 6th method, thus taking advantage of the TTL value being available [68] [69].

CHAPTER 5

# Paper B

## Using NetFlow to measure the impact of deploying DNS-based blacklists

**Main author:**
Martin Fejrskov
*Technology, IP Network and Core*
*Telenor A/S*
Aalborg, Denmark
mfea@telenor.dk

**Co-authors:**

Jens Myrup Pedersen
*Cyber Security Group*
*Aalborg University*
Copenhagen, Denmark
jens@es.aau.dk

Emmanouil Vasilomanolakis
*Cyber Security Group*
*Aalborg University*
Copenhagen, Denmark
emv@es.aau.dk

**Presented at:**
*EAI International Conference on Security and Privacy in Communication Networks (SecureComm) 2021)*

# Abstract

*To prevent user exposure to a wide range of cyber security threats, organizations and companies often resort to deploying blacklists in DNS resolvers or DNS firewalls. The impact of such a deployment is often measured by comparing the coverage of individual blacklists, by counting the number of blocked DNS requests, or by counting the number of flows redirected to a benign web page that contains a warning to the user. This paper suggests an alternative to this by using NetFlow data to measure the effect of a DNS-based blacklist deployment. Our findings suggest that only 38-40% of blacklisted flows are web traffic. Furthermore, the paper analyzes the flows blacklisted by IP address, and it is shown that the majority of these are potentially benign, such as flows towards a web server hosting both benign and malicious sites. Finally, the flows blacklisted by domain name are categorized as either spam or malware, and it is shown that less than 6% are considered malicious.*

**Keywords:** *Blacklist · DNS · NetFlow · Ipfix · ISP · RBL · Threat Intelligence*

## 5.1 Introduction

Threat Intelligence (TI) in the form of reputation-based blacklists of IP addresses and domain names have been made available by non-profit and commercial organisations for decades [70], and has later been the subject of academic research as well [71]. Improving the accuracy and completeness of the blacklists by the careful selection of entries to maximize the amount of true positives and minimize the amount of false negatives remains a continuous struggle. These metrics describe the blacklist itself, however they do not describe the actual impact of deploying a blacklist in practice. If there is not impact, the time and money spent by the user deploying the blacklist can be considered wasted. Therefore, we argue that the impact is an important metric from a practical perspective.

How to describe and measure the impact will naturally depend on the specific use case in which the blacklist is applied[1]. The most prevalent use cases for blacklists fall in two categories, offering protection to either the originating end of a connection (in antivirus software, in a web browser plugin, in a company firewall, in an Internet Service Providers (ISPs) DNS server, etc.) or the terminating end of a connection (a mail server, at a firewall protecting a web site, etc.). This paper focuses on the impact of deploying blacklists in DNS resolvers at ISPs. Deploying blacklists at ISPs is attractive as it can increase the security posture of all devices that default to use the ISP's DNS resolvers.

[1] The elaborated definition of impact used in this paper is presented in Section 5.4.3.

Informal conversations with blacklist vendors suggest that a common method for assessing the impact is to let the DNS resolver count the number of performed DNS queries that match an entry on a blacklist. Some ISPs and DNS security vendors even refer to this number directly as the number of blocked threats [72, 73]. This is similar to counting the number of emails flagged as phishing by an email server, or counting the number of requests towards a web server originating from an IP address known to be malicious. However a DNS request in itself is only a threat indicator. In order for a user to be at risk, an IP connection towards the malicious host is a minimum precondition, and we therefore consider an IP connection as a stronger threat indicator than a DNS resolution. In this paper, we propose a method based on NetFlow/IPFIX measurements to evaluate the impact of deploying blacklists at an ISP DNS resolver.

Assessing the network-level impact of applying a blacklist at a DNS server will, however, not in itself tell anything about the user-level impact perceived by the user. For instance, blocking a user's connection attempt towards a shared web hosting environment that incidentally also hosts a known spam sender, is likely to be perceived as a nuisance rather than as protection against a threat. On the other hand, connecting to a web server known to solely host malicious payloads represents a high risk to the user. To supplement the measured network-level impact, it is necessary both to identify the cause for the entry to be blacklisted in order to assess the risk level of connecting to the blacklisted entity, and to assess the risk that a connection is in fact made towards the malicious entity.

The contributions of this paper are twofold:

- We show how existing methods for measuring the impact of deploying domain and IP address blacklists in DNS resolvers can be improved by including NetFlow measurements.
- Using the NetFlow method, we quantify the number of malicious and non-malicious flows, and we quantify the number of flows blacklisted by IP address that may be benign.

The paper is organised in 7 sections: Section 5.2 gives an overview of related work. Section 5.3 describes the concept of blacklisted flows and the method for merging DNS, NetFlow and blacklist data to identify blacklisted flows. Section 5.4 describes the 3 data sources used in the paper and the application of the previously described merging method. Section 5.5 categorizes the blacklisted flows by the type of maliciousness and Section 5.6 identifies IP addresses that may contain multiple (and possibly both malicious and benign) endpoints. Section 5.7 combines the results from the previous sections to describe the network-level and user-level impact. Lastly, Section 5.8 summarizes and concludes the paper.

| | | Focus area | | | | | |
|---|---|---|---|---|---|---|---|
| **Author** | **Year** | Impact | Resolver BL | DNS data | NetFlow data | Endpoint | Maliciousness |
| Sheng et al. [77] | 2009 | | ✓ | ✓ | | | ✓ |
| Bermudez et al. [78] | 2012 | | | ✓ | ✓ | | |
| Connery [79] | 2012 | ✓ | ✓ | ✓ | | | |
| Zhang et al. [80] | 2013 | ✓ | | | ✓ | ✓ | ✓ |
| Kührer et al. [81] | 2014 | | ✓ | ✓ | | ✓ | ✓ |
| Foremski et al. [82] | 2014 | | | ✓ | ✓ | | |
| Satoh et al. [83] | 2019 | ✓ | | | | | |
| Spacek et al. [84] | 2019 | ✓ | ✓ | ✓ | | | |
| Wilde et al. [85] | 2019 | ✓ | ✓ | ✓ | | | |
| Li et al. [86] | 2019 | | | | ✓ | ✓ | ✓ |
| Telenor Norway [87] | 2020 | ✓ | ✓ | ✓ | | | |
| Griffioen et al. [88] | 2020 | ✓ | | ✓ | ✓ | ✓ | ✓ |

**Table 5.1:** *Related work and their focus areas*

## 5.2 Related work

As outlined in the introduction, the contribution of this paper is to show that existing measurement methods that *measure the impact* of implementing domain and IP address *blacklisting in DNS resolvers* can be improved by including *NetFlow data* based measurements in addition to *DNS data* based measurements. We use the proposed measurement method together with information about the *type of maliciousness* and knowledge about the *type of endpoint* to identify if the endpoint may host both benign and malicious sites simultaneously. The columns of table 5.1 represent each of the aspects highlighted in the above paragraph, and this section elaborates how related works cover some, but not all, of the aspects.

Many papers such as [58] focus on the creation, quality, accuracy or comparison of blacklists. Bouwman et al. focus on the differences between paid and free lists, and investigate the reasons (price, coverage, false positive rate, etc.) provided by operators/enterprises for choosing specific lists [74]. These topics are considered complementary to this paper, and such efforts will therefore not be the topic of this section. Similarly, papers such as [75, 76] focusing on using blacklists for spam filtering in mail servers are also considered complementary.

### 5.2.1 Network-level impact of blacklisting

Although not focusing on malware and blacklists, the authors of [78] observe that around 50% of DNS responses have an associated flow. This suggests that a flow cannot be assumed to be associated with all *blacklisted* DNS responses either. This forms a motivation for focusing on flows rather than DNS responses.

Zhang et al. measure the impact of applying several (IP address) blacklists on NetFlow records obtained from the routers of a large regional ISP [80]. The paper differentiates between different types of maliciousness and endpoints, and concludes that up to 17% of the traffic measured by volume could be considered tainted. Although this work blacklists NetFlow entries rather than DNS entries, we consider it to be one of the works that are most closely related to our paper.

Sheng et al. evaluate blacklists in browser plugins to protect against phishing websites [77]. This approach represents several advantages to DNS-based filtering, as lists of URLs rather than lists of domain names or IP addresses can be used. The approach is, however, by nature very application and browser specific, thus representing a disadvantage in relation to a DNS-based approach.

Li et al. use telescopes of scanning activities to determine list coverage, thus including some flow level data [86]. Furthermore, the paper uses IP ranges of known CDNs as a source to determine list accuracy. However, the focus is still on assessing the quality of the lists, rather than on the impact of applying them.

Spacek et al. describe many practical considerations in deploying DNS based blacklisting, and elaborates on some of the consequences to the user [84]. These consequences focus on feedback about the blocked site, difficulties in relation to the use of TLS, etc., and does not quantify the impact of the blacklist itself.

Deploying blacklists at an ISP or company DNS server is becoming a common security measure. Public statements such as [87] and [79] with limited descriptions of the impact of such measures exist. Both of these statements measure the impact in terms of visits to a website, to which a user is redirected instead of being blocked. Similarly, DNS firewall/resolver vendors, TI providers, etc. provide use case descriptions focusing on DNS-level measurements only. Furthermore, Wilde et al. examine the blocking behaviour of several publicly available resolvers and conclude that none of them block for security purposes [85]. They also use lists of URLs to quantify to which extent an RPZ enabled DNS resolver would block the list entries. However, no real world traffic is used in the quantification.

Academic papers describe the impact of blacklisting at the router and browser level, and to a certain extent at the DNS level, as outlined above.

However, we are not aware of any work that quantifies the impact at the NetFlow level.

### 5.2.2 User-level impact of blacklisting

Kuhrer et al. categorize both commercial and public blacklists entries to identify if an endpoint is a sinkhole or a parked domain [81]. The purpose of performing the categorization therefore relates more to the validity of a blacklist entry than to the impact experienced by the user. Furthermore, the paper evaluates the ability of blacklists deployed at a DNS server to detect known botnets.

Using DNS and flow information to determine the used application is the topic of [82]. The application of named flows (flows to which a DNS response can be associated) such as HTTP, Roblox and Skype is identified. This classification, however, focuses solely on the application rather than the type of endpoint.

Determining the type of maliciousness is the main focus of [83]. The authors use Word2Vec to group 388 malicious queries into three clusters, each comprising queries with a common cause. The study focus solely on DNS TXT records, which does not extend well to the majority of queries that do not have TXT records.

Some blacklist vendors and tools such as [89] provide the cause for an entry to be listed. This is in many cases directly related to the type of maliciousness.

Griffioen et al. present several aspects related to our paper [88]. Their main emphasis is to compare open source blacklists, including the impact metric. NetFlow information from a Tier 1 provider is used to assess the timeliness of entries on the lists, but is not used to assess the impact of deploying the lists, which is the main topic of our paper. Instead, information from authoritative DNS servers is used to evaluate the impact of deploying the lists, by analyzing how many domain names were pointing to a particular IP address on the day it was marked as malicious. We will extend this by including other aspects, beyond a high domain name to IP address ratio, and by analysing the domain names and IP addresses to identify different scenarios like shared web hosting.

Both [80] and [88] consider blacklisting on the IP level, for example in firewalls. In our paper, the focus is on invoking the blacklisting in a DNS server, thus considering both domain name and IP address based blacklists. Despite this conceptual difference, we consider these the most closely related to our paper.

### 5.2.3 Summary

Although related work exists, the idea of using NetFlow measurements for evaluating a DNS-based blacklist deployment seems to be unexplored, and this will therefore be the topic of Section 5.3-5.4. Categorizing existing blacklist entries by type of maliciousness does not seem to be receiving a lot of academic attention, maybe because the categorization can be available as a supplement to the blacklists. Using knowledge about the type of maliciousness and endpoint to provide a risk based view of the blacklisted flows will be the topic of Section 5.5 on page 55-5.6 on page 57.

## 5.3 Method for identifying blacklisted flows

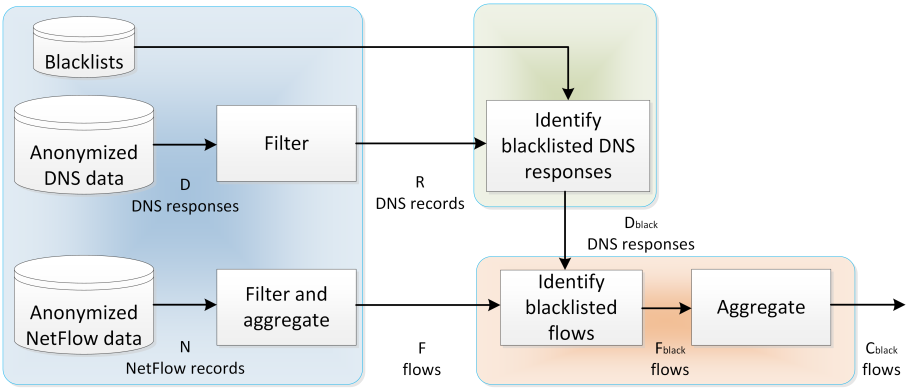

**Fig. 5.1:** *The overall dataflow to identify blacklisted flows.*

The concept of blacklisted flows is central to the flow based measurement method proposed in this paper. The method to identify blacklisted flows requires three data sources and is comprised of several steps, as illustrated in Figure 5.1. The three data sources are NetFlow data, DNS data and blacklists containing domain names and IP addresses. The first steps, relating to the practical collection and pre-processing of the three individual data sources are illustrated in blue in Figure 5.1 and elaborated in Section 5.4. Combining the three data sources involves two additional processing steps, elaborated in the following subsections. First, all blacklisted DNS records are identified (green in Figure 5.1). Then, all flows relating to the blacklisted DNS records are identified (red in Figure 5.1).

### 5.3.1 Blacklisted DNS data

All DNS records associated with a specific DNS response are considered blacklisted if *any* of these conditions are satisfied for any of the records:

- The Qname or Rname of the DNS record matches a blacklisted domain name
- The Rdata of the DNS record matches a blacklisted IP address or a blacklisted IP prefix

The result of this is that $D_{black}$ blacklisted DNS responses are identified.

### 5.3.2 Blacklisted flows

A flow is considered blacklisted by a specific, blacklisted DNS record if *all* of the following conditions are satisfied:

- The DNS record has $rtype = A$
- The DNS record and flow timestamps are less than 30 minutes apart (as elaborated below), $t_{DNS} \leq t_{NetFlow} < t_{DNS} + 30m$
- The blacklisted DNS record is the temporally closest DNS record where the two conditions below are satisfied
- The blacklisted DNS record client IP matches the flow source IP
- The blacklisted DNS record rdata matches the flow destination IP

This yields a number of blacklisted flows, $F_{black}$.

Both the use of temporal correlation and anonymized IP addresses can cause a number of false positives and false negatives that are not immediately quantifiable as no ground truth exists for verification. The limit of 30 minutes is based on an analysis of the time difference between the DNS record and the flow. This analysis suggests that the number of matched DNS records and flows converge towards 0 as a function of the time difference between the records, with few matches with a time difference of more than 15 minutes.

In case a flow matches two different DNS records where the only difference is the TTL, the DNS record with the highest TTL is considered a match.

The merging of NetFlow records into flows is described in Section 5.4.1. However, a NetFlow emitter may view a single, actual flow as two or more flows due to the use of aggressive timeouts for detection of flow end, especially for UDP traffic. Often this is referred to as flow splitting in related works. The effect is illustrated in Figure 5.2, where light blue represents the lifespan of actual flows and dark blue represents packets transmitted in the flow. Green represents the lifespan of flows as perceived and reported by

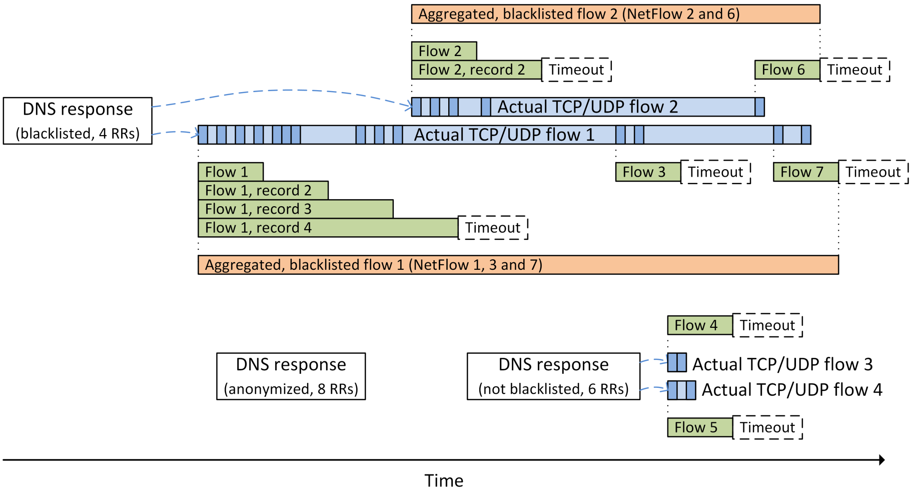

**Fig. 5.2:** *Flow aggregation illustration. In this example, D=3, R=10, $D_{black}$=1, N=11, F=7, $F_{black}$=5, $C_{black}$=2 and $C_{black,DNS}$=1*

the NetFlow emitter in successive flow records. Due to timeouts, the NetFlow emitter perceives the two actual flows as 5 different flows. Therefore, a further aggregation of flows is desirable.

We choose to aggregate all blacklisted flows that are blacklisted by the same DNS record (considered unique by the qname, timestamp and clientip) and that have the same 5-tuple into a single flow, producing $C_{black}$ flows. This aggregated entity is named an aggregated flow to distinguish it from the flow defined by the NetFlow emitter. The aggregated flows are represented in red in Figure 5.2, where two aggregated, blacklisted flows (red) related to 5 different NetFlows (green), related to 2 actual flows (blue), and related to the same (blacklisted) DNS response (white) are depicted. The aggregated flow record has a cumulative bytes/packet count and the flow start timestamp that is the earliest timestamp found in the related flows.

## 5.4 Data sources and processing

This section will provide details on the selection and pre-processing of the three data sources illustrated in blue in Figure 5.1 using data from Telenor Denmark's network (Sections 5.4.1 to 5.4.3). Furthermore, the section will describe the results of performing the steps described in Section 5.3 on the data (Sections 5.4.4 to 5.4.6).

The three data sources are all collected during two separate weeks for the

| Metric | Symbol | Week 1 | Week 2 |
|---|---|---|---|
| Total DNS responses | $D$ | $2,15 \cdot 10^{10}$ | $2,25 \cdot 10^{10}$ |
| Total relevant DNS records | $R$ | $1,85 \cdot 10^{10}$ | $1,88 \cdot 10^{10}$ |
| Blacklisted DNS responses | $D_{black}$ | $6,81 \cdot 10^{6}$ | $4,56 \cdot 10^{6}$ |
| Total NetFlow records | $N$ | $4,63 \cdot 10^{9}$ | $4,60 \cdot 10^{9}$ |
| Total relevant flows | $F$ | $3,92 \cdot 10^{8}$ | $3,94 \cdot 10^{8}$ |
| Blacklisted flows | $F_{black}$ | 185460 | 191923 |
| Blacklisted, aggregated flows | $C_{black}$ | 90796 | 86854 |
| Unique DNS responses in $C_{black}$ | $C_{black,DNS}$ | 78312 | 70134 |
| Blacklisted DNS response ratio | $\frac{D_{black}}{D}$ | 0,000317 | 0,000203 |
| Entries in $C_{black}$ matched by IP | $C_{ip}$ | 68045 | 62683 |
| Entries in $C_{black}$ matched by dom. | $C_{dom}$ | 22842 | 24486 |

**Table 5.2:** *DNS and NetFlow data metrics*

1,5M mobile and 100k broadband subscriptions of Telenor Denmark. Notice that multiple users can use the same subscription, such as a household where all members are the users of a single broadband subscription. The data set for week 1 represent 7 full days from 2020-10-29 to 2020-11-04, and the data set for week 2 represent 7 full days of from 2020-11-26 to 2020-12-02. Table 5.2 lists the key properties for data in these time periods and the following sections will elaborate on these numbers. The following sections will for readability refer to the data from week 1, unless explicitly stated otherwise.

### 5.4.1 NetFlow data

NetFlow data is collected at Telenor Denmark's Border Gateway Protocol (BGP) Autonomous System (AS) border routers, representing all Internet traffic entering and exiting Telenor's network, as depicted in Figure 5.3 on the facing page. As indicated in the figure, two primary types of internal traffic not crossing the border routers exist:

- User-to-user traffic: The amount of user-to-user traffic is considered negligible compared to the amount of traffic crossing the border router and is therefore similarly considered negligible for the purpose of this paper.
- User-to-CDN traffic: A number of Content Delivery Network (CDN) nodes are deployed internally, and these serve a significant volume of traffic. However the types of data hosted on these nodes (Netflix/Youtube videos and similar static content etc.) are considered irrelevant to this paper from a user threat and blacklist perspective.

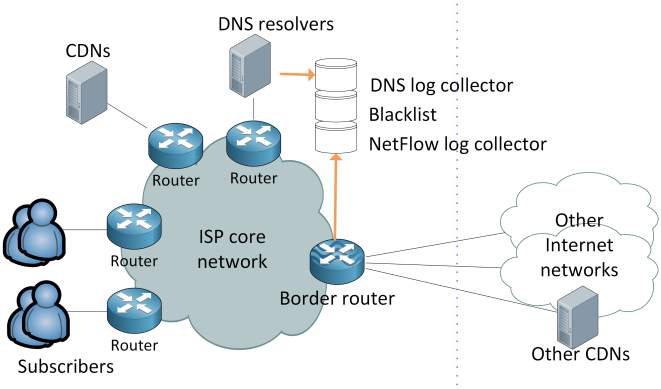

**Fig. 5.3:** *A conceptual view of the Telenor network indicating the sources of DNS and NetFlow data.*

| srcip | srcport | dstip | dstport | proto | packets | bytes |
|---|---|---|---|---|---|---|
| 129.142.227.0 | 56065 | 2.17.0.0 | 443 | TCP | 512 | 32768 |
| 83.73.228.0 | 49906 | 193.28.147.0 | 443 | TCP | 512 | 32768 |
| 85.80.228.0 | 45820 | 8.8.0.0 | 53 | TCP | 512 | 30720 |

**Table 5.3:** *Example NetFlow records. Timestamps are omitted for brevity.*

A (unidirectional) NetFlow record is created by the border routers at least every 60 seconds for each active 5-tuple flow in each flow direction. A sample rate of Q=512 is used, therefore NetFlow records represent data from $\frac{1}{Q}$ packets. The collected data contains $N = 4,63 \cdot 10^9$ NetFlow records.

For the purpose of this paper, only connections initiated by users as a result of a DNS lookup are relevant. Therefore, only NetFlow records with an internal source address are considered, and for TCP connections only flows in which a SYN packet is seen are considered, as this will make sure that the flow start time actually represents the beginning of the flow. Multiple NetFlow records belonging to the same flow (defined by similar start-time and 5-tuple) are aggregated. As a result of this data reduction, $F = 3,92 \cdot 10^8$ flows are available for comparison with blacklisted DNS records. No application layer proxies are deployed.

NetFlow data is anonymized for legal reasons by truncating the internal (user) IP address to a /24 prefix for non-NAT'ed users (or truncating the port for NAT'ed users), truncating the external IP address to a /16 prefix, reverse truncating the timestamp, as well as a number of other measures less relevant to this paper. The anonymization policy applied follows the guidelines of [90]. Table 5.3 contains a number of example NetFlow records.

### 5.4.2 DNS data

DNS data is collected at Telenor Denmark's DNS resolvers, as depicted in Figure 5.3 on the previous page. As the queried domain name is also a part of the DNS response packet, and as this study only focuses on syntactically valid DNS requests for which a response is always issued, only the response packets are collected (including for example NXDOMAIN responses). The resolvers are only accessible from Telenor Denmarks network, and are the default choice for all users. The collected data contains $D = 215 \cdot 10^8$ DNS responses. As a response can contain many Resource Records (RRs), the data is stored such that one record represents a unique RR augmented with the information common to all RRs in the same response.

There are no mechanisms preventing the use of 3rd party DNS resolvers residing outside the Telenor network, and therefore it is relevant to assess the prevalence of that type of traffic. NetFlow data contains $N_{DNS} = 5,92 \cdot 10^6$ records for traffic from users towards port 53 (DNS) and 853 (DNS-over-TLS) ($5,78 \cdot 10^6$ and $1,3 \cdot 10^5$ records respectively). It is not legally possible to inspect this traffic further to quantify how many and which queries this traffic represents. Assuming that one NetFlow record represents one DNS query (yielding the worst-case flow sample likelihood of 1:Q), the 3rd party DNS traffic represents only $\frac{QN_{DNS}}{(QN_{DNS}+D)} = 10,8\%$ of all queries. The traffic towards the Telenor DNS resolvers is therefore considered sufficiently representative of the total DNS traffic, and given the lack of legal basis for inspecting the 3rd party DNS resolver traffic, the 3rd party DNS traffic is disregarded for the purpose of this paper. Some anonymity services like TOR use private top level domains like '.onion'. These top level domains are not registered in the public DNS hierarchy. Therefore, such services are not considered relevant to this paper.

Only 0,1% of the DNS records, $R$, have an *rdata* field referring to a non-CDN IP address within the Telenor Denmark network. This supports the statement made in the NetFlow section that internal network traffic (both user-to-user and user-to-CDN) can be considered negligible to this paper.

DNS data is anonymized for legal reasons by truncating the client (user) IP address to a /24 prefix for non-NAT'ed users (or truncating the port for NAT'ed users), reverse truncating the timestamp, removing the domain name for the 15 most popular domains, and a number of other measures less relevant to this paper. The anonymization policy applied follows the guidelines of [90]. Discounting the anonymized records, $R = 185 \cdot 10^8$ records are therefore available for comparison with blacklists. Table 5.4 on the facing page contains a number of example records.

| clientip | qname/rname | rtype | rdata | ttl |
|---|---|---|---|---|
| 85.83.74.0 | a.config.skype.com.<br>l-0014.l-msedge.net. | A | 13.107.42.23 | 100-299 |
| 85.83.65.0 | log.tiktokv.com.<br>a2047.r.akamai.net. | A | 77.214.51.34 | 1-99 |
| 85.83.65.0 | log.tiktokv.com.<br>a2047.r.akamai.net. | A | 77.214.51.27 | 1-99 |

**Table 5.4:** *Example DNS records. The timestamp is omitted for brevity.*

### 5.4.3 Blacklists

Blacklists that are available for a fee generally outperform free lists [81]. Therefore, blacklists provided by two well-known, commercial DNS blacklist vendors are used for this paper. After a review of the paper, the vendors opted to stay anonymous. The vendors will therefore be referenced as $A$ and $B$, and the individual lists provided by each vendor as $A_1$, $A_2$, etc. The lists contain both IP addresses, IP prefixes and domains. Some of the lists are updated every minute, and the most realistic result would therefore be produced by doing a real-time correlation of DNS data and blacklists. However, as the DNS data is collected independently of the blacklists for operational and privacy reasons, this has not been possible in practice. Instead, the lists are collected at 23:00 CEST each day and the aggregated list is used for comparison for the whole period. In week 1, the aggregated lists contain 11878657 unique IP addresses, 3389 unique prefixes and 989490 unique domains. In week 2, the aggregated lists contain 16286208 unique IP addresses, 3320 unique prefixes and 1002913 unique domains.

