Master of Science in Engineering: Computer Security
June 2021

Collecting and analyzing Tor exit node traffic

Torbjörn Jonsson

Gustaf Edeby

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Engineering: Computer Security. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Torbjörn Jonsson
E-mail: tojf14@student.bth.se

Gustaf Edeby
E-mail: gued15@student.bth.se

University advisor:
Nurul Momen
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet	:	www.bth.se
Phone	:	+46 455 38 50 00
Fax	:	+46 455 38 50 57

Abstract

Background. With increased Internet usage occurring across the world journalists, dissidents and criminals have moved their operations online and in turn, governments and law enforcement have increased their surveillance of their country's networks. This have increased the popularity of programs masking users' identities online such as the Tor Project. By encrypting and routing the traffic through several nodes, the users' identity is hidden. But how are Tor users' utilizing the network, and is any of it in plain text despite the dangers of it? How has the usage of Tor changed compared to 11 years ago?
Objectives. The thesis objective is analyzing captured Tor network traffic that reveals what data is sent through the network. The collected data helps draw conclusions on Tor usage and is compared with previous studies.
Methods. Three Tor exit nodes is set up and operated for one week in the US, Germany, and Japan. We deploy packet sniffers performing a deep packet inspection on each traffic flow to identify attributes such as application protocol, number of bytes sent in a flow, and content-type if the traffic was sent in plain text. All stored data is anonymized.
Results. The results show that 100.35 million flows were recorded, with 32.47% of them sending 4 or fewer packets in total. The most used application protocol was TLS with 55.03% of total traffic. The HTTP usage was 15.91% and 16% was unknown protocol(s). The countries receiving the most traffic were the US with over 45% of all traffic, followed by the Netherlands, UK and Germany with less than 10% of recorded traffic as its destination. The most frequently used destination ports were 443 at 49.5%, 5222 at 12.7%, 80 with 11.9% and 25 at 9.3%