For this paper, the impact of a blacklist describes the effect derived from a specific blacklist deployment. The impact of a blacklist with perfect accuracy and perfect completeness will be zero if a user never visits a malicious website. Conversely, if the completeness of a list is low, but deploying the list in practice would block the majority of traffic anyways, the impact will be high.

### 5.4.4 Blacklisted DNS data

The result of the operation described in Section 5.3.1 is that $D_{black} = 6,81 \cdot 10^6$ blacklisted DNS responses are identified. This represents $\frac{D_{black}}{D}$=0,000317 of the total number of DNS responses. The impact of applying DNS based blacklisting is often measured by the magnitude of this number, with the interpretation that user were protected by $6,81 \cdot 10^6$ threats.

| qname | list | srcip/dstip | dport | proto | time diff |
|---|---|---|---|---|---|
| www-pf-dk.filesusr.com. | $A_2$ | 94.145.224.0<br>34.102.0.0 | 443 | TCP | 0 |
| collection.decibelinsight.net. | $A_2$ | 94.145.230.0<br>35.180.0.0 | 1789 | UDP | 434 |
| collection.decibelinsight.net. | $A_2$ | 94.145.230.0<br>35.180.0.0 | 0 | ICMP | 768 |
| wahoofitness.com. | $A_2$ | 2.130.11.0<br>151.101.0.0 | 443 | TCP | 93 |

**Table 5.5:** *Example of the most relevant columns from blacklisted communication records.*

### 5.4.5 Blacklisted flows

The result of the operation described in Section 5.3.2 is that $F_{black}$=185460 blacklisted flows are identified. After performing aggregation, a total of $C_{black}$=90796 blacklisted, aggregated flows are identified. Table 5.5 contains a number of examples of blacklisted, aggregated flows. $C_{black}$ represents the number of flows found in the sampled NetFlows that would have been blocked in the sample week, if DNS based blacklists had been activated for all users.

### 5.4.6 Discussion

The $C_{black}$=90796 blacklisted, aggregated flows contain $C_{black,DNS}$=78312 unique DNS responses (defined by DNS timestamp, clientip, qname and ipprotocol). This represents $\frac{C_{black,DNS}}{D_{black}} = 1,1\%$ of all blacklisted DNS responses. However, as packet sampling is employed, this only accounts for the number of observed flows, not the actual number of flows. Techniques exist for estimating the actual number of flows based on the observed number of flows [91]. However, this does not imply that $\frac{C_{black,DNS}}{D_{black}}$ can be scaled by the same techniques, as the non-observed flows could in theory all be related to the DNS responses already found in $C_{black,DNS}$. Therefore, the data available in this study does not allow any further conclusions on the magnitude of $\frac{C_{black,DNS}}{D_{black}}$.

The data sets from week 1 and 2 show that the amount of blacklisted DNS responses in each week differ significantly from $D_{black} = 6,81 \cdot 10^6$ to $4,56 \cdot 10^6$, a drop of 33%. The collected data cannot offer an explanation for this difference, which may simply be attributed to varying activity levels of the malicious actors. As a consequence of this, the fraction $\frac{D_{black}}{D}$ differ proportionately.

It is, however, interesting to note that although $D_{black}$ differ by 33%, the amount of observed flows blocked, $C_{black}$, only show a drop of 4%, from 90796 to 86854. The estimated ratio of blacklisted DNS requests that result in a TCP flow, $\frac{C_{black,DNS}}{D_{black}}$, does not vary much between the weeks either. This could indicate that the amount of blacklisted flows may be a temporally less variable metric than the amount of blacklisted DNS responses.

For readability, this paper will refer to the set of aggregated flows that are considered blacklisted because of an IP address entry on the blacklist as $C_{ip}$ (68045 entries), the set of aggregated flows that are considered blacklisted because of a domain name entry on the blacklist as $C_{dom}$ (22842 entries), and the set of aggregated flows that are considered blacklisted because of both a domain name and IP address entry on the blacklist as $C_{both}$ (91 entries), where $C_{ip} \cup C_{dom} = C_{black}$ and $C_{ip} \cap C_{dom} = C_{both}$. As $C_{both}$ contains an insignificant number of entries, this category will not be analysed separately in this paper.

## 5.5 Type of maliciousness

Two sets of blacklisted flows, $C_{ip}$ and $C_{domain}$, were identified in the previous section. These are the flows that would have been blocked if DNS based blacklists had been deployed, thus representing a network-level impact of blacklist deployment (subject to scaling due to NetFlow sampling). However, as outlined in the introduction, some blocked flows do not represent a threat to the user due to different types of maliciousness, and these may be seen as a nuisance instead. To quantify this user-level impact of blacklist deployment, this section will categorize the flows by the type of maliciousness.

Different types of malicious behaviour can cause a domain name or IP address to be blacklisted, but only some of the types should be considered a threat to the user connecting to the blacklist entry. The observations turn out to be different for $C_{ip}$ and $C_{domain}$, therefore the observations will be described separately.

### 5.5.1 Flows blacklisted by domain name

Both the *A* and *B* lists provide categories for phishing/malware/botnet related domains, as well as a more general spam category. The latter category includes for example domains in unsolicited mails promoting pills, counterfeits, dating sites etc., and is therefore in terms of badness distinct from malware/phishing domains. Although the sites and goods promoted in the spam category may not be desired to most users, they do not represent a cyber security threat. On the other hand, the phishing and malware related domains in what we will define as the malicious category clearly represent a cyber security threat to the user. In $C_{dom}$, 3% of the flows are in the malicious category, and the remaining 97% of the flows are in the spam category.

### 5.5.2 Flows blacklisted by IP address

Determining the type of maliciousness for entries in $C_{ip}$, requires different approaches for each list used.

The *B* lists provide a cause for an IP address to be blacklisted, and 99% of all IP address entries in the *B* lists are in the malicious category. However, only 5 entries in $C_{ip}$ are blacklisted by B list entries (109 entries in the week 2 data set). As this is an insignificant amount compared to the total amount of entries in $C_{ip}$, no further analysis of the type of maliciousness of these entries is made.

Two *A* lists contain IP addresses: The $A_1$ and $A_2$ lists. The $A_1$ registers only spam emitters, and the 18292 flows blacklisted only by the $A_1$ list (and not also the $A_2$ list) are therefore considered in the spam category.

The type of maliciousness is not immediately available for the $A_2$ list. Two distinct groups of $A_2$ related flows (including flows that relate to both $A_2$ and $A_1$) are therefore categorized by other means:

- A subset of $A_2$, called $A_3$ is available as a separate list. 3179 entries in $C_{ip}$ are marked by the $A_3$ (5%) and are therefore in the malicious category.
- A substantial amount of entries (13634, 20% of $C_{ip}$) relate to a single IP address owned by a laundry company. A manual lookup reveals that this IP address is in the spam category [89].

An informal conversation with list *A* representatives concluded that the vast majority of $A_2$ related flows not accounted for above are likely to be in the spam category as well. However, as we cannot quantify this, we will categorize the remaining flows as having unknown type of maliciousness.

### 5.5.3 Discussion

The type of maliciousness for the $C_{ip}$ and $C_{domain}$ flow sets are listed in Table 5.6. An important note is that if Telenor Denmark had only deployed domain name based blacklists, and only blocked the flows that are considered malicious to the user, a total of 1360 observed flows would have been blocked during week 2. The unknown $C_{ip}$ entries are expected to mostly be in the spam category, with an informed guess setting the fraction of malicious flows in $C_{ip}$ to less than 10%.

Some DNS based blocking implementations redirect the user to a web page warning the user that he has been blocked for security reasons. Web

| Type | $\mathbf{C_{ip}}$ | | $\mathbf{C_{dom}}$ | |
|---|---|---|---|---|
| | Week 1 | Week 2 | Week 1 | Week 2 |
| Spam | 31926 (47%) | 46918 (75%) | 22061 (97%) | 23126 (94%) |
| Malicious | 3184 (5%) | 1151 (2%) | 781 (3%) | 1360 (6%) |
| Unknown | 32935 (48%) | 14614 (23%) | 0 | 0 |

**Table 5.6:** *Type of maliciousness for blacklisted flow sets.*

| Type | $\mathbf{C_{ip}}$ | | $\mathbf{C_{dom}}$ | |
|---|---|---|---|---|
| | Week 1 | Week 2 | Week 1 | Week 2 |
| Spam | 20% | 13% | 40% | 37% |
| Malicious | 11% | 47% | 72% | 61% |
| Entire data group | 39% | 18% | 40% | 38% |

**Table 5.7:** *Port 80/443 (HTTP/HTTPS) fraction of flows. In the group of flows that are blacklisted by IP address (is in $C_{ip}$) in the week 2 data set, 13% of the spam-related flows in the group use port 80/443, and 18% of all flows in the group use port 80/443.*

traffic, defined as traffic towards port 80 and 443, accounts for 40% of the entries in $C_{dom}$, and 72% of malware/phishing entries in $C_{dom}$. Further numbers are available in Table 5.7. Measuring the impact of the DNS based blocking by the number of visits to the warning web site will therefore underestimate the efficiency.

## 5.6 Misaligned endpoints

In some scenarios where a user connects to a blacklisted IP address, there is a chance that the user does not in fact connect to the entity that caused the IP address to be blacklisted. A popular example is when a user connects to a web site hosted in a shared web hosting environment. The IP address of the shared hosting environment may be on the blacklist, but it may be included on the blacklist even though only one of the hosted sites serves malicious content. In this case, it is not possible to determine from either NetFlow or DNS data if the web site actually accessed by the user is benign or malicious. From a user perspective, this will likely be perceived as a nuisance, as the blacklist will then prevent access to benign sites not representing any risk. To assess the user-level impact of deploying DNS based blacklisting, it is therefore relevant to quantify the fraction of flows in $C_{ip}$ where the endpoint of the flow and the endpoint causing the IP address to be blacklisted can differ. We shall refer to these flows as potentially having misaligned endpoints.

An analysis of each individual endpoint IP prefix would be impractical. In order to identify the most prominent groups of $C_{ip}$ flows, we choose to focus the analysis on the groups of flows where:

- Many domain names are associated with a single destination IP prefix (high $qname/dstip$ ratio)
- Many destination IP prefixes are associated with a single domain name (high $dstip/qname$ ratio).
- A popular destination IP prefix is used (high $dstip$ count)
- A popular domain name is used (high $qname$ count)

Based on this analysis, three different scenarios that can cause misaligned endpoints has been identified in the $C_{ip}$ data set, and these three scenarios are elaborated in the following three subsections.

### 5.6.1 Shared content providers

The entries in $C_{ip}$ with a high $qname/dstip$ ratio all have a $dstip$ owned by a CDN, shared web hosting or similar cloud content provider like Amazon, Microsoft Azure, Google Cloud, DigitalOcean or Tencent. A total of 29556 entries ( 43% of $C_{ip}$) are related to such servers and we consider these flows to potentially having misaligned endpoints.

An number of $dstip$s are owned by Virtual Private Server (VPS) service providers and regular ISP customers. 516 entries are considered ISP customers as well, as they relate to a server with a dynamic IP address, identified by the use of a .duckdns.org domain name, a service used for assigning a permanent domain name to a dynamic IP address. These will not be considered as potentially misaligned endpoints.

It could be argued that all destination IP addresses could easily be enumerated by the use of BGP AS numbers. In practice, however, this turns out not to be viable for a number of reasons. First, only the /16 prefix address is available due to anonymization, and such a prefix may cover several AS numbers. Second, some providers share the AS number between the ISP and hosting part of the company (like OVH). Third, some providers reserve smaller prefixes for specific customers (like Amazon). Fourth, some providers use IP space assigned to other entities (like Tencent using ChinaUnicom owned IP prefixes).

### 5.6.2 VPN service providers

VPN service provider (PrivateInternetAccess, Hula, NorthGhost etc.) traffic identified by the $qname$ accounts for 12469 ( 18%) of all entries in $C_{ip}$. The specific implementations by the different providers is not known. However, it seems unlikely that a user creating a connection towards such an IP address

| Cause | Week 1 | Week 2 |
|---|---|---|
| Shared content providers | 29556 (43%) | 19370 (31%) |
| VPN service providers | 12469 (18%) | 13710 (22%) |
| NTP pool | 4006 (6%) | 3505 (6%) |
| Total | 45698 (67%) | 36336 (58%) |

**Table 5.8:** *Amount of entries in $C_{ip}$ with different causes for potential endpoint misalignment. Note that the total is less than the sum, as for example an NTP pool entry may also be a shared content provider entry, and this only counts as 1 in the total.*

will be relayed to a host residing behind the VPN service. A connection towards such a server seems more likely to be an attempt to use the service. The VPN provider IP addresses are likely to have been blacklisted because hosts using the service generated traffic that triggered a blacklisting. We shall therefore consider the VPN provider IP addresses as potentially having misaligned endpoints.

### 5.6.3 NTP Pool

Traffic towards hosts registered in the ntppool project[2] is identified by the *qname* containing .pool.ntp.org. This traffic accounts for 4006 ( 6%) of all entries in $C_{ip}$. A DNS request for a .pool.ntp.org domain will return a number of IP addresses, where each IP address belongs to a pool member. If the IP address of one of the pool members in the DNS response is blacklisted, the entire DNS response is considered blacklisted. Therefore, a connection towards a different pool member will be considered part of $C_{ip}$. This is not considered a flaw in the data analysis, as it reflects how DNS based blacklisting is implemented in practice. Blacklisting relates to the entire DNS request/response pair, not just to single response resource records. It is therefore likely that these flows have misaligned endpoints.

### 5.6.4 Discussion

Table 5.8 summarizes the amount of flows that may have misaligned endpoints and lists the 3 identified scenarios causing the potential misalignment. As seen in the table, we consider at least 45698 of 68045 $C_{ip}$ entries (67%) as potentially having misaligned endpoints. Blocking these flows involves a risk of blocking benign sites. Although the specific IP address on the blacklist may be correct by reflecting a malicious endpoint using that IP address (a true positive), the user may perceive it as a false positive. When a provider considers deploying DNS based blacklists that includes IP addresses, the willingness of both operators and users to accept this risk should therefore be carefully considered up front.

[2] https://www.ntppool.org/

Of the 45698 entries, only 5 are tagged by the $B$ lists, while the rest is tagged by $A$ lists. This highlights that the choice of blacklists represent an important limitation to the results presented in this section. The numbers presented are unlikely to be representative of other blacklists. However, looking outside the scope of this paper, this also suggests that when considering deploying DNS based blacklisting, the concept of IP address blacklists should not necessarily be deselected upfront. The risk of blocking benign sites due to endpoint misalignment can be decreased significantly by the careful selection of IP address blacklists.

As outlined in Section 5.2, we are only aware of one other paper that evaluates the risk of misaligned endpoints for the individual blacklist entries [88]. However, it is important to notice that the results presented in this section would not be directly comparable, as they relate to the blacklisted flows ($C_{black}$), not the blacklisted DNS responses ($D_{black}$ ) or the individual entries on a blacklist (the latter being the focus of [88]). The primary purpose of our work is to present the method of using NetFlow to measure the impact of deploying blacklists using a specific set of blacklists as examples, and not to compare blacklists. Hence, we consider evaluating a larger number of blacklists as an extension to this paper.

## 5.7 Impact

Sections 5.3-5.6 identify blacklisted flows, identify the type of maliciousness and assess the risk of endpoint misalignment. This section combines the result of the previous sections to quantify impact of deploying DNS-based blacklists seen from the network perspective and from the user perspective. Furthermore, this section describes interesting future works.

### 5.7.1 Network-level impact

The network-level impact of DNS based blocking is usually practically measured by counting the number of blocked DNS requests or by counting the number of visits to a warning page to which a user has been redirected. In this paper, the impact is instead measured using the number of blocked flows instead, and this reveals the fraction of web related flows.

- Approximately 0,02-0,03% of all DNS responses match a blacklist entry, and 1,1-1,5% of these blacklisted DNS responses can be associated with an observed flow, denoted a blacklisted flow. The use of sampled flow data was found to hinder the estimation of the actual fraction of blacklisted DNS responses that can be associated with a flow. Researchers or ISPs with access to non-sampled NetFlow and DNS data should assess the fraction of blacklisted DNS responses that can be associated with a flow. Given a known amount of blacklisted DNS responses, this would make it possible to more accurately assess impact of doing DNS based blocking.

- Some DNS based blocking implementations redirect the user to a website containing a message warning the user that he has been blocked for security reasons. Therefore, such implementations measure only the part of the traffic that is web traffic. Of the flows blacklisted by domain name, 38-40% are web traffic. Of the flows blacklisted by domain name *and* considered having a high threat level, 61-72% are web traffic. Therefore, this paper shows that measuring the impact of blacklisting by the number of visits to the warning web site underestimates the impact. ISPs and company system administrators should implement measures to also count non-web related connections, in order to get a more correct assessment of the blacklist impact.

These results are specific to a particular week, use particular blacklist vendors, and a particular ISP. Despite the listed limitations, we find the results significant enough to suggest that the method of using NetFlow to measure the impact of applying DNS based blacklists represents an improvement to existing methods.

### 5.7.2 User-level impact

Approximately 25% of the blacklisted flows relate to a blacklisted domain name, whereas the remaining 75% of the blacklisted flows relate to a blacklisted IP address.

- The flows blacklisted by domain name are, using the threat type categories provided by the blacklist vendors, divided into two groups. First, a group relating to general spam, considered a nuisance rather than a cyber security threat, accounts for 94-97% of the flows. Second, a group relating to phishing, malware and botnet accounts for the remaining 3-6%. When deploying DNS based blacklisting, it is therefore important to consider if both or only one of these types of traffic should be blocked, as this will have a significant impact on the amount of blocked connections experienced by the user.

- Of the flows blacklisted by IP address, this paper shows that 58-67% may be flows towards benign sites, primarily due to the prevalence of shared web hosting, whereby multiple web sites / domain share the same IP address. From a user and operator perspective, the willingness to risk blocking benign sites must be considered before deploying IP address based blacklists. This study shows that carefully selecting the IP address blacklist vendor can be a significant contribution to minimizing this risk.

These results are also specific to particular blacklist vendors and a particular ISP. Here, however, we show that the specific measurements depend a lot on the particular blacklist used, and therefore it is a clear limitation that these results cannot be generalized to different blacklists.

## 5.8 Conclusion

In this paper, we propose a method to measure the impact of deploying blacklists by combining NetFlow and DNS data. We evaluate the method on real data, containing anonymised NetFlow and DNS records collected by Telenor Denmark for two weeks, and combine these with blacklists containing IP addresses and domain names provided by two commercial vendors.

The measurements show that 0,02-0,03% of all DNS responses match a blacklist entry, however only 1,1-1,5% of these blacklisted DNS responses can be associated with an observed flow. Furthermore, only 38-40% of the blacklisted flows are web traffic. These observations suggest that the use of flow data can be used to make a more precise impact assessment than counting the amount of DNS responses matching a blacklist entry or counting the amount of visits to a warning web page.

For flows blacklisted by domain name, 3-6% of the flows related to phishing, malware and botnet domains, while the remaining flows relate to spam domains. For the flows blacklisted by IP address, 58-67% may be flows towards benign sites. These observations show that the careful consideration of the choice of blacklist type (domain name or IP address) and category (spam, malware etc.) before deployment is essential to avoid undesired impact seen from a user perspective when deploying DNS-based blacklists.

CHAPTER 6

# Paper C

## An uneven game of hide and seek: Hiding botnet CnC by encrypting IPs in DNS records

**Main author:**
Martin Fejrskov
*Technology, IP Network and Core*
*Telenor A/S*
Aalborg, Denmark
mfea@telenor.dk

**Co-authors:**

Jens Myrup Pedersen
*Cyber Security Network*
*Aalborg University*
Aalborg, Denmark
jens@es.aau.dk

Leon Böck
*Telecooperation Lab*
*Technische Universität Darmstadt*
Darmstadt, Germany
boeck@tk.tu-darmstadt.de

Emmanouil Vasilomanolakis
*Cyber Security Network*
*Aalborg University*
Copenhagen, Denmark
emv@es.aau.dk

**Presented at:**
*IEEE Conference on Communications and Network Security (CNS) 2021*

Funded by Telenor A/S and Innovation Fund Denmark.

# Abstract

*Botnets frequently use DGA and fast-flux techniques to ensure the availability of their command and control (CnC) infrastructure. However, the CnC IP addresses are still exposed in plain-text in publicly available DNS A records, which can be exploited by defenders to disrupt botnet operations. This paper presents the concept of the IP Generation Algorithm (IGA) as a novel method, usable by botmasters, to encrypt the CnC IP address in DNS records to avoid plain-text IP address exposure. This raises the bar for blacklisting malicious IP addresses, and can also be combined with existing techniques to further harden the CnC. For use by defenders, an IGA botnet detection method based on the combination of DNS and NetFlow data is presented and validated using an emulated botnet and an ISP data set.*

**Keywords:** *Botnet · NetFlow · DNS · encryption · detection*

## 6.1 Introduction

Many botnets use the DNS protocol and infrastructure to establish command and control (CnC) connections between a bot and the botmaster. Defenders can identify the botnet related domain names and block them in the DNS infrastructure. This is typically made significantly harder by the botmasters by using Domain Generation Algorithms (DGAs) to frequently create and register new domain names [92]. As the DNS responses to the DGA domains still reveal the IP address of the CnC host, defenders can choose to block DNS requests relating to that IP address, or block IP connections towards the CnC host IP address. This is made more difficult by the botnets by using fast-flux (FF) to frequently change the IP address registered with the DGA generated domain name.

The scarce resource in this game of hide-and-seek is the IP address. A usable IP address must represent an infected host, whereas the domain names can be freely chosen. For the botmaster, it would therefore be attractive not to expose the plain-text IP addresses of the CnC host in DNS records.

In this paper, we propose the IP Generation Algorithm (IGA) as a novel technique to encode the CnC host IP address. The botmaster would use the IGA to encode the plain-text CnC host IP address using a time-variable key, and register the domain with the encoded version. The bot would use the IGA to decode the retrieved address to reveal the plain-text IP address of the CnC host. Similar to a DGA, the purpose of the IGA is to generate random, legitimate looking IPs. However, as opposed to a DGA, the IGA is a two-way function.

A botmaster using the DGA and IGA techniques in combination could choose not to flux the IP address at all, effectively bypassing any FF detection techniques. Alternatively, the botmaster could use FF with a much higher frequency with IGA encoded IP addresses, as the amount of IP addresses available is not longer a limiting factor. In both cases, the botmaster does not reveal the plain-text IP addresses in the DNS messages.

The defender, however, faces potentially severe consequences. First, FF detection methods can be irrelevant when identifying malicious domains and IP addresses. Second, it would become impossible to generate IP address blacklists by only studying DNS data, as it does not reveal the plain-text IP address. Third, a defender unwittingly blocking an encoded IP address could result in the blocking of a benign host that matches the encoded address.

The primary contributions of this paper are:

- The IGA concept and a Python based implementation for encoding and decoding the CnC IP address.
- A DNS and NetFlow based IGA detection algorithm validated using Internet Service Provider (ISP) data and an emulated IGA botnet.

Section 6.2 of this paper introduces the threat model. Section 6.3 describes the IP Generation Algorithm, Section 6.4 describes an IGA detection method that is validated in Section 6.5. Section 6.6 summarizes related work and 6.7 concludes the paper.

## 6.2 Threat model

The goal of the botmaster is to use the DNS infrastructure to find one or more IP addresses AND subsequently create a connection towards the discovered IP address for CnC purposes. The botmaster is assumed:

- not to have control of any type of DNS servers.
- not to be able to actively obfuscate, spoof or modify the addresses and ports in the TCP/UDP/IP layers.
- to only use A-type DNS records (although the method presented in this paper could be extended to other types).
- to be able to create decoy connections towards any IP addresses present in clear-text in A-type DNS responses.

The defender is assumed to be able to inspect/modify/block DNS traffic at the application layer and CnC connections at the transport and network layer. Also, the defender is assumed not be obstructed by the use of DNS-over-TLS or DNSCrypt, such as the case where the defender controls the resolvers. Thus, a typical ISP can assume the role of defender.

The botmaster could choose to combine IGA with DGA/FF techniques and thereby be subject to existing detection methods for these techniques. This paper will assume that such techniques are not used, and will not depend on them for detection.

## 6.3 The IP Generation Algorithm

This section describes the proposed implementation of the IGA. The objective of the IGA is to encode a globally routed, plain-text IP address into an encrypted version, that can also be represented in the form of a globally routed IP address.

The proposed IGA encoding implementation contains three steps: ranking, encryption and mapping. The description in this section will, for brevity, focus on the encoding only. The decoding process is simply the reverse of the encryption process. These three steps are depicted in Figure 6.1 and elaborated in the following paragraphs. The four coloured dots in Figure 6.1 each represent a CnC IP address and visualize how the encryption process changes their position in the respective value ranges.

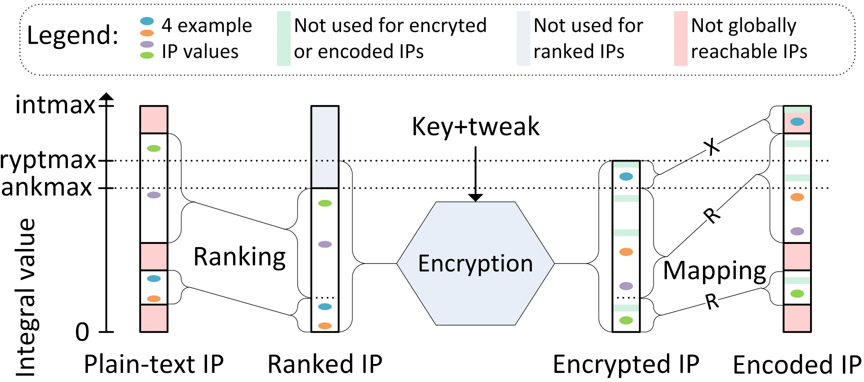

**Fig. 6.1:** *Conceptual illustration of the IGA encoding procedure containing three steps: ranking, encryption and mapping.*

### 6.3.1 Ranking

The purpose of ranking is to create a consecutive ordering of all globally reachable IP addresses in order to minimize the probability that the encryption and mapping steps will output an address that is not globally reachable. This excludes for example private, link-local, loopback nets etc. [93] [94]. An

IPv4 address is represented by a 32 bit integer in the range $[0..intmax]$ where $intmax = 2^{32} - 1$. A total of $R = 592708608$ IP addresses are not globally reachable, these are represented by red colour in Figure 6.1 on the preceding page. The pool of reachable IPs can be ranked by representing them in the interval $[0..rankmax]$ where $rankmax = intmax - R$. The values never provided as output by the ranking function are represented by blue colour in Figure 6.1 on the facing page, and the size of the blue area is the sum of sizes of the red areas.

### 6.3.2 Encryption

The input and output for the format-preserving encryption is not an integer, but two or more characters, each character consisting of a symbol from an alphabet. The number of different symbols in the alphabet is called the radix, radix $\in [2..2^{16}]$ [95]. Using this terminology, an IP address can be represented by four characters (bytes) and a radix of 256, as each byte can have a value in the range $[0..255]$. The integer representation of the input and output values is therefore the range $[0..cryptmax]$, where $cryptmax = radix^{characters} - 1$.

The number of characters and the radix should therefore be chosen such that $rankmax = cryptmax$, as this would ensure that a globally reachable IP address could be mapped one-to-one to another reachable IP address, but unfortunately this does not have an integral solution. Choosing $rankmax > cryptmax$ would make it impossible to encode a part of the reachable address space. This is very undesirable to a botmaster, as this would make it impossible to use those addresses for CnC. Choosing $rankmax < cryptmax$ is also undesirable, as the output of the encryption would then need to be mapped to a part of the IP space that is not reachable, indicating that the IP may be encrypted. For this paper, we will assume that botmasters prefer to create DNS A records with unreachable IP addresses rather than being unable to use certain IP addresses for CnC, and therefore opt for the $rankmax < cryptmax$ approach.

Solving $min(cryptmax - rankmax) \ll R$ for integral characters and radix reveal four options listed in Table 6.1 on the next page as well as the $cryptmax - rankmax$ difference. Two Python-based encryption/decryption libraries have been found. One supports only $radix < 37$ [96], which is not compatible with the options listed in Table 6.1 on the following page. Another supports only an even length and a $radix \leq 256$ [97], which matches only one possible option in Table 6.1, namely a length of 4 and a radix of 247 yielding $cryptmax = 247^4 - 1$. Note that a botmaster could choose to use a different library with different limitations, and therefore be able to choose a better length and radix combination.

| Characters | Radix | cryptmax-rankmax |
|---|---|---|
| 2 | 60847 | 98722 |
| 3 | 1547 | 35636 |
| 4 | 247 | 19839394 |
| 5 | 82 | 5139745 |

**Table 6.1:** *Options for the selection of radix and input/output length*

### 6.3.3 Mapping

The purpose of this step is to map the encrypted IP addresses ($[0..cryptmax]$) to a reachable IP addresses ($[0..intmax]$).

The green areas in Figure 6.1 on page 66 represent the output values of encrypting the values in the blue area between *rankmax* and *cryptmax*. This is illustrated by the summed size of the green areas being the same size as the blue area between *rankmax* and *cryptmax*. As values between *rankmax* and *cryptmax* are never output by the ranking step, the values in the green areas are never output by the encryption step (for a given key). However, as the specific, unused output values of the encryption depend on the key in use, the mapping function is unable to exclude these unused values before performing the mapping. Therefore, the mapping takes as input the range $[0..cryptmax]$ rather than $[0..rankmax]$.

If it was possible to set $rankmax = cryptmax$, the mapping operation would be the reverse of the operation performed by the ranking step. Instead, the mapping function is split into two functions. First, a function operating on encrypted IP values $\leq rankmax$ that performs the reverse operation of the ranking function, marked with an "R" in Figure 6.1 on page 66. Second, a function that operates on encrypted IP values $> rankmax$, marked with an "X" in the figure. The latter values are mapped to a segment of the IP address space that is not reachable, and for the implementation provided by this paper, the 224.0.0.0/4 subnet reserved for Multicast is used.

### 6.3.4 General notes

A note on the use of a tweak is important. A tweak can be considered as a non-secret key, that should vary with each instance of the encryption [95]. In an IGA context, an obvious choice for the tweak is the DGA generated domain name: This will ensure that a single (plain-text) IP address would be encoded into several different IP addresses, one for each domain name, even though the key is held constant. As the DGA algorithm already includes time variability, using the domain name as tweak also eliminates the need for time variability in the encoding/decoding step of the IGA.

| Plain-text | Encoded |
|---|---|
| 193.0.2.255 | 192.238.197.236 |
| 8.8.8.8 | 154.141.220.55 |
| 152.13.43.124 | 32.72.64.180 |
| 212.220.255.3 | 199.22.251.26 |

**Table 6.2:** *Examples of plain-text IPs and their associated encoded version.*

Without the ranking and mapping steps, the probability that the encryption would output a non-reachable IP address is $\frac{R}{intmax} = 0.138$. By including the ranking and mapping steps, this probability is reduced to $\frac{cryptmax-rankmax}{intmax} = \frac{(radix^{characters}-1)-(intmax-R)}{intmax} = 0.005$. With a crypto implementation supporting an odd number of characters (3) and a higher radix (1547) (as seen in Table 6.1 on the preceding page), this probability could be reduced even further to $8,3 \cdot 10^{-6}$.

Python code implementing the IGA algorithm described above is available in [98]. The implementation uses the FF1 encryption and decryption scheme provided by [97], which unfortunately does not support the use of tweaks. Table 6.2 contains examples of encoding using the key "someGoodKey".

Although the rank-and-encipher approach is proposed by [93] as a method for semantic-preserving encryption, our paper significantly extends it by considering multiple IP scopes and the associated ranking method, by introducing and solving the problems of differing set sizes for the ranking output and the encryption input, and by providing an implementation.

### 6.3.5 Summary

This section describes the IP Generation Algorithm and the three steps used to encode a globally routed, plain-text IP address into an encrypted version, that can in most cases also be represented in the form of a globally routed IP address. The encryption step maps a plain-text IP address into an encrypted IP address, and using the DGA generated domain name as tweak ensures time variability in the encrypted IP address. The ranking and mapping steps reduce the probability of the encoded IP being outside the globally reachable address space, thus decreasing the suspiciousness of the encoded IP.

## 6.4 Detecting IGA botnets

The purpose of this section is to outline a method to detect the presence of an IGA-based botnet by using DNS and NetFlow data. The key idea of the method is to identify sets of source IPs that exhibit the group behavior expected by IGA bots: They resolve the same set of domain names and then create unnamed flows towards the same set of destination IP addresses. Each of the steps are described in more detail in the following subsections. The results of applying the method to a real dataset will be the topic of Section 6.5 on page 76.

Throughout this section, it is assumed that the DGA used by the botnet will generate a domain per day, as this is a common domain validity period for time-dependent DGAs [92]. Adapting the proposed method to a different frequency should be trivial.

A flow is defined as the daily aggregation of all packets sharing the same 5-tuple (protocol and source/destination IP address/ports), timestamped using the first-seen timestamp observed in NetFlow records for that 5-tuple.

As the algorithm is based on identifying source IPs that exhibit the same behaviour over $k$ days, it is a requirement that DNS and NetFlow data is observed $\geq k$ days. From an adversary perspective, it is likely undesirable not to allow the majority of bots to establish CnC communication at least once a day. Therefore, the choice of $k$ can be based on data availability. It is demonstrated in Section 6.5 that $k \geq 3$ is a necessity to avoid a high number of false positives in the detection.

### 6.4.1 Dataset reduction

The first step in the detection algorithm is to apply systematic white-listing so that irrelevant data is disregarded (such as flows or DNS request related to popular domains or CDN IPs). The main purpose of this is to reduce the processing requirements for the following steps. Therefore, not all reduction operations may be relevant to apply for all datasets. The remaining parts of this paper will refer to the result of applying all the desired reduction operations as two sets, a set of DNS Resource Records (RRs), $D_{reduced}$, and a set of flows, $F_{reduced}$.

The detection method proposed in this paper is considered complementary to existing DNS-only or NetFlow-only methods, therefore a black-listing approach based on such methods is intentionally not used. The following notation is used:

- $D_{all}$: All collected A type DNS RRs.
- $D_{all}^{rdataIP}$: The unique rdata IPs in $D_{all}$
- $D_{all}^{FQDN}$: The unique qname FQDNs in $D_{all}$
- $D_{all}^{2ndlevel}$: The unique second level domains found by extracting these from all entries in $D_{all}^{FQDN}$
- $F_{all}$: The flows in all collected NetFlow packets.

#### 6.4.1.1 Outside originated flows

All flows originating from outside the network under observation should be white-listed, as the DNS requests related to these flows will not be observable. Notice that when using sampled NetFlow, it is often not possible to identify the flow origin.

#### 6.4.1.2 Frequent second level domains

Two techniques can be used to white-list both DNS RRs and flows based on frequently seen second level domains:

A popular white-listing technique is to disregard any domain names found in lists of popular second level domains (such as the Alexa Top list), which will be denoted $W_{2ndlevel}$. The rationale of this is that is unlikely that a botmaster can retain long-term, unnoticed control of either domains or servers used by such high-volume organisations. The list is used to find the white-listable RRs, $D_{whitelisted} = D_{all} \ltimes W_{2ndlevel}$, which can then be used to find two reduced sets, $D_{reduced} = D_{all} \setminus D_{whitelisted} = D_{all} \triangleright W_{2ndlevel} = D_{all} \triangleright D^{2ndlevel}_{whitelisted}$ and $F_{reduced} = F_{all} \triangleright D^{rdataIP}_{whitelisted}$, that do not include RRs or flows relating to the white-listed second level domain names. The threshold $T_{secondlevel} = \frac{D_{reduced}}{D_{all}}$ is defined to indicate how large a fraction of DNS RRs is retained.

Another technique is to white-list any DNS RRs with qnames or rnames containing FQDNs or second level domains, that are requested by more than $T_{maxbots}$ source IPs, the set of white-listed DNS RRs denoted $W_{2ndlevel}(T_{maxbots})$. The rationale of this is that a specific DGA FQDN will never be requested by a benign client, therefore FQDNs requested by more than $T_{maxbots}$ source IPs will be benign, if the number of bots in the observed network is $\leq T_{maxbots}$. This can be used to construct $D_{reduced}$ and $F_{reduced}$ in a similar way as described above. This technique should only be used to white-list flows if the botnet is not assumed to deploy CnC hosts on servers also serving benign content.

Although these two techniques overlap, practical experiments show that both provide distinct dataset reductions.

#### 6.4.1.3 Frequent rdata IPs

It seems tempting to also white-list DNS RRs and flows based on frequently observed rdata IP addresses. This is, however, not a feasible strategy, as the botmaster has unlimited control over the contents of the rdata IP address, and could therefore choose to implement an IGA, that only uses frequently seen IP addresses (CDN IPs etc.) as output space for the IGA algorithm.

#### 6.4.1.4 Frequent destinations

Two techniques can be used to white-list flows based on frequently seen flow destinations:

For NetFlow data, a popular technique is to disregard Content Delivery Network (CDN) IPs, denoted $W_{CDN}$, solely serving static/benign content, such as Youtube, Facebook and Akamai CDNs (but not for example Amazon or Azure CDNs, as these can host private virtual servers). The rationale of this is that it is unlikely that a botmaster can retain long-term, unnoticed control of such servers. Notice that as ISPs deploy CDNs locally in their networks, the CDN IPs will likely differ between ISPs. This can be used to construct a reduced set of flows, $F_{reduced} = F_{all} \triangleright W_{CDN}$.

Another technique is to construct a set of white-listed flow destination endpoints (defined by a destination IP, protocol and port combination), denoted $W_{destination}(T_{maxbots})$, such that a flow is in this set if more than $T_{maxbots}$ source IPs contact a specific endpoint. The rationale of this is that a specific botnet CnC endpoint will never be requested by a benign client (assuming that the CnC software does not share the destination port with benign software, such as Apache). This can be used to construct a reduced set of flows, $F_{reduced} = F_{all} \triangleright W_{destination}(T_{maxbots})$.

Although these two techniques also overlap, practical experiments show that both provide distinct dataset reductions.

#### 6.4.1.5 Domains seen yesterday

FQDNs and second level domain names seen in DNS requests the day before the day under analysis form a large white-list, $W_{yesterday}$. The rationale of this is that the purpose of a DGA algorithm is to create a new and unique FQDNs each day, so that a listing of the domain name on a free or commercial domain name blacklists becomes irrelevant. This can reduce the set of DNS RRs to $D_{reduced} = D_{all} \triangleright W^{2ndlevel}_{yesterday} \triangleright W^{FQDN}_{yesterday}$.

### 6.4.2 Unnamed flows

It is an inherent property of using an IGA that the DNS RRs and flows will only be explicitly related by the source IP address. The A record IP in the DNS response is encoded, and therefore not identical to the destination IP in the flow.

Any flows for which a matching DNS record can be found is called a named flow, $F_{named} = F_{reduced} \ltimes D_{all}$. These flows are not relevant for IGA detection and can be white-listed, leaving only the unnamed flows, $F_{unnamed} = F_{reduced} \setminus F_{named} = F_{reduced} \triangleright D_{all}$. The matching criteria are source IPs, timestamps, rdata record and destination IP.

When identifying named flows, the aspect of time is relevant. A flow could be named by the first preceding DNS record, or by all preceding DNS records within a certain time window, for example the window set by the TTL or the window of the current day. In order to make $F_{unnamed}$ as small as possible by making the white-list $F_{named}$ as large as possible, a time window of the current day seems to be a reasonable approach.

Notice that DNS requests for which a related NetFlow record can be found should not be white-listed, as the botmaster could choose to initiate traffic towards the encoded IP address in the DNS request as a decoy to avoid detection.

### 6.4.3 Re-named flows

Given a set of flows, $F_{unnamed}$, that have no related DNS response, and given a set of DNS responses, $D_{reduced}$, the purpose of the remaining steps of the detection algorithm is to identify the specific entries from each set that are actually created by the IGA botnet. To facilitate this, this section introduces the concept of a re-named flow.

A re-named flow is a flow from $F_{unnamed}$ augmented with a relevant DNS RR from $D_{reduced}$. A DNS RR is considered relevant, if the flow and the DNS request originate from the same source IP address, and if the start time of the flow is within a certain time window, $T_{delaymax}$, of the DNS response. The set of re-named flows, $R$, is therefore given by the theta-join, or selected cartesian product, $R = F_{unnamed} \bowtie_{\theta} D_{reduced} = \sigma_{\theta}(F_{unnamed} \times D_{reduced})$, the $\theta$ condition being the source IP and time window constraints.

The choice of $T_{delaymax}$ requires further consideration, as a DNS request should not be expected to be immediately followed within few seconds by an observed NetFlow record. There are multiple reasons for this: A botmaster could deliberately introduce a variable time delay between the DNS lookup and the CnC connection in order to evade detection, including a delay exceeding the TTL of the DNS RR. Also, the NetFlow data used can be sampled, and therefore the observed flow start timestamp is not necessarily equal to the actual flow start time.

Note that DGA domains are typically registered with a low TTL in order to enable fast-flux. However, as the IGA eliminates the need for fast-flux in the CnC phase, such low TTLs cannot be assumed to be used for IGA based botnets.

$R$ contains a number of entries that are irrelevant for the following IGA detection steps. Only the qname, source IPs and the destination IP are relevant, and $R_{reduced}$ is constructed from $R$ by removing any other information and removing duplicates.

Following the assumption that DGA algorithms generate a new domain name on a daily basis, the processes of reducing the dataset, and identifying unnamed and re-named flows can be performed on a daily basis as well. Therefore, given that DNS and NetFlow data is available for $k$ days, $k$ sets of re-named flows, $R^{1..k}_{reduced}$, can be constructed on an individual basis to form $U = \cup^{k}_{i=1} R^{i}_{reduced}$.

### 6.4.4 Vertices and edges

The next step of the IGA detection algorithm is to build a graph based on all of the re-named flows, $U$. The purpose of the graph is to identify sets of source IPs that exhibit the same behaviour by resolving the same domain names and connecting to the same destination IPs.

#### 6.4.4.1 Vertices

Subsets of $U$ are created, where a subset consists of the entries of $U$ that share a specific combination of qname, destination IP and day. The size of a subset is equal to the number of unique source IP addresses with this combination. Subsets containing $< T_{minbots}$ source IPs are discarded. Subsets containing $\geq T_{minbots}$ entries form the set $V$, after removing the destination IP information and duplicates. $V$ is therefore a set of source IP, qname and day triplets.

Subsets $v_1...n$ of $V$ are created, where a subset $v_i$ consists of all the source IPs that share a specific combination of qname and day. In other words, $v_i$ represents the source IPs that resolve the same domain name on a specific day. Each of the subsets of $V$ are represented by vertices in a graph.

#### 6.4.4.2 Edges

A bidirectional edge connects two vertices if the Jaccard similarity, $J()$, of the two sets of source IPs, $v_i$ and $v_j$, is larger than a given threshold, $T_{jaccard}$ determined as $T_{jaccard} > J(v_i, v_j) = \frac{v_i \cap v_j}{v_i \cup v_j}$. Calculating this only for vertices representing different days represents the expected behaviour of a set of IGA bots (represented by their source IPs) resolving a different domain each day.

The similarity threshold $T_{jaccard}$, $0 \geq T_{jaccard} \geq 1$ should be chosen sufficiently high to eliminate false positives and sufficiently low to make it unattractive for the botmaster to try to avoid detection by generating many DGA domains each day, or by instructing too high a fraction of bots to not create CnC connections each day.

### 6.4.5 Cliques and communities

Having constructed a graph where two vertices are connected if they share a certain fraction of their associated source IPs enables the final steps of the detection algorithm: Clique and community detection.

A k-clique is a set of vertices that are fully connected to at least $k - 1$ other vertices, representing sets of source IP addresses that resolve the same set of domain names and then create unnamed flows towards the same set of destination IP addresses across all $k$ days.

As several DGA domains could be created each day by a botnet, a botnet may be represented by several k-cliques. Therefore, the graph is used to identify k-communities: A k-community is the union of all cliques of size k that can be reached through adjacent (sharing k-1 nodes) k-cliques [99].

A k-community could represent an IGA botnet, where the source IPs related to the community represents the bots. However, a community could also represent other structures than IGA botnets, such as regular botnets generating traffic to some other, common destination (e.g. for attack purposes) based on information obtained through the CnC channel (thereby creating an unnamed attack flow).

### 6.4.6 Summary

This section describes the steps of the IGA detection algorithm, which includes a number of possible data reduction techniques. Based on DNS responses and NetFlow records collected over k days, a number of k-cliques are found that may represent an IGA botnet. Several properties are worth noting:

- No assumptions are made about the similarity of two different qnames when identifying cliques, as is the case for example in semantic based DGA detection methods.

- The rdata IP value is only used for data minimization and to identify unnamed flows, but is not used in the re-naming of flows or the clique identification. The rdata IP value is often key in DGA detection methods for example when identifying IP addresses with many associated qnames.

- No assumptions are made about the distribution or similarity of source or destination port numbers (except for when performing white-listing), which is often the case for NetFlow-based detection methods.

- The flow sizes (packets, bytes or time) are not used, which is often the case for NetFlow-based detection methods.

As the botmaster has almost full control of all of the aforementioned features, these properties should be considered important to any detection algorithm.

## 6.5 Validation of detection method

In this section, IGA botnet traffic will be injected into DNS and NetFlow data from an ISP to validate the IGA botnet detection method described in the previous section. The following subsections describe in further detail the ISP data available, how emulated IGA traffic is injected into the ISP data, and the specific values chosen for the various thresholds used in the detection algorithm. Finally, the results of running the detection algorithm are presented and discussed.

### 6.5.1 ISP DNS and NetFlow data

DNS and NetFlow data for the 1,5M mobile and 100k broadband subscribers of Telenor Denmark is used for validation. NetFlow data is collected at the Border Gateway Protocol (BGP) Autonomous System (AS) border routers using a sample rate of 1:512. This traffic therefore represents all Internet traffic entering and exiting Telenor Denmark's network. DNS data is collected at DNS resolvers by collecting all DNS response packets. The resolvers are only accessible to Telenor Denmark subscribers, and they are the default choice for all subscribers.

NetFlow and DNS data are anonymized for legal reasons by truncating the internal (subscriber) IP to a /24 prefix for non-NAT'ed subscribers (or truncating the port for NAT'ed subscribers) as well as a number of other measures less relevant to this paper. The anonymization policy applied follows the guidelines of [90] except that varying levels of anonymization is applied to the NetFlow destination IPs and the DNS rdata IPs in order to evaluate the effect of anonymization on the results. Due to anonymization of source IPs, all traffic will originate from only approximately 15k prefixes, each representing somewhere between 0 and 256 customers.

### 6.5.2 Proof-of-concept IGA botnet data

To emulate the behaviour of an IGA botnet, the Bash script available in Listing 6.1 on the facing page is used. The script is run once every day to emulate the behaviour of an IGA botnet consisting of 30 bots among the Telenor customers, each of which contact the botnet CnC infrastructure once a day by means of a single DNS lookup and a TCP flow.

The fixed set of 30 IP addresses used as faked source IP addresses are selected from the various prefixes used by Telenor for customers. For each source IP, a single DNS request is created of type A for the domain bottest.testlab.telenor.dk, giving the response value 192.0.2.3. For each source IP, 512 TCP packets are then created with a random source port, and destination 12.34.56.78:23. This emulates the IGA bot behaviour of decoding the

**Listing 6.1:** *IGA traffic emulation script.*

```
for sourceip in $sourceiplist; do
 dnsflood -n 1 -s $sourceip $botdomain $resolver
 sleep 1s
done
sleep 10s
let sourceport=$RANDOM
for sourceip in $sourceiplist; do
 for ((i=1;i<=512;i++)); do
  sendip -p ipv4 -is $sourceip -p tcp -ts $sourceport -td 23 -tfs 0 12.34.56.78
 done
 let sourceport++
 sleep 1s
done
```

IP address 192.0.2.3 to the plain-text IP address 12.34.56.78, and knowing the destination port number by some other means. The choice 512 packets is made in order to increase the probability that at least one of the packets from each bot is represented in the collected NetFlow records that use a sample rate of 1:512.

As the domain registration process is not scripted, the emulated botnet will always resolve the name bot-test.testlab.telenor.dk. This specific domain is therefore made exempt to the minimization step that removes domains seen the day before. This is implemented in practice by prefixing the domain name with the number of the day of the observation, such as 1.bot-test.testlab.telenor.dk.

### 6.5.3 Detection algorithm parameters

All of the recommended methods for reducing the datasets described in Section 6.4.1 on page 70 are applied. The choices of the detection threshold parameters used for validation are summarized in Table 6.3 on the following page, and where relevant, the choice of each parameter is elaborated in the following paragraphs.

#### 6.5.3.1 *k*

Data is collected in two periods, $p_1$ from 20210318 to 20210321 (4 full days) and $p_2$ from 20210418 to 20210418 (5 full days). Values of k from 3 to 5 are used, always starting at the first day of the period.

| **Metric** | **Symbol** | **Value** |
|---|---|---|
| Number of days observed | $k$ | 3-5 |
| Ratio of retained DNS RRs | $T_{secondlevel}$ | 0,05 |
| Number of whitelisted IPs | $\|W_{cdnip}\|$ | $1{,}3{\cdot}10^6$ |
| Maximum number of bots expected | $T_{maxbots}$ | 500 |
| Maximum delay from DNS request to flow | $T_{delaymax}$ | 10 min |
| Jaccard similarity threshold | $T_{jaccard}$ | 0,24 |
| Minimum number of bots expected | $T_{minbots}$ | 4 |

**Table 6.3:** *Thresholds used for validation.*

#### 6.5.3.2 $T_{secondlevel}$

To white-list frequent second level domain names, the 1500 most popular second-level qnames and the 200 most popular second-level rnames (where rnames and qnames differ) were white-listed. The qname approach causes approximately 95% of all responses to be white-listed. This is approximately equal to removing second-level domains for which there is more than 100k queries per day. The rname approach causes approximately 95% of all RRs to be white-listed, yielding $T_{secondlevel} = 0,05$. This is approximately equal to removing RRs where more than 170k RRs per day contain a particular second level domain.

#### 6.5.3.3 $T_{jaccard}$

For the Jaccard similarity threshold, a value of $T_{jaccard} = 0,24$ is chosen, meaning that at least a fourth of the source IP addresses in two vertices must be common to the two vertices, for the vertices to be considered connected.

### 6.5.4 Results

Seven different result sets are collected and summarized in Table 6.5 on the next page. The different result sets vary in which of the two time periods are used, how many days of data is used. Also, two result sets are using a truncated version of the destination IP addresses (using the /24 and /16 version of the IP address), in order to evaluate the effect of anonymizing the destination IP address. Example reference metrics can be found in Table 6.4 on the facing page.

The full graph for result set 2 is found in Figure 6.2 on page 80. The graph for result set 1 (k=3) is structurally similar. The IGA detection algorithm detects two communities. The community in cluster 1 contains the vertexes representing the 4 domains and 26 of the 30 source IPs used in the emulated IGA botnet. The community in cluster 2 contains the vertices representing

| Metric | Symbol | Count |
|---|---|---|
| Total DNS responses | $\|D_{all}\|$ | $3,64 \cdot 10^9$ |
| Total flows | $\|F_{all}\|$ | $150 \cdot 10^6$ |
| Minimized DNS responses | $\|D_{reduced}\|$ | $631 \cdot 10^3$ |
| Minimized flows | $\|F_{reduced}\|$ | $17,9 \cdot 10^6$ |
| Unnamed flows | $\|F_{unnamed}\|$ | $13,3 \cdot 10^6$ |
| Relevant re-named flows | $\|R_{reduced}\|$ | $33,0 \cdot 10^6$ |

**Table 6.4:** *Example validation data metrics from the first day of the first time period.*

| **Result set** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **Period** | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| **k** | 3 | 4 | 3 | 4 | 5 | 4 | 4 |
| **Destination IP** | /32 | /32 | /32 | /32 | /32 | /24 | /16 |
| **Vertices** | 333 | 395 | 530 | 691 | 853 | 1037 | 598 |
| **k-cliques** | 4k | 24k | 16k | 49k | 111k | 60k | 22k |
| **k-communities** | 6 | 2 | 9 | 5 | 18 | 22 | 8 |

**Table 6.5:** *Validation data detection results.*

124 domains (for example ecy.eu, rfn.de, rae.biz, pms.mx, egln.vg, pbp.ru) and 7 source IPs. Although other clusters exist, they are not 4-communities.

The graphs for result set 3-7, which are all from the same time period, are structurally similar to each other. They all depict the IGA botnet as a separate cluster, less than 10 smaller, non-community clusters, and finally one very large cluster containing the remaining vertices and communities.

### 6.5.5 Discussion

For all result sets, the IGA detection algorithm successfully detects the k-community (containing a single k-clique) with the bot-test.testlab.telenor.dk domains that represent the emulated botnet (cluster 1 in Figure 6.2). However, additional communities are also detected in all result sets, showing that further data processing is needed. Although this does indicate a high false positive rate for the detection algorithm, we still consider the detection algorithm successful, as it reduces a nationwide traffic data set to a manageable number of 6-22 positives.

As can be deduced from Figure 6.2 that depicts k=4, using k=2 would provide many false positives. Using k=3 results in six 3-communities and using k=4 results in two 4-communities, which we consider a quite low number given the size of the observed network. As expected, the number of vertices and cliques seem to grow when data from additional days is used. Interestingly, the number of communities detected is lower for $k = 4$ than for $k = 3$

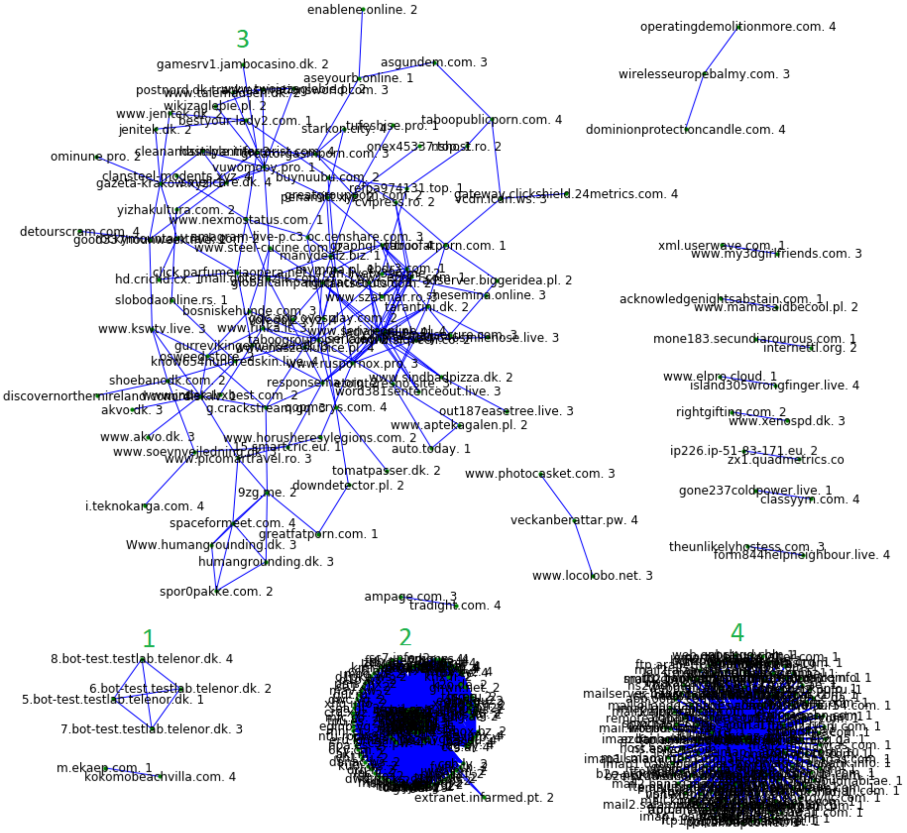

**Fig. 6.2:** *A graph depicting the results of the IGA detection algorithm for result set 2, using $p_1$, $k = 4$ and no destination IP address truncation. Green numbers identify clusters for reference.*

or $k = 5$. This could indicate a sweet spot in the balance of too little or too much data.

Anonymizing the destination IP address by removing the last octet yields result set 6 and removing the two last octets as recommended by [90] yields result set 7. In both cases, the emulated botnet is identified as a k-community and as a distinct cluster. This, combined with the total number of k-communities still being relatively low, could indicate that the IGA algorithm may be feasible to run on anonymized data.

Cluster 2 in Figure 6.2 could be an IGA botnet, however further investigations in this area were inconclusive. A cluster with similar domain names could not be found in period 2.

Some of the non-community clusters, such as cluster 3, include domain names that look like they could be created by a DGA, and these are probably regular (non-IGA) botnets. Although regular bots do not produce unnamed flows, they produce a lot of DNS requests, and these may by random chance be attributed to non-white-listed, unnamed flows towards common destinations. This indicates that the IGA detection could be an novel method for identifying non-IGA botnets as well.

Cluster 4 contains a lot of mailserver-related names. Although this is not a 4-community, it is surprisingly densely connected. By eliminating day 1 and only looking at at dataset for days 2 to 4, this cluster is reduced to a much smaller cluster of 6 vertices. The cluster is not found in the dataset for period 2. Although this cluster could be non-IGA botnet related as well, it could also belong to SMTP mail servers lookup up the IP address of the sending domain in order to verify the sender, verify SPF/DMARC records or similar. As for the non-IGA botnets, this would create a lot of DNS requests that are by random chance attributed to non-white-listed, unnamed flows.

## 6.6 Related work

The related work falls into current and proposed IGA implementations and IGA detection techniques. Both categories have related work focusing on DNS tunnelling techniques, however, these presuppose that the botmaster has control of the DNS server infrastructure, and are therefore incompatible with the threat model of this paper.Similarly, work focusing on DNS-over-TLS or DNSCrypt techniques is not considered relevant. These techniques describe application-layer encryption between the client and the resolver, whereas the IGA technique describes record-level encryption between the client and the botmaster.

### 6.6.1 IGA implementations and encoding schemes

The Sage 2.0 botnet uses a conceptually different, but similarly named IP Generation Algorithm (using the IPGA acronym) to randomly contact CnC servers among 7702 addresses within four predefined /16 subnets [100]. Most of these addresses are expected to be benign, and the actual CnC IP addresses are not conveyed through the DNS system as suggested in this paper. Therefore, the IPGA and suggested IGA techniques differ significantly.

The purpose of Cryptographically Generated Addresses (CGAs) is to generate an IPv6 address with an embedded public key [101]. This scheme transforms the IPv6 address, but does not hide the real IP address. This technique is therefore also of less relevance to our paper.

Much work is available various techniques to anonymize IP addresses [14] [12]. The focus in these papers is typically privacy preservation using one-way mapping functions, such as truncation or hashing. However, the IGA technique requires a two-way mapping as well as an encryption/decryption key, as bots need to be able to obtain the plain-text version of the IP address. This can be achieved using semantic- and format-preserving encryption as described in [93] using cryptographic algorithms as described in [102] and [95]. Preserving semantics is important, as not all possible IP addresses should appear in DNS records. Preserving the format is important, as an encoded IP address must consist of exactly 4 bytes, the length used for IPv4 addresses. If an encoded IP address does not adhere to normal semantics and formats, it will be easy to detect. A different encryption scheme focusing on prefix-preservation is introduced in Crypto-PAN [103], however, the prefix-preservation is an undesired property in our use case, and furthermore the scheme is not semantics-preserving.

A technique for using DNS resolvers as bridge in a two-way communication between two hosts without control of the DNS infrastructure is presented in [104]. However, only clients sharing the same DNS resolver can communicate.

### 6.6.2 IGA detection

To show that a DNS record is implicitly, but not explicitly related to a CnC flow, it is necessary to analyse both DNS and flow data. For this paper, NetFlow/IPFIX data will be used to represent flow data, as this is the simplest, standardized method.

Although commercially available products are difficult to survey due to lack of detailed information, it is appropriate to include for completeness. For NetFlow based detection, Cisco Stealthwatch provides by far the most in-depth documentation of detection capabilities and methods [41]. For DNS based detection, the deepest documentation seems to be provided by Cisco

Umbrella (formerly OpenDNS) [50], Infoblox Advanced DNS protection [52] and HP Arcsight DNS Malware Analytics (formerly Damballa) [54]. None of the surveyed products document the ability to perform behavioural analysis on the combination of DNS and NetFlow features, which suggests that commerciall products will not be able to detect IGAs.

Based on academic survey papers, it seems that combining the DNS and NetFlow feature sets may not be a widespread approach. Some papers use NetFlow analysis on DNS packets only, by simply detecting an abnormal amount of DNS traffic within a specific time period for specific hosts [64] [63]. This is still considered a NetFlow-only based approach not applicable to IGA detection, as no layer 7 information is used from the DNS packets, only general knowledge about the DNS protocol.

IP source address entropy derived from NetFlow logs and the ratio of nxdomain responses derived from DNS logs are used by [65] as detection and validation methods. It is, however, not clear if the two methods are used in combination or independently, thus effectively being a NetFlow based method combined with an DNS based method, rather than a method that combines DNS and NetFlow data before applying the method.

Fuzzy pattern recognition is applied by [67] to both DNS and network flows in a two-stage approach. The DNS related features used are based on the inter-arrival time of requests, and total and failed number of responses. The NetFlow related features used for each destination address are based on the request-response time interval, the number of requests and the payload size. The pattern recognition is then applied in each of these phases, thus analyzing DNS and flow data separately.

IP flows that can not be related to a previous DNS lookup (denoted unnamed flows or non-DNS connections) is one of the topics of [68] and [105]. Such flows account for 5-10% of all internally originated flows in one of the available datasets [68]. As IGA flows will appear unnamed, this property is clearly relevant to exploit in IGA detection.

The topic of [106] is to use traffic analysis on encrypted DNS traffic (DNS-over-TLS etc.) to identify nxdomain response patterns from DGA based botnets. The paper shows that time series analysis and packet size diversity can be used to create IoCs for several specific botnet families. The presented techniques could make it possible to extend the threat model of our paper to allow encrypted DNS traffic as well.

## 6.7 Conclusion

This paper presents the novel concept of the IP Generation Algorithm (IGA) as a method usable by botmasters to avoid exposing the CnC IP address in plain text in DNS A records. An implementation of the concept is provided, and a detection method is presented and validated using an emulated botnet and data from Telenor Denmark's network. Although the results do not indicate that any botnets currently use the IGA method, the method could in the future potentially supplement or replace existing DGA and fast-flux methods.

Modifications to the detection algorithm, or entirely different detection algorithms, suitable for real-time threat prevention by firewalls should be developed, as the method outlined in this paper is very reactive, as it requires several days of retained data for detection.

Looking further into the detecting and eliminating potential decoy flows or applying existing DGA detection methods on top of the detection method described in this paper could potentially reduce the number of false positives, and could therefore also be interesting topics to address in future work.

CHAPTER 7

# Paper D

# A study on the use of 3rd party DNS resolvers for malware filtering or censorship circumvention

**Main author:**
Martin Fejrskov
*Technology, IP Network and Core*
*Telenor A/S*
Aalborg, Denmark
mfea@telenor.dk

**Co-authors:**
Emmanouil Vasilomanolakis
*Cyber Security Group,*
*Aalborg University*
Copenhagen, Denmark
emv@es.aau.dk

Jens Myrup Pedersen
*Cyber Security Group,*
*Aalborg University*
Copenhagen, Denmark
jens@es.aau.dk

**Accepted at:**
*International Conference on ICT Systems Security and Privacy Protection (IFIP SEC) 2022*

Funded by Telenor A/S and Innovation Fund Denmark.

# Abstract

*DNS resolvers perform the essential role of translating domain names into IP addresses. The default DNS resolver offered by an Internet Service Provider (ISP) can be undesirable for a number of reasons such as censorship, lack of malware filtering options and low service quality. In this paper, we propose a novel method for estimating the amount of DNS traffic directed at non-ISP resolvers by using DNS and NetFlow data from an ISP. This method is extended to also estimate the amount of DNS traffic towards resolvers that offer malware filtering or parental control functionality. Finally, we propose a novel method for estimating the amount of DNS traffic at non-ISP resolvers that would have been censored by ISP resolvers. The results of applying these methods on an ISP dataset shows to which extent 3rd party resolvers are chosen by users for either malware filtering or censorship circumvention purposes.*

**Keywords:** *DNS · NetFlow · resolver · ISP · filtering · censorship*

## 7.1 Introduction

The DNS resolver service has traditionally been provided to customers by Internet Service Providers (ISPs). Recently, providers of public DNS resolver services, such as Google and Cloudflare, have gained popularity, and are estimated by Radu et al. to handle more than 50% of all DNS resolutions globally [107]. Although Radu et al. discuss the possible reasons users can have for choosing public DNS services, the authors remain at speculations on this topic.

Some equipment vendors (e.g. webcams) use 3rd party DNS resolvers as a default setting in products. Three main reasons for a user to *actively* choose a 3rd party DNS resolver are presented by web pages containing security advice:

- Service quality: Speed, reliability, and basic security features such as DNS-over-TLS (DoT), DNS-over-HTTPs (DoH) and DNSSEC validation.
- Privacy: Adherence to more strict privacy principles and no modification of the responses, for example to inject ads in NXDOMAIN responses [108].
- Filtering/censoring: The 3rd party provider does not follow government orders to censor responses. Conversely, the 3rd party provider may offer filtering of domains related to malware, porn, drugs, etc. as an add-on service.

As ISPs can deploy resolvers topologically closer to the end users than any 3rd party resolver, an ISP will always be able to offer a faster resolver service than any 3rd party resolver. As a fast DNS resolution can make an Internet connection appear faster, this represents a competitive advantage to an ISP. A competitive ISP can therefore be assumed to offer DNS resolvers with good service quality (although examples of ISPs not having this focus do exist [109]). European Union legislation forbids ISPs to collect personal information, and forbids ISPs to modify DNS responses for ad injection. Therefore a rational customer at a competitive, European ISP should not be inclined to use service quality or privacy as the main reason for choosing a 3rd party DNS resolver.

Following the arguments presented above, and assuming a rational customer and a competitive, European ISP, only the third category, filtering/-censoring, is relevant, which will therefore be the focus of this paper. We recognize that there can be a difference between perceived privacy and actual privacy, as well as a difference between perceived and actual service quality, however we consider this topic out of scope of our paper. The contribution of the paper is the methods and measurements needed to answer the following research questions:

- RQ1: To which extent are 3rd party resolvers used compared to the default ISP resolvers?
- RQ2: To which extent are 3rd party resolvers that offer malware filtering or parental control used?
- RQ3: To which extent are 3rd party resolvers used to circumvent censorship?

These methods and associated results can be relevant for ISPs to assess the business case for offering DNS based filtering services. The results can also be relevant to regulatory bodies to assess the effect of DNS based censorship.

Section 7.2 introduces related work and other background information. The three following sections (7.3, 7.4 and 7.5) each answer one of the research questions outlined above. Section 7.6 summarizes the answers and concludes the paper.

## 7.2 Background and related work

### 7.2.1 Data availability

The simplest way to examine how much and which DNS traffic is directed at 3rd party resolvers is to ask the operators of those services. The privacy policies of the five major public DNS resolver providers (according to Radu et al.) reveal that the providers store data that could answer the question in either anonymized or non-anonymized form, however, they are generally not willing to share the data [110–114]. Another approach is to collect data by interacting with user equipment. One example is the use of apps as probes by the Open Observatory of Network Interference (OONI) project. A second example is the use of advertisement campaigns (or similar mechanisms) that trigger a resolver to query observer-controlled authoritative servers [115]. These approaches can measure which resolvers are used relative to other resolvers, but do not quantify the amount of traffic from each client towards each resolver, which is the purpose of our paper.

Although ISPs are not legally allowed to inspect the DNS traffic to 3rd party resolvers, Fejrskov et al. describe that DNS data from the ISPs own resolvers as well as sampled NetFlow data (that includes 3rd party resolver traffic) can be used in anonymized form even when considering European Union legislation [90]. In our paper the ISP approach is adopted, and data from Telenor Denmark, a national ISP in Europe with 1,5M mobile and 100k broadband subscriptions, is used. Their DNS resolvers adhere to the service quality and privacy criteria mentioned in the introduction, and provide no add-on block offerings.

### 7.2.2 Estimating DNS traffic based on NetFlow data

Konopa et al. suggest a method to detect DoH traffic based on NetFlow records [116]. However, the method relies on access to unsampled NetFlow records which is not available in our paper. Although some papers discuss using NetFlow to identify specific applications, we are not aware of any other papers that directly focus on estimating the amount of DNS traffic. An intermediate step is to use the NetFlow records to estimate the actual number of UDP or TCP flows, a technique often referred to as flow inversion. Several papers, most recently [117], estimate the flow size distribution using various sampling methods, different traffic models, and uses different information from the sampled packets, such as the presence of TCP SYN packets and sequence numbers. Duffield et al. describe and validate a simpler technique that estimates the actual amount of TCP flows as the multiplication of the sample rate and the observed number of flows for which the initial SYN packet was observed [91]. Neither paper present any methods that are applicable to this paper for estimating the amount of UDP flows.

### 7.2.3 DNS Response manipulation

Several studies characterize the use of response manipulation in resolvers [118–120], including both filtering, censoring, injection, etc. Most papers consider response manipulation as an undesired feature as opposed to something positive that the user has actively chosen to gain features such as malware protection. In all papers, the characterization of servers is based on whether or not the server actually performs response manipulation, independently of whether it is advertised or not. In our paper, we therefore find it interesting to characterize resolvers based on whether they advertise themselves as filtering or not, in order to investigate to which extent such functionality is desirable by users.

### 7.2.4 Censorship and circumvention detection

The legislation in Denmark requires ISPs to perform DNS based blocking of certain domains in 7 different categories [121]. In our paper, all categories are included with no distinction between them, giving a total of approximately 800 domains that have a DNS A record. The legislation (and following public discussion) is about blocking web pages, and DNS is seen as the tool that can implement this [122].

Related work on censorship fall in four categories: Techniques for implementing censorship, detecting censorship, circumventing censorship, and measure circumvention attempts. Only the last category is relevant to this paper, and this seems to be the topic of only a few papers. Three of these focus on the use of specific tools or apps like TOR [123], an app for changing DNS resolver [124], and on the use of DNS servers owned by VPN providers [125]. Our focus is only on circumvention that involves the use of 3rd party resolvers, not on specific tools.

The Danish Rights Alliance, an organisation focusing on copyright and other conditions for content creators, measures the effect of DNS based blocking by analysing web site visits [126]. They concluded that the effect of blocking a specific site through DNS blocking reduces the number of visits to the specific site by up to 75% after 4-5 months. In our paper, it is not a requirement that the censored sites consent to embedding code in their web page that measure usage statistics, and the focus is not limited to copyright.

Callejo et al. conclude that 13% of the global DNS queries are resolved by 3rd party rather than by ISP-provided DNS resolvers [115]. They also conclude that the use of 3rd party providers is more frequent in countries with a high level of censorship (a poor rating by the Reporters Without Borders' (RWB) World Press Freedom Index). Their approach relies on serving ads through browsers, and for the reasons mentioned initially in this section, the approach is not applicable for our paper. However, they conclude that the use of 3rd party resolvers in countries rated as Good by RWB is around 7-11% of the total traffic, which is an interesting figure to compare to our results.

## 7.3 Prevalence of 3rd party resolvers

This section presents a method for estimating the number of DNS responses represented by a set of sampled NetFlow records towards 3rd party DNS resolvers. The method consists of three steps that are described in more detail in the following three subsections. The number of 3rd party DNS responses is compared to the number of responses served by Telenor Denmark's DNS resolvers to answer the first question (RQ1) posed in the introduction.

Four different DNS traffic types are considered in this section: DNS over UDP and TCP, DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH). As DNS requests can potentially be malformed, and as only requests that result in a response are relevant from a user perspective, this study will focus on the number of responses rather than the number of requests.

### 7.3.1 Identifying relevant Netflow records

The first step is to identify the NetFlow records that represent 3rd party DNS resolver traffic. In this paper, it is a precondition that the available NetFlow records represent a view of all flows crossing a well-defined network boundary. Users and the default DNS resolvers are defined to be on the internal of the network boundary, 3rd party resolvers and other servers are defined to be on the external side. The NetFlows are considered sampled with a rate of 1:$Q$.

For an external IP address to be considered a potential 3rd party DNS resolver, and to filter away irregular and irrelevant traffic such as that originating from DDoS attacks and port scanning, some TCP or UDP traffic must be observed on port 53 or 853 in both directions, that is both to and from the server. However, due to the use of sampled NetFlow, observing records that form a bidirectional flow is not required, as both directions of the same flow will rarely be sampled given a high sample rate. TCP and DoT records originating from the potential resolver IP must report a packet size of at least 54 bytes to ensure that the response is at least large enough to contain a valid IP, TCP and DNS header. Therefore, packets only containing, for example, a TCP Reset flag indicating that no service is available do not qualify. This packet size criterion is not necessary for UDP based flows, as a server with no UDP service will respond with an ICMP packet instead of a UDP packet.

TCP port 443 traffic towards the resolvers outlined above is considered DoH traffic. We recognize that operators could run both DoH and Web services on the same IP address, and therefore the amount of DoH traffic estimated using this method should be considered as an upper bound rather than an exact number.

Traffic towards authoritative servers also satisfies the aforementioned criteria for a potential resolver, and these flows must be disregarded. Any of the following criteria are used to identify authoritative server IPs:

- The server returns an error code when resolving a well-known domain name, but answers succesfully when resolving the domain name found in the server's reverse/pointer (PTR) record.
- The IP address of the server is identical to any IP address with which the default resolvers communicate.
- The PTR record of the server IP reveals that the server is a well-known authoritative server, such as the DNS root servers or the authoritative servers of major commercial DNS providers.

As a result of the selection process described above, $N$ NetFlow records are considered to represent user-initiated traffic to/from 3rd party resolvers, and only these records are considered for further analysis.

### 7.3.2 Average number of flows per Netflow record

Having identified a number of NetFlow records that represent a number of observed flows towards 3rd party resolvers, the next step is to estimate the number of actual flows. This requires different approaches for TCP and UDP traffic.

As outlined in Section 7.2, the estimated number of actual TCP flows, $\hat{F}_{TCP}$, can be found by multiplying the NetFlow sample rate with the number of flows in which a SYN packet is observed, $\hat{F}_{TCP} = Q \cdot F_{SYN}$. The number of observed SYN flows, $F_{SYN}$, is determined by aggregating the observed response SYN records, $N_{SYN}$, by the 6-tuple of observed flow start time, source and destination IP address, source and destination port number and protocol. For a $Q$ much larger than the expected number of packets in a TCP flow, it is only expected that each TCP flow is sampled once, and in that case $\hat{F}_{SYN} = N_{SYN}$, which is demonstrated as a valid practice in Section 7.3.5.

To estimate the number of actual UDP flows, we use the property that a DNS request or response is always contained within a single UDP packet, and the property that a new UDP flow is made for each request due to the prevalence of source port randomization [127]. In other words, one UDP NetFlow record represents one flow and one DNS response. Therefore, the estimated number of UDP flows, $\hat{F}_{UDP}$, is given by the number of observed UDP response records multiplied by the NetFlow sample rate, $\hat{F}_{UDP} = Q \cdot N_{UDP}$. Note that although a response is always contained within a single UDP packet, this packet may be split into several IP packets due to fragmentation. In this case, only the first IP packet will contain UDP headers, and therefore only the first packet will be considered a UDP packet by the NetFlow emitting router. Therefore, the assumption of a one-to-one relation between DNS responses and UDP packets should be considered valid when using NetFlow as measurement method.

### 7.3.3 Average number of DNS responses per flow

Having estimated the number of actual TCP/UDP flows represented by NetFlow records, the next step is to identify the number of DNS responses per flow. For this purpose, it is assumed that the average number of responses per TCP flow for 3rd party resolvers and for the default resolvers are similar, that the average number of responses per DoT flow for 3rd party resolvers and for the default resolvers are similar, and that these numbers can be calculated from the collected data from the default resolvers. Different collection methods will allow for different methods for calculating the numbers, and the method described below reflects an approach applicable to our data set.

To estimate the average number of responses per TCP/DoT/DoH session, DNS response data from the default resolvers that include the ports of the response is used. The minimum time between flow closure and the allowed reuse of the related source port from the same request source IP address is denoted $t_{graceperiod}$. The longest allowed time for a TCP session to be open is denoted $t_{maxsessionlength}$, and therefore should be true that $t_{maxsessionlength} > t_{graceperiod}$. A response, $c$ is considered belonging to the same flow as another response $b$, if the two responses are less than $t_{graceperiod}$ apart ($t_b + t_{graceperiod} > t_c$), and if the response $c$ and the first response in the flow, $a$, are less than $t_{maxsessionlength}$ apart ($t_a + t_{maxsessionlength} > t_c$).

It should be noted that the specific values of both $t_{maxsessionlength}$ and $t_{graceperiod}$ can differ among clients and servers, as such settings can be either operating system, application or deployment specific. The choice of values for these will therefore depend on the specific DNS server software settings.

Using this method to estimate which DNS responses belong to the same flow makes it possible to calculate an estimated, average number of responses per TCP flow, $\hat{R}_{TCP}$, and an estimated, average number of responses per DoT flow, $\hat{R}_{DoT}$. Notice that the similar number for UDP flows, $\hat{R}_{UDP}$, is always 1 for the reasons outlined in Section 7.3.2.

### 7.3.4 Method summary

The number of DNS responses from 3rd party DNS resolvers, $\hat{D}$, is estimated using NetFlow records as

$$\begin{aligned}\hat{D} =& \hat{D}_{UDP} + \hat{D}_{TCP} + \hat{D}_{DoT} + \hat{D}_{DoH}\\ =& \hat{F}_{UDP} \cdot \hat{R}_{UDP} + \hat{F}_{TCP} \cdot \hat{R}_{TCP} + \hat{F}_{DoT} \cdot \hat{R}_{DoT} + \hat{F}_{DoH} \cdot \hat{R}_{DoH}\\ =& Q(N_{UDP} + N_{TCP,SYN} \cdot \hat{R}_{TCP} + N_{DoT,SYN} \cdot \hat{R}_{DoT} + N_{DoH,SYN} \cdot \hat{R}_{DoH})\end{aligned}$$

| Metric | Symbol | Count |
|---|---|---|
| Total NetFlow records | $n$ | $2,75 \cdot 10^9$ |
| Relevant NetFlow records | $N$ | $3,32 \cdot 10^6$ |
| NetFlow UDP records | $N_{UDP}$ | $2,85 \cdot 10^6$ |
| NetFlow TCP SYN records | $N_{TCP,SYN}$ | $98,9 \cdot 10^3$ |
| NetFlow TCP SYN flow | $\hat{F}_{TCP,SYN}$ | $98,5 \cdot 10^3$ |
| NetFlow DoT SYN records | $N_{DoT,SYN}$ | $12,6 \cdot 10^3$ |
| NetFlow DoT SYN flow | $\hat{F}_{DoT,SYN}$ | $12,6 \cdot 10^3$ |
| NetFlow DoH SYN records | $N_{DoH,SYN}$ | $15,9 \cdot 10^3$ |
| NetFlow DoH SYN flow | $\hat{F}_{DoH,SYN}$ | $15,9 \cdot 10^3$ |
| Max TCP session length | $t_{maxsessionlength}$ | 100 s |
| TCP source port grace period | $t_{graceperiod}$ | 30 s |
| DNS responses per TCP flow | $\hat{R}_{TCP}$ | 1,19 |
| DNS responses per DoT flow | $\hat{R}_{DoT}$ | 11,3 |

**Table 7.1:** *Metrics for 3rd party DNS resolver traffic estimation.*

for a large NetFlow sample rate $Q$, the number of relevant UDP NetFlow records, $N_{UDP}$, the number of relevant NetFlow records observing a SYN packet, $N_{TCP,SYN}$, $N_{DoT,SYN}$ and $N_{DoH,SYN}$, and the estimated, average number of DNS responses per TCP/DoT/DoH flow, $\hat{R}_{TCP}$, $\hat{R}_{DoT}$ and $\hat{R}_{DoH}$.

### 7.3.5 Measurements and discussion

Anonymized DNS and NetFlow data collected over a period of 4 days (covering both weekdays and weekend) from 2021-08-08 to 2021-08-11 from Telenor Denmark's network is used to demonstrate the use of the estimation method elaborated in the previous section. The DNS data is derived from the response packets for all DNS queries towards the default DNS resolvers. The NetFlow data is derived from traffic passing the BGP AS border with sample rate $Q = 512$. Metrics are summarized in Table 7.1. Although the data set only contains 4 days of data, we consider it to be representative, as DNS services are used on a daily basis, and as the amount of users is large ( 1,6M). The internal IP addresses in the data are anonymized by truncation to a /24 prefix, and the AM/PM information of the timestamps is truncated as suggested by Fejrskov et al. [90].

The NetFlow sample rate, $Q$=512, is higher than the expected number of packets in a DNS TCP flow. Therefore the number of observed flows is almost identical to the number of NetFlow records ($\hat{F}_{TCP,SYN} \approx N_{TCP,SYN}$ and $\hat{F}_{DoT,SYN} \approx N_{DoT,SYN}$) as anticipated in Section 7.3.2.

232 NetFlow records relating to UDP traffic on port 853 were observed. This could represent DNS-over-DTLS (DNSoD) traffic [128]. Due to the small

|  | UDP | TCP | DoT | DoH | Sum |
|---|---:|---:|---:|---:|---:|
| **Default** | $15,2 \cdot 10^9$<br>87,67% | $10,9 \cdot 10^6$<br>0,06% | $446 \cdot 10^6$<br>2,57% | 0<br>0% | 90,31% |
| **3rd party** | $1,46 \cdot 10^9$<br>8,39% | $60,3 \cdot 10^6$<br>0,35% | $73,2 \cdot 10^6$<br>0,42% | $92,3 \cdot 10^6$<br>0,53% | 9,69% |

**Table 7.2:** *Number of responses observed on the default resolvers and estimated from 3rd party resolvers. Notice that the DoH number should be considered an upper bound.*

amount and the experimental status of the DNSoD standard, we disregard these records.

Moreover, $43,2 \cdot 10^3$ NetFlow records relating to UDP traffic (from port 53) report more than one packet per flow, which seems to contradict the assumption of one UDP packet per flow made in Section 7.3.2. Although an experimental IETF RFC from 2016 [129] describes the use of multiple UDP packets for responses, it seems unlikely that this should be implemented in several 3rd party resolvers. We therefore believe that a more plausible explanation is that this is caused by re-transmission of requests and responses. As re-transmissions are of no interest to this paper, a UDP NetFlow record (from port 53) reporting more than one packet will only be counted as one packet, and therefore as one request or response.

The value of $t_{graceperiod}$=30 seconds is chosen to match the default tcp-idle-timeout value of the Bind software running on the default DNS resolvers. The value of $t_{maxsessionlength}$=100 seconds is chosen arbitrarily to a value larger than $t_{graceperiod}$. Experiments show that choosing a significantly higher value, $t_{maxsessionlength}$=1000 seconds, does not change the estimated average number of requests per flow significantly.

The estimated 3rd party DNS resolver traffic is summarized in Table 7.2 in comparison to the amount of traffic at Telenor Denmark's default DNS resolvers. As Telenor Denmark's default DNS resolvers do not offer DoH service, the 3rd party DoH number is calculated by assuming that $\hat{R}_{DoH} = \hat{R}_{DoT}$.

Note that the estimated number of DNS responses from 3rd party resolvers listed in Table 7.2 also include responses for servers that could not be explicitly identified as either authoritative or resolving. This is applicable to approximately 0,79% of the listed responses from 3rd party resolvers.

Some customers use VPN services for connecting to their employer's VPN gateway or for keeping the traffic private. We consider it most likely that such traffic will use the 3rd party resolvers operated by the VPN gateway operator, that this operator is located outside Telenor Denmark's network, and that the DNS traffic is therefore not visible in the data set used for this study. Although a study of how widespread the use of VPN services is could be interesting, we consider it complementary to the scope of this paper.

The first question posed in the introduction (RQ1) asks to which extent the DNS traffic is directed at 3rd party resolvers. In Table 7.2 it can be seen that the fraction of the total DNS traffic that is directed at 3rd party resolvers is estimated to be between 9,69-0,79=8,90% and 9,69%. These results are in line with the 7-11% measured by Callejo et al. [115].

## 7.4 Prevalence of filtering 3rd party resolvers

The second research question (RQ2) asks to which extent 3rd party resolvers that offer desirable filtering services (such as malware filtering or parental control features) are used. In this section, the data presented in Section 7.3.5 is further enriched by adding information about which organisation runs the resolver, whether the resolver is public or private, and whether or not the resolvers are advertised by the owners as filtering.

### 7.4.1 Method

To identify if a 3rd party resolver is private or public, two methods are used:

- The resolver is queried with a popular domain name. If this query returns the correct result, the resolver is considered public. If no response is received, the server is considered private.
- If the owner of the resolver is known to only run private resolvers, the resolver is marked as private. These include the resolvers of other ISPs, some VPN services, as well as commercial DNS resolver companies known for only providing private services.

To identify the owner of a resolver, simple methods such as resolving the PTR record of the server, performing a Google or Whois search, are used.The owner's web page is then used to determine if the resolver offers filtering functionality.

Some DNS resolvers exist with the purpose of enabling the user to circumvent some restrictions put in place by web site owners, such as enabling the user to view TV shows that are only broadcasted in some countries due to copyright restrictions. Some, but not all, of these resolvers are associated with VPN services. For the purpose of this paper, we consider these as non-filtering resolvers, as actively choosing these resolvers is conceptually more similar to trying to avoid censorship, than to desire additional filtering.

Another category of resolvers are those that are associated with DNS hijacking malware that changes the DNS resolver settings on a device to point to a resolver under control of a malicious party. This resolver will then most likely manipulate the DNS response to achieve the purpose of the malicious

| | Public | Private | Unknown | Sum |
|---|---|---|---|---|
| **Filtering** | $202 \cdot 10^6$<br>12,02% | $6,41 \cdot 10^6$<br>0,38% | $101 \cdot 10^3$<br>0,01% | 12,41% |
| **Non-filtering** | $1,37 \cdot 10^9$<br>81,11% | $53,5 \cdot 10^6$<br>3,18% | $204 \cdot 10^3$<br>0,01% | 84,30% |
| **Unknown** | $16,0 \cdot 10^6$<br>0,95% | $26,4 \cdot 10^6$<br>1,57% | $12,9 \cdot 10^6$<br>0,77% | 3,29% |

**Table 7.3:** *Categorization of 3rd party DNS responses.*

actor. For the purpose of this paper, we consider these resolvers non-filtering, as they are unlikely to perform any kind of filtering that is considered desirable by the user.

### 7.4.2 Measurements and discussion

The result of identifying server owner, advertised filtering features and private/public category is summarized in Table 7.3. Unknown filtering status represents that we were not able to identify the owner/operator of the resolver. Unknown public/private status is typically caused by the server sending back a wrong answer or an error, such as REFUSED, NXDOMAIN or SERVFAIL.

A key finding is that between 12,41% and 12,41+3,29 =15,70% of traffic for 3rd party resolvers is for filtering resolvers. This suggests that malware filtering, etc., is not likely to be the primary motivation for using 3rd party resolvers.

In Section 7.3.5 on page 93, it was concluded that the amount of 3rd party resolver responses is between 8,90% and 9,69% of all responses. In other words, the total fraction of responses that originate from filtering DNS resolvers is between $8,90\% \cdot 12,41\% = 1,10\%$ and $9,69\% \cdot 15,70\% = 1,52\%$, which answers the second research question. This shows that the use of filtering resolvers is not prevalent among Telenor Denmark's customers.

## 7.5 Censorship avoidance detection

The third question posed in the introduction (RQ3) asks if 3rd party resolvers are used to circumvent censorship. It is a prerequisite that the ISP's default DNS servers censor some domains based on national legal requirements, and that these are not censored by 3rd party resolvers. This section presents a method that uses ISP data to estimate how many DNS responses for censored domains are sent by 3rd party resolvers, and the results obtained by applying the method.

|  |  |  |  | **Default** | **3rd p.** | **None** |
|---|---|---|---|---|---|---|
| **W** | Tainted | Shared | Cens. dom. | $W_1 = \emptyset$ | $W_5$ | $W_9 = \emptyset$ |
|  |  |  | Non-cens. dom. | $W_2$ | $W_6$ | $W_{10}$ |
|  |  | Non-Shared | Cens. dom. | $W_3 = \emptyset$ | $W_7$ | $W_{11} = \emptyset$ |
|  | Non-Tainted |  | (Non-cens. dom.) | $W_4$ | $W_8$ | $W_{12}$ |

**Table 7.4:** *Categorization of the set of all web flows, W.*

### 7.5.1 Method

As elaborated in Section 7.2 on page 87, the censorship focuses on web domain names, and in contrast to the two previous sections that considered flows related to DNS servers, this section focuses on flows related to web servers only.

The core idea of the estimation method is to categorize the web flows seen in NetFlow records, use this categorization to estimate the fraction of the web flows that are towards censored sites, and then use this number of web flows to estimate the number of related DNS queries at 3rd party resolvers for censored domains. The categorization of flows is illustrated in Table 7.4 and elaborated in the following paragraphs. The lowercase $w_1$ to $w_{12}$ represent the count of the flows within each category, and the uppercase $W_1$ to $W_{12}$ represents the sets of flows within each category.

The (uncensored) A records of all the censored domains contain a number of IP addresses, which will be referred to as tainted IP addresses. Some of the tainted IP addresses are assigned to servers that serve both censored and non-censored domains, and these addresses will be referred to as shared IP addresses. Flows relating to these servers are in categories $W_1$,$W_2$,$W_5$,$W_6$,$W_9$ and $W_{10}$). Conversely, some servers with tainted IP addresses only serve censored domains (no non-censored domains), and the IP addresses of these servers are referred to as non-shared IPs. Flows relating to these servers are in categories $W_3$,$W_7$ and $W_{11}$. Finally, the web flows that do not relate to any server IP found in the A record of any censored domain are referred to as non-tainted (categories $W_4$,$W_8$ and $W_{18}$). Some web flows are created following a DNS lookup at the default resolver (categories $W_1$ to $W_4$ in Table 7.4), some web flows are created following a DNS lookup at a 3rd party resolver ($W_5$ to $W_8$), and some web flows are created without any preceding DNS lookup ($W_9$ to $W_{12}$).

As queries for censored domains towards the default DNS server result in a censored response, such queries will not cause a subsequent flow to be created to the web server, therefore by definition $W_1 = \emptyset$ and $W_3 = \emptyset$. As the censoring is based on domain names only, we find it reasonable to assume that flows towards censored sites must be preceded by a DNS lookup, therefore in addition $W_9 = \emptyset$ and $W_{11} = \emptyset$. The number of flows towards censored sites created after a DNS lookup to a 3rd party resolver would be $w_5 + w_7$, and this is the interesting number to estimate.

By definition all web servers are located on the outside of the NetFlow boundary, and all clients on the inside of the NetFlow boundary. The set of relevant flows, $W$, is found using two criteria: First, only records relating to server TCP/UDP port 80 or 443 are considered. Second, only servers for which traffic both from and to the server is observed are considered, although the to/from traffic can relate to different flows to mitigate the effects of NetFlow sampling, following the same arguments as for DNS flows in Section 7.3.1. Flows are thereafter defined by aggregating NetFlow records by 5-tuple on a daily basis, and timestamped with the earliest timestamp on that day.

To estimate $w_5 + w_7$, the following steps are needed. Please refer to Table 7.4 for an overview of the different flow categories. An initial step is to identify the set of tainted and the set of shared IP addresses:

- $T_{ip}$: Let $T_{ip}$, the set of tainted IPs, be the set of DNS A record IPs returned by doing a DNS lookup towards a non-censoring DNS resolver of all the censored domains.

- $S_{ip}$: Let $R_{ip}$ denote the set of IP addresses found in the Rdata field of A records of all responses from the default resolvers. As this because of the censoring will not include any non-shared IPs, $R_{ip}$ thus contains all the non-tainted and all the shared IP addresses. The set of shared IP addresses, $S_{ip}$, can then be found as the subset of the tainted addresses, $T_{ip}$, that are also found in $R_{ip}$, $S_{ip} = T_{ip} \ltimes R_{ip}$.

These two IP address sets are then used split the full set of web flows $W$ into sets of tainted, non-tainted, shared and non-shared flows corresponding to the four main categories $(T, NT, S, NS)$ in Table 7.4 on the previous page:

- $T$ and $NT$: Split the full set of flows, $W$, into the set of tainted flows $T = W_1 \cup W_2 \cup W_3 \cup W_5 \cup W_6 \cup W_7 \cup W_9 \cup W_{10} \cup W_{11}$ and the set of non-tainted flows $NT = W_4 \cup W_8 \cup W_{12}$. These can be determined based on whether or not one of the flow IP addresses can be found in $T_{ip}$ such that $T = W \ltimes T_{ip}, NT = W \triangleright T_{ip}$.

- $S$: Find the set of shared flows, $S = w_1 \cup w_2 \cup w_5 \cup w_6 \cup w_9 \cup w_{10}$. This can be found using $T$ as a tainted flow address is shared, if the server IP can be found in the default DNS responses, $S = T \ltimes S_{ip}$.

- $NS = W_7$: Find the number of non-shared (and by definition, censored) flows preceded by a 3rd party DNS lookup, $W_7$, by finding the total number of non-shared flows, $NS$, and exploiting that that $W_3 = \emptyset$ and $W_{11} = \emptyset$. $NS = W_7$ can be found using $T$ as a tainted flow address is non-shared, if the server IP can not be found in the default DNS responses, $W_7 = NS = T \triangleright S_{ip} = T - S$.

The set of shared flows, $S$, consists of two subsets of flows, related to censored domains, $W5$, and non-censored domains, $W_2 \cup W_6 \cup W_{10}$. The next steps of the method focus on identifying which flows belong to which of these two subsets by various means. For this purpose, the concept of flow renaming will be used several times to determine which web flows are associated with which DNS responses. In our paper, flows and DNS responses are considered associated, if a flow is created no longer than $\theta$ minutes after the DNS lookup, if the client IP addresses match, and if the server IP of the flow is the IP found in the Rdata record of the DNS response. The effect of DNS caching at the user is assumed to be mitigated by the aggregation of flow records to the earliest timestamp during a specific day as mentioned above.

- $W_2$: Find the set of tainted, shared, non-censored flows preceded by a DNS lookup at the default servers, $W_2$. As $W_1 = \emptyset$ and $W_3 = \emptyset$ this can be found by renaming the flows of $S$ by using all entries in the DNS response log, $D$, such that $W_2 = S \ltimes_\theta D$. The same method can in theory be applied to the set of non-tainted flows, $NT$, to find the untainted set $W_4$. However, the amount of data can be large, and the following steps therefore do not depend on the feasibility in practice of using renaming to distinguish between $W_4$ and $W_8 \cup W_{12}$.

- $w_6 + w_{10}$: The fraction of re-nameable flows within the non-tainted flow set and within the non-censored flow set is assumed to be the same, as none of these flows are censored. Therefore, $\frac{w_6+w_{10}}{w_2} = \frac{w_8+w_{12}}{w_4}$, where $w_6 + w_{10}$ is then easily found as $w_2$ is already known. Although $W_4$, $W_8$ and $W_{12}$ cannot be identified (as elaborated above), the ratio $\frac{w_8+w_{12}}{w_4}$ can be found by renaming a sampled set of non-tainted flows, $\frac{w_8+w_{12}}{w_4} = \frac{w_{8_s}+w_{12_s}}{w_{4_s}} = \frac{nt_s-w_{4_s}}{w_{4_s}}$ where $sample(NT) = NT_s = W_{4_s} \cup W_{8_s} \cup W_{12_s}$ , $W_{4_S} = NT_s \ltimes_\theta D$.

- $w_5$: Find the number of shared, censored flows preceded by a 3rd party DNS lookup, $w_5$, by subtraction: $w_5 = s - (w_2 + w_6 + w_{10})$

These steps provide the necessary values to calculate $w_5 + w_7$ which is the estimated number of flows towards censored sites that are associated with a DNS lookup to a 3rd party resolver.

Flow renaming is performed in the steps for finding $W_2$ and $w_6 + w_{10}$, and we consider this mechanism to be the largest cause of uncertainty to the result. The method as used in this paper is greedy in the sense that too many flows will be considered re-nameable and therefore as non-censored, both because flows and DNS responses are considered related based on a time interval (larger time interval is more greedy), but also because user IP addresses are anonymized by truncation. Therefore, the estimated value of $w_5 + w_7$ should be considered as the lower boundary of the real value. As shown in a later subsection, the estimation of the lower boundary instead of the actual value turns out to be a sufficient metric to support our conclusions.

The next step is to calculate the number of estimated, actual DNS responses, $\hat{p}$, that relate to the estimated, observed, flows $w_5 + w_7$. The techniques described in Section 7.2 on page 87 for estimating the actual number of flows based on the observed number of flows are not applicable in this case, as they depend on the availability of NetFlow records and not just the availability of an estimated flow count. Instead, we propose to identify all servers for which only port 80/443 flows are observed, let $w_{web}$ denote the number of flows towards these servers and let $p_{web}$ denote the count of DNS responses with an A record containing the IP addresses of these servers. Then we will estimate the number of DNS responses related to the censored flows as $\hat{p} = \frac{p_{web}}{w_{web}}(w_5 + w_7)$. As the value of $w_5 + w_7$ is considered a lower boundary, the value of $\hat{p}$ should also be considered a lower boundary.

### 7.5.2 Measurements and discussion

The estimation method detailed above is applied to DNS and NetFlow data from Telenor Denmark's network collected over a period of 4 days from 2021-09-23 to 2021-09-26. The most interesting metrics are summarized in Table 7.5. 1:1000 of the non-tainted flows are used to estimate $w_6 + w_{10}$. Results for two different values, $\theta = 1min$ and $\theta = 60min$, of the time interval allowed in the renaming process are presented in order to illustrate the importance of this parameter as discussed above. A $\theta > 60min$ does not give significantly different results.

In summary, we estimate that at least $\hat{p} = 477 \cdot 10^3$ DNS responses for censored domains have been answered by 3rd party DNS resolvers. This number can be compared to the number of censored DNS responses served by the default resolvers, $44,6 \cdot 10^3$, and the ratio between these numbers is $r = 10,7$.

Section 7.3.5 on page 93 concluded that approximately 9% of the total DNS traffic was from 3rd party resolvers. If 3rd party resolvers were not used to circumvent censorship, it would be expected that $r \approx 0,09$. Censored 3rd party resolver responses are therefore at least two orders of magnitude more prevalent than expected, which suggests that 3rd party DNS resolvers are chosen to circumvent censorship. It is more challenging to consider if censorship circumvention is the *primary* reason for a user to choose a 3rd party resolver. Hypothetically, even if this was the only reason for choosing 3rd party resolvers, the number of censored domains would still only be a small fraction of the total responses, as individual users will then also use

| Metric | Symbol | Count | |
|---|---|---|---|
| Relevant flows | $w$ | $1,03 \cdot 10^9$ | |
| Shared flows | $s$ | $196 \cdot 10^3$ | |
| Non-shared flows | $ns = w_7$ | $7,40 \cdot 10^3$ | |
| Ratio of responses and flows | $\frac{p_{web}}{w_{web}}$ | $18,1$ | |
| Censored responses at default DNS resolvers | $d_{censored}$ | $44,6 \cdot 10^3$ | |
| Renaming interval | $\theta$ | 1 min. | 60 min. |
| Shared, non-censored flows preceded by default lookup | $w_2$ | $103 \cdot 10^3$ | $166 \cdot 10^3$ |
| Shared, non-cens. flows not preceded by def. lookup | $w_6 + w_{10}$ | $28,0 \cdot 10^3$ | $11,1 \cdot 10^3$ |
| Shared, censored flows preceded by 3rd party lookup | $w_5$ | $65,5 \cdot 10^3$ | $19,0 \cdot 10^3$ |
| Estimated DNS responses related to censored flows | $\hat{p}$ | $1,32 \cdot 10^6$ | $477 \cdot 10^3$ |
| Ratio of censored responses at default and 3rd party | $r$ | $29,6$ | $10,7$ |

**Table 7.5:** *Metrics for censorship evasion estimation.*

the 3rd party resolver for non-censored domains.

As the number of censored responses from 3rd party servers is only an estimated number, it is not possible to assess how many users resolve censored domains using this method either. Even if this was possible, it would not be meaningful to compare this number of users to the number of users receiving censored responses from the default resolvers, without knowing more about the intentions of these users. One may argue that all of the responses from the default servers are caused by unintentional web page visits that will not be repeated by a user, whereas all the responses from the 3rd party servers could be caused by deliberate web page visits that will most likely be repeated by the user.

Although the results in this paper are based on only a single dataset, we find that the methods are independent of the dataset, and that the temporal length of the dataset is sufficient to present valid results for Telenor Denmark. We fully recognize that using the dataset of another ISP in another country could yield different results, both for technical reasons (such as differences in default DNS resolver setup) and cultural reasons (desire to circumvent censorship etc.).

## 7.6 Conclusion

In this paper we propose a method for estimating the amount of TCP/UDP/DoT/DoH DNS responses by using information from NetFlow records. This method is applied to estimate how much of the DNS traffic in an ISP is from 3rd party resolvers instead of the ISP's default resolvers. Using data from Telenor Denmark it is concluded that 8,9-9,7% of the total DNS traffic is from 3rd party resolvers (RQ1). This result supports and is supported by the most recent related work that uses a completely different method for obtaining the results [115]. Also, it is concluded that 1,1-1,5% of the total DNS traffic is from filtering resolvers (RQ2). Although it is expected that some traffic is from filtering resolvers, the specific number is not quantified by any existing research that we are aware of. The low number suggests that filtering resolvers are not commonly used by Telenor's customers, and this could represent an unexploited business opportunity to promote the use of such services.

Furthermore, we propose a NetFlow based method for estimating the amount of DNS responses from 3rd party resolvers that would have been censored by the ISP's default DNS resolvers. Using data from Telenor Denmark, it is concluded that DNS responses for censored domains are at least two orders of magnitude more prevalent at 3rd party resolvers than at the ISP's default resolvers (RQ3). We are not aware of any related work quantifying this number on an ISP scale. The high number suggests that 3rd party resolvers are actively chosen in order to circumvent censorship, which should be considered when the censorship legislation is up for evaluation.

It is correct that we only rely on a single dataset, however, we believe that the methods are independent of the dataset, and that the single dataset used is sufficiently large to present valid results for the specific ISP. We fully recognize that using the dataset of another ISP in another country could yield different results. This is, however, more likely attributed to cultural differences (knowledge about cyber security in the population, the desire/need to circumvent censorship in a particular country, etc.) rather than the merits of the presented method.

The focus of this paper is purely technical, however for future work it could be interesting to compare the obtained results with a user questionnaire asking for the user's primary motivation for actively choosing 3rd party servers.

Although the specific results presented in this paper applies only to Telenor Denmark's customers, the methods are general, and it is our hope that they will be used by other ISPs and organisations to identify both business opportunities and regulatory challenges.

CHAPTER 8

# Paper E

## Detecting DNS hijacking by using NetFlow data

**Main author:**
Martin Fejrskov
*Technology, IP Network and Core*
*Telenor A/S*
Aalborg, Denmark
mfea@telenor.dk

**Co-authors:**

Jens Myrup Pedersen
*Cyber Security Group*
*Aalborg University*
Aalborg, Denmark
jens@es.aau.dk

Emmanouil Vasilomanolakis
*Cyber Security Group*
*Aalborg University*
Copenhagen, Denmark
emv@es.aau.dk

**Submitted to:**
*International Workshop on Security (IWSEC) 2022*

Funded by Telenor A/S and Innovation Fund Denmark.

## Abstract

*DNS hijacking represents a security threat to users because it enables bypassing existing DNS security measures. Several malware families exploit this by changing the client DNS configuration to point to a malicious DNS resolver. Following the assumption that users will never actively choose to use a resolver that is not well-known, our paper introduces the idea of detecting client-based DNS hijacking by classifying public resolvers based on whether they are well-known or not. Furthermore, we propose to use NetFlow-based features to classify a resolver as well-known or malicious. By characterizing and manually labelling the 405 resolvers seen in four weeks of NetFlow data from a national ISP, we show that classification of both well-known and malicious servers can be made with with an AUROC of 0.85.*

**Keywords:** *NetFlow · IPFix · DNS · hijacking · malware*

## 8.1 Introduction

The integrity protection offered by Domain Name System (DNS) security measures, such as DNS-over-TLS and DNSSec, can be completely circumvented by changing the configuration of DNS clients to use malicious DNS resolvers instead of trustworthy resolvers. This approach has therefore historically been used by several malware families such as DNSChanger, DNSUnlocker, Koobface and others for diverse purposes such as pushing adware, redirecting to phishing or malware web pages, etc. [130] [131] [132]. Although these malware families target Windows machines, taking control of home routers in order to use DHCP to extend the malicious DNS configuration to all devices in a household is also an approach used in practise for example by the GhostDNS malware or in on-premises attacks [133] [134] [135].

The DHCP based approach limits the malware detection options, as typical IoT devices and home routers do not support host-based detection mechanisms such as anti-virus software available for mainstream operating systems. As an alternative to host-based detection, network-based detection mechanisms that work by passively inspecting the payload of the DNS traffic between the home router and 3rd party resolvers could be deployed by an Internet Service Provider (ISP) [120]. This is, however, not legal to implement in the European Union for privacy reasons [8]. For purposes of malware detection, ISPs are only allowed to process data found in customer traffic, if the data is already processed for transmission purposes (such as the information found in NetFlow records), and only if the data is anonymized before processing [90]. NetFlow records are emitted by routers, and typically contain information about the flows observed on a particular router interface, such as timestamp, source/destination IP address, TCP/UDP source/destination ports and similar flow-level information. An anonymized NetFlow based approach is therefore a legally viable option, and this detection approach will therefore be pursued in our paper.

Although many papers analyse the maliciousness of the DNS traffic itself (such as DNS traffic used in DDoS attacks), including some that are based on NetFlow level information [63] [64] [65], we are not aware of any work that only use NetFlow level features to assess whether a resolver performs record manipulation with either benign or malicious intent. Determining maliciousness solely based on NetFlow features makes could present a simpler (and therefore more desirable) option to an ISP, as the ISP would then not need to rely on procuring additional threat intelligence for resolver labelling. This observation provides the base for the first research question (RQ) examined in this paper:

*RQ1: Can public resolvers be correctly classified as either malicious or non-malicious using ISP-level NetFlow data?*

Most users do not know or care about which resolver they use, and as a result, they use the default resolver assigned by equipment manufacturers or ISPs. For this paper it is assumed that if a user (or equipment manufacturer) should actively choose which resolver to use, the user will choose a well-known resolver operator. Well-known resolvers are defined as all public DNS resolvers that are known (through an associated web page or similar) to be run by a publicly known organisation, no matter the amount of filtering or censoring applied for benign/desirable/regulatory purposes. This assumption and definition provides the base for the second research question examined in this paper:

*RQ2: Can public resolvers be correctly classified as either well-known or not using ISP-level NetFlow data?*

RQ1 and RQ2 classify resolvers in one of four classes, depending on whether they are considered well-known or not, and if they are considered malicious or not. This resolver categorization can potentially be used in firewalls by ISPs to black/white-list resolver IPs on behalf of a group of consenting users, or the categorization can be combined with user-specific NetFlow records to discover and notify consenting users of a potential malware infection.

The contributions of the paper are therefore twofold: First, we introduce the concept of using whether a resolver is well-known or not for classification. Second, we show how accurately NetFlow data can be used to identify well-known and/or malicious resolvers.

This remaining part of the paper is organized as follows: Section 8.2 describes the method used to answer the research questions. Section 8.3 shows the result of applying the method, and the results are discussed in Section 8.4. Section 8.5 describes related work and Section 8.6 concludes the paper.

## 8.2 Method

To answer the research questions posed in the introduction, we apply the following four steps, which are described in further details in this section. First, the IP addresses of the DNS servers that are considered public resolvers are identified, and the NetFlow records related to any other IP addresses are discarded. Second, a number of features are extracted from the remaining NetFlow records and auxiliary features such as the DNS PTR records of the resolver IP addresses are added. The third step is to establish a set of labels that are used as ground truth for supervised machine learning. The fourth step is to apply machine learning to show if the classification is feasible, thereby answering the research questions posed in the introduction.

### 8.2.1 Identifying public resolvers

Some DNS servers assume the role of both public resolver and authoritative servers, and for the purpose of this paper, these are considered public resolvers and their authoritative role is ignored. An IP address is considered to host a public resolver if all of the following three criteria are satisfied.

First, NetFlow data relating to port 53 or 853 must show unidirectional TCP or UDP traffic flows both to and from the IP address. Due to sampling it is not a requirement that the unidirectional flows are related. TCP traffic must contain more than 54 bytes. The purpose of these criteria is to eliminate traffic related to port scans, TCP connections/handshakes with no DNS payload, DDoS amplification attacks and other irregular or irrelevant use cases.

Second, a DNS A-type query is issued towards the IP address using both DNS and DNS-over-TLS for a domain name under our control, where the valid response and authoritative servers are therefore known. The response must contain a syntactically valid no-error response record that contains an IP address. The purpose of this is to eliminate private resolvers and authoritative-only servers.

Third, the Recursion Available flag is considered. If the Recursion Available flag is set to False, the IP address in the response must be correct for the server to be considered a resolver. This is done in order to eliminate a number of authoritative-only servers that respond non-authoritatively to queries (for example with an IP address hosting a web page with a "This page does not exist" banner instead of providing an NXDOMAIN response) and to avoid eliminating resolvers that answer with the correct response record, but try to evade detection by setting the RA flag to false.

### 8.2.2 Features used to characterize public resolvers

The features chosen to characterize the public resolvers are listed in Table 8.1 on the following page. Two features warrant further elaboration in the follow paragraphs.

**ResolverPrefix**: During a preliminary data analysis, we found several servers (both benign and malicious) within the same prefix. To exploit this as a feature, we choose to use a /24 prefix as a more narrow feature than the more traditional measures such as geographical location or BGP AS number.

**PtrCategory**: Many benign DNS resolvers have a valid PTR record indicating the role as DNS server. Similarly, many ISPs create a default PTR record for all their customers, that contains the IP address itself. To exploit the PTR record as a feature, the PtrCategory of a server is set to "DNS" if the PTR record contains the words "dns", "ns[1-4]", "ns0[1-2]", "resolver" or starts with "ns.". The PtrCategory is set to "IP" if the record contains four numbers separated by ".". The PtrCategory is "NoPTR" when no PTR record exists, and "Uncategorized" if none of the above applies.

Some features are for various reasons intentionally not used to describe DNS resolvers, such as

- features directly or indirectly controllable by the malicious actor. This includes many NetFlow features such as packet/byte counts, query source port number and TCP vs. UDP. It also includes features found by actively probing the DNS resolver, whether the resolver is also an authoritative server, or whether there are any services available on other TCP/UDP ports on the resolver server.

- features unavailable due to anonymization requirements, such as an mxtoolbox.com lookup of the client IP address. Mxtoolbox.com provides information about whether a particular IP address is listed in popular public/commercial threat intelligence databases.

- features unavailable due to the use of a high sampling rate when creating NetFlow records. This includes features that require that several related flows are all observed in NetFlow records, such as if a flow towards a resolver is preceded by a DNS query towards the ISP's default DNS resolvers, or if a certain sequence of flows is always observed towards certain resolvers.

### 8.2.3 Features used to label public resolvers

Table 8.2 on page 109 lists the labels used as ground truth. Three features warrant further elaboration below.

**AdResponse**: The purpose of some benign DNS resolvers is to remove advertisements. To identify these, the AdResponse feature denotes if an

| Source | Name | Description | Value type |
|---|---|---|---|
| NetFlow | ResolverPrefix | The /24 prefix of the resolver | Integral |
| | ClientCount | Number of unique client /24 IP prefixes seen in NetFlow records related to the resolver. | Integral |
| | DayCount | Number of days in which traffic from/to the resolver is observed | Integral |
| | RecordCount | The $log_{10}$ count of NetFlow records related to the resolver | Integral |
| | RecordCountPerClient | The $log_{10}$ count of NetFlow records related to the resolver divided by the number of clients | Integral |
| Auxillary | PtrCategory | The category of the resolver's PTR record. Feature values: DNS/ IP/ NoPTR/ Uncategorized | Categorical |
| | Qname | True if an A record with the resolver's IP observed by the ISPs own DNS resolvers | Boolean |

**Table 8.1:** *Feature overview. The following features are used to describe each public resolver. The IP address of the resolver identifies the resolver, but is not used as a feature, and is therefore omitted from this list.*

| Source | Name | Description | Value type |
|---|---|---|---|
| Probing | AdResponse | Indicates if the resolver answered with the correct IP address for a number of advertisement hosts. Feature values: Correct/Incorrect/Inconsistent/Malicious | Categorical |
|  | UpdateResponse | Indicates if the resolver answered with the correct IP address for domains hosting software updates. Feature values: Correct/Incorrect/Inconsistent/Malicious | Categorical |
| Auxillary | Blacklisted | Indicates if the resolver IP is listed on a blacklist according to a lookup on mxtoolbox.com (excluding the SpamHaus PBL that simply lists IPs assigned to broadband customers) | Boolean |
|  | Webreference | Indicates if the resolver is referenced on a website. Feature values: Benign/Unknown/Malicious | Categorical |
| Inferred | Wellknown | Indicates if Webreference is Benign or not. Label for RQ2. | Boolean |
|  | Malicious | Indicates if either Blacklisted has value True (and Webreference is not Benign), Webreference has value Malicious, or AdResponse or UpdateResponse has value Malicious. Label for RQ1. | Boolean |

**Table 8.2:** *Label overview. Input from four different features are combined to form the labels used as ground truth for each research question.*

A record query for 6 popular ad hosts[1] return IPs owned by Doubleclick-/Google, as identified by a PTR record ending in "1e100.net.". If the IP contained in any A record is listed on any blacklist on mxtoolbox.com, the AdResponse feature is set to "malicious". If some A records are correct, and some are incorrect (but not blacklisted), the AdResponse feature is set to "inconsistent".

**UpdateResponse**: Malicious DNS resolvers could block access to the update servers of anti-virus products, operating systems or similar, in order to avoid that any updates to these products would trigger a malware detection or detection of a choice of malicious resolver. To identify such resolvers, the same approach as for the AdResponse is used for 8 domains[2].

**Webreference**: This feature indicates the result of a manually performed search for *all* of the IPs identified as resolvers IP and their associated PTR record using Google, the Whois database and various publicly available lists of resolvers IPs. A resolver is classified as Benign if it satisfies the criteria for a well-known resolver as defined in Section 8.1, as Malicious if the search indicates that the IP belongs to a malicious resolver, and as Unknown if the search did not provide any further insight.

The approach for the AdResponse and UpdateResponse features are inspired by Kührer et al. [136]. Although they include more feature categories (without disclosing the exact domains used), we consider AdResponse and UpdateResponse the most relevant to our paper. The specific choice of domain names are based on the market prevalence of the related companies, the company's documentation about which domains are used for the purposes, and the prevalence of the domains as observed in Telenor Denmark's DNS resolvers.

The features AdResponse, UpdateResponse, Blacklisted and Webreference are combined into two binary labels, called Wellknown and Malicious, as elaborated in Table 8.2. The use of the combination of the Blacklisted and Webreference features to construct the Malicous feature is necessary as known resolvers such as CloudFlare's 1.1.1.1 and several DNS resolvers related to VPN services can be found on multiple blacklists. This could be caused by the VPN DNS service being located on the same IP/prefix as the VPN outlet, if any VPN customers are exhibiting malicious behaviour.

[1] ad.doubleclick.net, www.google-analytics.com, googlesyndication.com, googleads.g.doubleclick.net, tpc.googlesyndication.com and pagead2.googlesyndication.com

[2] sadownload.mcafee.com, ncc.avast.com, ds.kaspersky.com, dc1.ksn.kaspersky-labs.com, dci.sophosupd.com, liveupdate.symantec.com, ctldl.windowsupdate.com and download.windowsupdate.com.

### 8.2.4 Algorithm

The features outlined above are both categorical and numerical in nature, and data is labelled and binary, which suggest that a supervised classification algorithm within the class of decision trees such as Random Forest (RF) or Gradient Boosted Trees (GBT) should be the most appropriate. The Area Under Receiver Operating Characteristic (AUROC) is used as hyper-parameter optimization metric through a 5-fold cross-validation using all combinations of three hyper-parameters: Tree depth (5, 10, 15, 20, 30), maximum number of bins (discretize continuous features) (10, 50, 100, 150) and number of trees (10, 20, 30, 40). 80% of the resolvers are used for model training and cross-validation, 20% of are used for test/prediction/evaluation.

The number of indices for the categorical ResolverPrefix feature will probably be close to the number of observed resolvers. To avoid such a large number of indices, and to avoid the large number of feature columns created by a one-hot-encoding, the prefix is converted to its integral representation and considered as a continuous feature instead. As a continuous feature, the ResolverPrefix feature will be binned, making it more likely that numerically close prefixes will be classified similarly by the model. This seems like a reasonable approach given that organisations are typically allocated larger IP prefixes than individual /24 prefixes.

## 8.3 Results

This section describes the results of applying the method described in Section 8.2. In subsection 8.3.1, a description is provided of the data used, as well as how the data is characterized in terms of the features and labels introduced in Section 8.2. In subsection 8.3.2, the results of applying machine learning algorithms to predict labels are presented. The results are discussed in Section 8.4.

### 8.3.1 Data characteristics

The primary data source used in this paper is four weeks of NetFlow data collected from 2021-11-25 to 2021-12-22 with a sample rate of 1:1024 at the Border Gateway Protocol (BGP) Autonomous System (AS) border routers of Telenor Denmark. Telenor Denmark is a national ISP in Europe with 1,5M mobile and 100k broadband subscriptions.

For legal end ethical reasons, the client (Telenor customer) IP is anonymized to a /24 prefix in each NetFlow record before any further processing. Features are extracted at least every 5 days during the collection period, after which all NetFlow records (including the anonymized client IP) are dis-

| **Label** | **Algo** | **Best model** | | | |
|---|---|---|---|---|---|
| | | AUROC | MaxDepth | MaxBins | NumTrees |
| Wellknown | RF | 0,85 | 5 | 50 | 10 |
| | GBT | 0,77 | 5 | 100 | N/A |
| Malicious | RF | 0,83 | 5 | 100 | 30 |
| | GBT | 0,85 | 5 | 100 | N/A |

**Table 8.3:** *Hyperparameters yielding the best AUROC for each label and algorithm.*

carded. Therefore, the resulting dataset used for analysis in this paper does not depend on storing any Personal Identifiable Information (PII) relating to the clients.

The criteria listed in Section 8.2.1 for identifying public resolvers in actual use by Telenor customers are satisfied by 405 IP addresses during the data collection period. Of the 405 resolvers, 62 have the label Malicious set to True, and 259 have the label Wellknown set to True. None of the resolvers have both labels set to True, and we will therefore for the ease of reference refer to the resolvers as being either malicious, wellknown or unknown. Of the 62 malicious resolvers, 5, 2 and 25 are categorized as Malicious in the AdResponse, UpdateResponse or Webreference features, respectively. 35 of the 62 are categorized as True in the Blacklist feature.

Figure 8.3-8.7 in the Appendix illustrate the number of resolvers that are tagged with which label for each of the features used to describe the resolvers (as described in Table 8.1).

### 8.3.2 Label prediction

As outlined in Section 8.2, both the Malicious and Wellknown labels are predicted by either a Random Forest or Gradient Boosted Tree algorithm. The results in this paper are found using the implementation provided by PySpark.

The set of hyperparameters with the highest AUROC for each label and each algorithm are listed in in Table 8.3. Unless noted otherwise, the rest of this paper only presents details relating to the best model found for each label, which is an RF based model for the Wellknown label and a GBT based model for the Malicious label.

The ROC curves can be found in Figure 8.1 and 8.2. These show the True Positive Rate and False Positive Rate at various probability threshold settings.

A confusion matrix for each of the models can be found in Table 8.4 and 8.5. The confusion matrix for the Wellknown label shows how many of the resolvers in the test set were predicted to have label Wellknown set true or false, as compared to the curated label. Both matrices are created using a

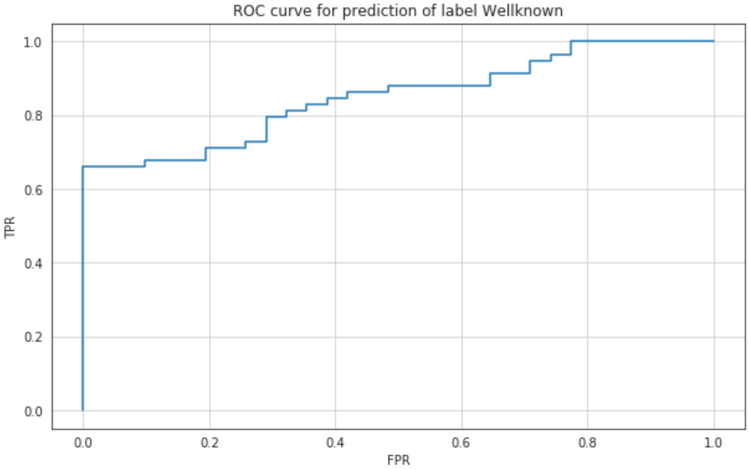

**Fig. 8.1:** *ROC curve for the prediction of the label Wellknown.*

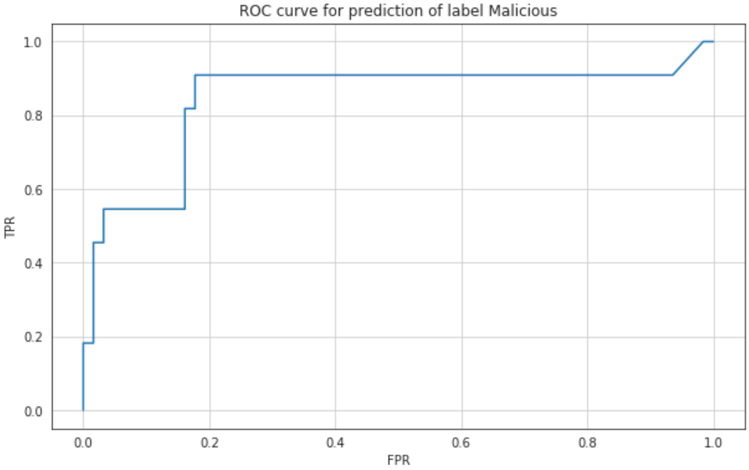

**Fig. 8.2:** *ROC curve for the prediction of the label Malicious.*

| | Predicted True | Predicted False |
|---|---|---|
| **Labelled False** | 15 | 16 |
| **Labelled True** | 51 | 8 |

**Table 8.4:** *Confusion matrix for the prediction of the label Wellknown.*

| | Predicted True | Predicted False |
|---|---|---|
| **Labelled False** | 4 | 58 |
| **Labelled True** | 6 | 5 |

**Table 8.5:** *Confusion matrix for the prediction of the label Malicious.*

probability threshold of 0,5, as this threshold is among the set of threshold values that provide a high F-score. The accuracy ($\frac{TP+TN}{P+N}$) represented by the two matrices is 0,74 and 0,87.

The feature importances for each of the models can be found in Table 8.6. This quantifies the importance of a particular feature as an average across all trees in the model [137].

## 8.4 Discussion

The exclusion of authoritative-only servers based on the Recursion Available flag did not increase the accuracy of the model as much as expected. Running the model training without considering the RA flag yields an AUROC of 0,84 and 0,82 (instead of 0,85) based on 455 servers instead of 405 servers. As the RA value returned by a malicious resolver can be controlled by the malicious actor, it might be a better option not to consider this flag at all.

The AUROC values and the confusion matrices indicate that labels can indeed be predicted based on NetFlow data, although we do not consider the AUROC values high enough for operational/production use. During the

| Feature | Wellknown label | Malicious label |
|---|---|---|
| ResolverPrefix | 0,18 | 0,47 |
| ClientCount | 0,07 | 0,06 |
| DayCount | 0,17 | 0,09 |
| RecordCount | 0,24 | 0,10 |
| RecordCountPerClient | 0,15 | 0,13 |
| PtrCategory | 0,19 | 0,13 |
| Qname | 0,01 | 0,02 |

**Table 8.6:** *Feature importances.*

data analysis we observed that the AUROC values of 0,85 can vary depending on the specific sampling in the training/test data split. This variance is not systematically analyzed, however, the reported value of approximately 0,85 seems to appear often, but values as low as 0,80 and as high as 0,87 have also been observed.

Neither label is well balanced (259 well-known resolvers and 62 malicious resolvers from a total of 405 resolvers) and therefore the class imbalance problem needs to be considered. For this purpose we repeated the model training with undersampling, and results indicated that this yielded slightly lower AUROC values (in the range of 0,78 to 0,82) for both labels and both algorithms. It seems reasonable to assume that this is caused by the relatively small dataset available for training. It could therefore be interesting to repeat the experiment on an even larger dataset, preferably with a larger fraction of of malicious resolvers.

The feature importances listed in Table 8.6 show that most features contribute to the model. The Qname feature, indicating if the ISP's resolvers have seen the public resolvers IP address in a DNS response record, is the least significant feature, also when training with undersampling. This is surprising given that Figure 8.6 indicates that most well-known resolvers have Qname=True, and most malicious resolvers have Qname=False. We have no credible explanation for this discrepancy.

## 8.5 Related work

The general topic of DNS hijacking can be be split into 4 different subtopics, where the hijacking is implemented by manipulating response records in (1) resolvers and forwarders, (2) middleboxes such as firewalls, (3) authoritative name servers, or (4) by manipulating the DNS resolver IP address configuration on client devices to direct DNS traffic to malicious resolvers [138]. As outlined in the introduction, the focus of our paper is restricted to *malicious* resolvers in the *client hijacking* use case and the *passive* collection of *NetFlow* data. The related work is described in this section and the properties highlighted above are summarized in Table 8.7.

Some papers measure which resolvers are used by introducing *observer-controlled authoritative servers* and zones. These are combined with advertisement campaigns [115], visits to self-owned websites [141] or a large number of remotely controlled clients in various world regions that send DNS requests [142]. Approaches relying on data from browsers or the installation of specific apps will not capture any traffic from unsupported device types, such as IoT devices, home routers etc., which is central to our paper.

Other papers focus on inspecting DNS data obtained by passively mirroring *DNS traffic at the application layer* at a non-authoritative point in the DNS

| | **Related Work** | | | | |
|---|---|---|---|---|---|
| **Aspect** | 120,139 | 136,140 | 107,115,141,142 | 143 | 138,144 |
| Client hijack | ✓ | | | | |
| Passive approach | ✓ | | ✓ | ✓ | ✓ |
| NetFlow data | | | | ✓ | |
| Maliciousness | ✓ | ✓ | | | ✓ |

**Table 8.7:** *Notable related work and aspects in focus*

chain. This includes dump of DNS flows at application layer at an ISP [120], at a LAN gateway [144], at a campus gateway [139] and using DNS data from the Farsight database [138]. It should be noted that the Farsight database is built upon voluntary participation by resolver owners, and so it is unlikely that queries towards intentionally malicious resolvers would be represented in this database.

Finally, use of *data from clients* collected in the Open Observatory of Network Interference (OONI) database is used by Radu et al. [107], and passively mirroring *DNS traffic at the network/transport layer* through NetFlow at a national ISP is used by Fejrskov et al. [143]. Their focus is, however, on the use of major/well-known 3rd party resolvers, not on the more rarely used, potentially malicious resolvers.

The maliciousness of DNS responses are evaluated using various techniques, such as by use of open threat intelligence [140] [139], by probing HTTP/POP3/ IMAP/SMTP services on the resolved IPs [136] [144], by detecting differences in responses for similar queries [120] and by detecting NS record changes [138].

Of the papers mentioned above, only Dagon et al. and Trevisan et al. focus on the use case of changing the DNS resolver IP [139] [120]. Interestingly, the techniques used by the two papers can be used for detecting all the DNS hijacking use cases mentioned initially in this section, not only for detecting malicious DNS resolver IP changes. As mentioned above, these papers inspect DNS data to achieve their results and are therefore fundamentally different from our paper.

## 8.6 Conclusion

This paper investigates if it is possible to classify public resolvers as malicious and/or well-known using features derived from NetFlow data. Our suggested NetFlow based approach comes with a number of advantages compared to existing methods: *i)* it is legal to be deployed by ISPs in the EU, *ii)* it does not rely on excessive Internet-wide scanning, *iii)* it does not rely on features that are controllable by a malicious actor.

Using Random Forest and Gradient-Boosted Trees on 7 different NetFlow-related features we show that it is indeed possible to classify a resolver as well-known or malicious with an AUROC of 0,85 (around 0.80 using under-sampling). This shows that NetFlow features can indeed contribute to the classification, although the value is not high enough for a NetFlow-only approach to be considered for operational use. It may be possible to create a better model if a larger data set is used, especially if the dataset has a better balance between malicious and benign resolvers.

In our paper, active probing of resolvers is intentionally only used for labelling / model training purposes, not as features, as the purpose is to investigate the value of a NetFlow-only approach. To increase the accuracy of the classification, we consider a hybrid approach adding such features as the most interesting approach for future work.

## 8.7 Appendix

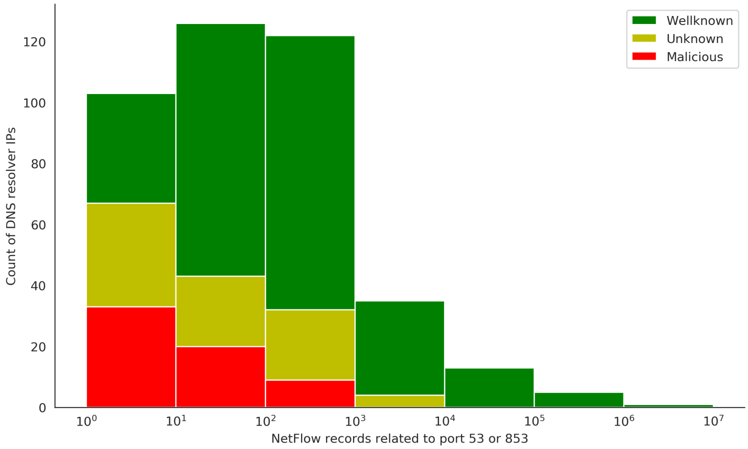

**Fig. 8.3:** *Histogram showing the count of DNS resolver IP addresses for which a certain number of NetFlow records are reported. As an example, between 10 and 100 NetFlow records are observed from approximately 125 different DNS resolvers IPs.*

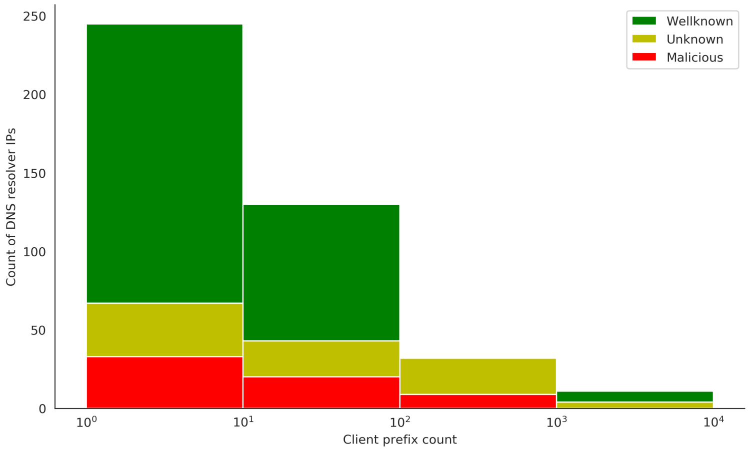

**Fig. 8.4:** *Histogram showing the count of DNS resolver IP addresses that are observed to be used by a certain number of client /24 prefixes. As an example, approximately 30 DNS resolvers IPs are used by between 100 and 1000 client prefixes.*

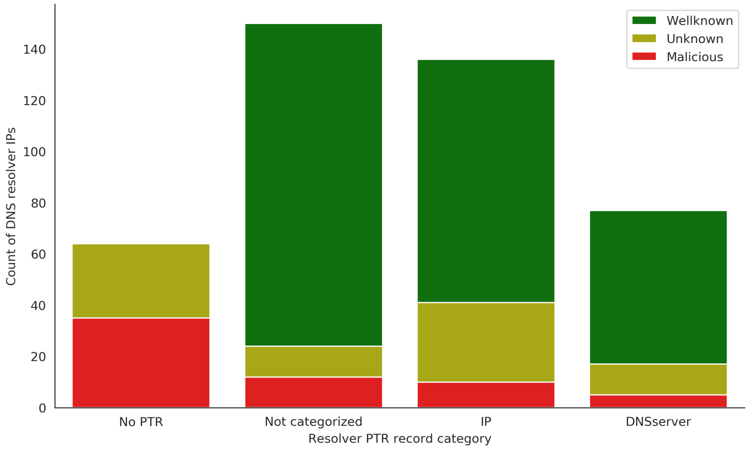

**Fig. 8.5:** *Histogram showing the count of DNS resolver IP addresses that have a PTR record in a certain category. As an example, approximately 75 IP addresses have a PTR record whose name hints that this could be a DNS server.*

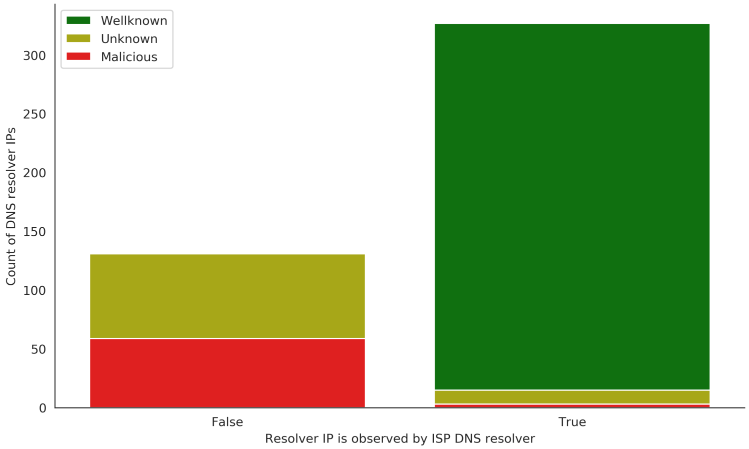

**Fig. 8.6:** *Histogram showing the count of DNS resolver IP addresses that are also observed in response records in Telenor's default DNS resolvers. As an example, 130 resolver IPs were not observed in Telenor's DNS resolver response records.*

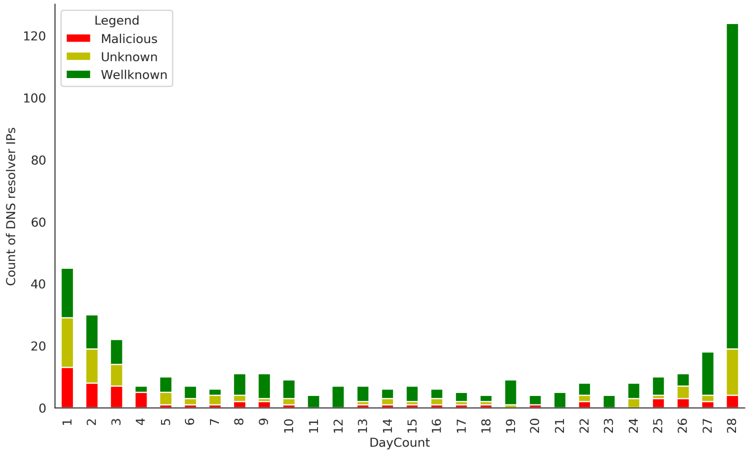

**Fig. 8.7:** *Histogram showing the count of DNS resolver IP addresses that are observed in a certain number of days. As an example, approximately 20 different DNS resolvers IPs are observed for a total of exactly 3 days during the data collection period.*

CHAPTER 9

# Discussion

The papers presented in the previous chapters are part of the same subject area and are based on the same dataset. Therefore they share a number of common properties that warrant an extended discussion and comparison, which is the topic of the following sections in this chapter.

## 9.1 Feasibility of proposed anonymization policy

An anonymization policy is proposed in Paper A, and this policy has been applied on the data in the remaining papers, with some exceptions, as listed in Table 9.1.

The client IP addresses (or ports for CGNAT'ed clients) are anonymized by truncation in all papers. In practice, this means that the traffic of up to 256 subscribers is considered to originate from a single prefix. From the perspective of attributing traffic (malicious or not) to a specific subscriber, this privacy-yielding anonymization is obviously an obstacle. However, as shown by the papers in this thesis, the client IP anonymization does not make it impossible to draw conclusions from the data. Indeed, any network traffic analysis methods that operate on data in which multiple users/clients share the same IP address (such as the NAT employed by most broadband deployments and the CGNAT employed by many mobile deployments) are subject to a similar type of limitation, due to the implicit anonymization provided by the NAT functionality. However, from an ISP business and ISP customer

| **Feature** | **Data type** | **Paper B** | **Paper C** | **Paper D** | **Paper E** |
|---|---|---|---|---|---|
| Client IP/port | DNS and NetFlow | ✓ | ✓ | ✓ | ✓ |
| Sample rate | NetFlow | 1:512 | 1:512 | 1:512 | 1:1024 |
| External IP | NetFlow | ✓ | ✓/ - | - | - |

**Table 9.1:** *Comparison of which of the most relevant features are anonymized according to the anonymization policy proposed in Paper A. Paper B follows the proposed anonymization policy completely, whereas the remaining papers omit some features from being anonymized.*

perspective, the detection result therefore has significantly less value, as the derived intelligence is then not actionable, as customers cannot be warned, and an attack cannot be mitigated.

Both Paper B, D and E involve comparing either the external IP address of a NetFlow record or the IP address found in the A record of a DNS response to an external list of ground truth, such as a blacklist or a list of known, public resolvers. There is, however, at crucial difference between Paper B on one side, and Paper D and E on the other side. In Paper B, the blacklist is available as a list of IP addresses that can also be truncated, and then compared to the anonymized IP addresses in the NetFlow data. In Paper D and Paper E the method is based on using the external IP from the NetFlow data to perform a search in an external database, and this naturally requires the IP address to be available in a non-anonymized version. This difference is the reason for the anonymization policy to be different for the two groups of papers. A key takeaway is therefore that the availability of curated lists of ground truth can potentially make it possible to apply a stricter anonymization policy on the external IP address in NetFlow records.

The correlation of DNS and NetFlow traffic is a key step in the methods described in Paper B, C and D. Of these papers, only Paper C does not rely on the comparison with any lists of ground truth as described above. Although the IP address available in the A record of a DNS response is not anonymized (as it is public information), the anonymization of the external IP address in the NetFlow records requires the correlation of DNS and NetFlow records to be performed on the anonymized version of the IP address found in the A record. For this reason, experiments with different levels of anonymization of the external IP address was performed in Paper C to show the effect on the results of anonymization on the renaming process. Specifically in Paper C, the effect of applying anonymization was an increase in false positive results. Further studies are needed to provide a more general conclusion on the effect of anonymization on the renaming accuracy, and ideally the contribution made by anonymizing the client IP addresses should be considered in such a study as well.

The NetFlow sample rate was changed during the project to address capacity limitations in other NetFlow reception systems at Telenor. From a data visibility perspective, it is obvious that a coarse sample rate can cause inaccuracies in results, that can only in some cases be compensated for by collecting data from more clients or for a longer time period. From an anonymization perspective, however, the sample rate is primarily relevant in order to degrade the precision of byte and packet counts. As these counts can to some extent be controlled by a malicious actor, they are not given a prominent role in this thesis.

In summary, the feasibility of the anonymization policy suggested in Paper A should, as expected, be evaluated on a case-by-case basis. Unsurprisingly, the use of NAT in a network presents some of the same challenges as the anonymization of the client IP address, so methods designed to handle NAT'ed traffic could also work well with anonymized traffic, and vice versa. The anonymization of the server IP address can be a limiting factor, however, this can to some extent be mitigated by the availability of a non-anonymized ground truth base.

## 9.2 Large scale data versus ISP data

Although this thesis is focused on ISP data, it is interesting to briefly consider which of the methods and results described in the thesis papers are relevant outside of an ISP context, and if so, whether an ISP would still be the best choice of data source, even despite regulatory restrictions.

The anonymization policy presented in Paper A is designed to comply with the ePrivacy Directive, which only applies to ISPs. Although the policy could in principle also be applied by non-ISPs to comply with the GDPR, it seems more likely that such organisations would not limit themselves to only collect NetFlow and DNS data, and therefore the practical application outside of the ISP domain is probably limited.

The method presented in Paper B to estimate the impact of DNS-based blacklists can in principle be applied to traffic from a single user, and could also be applied on data in which all ISO layers are available (instead of solely DNS and NetFlow data), for example PCAP dumps.

Detecting IGA botnets using the method presented in Paper C requires a certain number of bots within a network, as the method is based on identifying the group behaviour of the bots. Therefore, the method is likely only applicable in networks with a large number of users that at the same time have a sufficiently large diversity in security posture. As an example, a well-controlled, large corporate network with enforced anti-virus application policies and DNS traffic inspection/blocking is probably less likely to contain a large number of infected clients than an ISP with the same number of users. Therefore, due to the diversity of the security posture of the users, the method is probably more interesting in practice to apply on an ISP network than to a large corporate network, although the use case itself is valid outside of an ISP context.

Measuring the use of 3rd party resolvers for either malware filtering or censorship circumvention purposes as performed in Paper D is topic-wise probably only interesting to ISPs. If this should be interesting to for example corporate networks, it seems reasonable to assume that this can be performed in a more simple manner by using statistics from firewalls or other traffic inspection devices already in place.

The method to characterize resolvers described in Paper E only provides

| Paper | A | B | C | D | E |
|---|---|---|---|---|---|
| Use case is relevant outside ISP context | No | Yes | Yes | No | Yes |
| ISP data could still be preferable | - | No | Yes | - | Yes |

**Table 9.2:** *Relevance of use cases and ISP data outside of an ISP context.*

value in environments where traffic towards 3rd party resolvers is not blocked. Furthermore, as the method is based on machine learning, a data set where many resolvers are represented is needed. In practice, although relevant outside of an ISP context, these properties are probably mostly found in ISP networks.

As summarized in Table 9.2, some methods presented in this thesis are indeed relevant outside of an ISP context. Some of the methods that are relevant outside of an ISP context could potentially benefit from using a larger ISP data set anyway, despite the anonymization applied. For other uses cases that are relevant outside of an ISP context, a different type of data set may be available and/or simpler methods of obtaining similar results may be applicable.

## 9.3 Project success criteria and business value

Five success criteria were defined prior to the initiation of the Ph.D. project (quoted here in italics), and a small discussion of these criteria is appropriate. *The project is perceived as a success if it succeeds in:*

- *Using the developed methods/tools to find and confirm attacks against at least 10 identifiable customers who had otherwise not discovered the attack.* Due to the anonymization requirements, it has not been legally possible to identify individual customers and therefore not possible to confirm the attacks, and ascertain whether the attacks were successful or not. However, based on Paper B and Paper E, it is still possible to find potential attack attempts. The anonymization causes traffic to appear to originate from approximately 6400 source prefixes. Data used for week 2 in Paper B reveal that 680 unique prefixes create flows towards malicious domains, and these flows would have been blocked if a network-wide DNS resolver based blacklisting had been in place. Similarly, data used for Paper E show that 207 unique prefixes receive DNS responses from DNS resolvers known to be malicious. It seems likely that these prefixes represent at least 10 customers, and therefore this success criterion is considered fulfilled to the extent permitted by law.

- *Contribute to the development of at least one new product at Telenor.* The process of working on Paper A has contributed with general knowledge about the legal conditions under which various data types can be used. This is useful as basic knowledge for all product development (not just malware and security related) that intends to use customer related data. More specifically, three papers have contributed to assessing the business case of deploying blacklists in DNS resolvers. Paper B demonstrates the possible impact of DNS blacklisting with massive adoption, such as in a free-for-all business model, and highlights the steps that can be taken to avoid undesired impact. Paper D maps the current market adoption of competing 3rd party resolvers that offer filtering of malicious domains and/or parental control functionality. An ISP could offer an option to block a customer's DNS traffic towards all 3rd party resolvers, so that malware is not able use a non-filtering or even malicious 3rd party resolver. This could be offered both in combination with a DNS resolver based filtering mechanism and as a stand-alone option. As mentioned above, Paper E contributes by showing that such an option could provide immediate value to customers in 207 prefixes. This success criterion is considered fulfilled to the extent made possible within Telenor's current commercial security product landscape.

- *Demonstrate that the developed methods can significantly improve detection compared to existing statistical methods.* Although Paper B does not provide an improved *detection* method, it does provide an improved method for estimating the *impact* of deploying DNS-based blacklists. A detection method for a proposed new type of botnet is described in Paper C. However, as no existing methods exist for detecting specifically this type of botnet, the described method can strictly speaking not be considered an improvement to existing methods. Detecting malicious DNS resolvers and the use thereof is performed by existing method through the use of DNS data. However, the method proposed in Paper D uses NetFlow data, and should therefore be considered complimentary to existing methods, rather than an improvement. Although none of the papers in the thesis address *all* of the elements in the specific formulation of the criterion, the thesis as a whole is considered to fullfill the criterion.

- *Result in at least 2 publications in top-tier conferences/journals.* The co-authored Paper F titled "Processing of Botnet Tracking Data under the GDPR" is published in Journal of Computer Law & Security Review, and this journal is considered the top ranking journal in the category "Technology Law" by Google Scholar [145]. Paper D and E are accepted in / submitted to conferences (IFIP SEC and IWSEC) that are considered B rank by the Core2021 ranking [146]. A Core2021 B rank conference is not considered top tier, but is considered a good and well regarded conference. Paper B is published in the SecureComm conference, which is currently a C rank conference, but was considered a B rank conference by Core2020. The conferences in which Paper A and C are published (Cyber Security and CNS) are not ranked by the Core ranking association, however the CNS conference is by some considered on par with the other B rank conferences mentioned above [147]. This success criterion is therefore considered partially fulfilled.

- *Develop a dashboard that can continuously display analytical results and provide ongoing overview of cyberattacks and malware infections among Telenor's customers.* A dashboard as depicted in Figure 9.1 on the facing page was created when working on Paper B. The original intention of the dashboard was for Telenor to be able to provide a warning to infected and/or attacked customers, and to gain a network-level overview of the number of attacked/infected customers in different customer segments. As the anonymization does not allow precise customer identification and therefore not an accurate count or customer segment attribution, the value of the dashboard is limited in practice. Therefore, the dashboard idea was not pursued further in the remaining part of the project. This success criterion is, however, considered fulfilled to the extent that it provided project value.

In summary, all criteria are considered fulfilled or partially fulfilled.

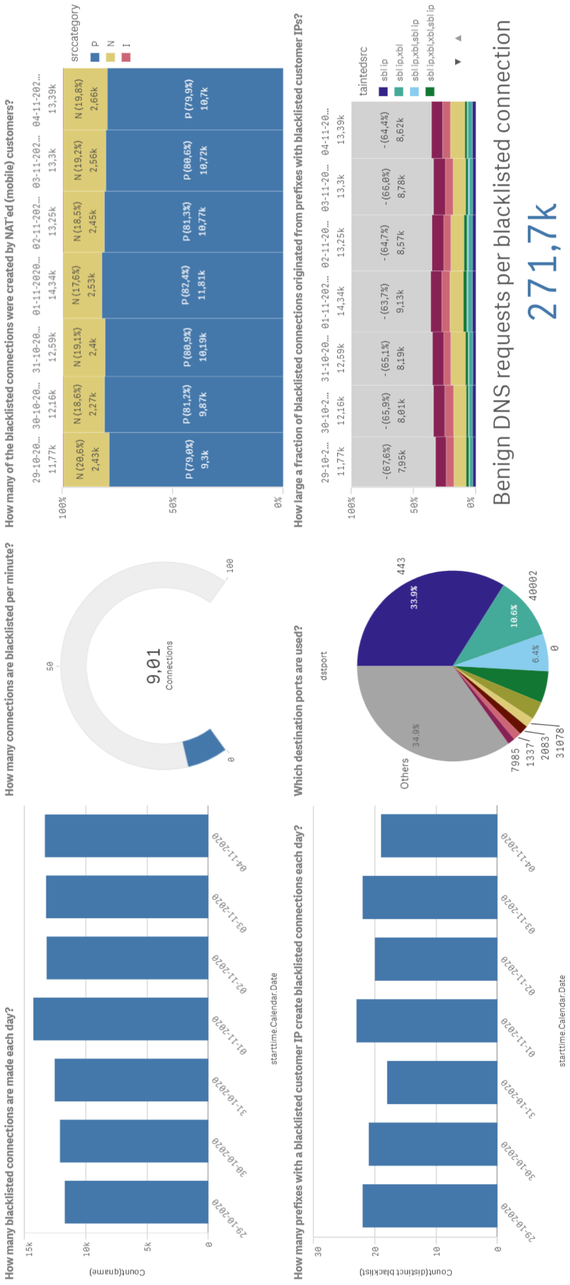

**Fig. 9.1:** *Dashboard showing some of the results of the continuous data analysis using the methods described in Paper B.*

# CHAPTER 10
# Conclusion

The starting point of this thesis is the unique value proposition that ISPs have in terms of access to the customers' internet traffic. As outlined in the introduction, this lead to the following thesis statement: "Compared to only using IP traffic data, it is advantageous to use data from ISPs to identify cyber attacks against customers and to identify customers infected with malware."

To explore this thesis statement, the technical and legal availability of ISP data is explored, and various use cases for the data are presented in separate papers with individual conclusions in chapter 2-8.

## 10.1 Contribution summary

Apart from the overall conclusion described in the next section, the specific scientific contributions from this thesis can be briefly summarized as follows:

- An analysis of which ISP data are relevant to cyber security research, an overview of the legal aspects of using such data, and an anonymization policy for DNS and NetFlow data.
- A method to estimate the impact of applying blacklists in DNS resolvers that represents an improvement to existing methods.
- A new type of botnet (and associated detection method) that does not reveal the CnC IP address in clear text in DNS records.
- Methods that show the prevalence of the use of 3rd party resolvers in an ISP network, and whether these resolvers are chosen for malware filtering or censorship evasion purposes.
- A method for classifying a resolver as either well known or malicious based on applying machine learning to NetFlow data.

As discussed in Section 9.3, these scientific contributions have throughout the project been used to provide business value to Telenor Denmark as well.

## 10.2 Overall conclusion

A key aspect is that data from European ISPs are by law subject to strict anonymization requirements. In practice this makes it hard, if not impossible, to use and/or correlate several otherwise interesting data sources, even when considering different levels of anonymization as discussin in Section 9.1. Therefore, the only data sources actually available (and relevant to malware detection) are DNS logs and NetFlow records, which can be considered a subset of the IP traffic data. This aspect seems at first glance to negate the thesis statement, at least within the jurisdiction of the European Union.

As discussed in Section 9.2, use cases that are only relevant in an ISP context do exist, such as the use case presented in Paper D. For such use cases, the only way forward is to use ISP data, and tolerate the consequences of anonymization required by current legislation. Other use cases, such as the use cases presented in Paper C and E, rely on datasets with the large-scale user and diversity properties that are primarily found in ISP data sets, making it likely that ISP data is the preferable data set to use, even in the presence of anonymization. This supports the thesis, albeit under specific circumstances only.

The overall recommendation from the work of this thesis is that within the current EU jurisdiction, ISP data should only be used as a last resort for malware and cyber attack detection. Other data sources, like data from a university campus, a large company, or non-EU ISPs, should be considered preferred choices. From a broader scientific perspective, this conclusion makes it seem likely that ISP-scale cyber security research will decrease in the EU region in the future. Not exploiting the vast academic and financial resources of the European Union for cyber security research purposes could be detrimental to the global security posture, as the field of cyber security is a constantly evolving arms race against malicious actors.

## 10.3 Future directions

The papers of the thesis suggest the direction of future work within the scope of the individual papers. From a larger perspective, I consider the following directions within the regulatory, commercial and academic domains as the most interesting to pursue:

**Regulatory** A revised ePrivacy Regulation is being proposed [148], that will among other extend the scope of the ePrivacy Directive [8] to not only apply to ISPs, but to also apply to Over-The-Top service providers of electronic communication like Skype, Messenger and even in-game messaging systems. Many definitions and paragraphs have been reformulated to support the understanding of the legislation, for example the definitions in Article 4 on the definition of data, content and metadata. More interestingly, Article 6 states that data can be processed if it is necessary to maintain or restore the security of electronic communications networks and services for the duration necessary for that purpose. Although this is a rather general and unspecific statement, it may open up for more flexible interpretations in the future than what the Directive allows for.

**Commercial** As outlined in Chapter 3.1 on page 25, none of the commercially available malware detection products seem to be able to operate on anonymized data. Although the focus of this thesis is data from ISPs and the related ePrivacy directive, other business types in the EU are still subject to the rules of the GDPR, where anonymization can in some cases also be used as a tool to enable compliance. Enabling products to be able to work on anonymized data could therefore become a competitive advantage to a vendor in several different business segments. An obvious example is to make data analysis products aware that certain ingested data types, for example IP addresses, are anonymized in a specific way. A side effect of this is that methods designed for non-NAT'ed data may be simpler to extend to also apply to NAT'ed data, thus expanding the applicability of the product. Another example could be to enable routers to apply a anonymization policy to NetFlow records before emission, so that non-anonymized NetFlow records are never emitted.

**Academic** The argument of competitive advantage presented for commercial contributions can also to some extent be applied to academic contributions. First of all, a contribution that allows for the use of anonymized data could be more likely to be applicable in practice. Second, applying anonymization in a contribution even though non-anonymized data is available could represent an opportunity instead of a challenge, as this could be an unexplored niche in some fields. As a specific example, many of the papers presented in this thesis use DNS data to rename flows found in NetFlow data, as described in chapter 4 on page 35. Although papers by other authors apply the renaming process on non-anonymized data, the quantification of the consequences of applying renaming on anonymized data seems unexplored. As another example, it could be interesting to assess the extent to which the accuracy of existing malware detection methods proposed in the academic literature would be decreased when applying anonymization on the input data. Such studies will naturally require access to a non-anonymized data set, but could potentially also require access to individual hosts in order to pin individual DNS requests and the following data flows to individual applications rather than to individual hosts.

Although the conclusion that other data sources should currently be preferred over ISP data may seem a rather disappointing conclusion from a security point of view, it could also be considered positive from a privacy point of view. How to set the right balance between privacy and security is not merely a technical discussion, but rather a political and philosophical question. Current legislation within this area is almost exclusively favouring privacy, however as a subjective opinion, I find that society as a whole would gain by leaning the balance a bit more towards the side focusing on security. Although this thesis provides a tiny piece in this puzzle, the debate on privacy versus security will probably still be a topic of political debate for many years to come.

# References

[1] B_A, "Untitled," Pixabay, 2022. [Online]. Available: https://pixabay.com/photos/hacker-silhouette-hack-anonymous-3342696/

[2] Geralt, "Untitled," Pixabay, 2022. [Online]. Available: https://pixabay.com/illustrations/binary-one-cyborg-cybernetics-1536651/

[3] L. Böck, M. Fejrskov, K. Demetzou, S. Karuppayah, M. Mühlhäuser, and E. Vasilomanolakis, "Processing of Botnet Tracking Data under the GDPR," Journal of Computer Law & Security Review, 2022. [Online]. Available: https://doi.org/10.1016/j.clsr.2021.105652

[4] Danish Defence Intelligence Service, "Intelligence Outlook 2021," 2021. [Online]. Available: https://www.fe-ddis.dk/globalassets/fe/dokumenter/2021/udsyn/-fe-udsyn-uk_final_samlet_fredag-.pdf

[5] Gartner, "Gartner Survey of Over 2,000 CIOs Reveals the Need for Enterprises to Embrace Business Composability in 2022," 2021. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-10-18-gartner-survey-of-over-2000-cios-reveals-the-need-for-enterprises-to-embrace-business-composability-in-2022

[6] B. Rowe and D. Wood, "Are Home Internet Users Willing to Pay ISPs for Improvements in Cyber Security?" Economics of Information Security and Privacy III, 2013. [Online]. Available: https://doi.org/10.1007/978-1-4614-1981-5_9

[7] D. Singh, "The Changing Consumer Landscape - Telco Strategies for Success," 2021. [Online]. Available: https://www.telecompaper.com/industry-resources/the-changing-consumer-landscape-telco-strategies-for-success--1388792

[8] The European Parliament and of the Council, "Directive 2002/58/ec (the ePrivacy directive)," 2002. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0058

[9] ——, “Regulation (eu) 2015/2120 (the Net Neutrality Regulation),” 2015. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015R2120

[10] ——, “Regulation (eu) 2016/679 (the General Data Protection Regulation, GDPR),” 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[11] The Working Party on the Protection of Individuals With Regard to the Processing of Personal Data, “Opinion 05/2014 on Anonymisation Techniques,” 2014. [Online]. Available: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

[12] E. Boschi and B. Trammel, “IP Flow Anonymization Support, RFC 6235,” 2011. [Online]. Available: https://doi.org/10.17487/RFC6235

[13] D. Riboni, A. Villani, D. Vitali, C. Bettini, and L. V. Mancini, “Obfuscation of sensitive data for incremental release of network flows,” IEEE/ACM Transactions on Networking, 2014. [Online]. Available: https://doi.org/10.1109/TNET.2014.2309011

[14] N. Dijkhuizen and J. Ham, “A survey of network traffic anonymisation techniques and implementations,” ACM Computing Surveys, 2018. [Online]. Available: https://doi.org/10.1145/3182660

[15] W. John, S. Tafvelin, and T. Olovsson, “Passive internet measurement: Overview and guidelines based on experiences,” Computer Communications, 2009. [Online]. Available: https://doi.org/10.1016/j.comcom.2009.10.021

[16] D. Herrmann, C. Banse, and H. Federrath, “Behavior-based tracking: Exploiting characteristic patterns in dns traffic,” Computers & Security, 2013. [Online]. Available: https://doi.org/10.1016/j.cose.2013.03.012

[17] D. W. Kim and J. Zhang, “Deriving and measuring dns-based fingerprints,” Journal of Information Security and Applications, 2017. [Online]. Available: https://doi.org/10.1016/j.jisa.2017.07.006

[18] D. Sauter, M. Burkhart, D. Schatzmann, and B. Plattner, “Invasion of privacy using fingerprinting attacks,” 2009. [Online]. Available: https://pub.tik.ee.ethz.ch/students/2008-HS/MA-2008-22.pdf

[19] S. Dickinson, B. Overeinder, R. van Rijswijk-Deij, and A. Mankin, “Recommendations for DNS Privacy Service Operators,” 2019. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-dprive-bcp-op-07

[20] M. Burkhart, D. Brauckhoff, M. May, and E. Boschi, "The risk-utility tradeoff for ip address truncation," CCS: Computer and Communications Security, ACM workshop on Network data anonymization, 2008. [Online]. Available: https://doi.org/10.1145/1456441.1456452

[21] M. Burkhart, D. Schatzmann, B. Trammell, E. Boschi, and B. R. Plattner, "The role of network trace anonymization under attack," ACM SIGCOMM Computer Communication Review, 2010. [Online]. Available: https://doi.org/10.1145/1672308.1672310

[22] K. Lakkaraju and A. Slagell, "Evaluating the utility of anonymized network traces for intrusion detection," SecureComm: International conference on Security and privacy in communication networks, 2008. [Online]. Available: https://doi.org/10.1145/1460877.1460899

[23] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter, "On web browsing privacy in anonymized netflows," USENIX Security Symposium, 2007. [Online]. Available: https://www.cs.unc.edu/~fabian/papers/usenix07b.pdf

[24] R. van Rijswijk-Deij, G. Rijnders, M. Bomhoff, and L. Allodi, "Privacy-Conscious Threat Intelligence Using DNSBLoom," IFIP/IEEE International Symposium on Integrated Network Management, 2019. [Online]. Available: http://dl.ifip.org/db/conf/im/im2019/189282.pdf

[25] D. C. Ferreira, M. Bachl, G. Vormayr, F. Iglesias, and T. Zseby, "A meta-analysis approach for feature selection in network traffic research," ACM SIGCOMM Reproducibility Workshop, 2017. [Online]. Available: https://doi.org/10.1145/3097766.3097771

[26] Nmap.org, "Tcp/ip fingerprinting methods supported by nmap," 2019. [Online]. Available: https://nmap.org/book/osdetect-methods.html

[27] M. Larsen and F. Gont, "Recommendations for Transport-Protocol Port Randomization, RFC 6056," 2011. [Online]. Available: https://doi.org/10.17487/RFC6056

[28] M. Singh, M. Singh, and S. Kaur, "Issues and challenges in dns based botnet detection: A survey," Computers & Security, 2019. [Online]. Available: https://doi.org/10.1016/j.cose.2019.05.019

[29] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: a passive dns analysis service to detect and report malicious domains," Computers & Security, 2014. [Online]. Available: https://doi.org/10.1145/2584679

[30] Wikipedia, "Network behavior anomaly detection," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Network_behavior_anomaly_detection

[31] Gartner Research, "Market guide for network traffic analysis," 2019. [Online]. Available: https://www.gartner.com/en/documents/3902353/market-guide-for-network-traffic-analysis

[32] Cisco, "Cisco Stealthwatch," 2019. [Online]. Available: https://www.cisco.com/c/dam/en/us/td/docs/security/stealthwatch/management_console/smc_users_guide/SW_7_1_Stealthwatch_Desktop_Client_User_Guide_DV_1_0.pdf

[33] McAfee, "Network threat behavior analysis appliance administration guide," pp. 15–17, 2019. [Online]. Available: https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/27000/PD27258/en_US/NSP_91_NTBA_Administration_Guide_revE_en-us.PDF

[34] ManageEngine, "Security problem and class catalog," 2019. [Online]. Available: https://www.manageengine.com/products/netflow/problem-class-catalogue.html

[35] Redsocks (bought by Bitdefender), "Bitdefender network traffic security analytics," 2019. [Online]. Available: https://www.bitdefender.com/business/enterprise-products/network-traffic-security-analytics.html

[36] FlowTraq (bought by Riverbed), "Flowtraq 18.12 documentation," 2018. [Online]. Available: http://support.flowtraq.com/Documentation/18.12/webhelp/content/ch03s03.html

[37] Solana Networks, "Smartflow," 2019. [Online]. Available: https://www.solananetworks.com/sites/default/files/resources/smartflow-brochure.pdf

[38] AdvaICT, "Flowmon ads user guide," 2012. [Online]. Available: http://www.orsenna.org/ftp/Documentation/Invea/flowmon_ads_user_guide_en_2012.pdf

[39] R. Hofstede, "Flow-based compromise detection," Centre for Telematics and Information Technology, 2016. [Online]. Available: https://doi.org/10.3990/1.9789036540667

[40] ——, "List of publications," 2019. [Online]. Available: https://scholar.google.com/citations?user=_GyDVoMAAAAJ

[41] Cisco, “Threat detection, Cisco Stealthwatch at work,” 2019. [Online]. Available: https://cisco.bravais.com/s/SXhrFcFSKfsqJnOUyF2J

[42] ——, “Cisco Stealthwatch security events and alarm categories 7.0,” 2018. [Online]. Available: https://www.cisco.com/c/dam/en/us/td/docs/security/stealthwatch/management_console/securit_events_alarm_categories/SW_7_0_Stealthwatch_Security_Events_and_Alarm_Categories_DV_1_0.pdf

[43] ——, “Cisco security analytics,” 2018. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/security/stealthwatch/white-paper-c11-740605.pdf

[44] Plixer, “Flowpro features and functionality,” 2019. [Online]. Available: https://docs.plixer.com/projects/flowpro/en/latest/flowProFeaturesAndFunctionality.html

[45] PowerDNS, “Security of the powerdns recursor,” 2019. [Online]. Available: https://doc.powerdns.com/recursor/security.html

[46] Palo Alto Networks, “Dns security service,” 2019. [Online]. Available: https://www.paloaltonetworks.com/resources/datasheets/dns-security-service

[47] Akamai, “Akamai threat avert product brief,” 2019. [Online]. Available: https://www.akamai.com/uk/en/multimedia/documents/product-brief/sps-threatavert-product-brief.pdf

[48] AlphaSOC, “Network behaviour analytics for splunk,” 2019. [Online]. Available: https://alphasoc.com/docs/nba-introduction

[49] F5, “DNS security agility,” 2018. [Online]. Available: https://www.f5.com/pdf/agility2018/dns_security.pdf

[50] Cisco, “Cisco Umbrella Investigate,” 2019. [Online]. Available: https://docs.umbrella.com/investigate-ui/docs/

[51] D. Mahjoub, “Finding the patterns in a mysterious new DGA,” 2013. [Online]. Available: https://umbrella.cisco.com/blog/2013/10/24/mysterious-dga-lets-investigate-sgraph/

[52] Infoblox, “Infoblox advanced DNS protection,” 2019. [Online]. Available: https://www.infoblox.com/wp-content/uploads/infoblox-datasheet-infoblox-advanced-dns-protection.pdf

[53] J. M. Kuo, R. Nagy, and C. Liu, “DNS security for dummies,” 2018. [Online]. Available: https://www.infoblox.com/wp-content/uploads/infoblox-ebook-dns-security-for-dummies.pdf

[54] HP, "HP ArcSight DNS malware analytics datasheet," 2015. [Online]. Available: http://www.hp.com/sbso/hpinfo/newsroom/DNSMalwareAnalyticsDataSheet.pdf

[55] M. C. Mont and Y. Beresna, "Big data for security: Usable DNS data analytics at scale," 2016. [Online]. Available: https://www.labs.hpe.com/techreports/2016/HPE-2016-99.pdf

[56] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Springer CyberSecurity, 2019. [Online]. Available: https://doi.org/10.1186/s42400-019-0038-7

[57] S. García, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," Security and Communication Networks, 2013. [Online]. Available: https://doi.org/10.1002/sec.800

[58] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," ACM Computing Surveys, 2018. [Online]. Available: https://doi.org/10.1145/3191329

[59] K. Alieyan, A. Almomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," Neural Computing and Applications, 2017. [Online]. Available: http://dx.doi.org/10.1007/s00521-015-2128-0

[60] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi, "Detecting internet abuse by analyzing passive DNS traffic: A survey of implemented systems," IEEE Communications Surveys & Tutorials, 2018. [Online]. Available: https://doi.org/10.1109/COMST.2018.2849614

[61] M. Stevanovic and J. M. Pedersen, "On the use of machine learning for identifying botnet network traffic," Journal of Cyber Security and Mobility, 2016. [Online]. Available: https://doi.org/10.13052/jcsm2245-1439.421

[62] A. H. Lashkari, G. D. Gil, J. E. Keenan, K. F. Mbah, and A. A. Ghorbani, "A survey leading to a new evaluation framework for network-based botnet detection," ICCNS: International Conference on Communication and Network, 2017. [Online]. Available: https://doi.org/10.1145/3163058.3163059

[63] D. Huistra, "Detecting reflection attacks in DNS flows," 2013. [Online]. Available: https://pdfs.semanticscholar.org/4ad8/24537f212f70e25e4cbab55498f5a8e43942.pdf

[64] M. Grill, I. Nikolaev, V. Valeros, and M. Rehak, “Detecting DGA malware using NetFlow,” IFIP/IEEE International Symposium on Integrated Network Management, 2015. [Online]. Available: https://doi.org/10.1109/INM.2015.7140486

[65] R. Hananto, C. Lim, and H. P. Ipung, “Detecting network security threats using domain name system and NetFlow traffic,” ICCSP: International Conference on Cryptography, Security and Privacy, 2018. [Online]. Available: https://doi.org/10.1145/3199478.3199505

[66] B. Anderson and D. McGrew, “Identifying encrypted malware traffic with contextual flow data,” CCS: Computer and Communications Security, ACM Workshop on Artificial Intelligence and Security, 2016. [Online]. Available: https://doi.org/10.1145/2996758.2996768

[67] K. Wang, C.-Y. Huang, S.-J. Lin, and Y.-D. Lina, “A fuzzy pattern-based filtering algorithm for botnet detection,” Computer Networks, 2011. [Online]. Available: https://doi.org/10.1016/j.comnet.2011.05.026

[68] M. Janbeglou, “Understanding and Controlling Unnamed Internet Traffic,” The University of Auckland, 2017. [Online]. Available: https://researchspace.auckland.ac.nz/handle/2292/36323

[69] K. Hageman, E. Kidmose, R. R. Hansen, and J. M. Pedersen, “Understanding the Challenges of Blocking Unnamed Network Traffic,” To appear in IEEE/IFIP Network Operations and Management Symposium, 2022. [Online]. Available:

[70] Wikipedia, “Domain Name System-based Blackhole List,” 2020. [Online]. Available: https://en.wikipedia.org/wiki/Domain_Name_System-based_Blackhole_List

[71] J. Jung and E. Sit, “An empirical study of spam traffic and the use of DNS black lists,” IMC: Internet Measurement Conference, 2004. [Online]. Available: https://doi.org/10.1145/1028788.1028838

[72] R. Williamson, “What do Canada and New Zealand have in common?” 2020. [Online]. Available: https://internetnz.nz/blog/dns-firewall-what-do-canada-and-new-zealand-have-in-common/

[73] Cisco Umbrella, “Better intelligence drives better security,” 2020. [Online]. Available: https://umbrella.cisco.com/solutions/reduce-security-infections

[74] X. Bouwman, H. Griffioen, J. Egbers, C. Doerr, B. Klievink, and M. van Eeten, “A different cup of TI? The added value of commercial threat intelligence,” USENIX Security Symposium, 2020. [Online]. Available: https://www.usenix.org/system/files/sec20-bouwman.pdf

[75] S. Sinha, M. Bailey, and F. Jahanian, “Shades of grey: On the effectiveness of reputation-based "blacklists",” MALWARE: International Conference on Malicious and Unwanted Software, 2008. [Online]. Available: https://doi.org/10.1109/MALWARE.2008.4690858

[76] A. Ramachandran and N. Feamster, “Understanding the Network-Level Behavior of Spammers,” ACM SIGCOMM Computer Communication Review, 2006. [Online]. Available: https://doi.org/10.1145/1159913.1159947

[77] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” CEAS: Conference on Email and Anti-Spam, 2009. [Online]. Available: https://doi.org/10.1184/R1/6469805.V1

[78] I. N. Bermudez, M. Mellia, M. M. Munafò, R. Keralapura, and A. Nucci, “DNS to the Rescue: Discerning Content and Services in a Tangled Web,” IMC: Internet Measurement Conference, 2012. [Online]. Available: https://doi.org/10.1145/2398776.2398819

[79] H. Connery, “DNS: Response Policy Zone,” 2012. [Online]. Available: https://dnsrpz.info/spamhaus-rpz-case-study.pdf

[80] J. Zhang, A. Chivukula, M. Bailey, M. Karir, and M. Liu, “Characterization of Blacklists and Tainted Network Traffic,” PAM: International Conference on Passive and Active Network Measurement, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-36516-4_22

[81] M. Kührer, C. Rossow, and T. Holz, “Paint It Black: Evaluating the Effectiveness of Malware Blacklists,” RAID: Research in Attacks, Intrusions and Defenses, 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-11379-1_1

[82] P. Foremski, C. Callegari, and M. Pagano, “DNS-Class: immediate classification of IP flows using DNS,” ACM Internation Journal of Network Management, 2014. [Online]. Available: https://doi.org/10.1002/nem.1864

[83] A. Satoh, Y. Nakamura, Y. Fukuda, K. Sasai, and G. Kitagata, “A Cause-Based Classification Approach for Malicious DNS Queries Detected Through Blacklists,” IEEE Access, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2944203

[84] S. Spacek, M. Lastovicka, M. Horak, and T. Plesnik, “Current Issues of Malicious Domains Blocking,” IFIP/IEEE International Symposium on Integrated Network Management, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8717891

[85] N. Wilde, L. Jones, R. Lopez, and T. Vaughn, "A DNS RPZ Firewall and Current American DNS Practice," ICISA: International Conference on Information Science and Applications, 2019. [Online]. Available: https://doi.org/10.1007/978-981-13-1056-0_27

[86] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, and K. Levchenko, "Reading the Tea leaves: A Comparative Analysis of Threat Intelligence," USENIX Security Symposium, 2019. [Online]. Available: https://www.usenix.org/system/files/sec19-li-vector_guo.pdf

[87] Telenor Norway, "Stanset over 80.000 besøk på falske nettsider på én måned," 2020. [Online]. Available: https://www.mynewsdesk.com/no/telenor/pressreleases/stanset-over-80-dot-000-besoek-paa-falske-nettsider-paa-en-maaned-2986773

[88] H. Griffioen, T. Booij, and C. Doerr, "Quality Evaluation of Cyber Threat Intelligence Feeds," ACNS: International Conference on Applied Cryptography and Network Security, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-57878-7_14

[89] MXToolbox, "Blacklist check," 2021. [Online]. Available: https://mxtoolbox.com/blacklists.aspx

[90] M. Fejrskov, J. M. Pedersen, and E. Vasilomanolakis, "Cyber-security research by ISPs: A NetFlow and DNS Anonymization Policy," International Conference on Cyber Security And Protection Of Digital Services, 2020. [Online]. Available: https://doi.org/10.1109/CyberSecurity49315.2020.9138869

[91] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," IMW: ACM SIGCOMM Internet Measurement Workshop, 2002. [Online]. Available: https://doi.org/10.1145/637201.637225

[92] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A Comprehensive Measurement Study of Domain Generating Malware," USENIX Security Symposium, 2016. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_plohmann.pdf

[93] G. Ládi, "Semantics-Preserving Encryption for Computer Networking Related Data Types," AIS: International Symposium on Applied Informatics and Related Areas, 2017. [Online]. Available: https://www.crysys.hu/publications/files/Ladi2017ais.pdf

[94] Internet Assigned Numbers Authority, "IANA IPv4 Special-Purpose Address Registry," 2020. [Online]. Available: https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

[95] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption," National Institute of Standards and Technology, 2016. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-38G

[96] K. P. Dyer, "FFX," 2018. [Online]. Available: https://github.com/kpdyer/libffx

[97] W. J. Buchanan, "FFX schemes," 2020. [Online]. Available: https://asecuritysite.com/encryption/ffx

[98] M. Fejrskov, "IGA: A python module for format- and semantics preserving encryption and decryption of IP addresses," 2021. [Online]. Available: https://github.com/Fejrskov/IGA

[99] NetworkX, "NetworkX algorithms, k clique communities," 2002. [Online]. Available: https://networkx.org/documentation/stable//reference/algorithms/generated/networkx.algorithms.community.kclique.k_clique_communities.html

[100] Swiss Government Computer Emergency Response Team, "Sage 2.0 comes with IP Generation Algorithm (IPGA)," 2017. [Online]. Available: https://www.govcert.admin.ch/blog/27/sage-2.0-comes-with-ip-generation-algorithm-ipga

[101] T. Aura, "Cryptographically Generated Addresses (CGA)," 2005. [Online]. Available: https://tools.ietf.org/html/rfc3972

[102] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-Preserving Encryption," SAC: Selected Areas in Cryptography, 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-05445-7_19

[103] J. Xu, J. Fan, M. Ammar, and S. Moon, "Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme," IEEE International Conference on Network Protocols, 2002. [Online]. Available: https://doi.org/10.1109/ICNP.2002.1181415

[104] D. Bernát, "Domain Name System as a Memory and Communication Medium," SOFSEM 2008: Theory and Practice of Computer Science, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-77566-9_49

[105] B. Rahbarinia, R. Perdisci1, A. Lanzi, and K. Li, “PeerRush: Mining for Unwanted P2P Traffic,” Journal of Information Security and Applications, 2014. [Online]. Available: https://doi.org/10.1016/j.jisa.2014.03.002

[106] C. Patsakis, F. Casino, and V. Katos, “Encrypted and covert DNS queries for botnets: Challenges and countermeasures,” Computers & Security, 2019. [Online]. Available: https://doi.org/10.1016/j.cose.2019.101614

[107] R. Radu and M. Hausding, “Consolidation in the DNS resolver market – how much, how fast, how dangerous?” Journal of Cyber Policy, 2019. [Online]. Available: https://doi.org/10.1080/23738871.2020.1722191

[108] The ICANN Security and Stability Advisory Committee (SSAC), “SAC 032 - Preliminary Report on DNS Response Modification,” 2008. [Online]. Available: https://www.icann.org/en/system/files/files/sac-032-en.pdf

[109] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, “Comparing DNS Resolvers in the Wild,” IMC: ACM SIGCOMM conference on Internet measurement, 2010. [Online]. Available: http://dx.doi.org/10.1145/1879141.1879144

[110] Google, “Your privacy,” 2021. [Online]. Available: https://developers.google.com/speed/public-dns/privacy

[111] Cisco, “Cisco Umbrella Privacy data sheet,” 2021. [Online]. Available: https://trustportal.cisco.com/c/dam/r/ctp/docs/privacydatasheet/security/umbrella-privacy-data-sheet.pdf

[112] Cloudflare, “1.1.1.1 Public DNS Resolver,” 2020. [Online]. Available: https://developers.cloudflare.com/1.1.1.1/privacy/public-dns-resolver

[113] Quad9, “Data Privacy Policy,” 2021. [Online]. Available: https://www.quad9.net/privacy/policy/

[114] Yandex, “Terms of use of the Yandex.DNS service,” 2021. [Online]. Available: https://yandex.com/legal/dns_termsofuse/

[115] P. Callejo, R. Cuevas, N. Vallina-Rodriguez, and Ángel Cuevas Rumin, “Measuring the Global Recursive DNS Infrastructure: A View From the Edge,” IEEE Access, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2950325

[116] M. Konopa, J. Fesl, J. Jelínek, M. Feslová, J. Cehák, J. Janeček, and F. Drdák, "Using Machine Learning for DNS over HTTPS Detection," European Conference on Cyber Warfare and Security, 2020. [Online]. Available: http://dx.doi.org/10.34190/EWS.20.001

[117] N. Antunes, V. Pipiras, and G. Jacinto, "Regularized inversion of flow size distribution," INFOCOM: IEEE Conference on Computer Communications, 2019. [Online]. Available: https://doi.org/10.1109/INFOCOM.2019.8737406

[118] A. Khormali, J. Park, H. Alasmary, A. Anwar, and D. Mohaisen, "Domain Name System Security and Privacy: A Contemporary Survey," Computer Networks, 2021. [Online]. Available: https://doi.org/10.1016/j.comnet.2020.107699

[119] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson, "Global Measurement of DNS Manipulation," USENIX Security Symposium, 2017. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-pearce.pdf

[120] M. Trevisan, I. Drago, M. Mellia, and M. M. Munafò, "Automatic Detection of DNS Manipulations," IEEE International Conference on Big Data, 2017. [Online]. Available: https://doi.org/10.1109/BigData.2017.8258415

[121] Telecom Industry Association Denmark, "Blokeringer," 2021. [Online]. Available: https://www.teleindu.dk/brancheholdninger/blokeringer-pa-nettet/

[122] Danish Ministry of Justice, "Lov om ændring af retsplejeloven og forskellige andre love," 2017. [Online]. Available: https://www.retsinformation.dk/eli/ft/201612L00192

[123] H. Roberts, E. Zuckerman, J. York, R. Faris, and J. Palfrey, "2010 Circumvention Tool Usage Report," The Berkman Center for Internet & Society, 2010. [Online]. Available: https://cyber.harvard.edu/sites/cyber.harvard.edu/files/2010_Circumvention_Tool_Usage_Report.pdf

[124] A. D. Florio, N. V. Verde, A. Villani, D. Vitali, and L. V. Mancini, "Bypassing Censorship: a proven tool against the recent Internet censorship in Turkey," IEEE International Symposium on Software Reliability Engineering Workshops, 2014. [Online]. Available: https://doi.org/10.1109/ISSREW.2014.93

[125] O. Farnan, A. Darer, and J. Wright, "Analysing Censorship Circumvention with VPNs via DNS Cache Snooping," IEEE

Security and Privacy Workshops (SPW), 2019. [Online]. Available: http://dx.doi.org/10.1109/SPW.2019.00046

[126] The Danish Rights Alliance, "Report On Share With Care 2," 2020. [Online]. Available: https://rettighedsalliancen.dk/wp-content/uploads/2020/06/Report-On-Share-With-Care-2_Final.pdf

[127] A. Hubert and R. van Mook, "RFC 5452: Measures for Making DNS More Resilient against Forged Answers," 2009. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc5452

[128] T. Reddy.K, D. Wing, and P. Patil, "RFC 8094: DNS over Datagram Transport Layer Security (DTLS)," 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8094.html

[129] M. Sivaraman, S. Kerr, and L. Song, "DNS message fragments," 2016. [Online]. Available: https://www.ietf.org/staging/draft-muks-dnsop-dns-message-fragments-00.txt

[130] V. Verónica and R. Gibb, "Adware's new upsell: malware," BSides Calgary, 2016. [Online]. Available: http://dx.doi.org/10.13140/RG.2.1.1218.1365

[131] S. Alrwais, A. Gerber, C. Dunn, O. Spatscheck, M. Gupta, and E. Osterweil, "Dissecting Ghost Clicks: Ad Fraud Via Misdirected Human Clicks," ACM Annual Computer Security Applications Conference (ACSAC), 2012. [Online]. Available: http://dx.doi.org/10.1145/2420950.2420954

[132] J. Baltazar, J. Costoya, and R. Flores, "The real face of KOOBFACE: The largest Web 2.0 botnet explained," Trend Micro Threat Research, 2009. [Online]. Available: http://www.trendmicro.com.ph/cloud-content/us/pdfs/security-intelligence/white-papers/wp_the-real-face-of-koobface.pdf

[133] G. Ye, "70+ different types of home routers(all together 100,000+) are being hijacked by GhostDNS," Netlab 360, 2018. [Online]. Available: https://blog.netlab.360.com/tag/ghostdns/

[134] Z. Yin, "Domain Resolution in LAN by DNS Hijacking," International Conference on Computer Engineering and Networks (CENet), 2020. [Online]. Available: https://doi.org/10.1007/978-981-15-8462-6_98

[135] J. Galloway, "AirBNBeware: Short term rentals, long term pwnage," Blackhat, 2016. [Online]. Available: https://www.blackhat.com/us-16/briefings/schedule/#airbnbeware-short-term-rentals-long-term-pwnage-2891

[136] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," IMC: Internet Measurement Conference, 2015. [Online]. Available: http://dx.doi.org/10.1145/2815675.2815683

[137] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," Springer Series in Statistics, 2009. [Online]. Available: https://doi.org/10.1007/978-0-387-84858-7

[138] R. Houser, S. Hao, Z. Li, D. Liu, C. Cotton, and H. Wang, "A Comprehensive Measurement-based Investigation of DNS Hijacking," International Symposium on Reliable Distributed Systems (SRDS), 2021. [Online]. Available: https://cpb-us-e2.wpmucdn.com/faculty.sites.uci.edu/dist/5/764/files/2021/10/srds21.pdf

[139] D. Dagon, N. Provos, C. P. Lee, and W. Lee, "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority," Network and Distributed System Security Symposium, 2008. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/Corrupted-DNS-Resolution-Paths-The-Rise-of-a-Malicious-Resolution-Authority-paper-David-Dagon.pdf

[140] J. Park, R. Jang, M. Mohaisen, and D. Mohaisen, "A Large-Scale Behavioral Analysis of the Open DNS Resolvers on the Internet," IEEE/ACM Transactions on Networking, 2021. [Online]. Available: https://doi.org/10.1109/TNET.2021.3105599

[141] C. A. Shue and A. J. Kalafut, "Resolvers Revealed: Characterizing DNS Resolvers and their Clients," ACM Transactions on Internet Technology, 2013. [Online]. Available: http://dx.doi.org/10.1145/2499926.2499928

[142] B. Liu, C. Lu, H. Duan, Y. Liu, Z. Li, S. Hao, and M. Yang, "Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path," USENIX Security Symposium, 2018. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-liu_0.pdf

[143] M. Fejrskov, E. Vasilomanolakis, and J. M. Pedersen, "A study on the use of 3rd party DNS resolvers for malware filtering and censorship circumvention," To appear in International Conference on ICT Systems Security and Privacy Protection (IFIP SEC), 2022.

[144] C. Huang, P. Zhang, Y. Sun, Y. Zhu, and Y. Liu, "SFDS: A Self-Feedback Detection System for DNS Hijacking Based on Multi-Protocol Cross Validation," International Conference on Telecommunications (ICT), 2019. [Online]. Available: https://doi.org/10.1109/ICT.2019.8798832

[145] Google, "Top publications," 2022. [Online]. Available: https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_technologylaw

[146] The Computing Research and Education Association of Australasia, CORE, "Core conference portal," 2022. [Online]. Available: http://portal.core.edu.au/conf-ranks/?search=4604&by=all&source=all&sort=arank

[147] G. Gu, "Computer Security Conference Ranking and Statistic," 2021. [Online]. Available: https://people.engr.tamu.edu/guofei/sec_conf_stat.htm

[148] The European Parliament and of the Council, "Proposal for ePrivacy Regulation," 2017. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52017PC0010

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-905-9

AALBORG UNIVERSITY PRESS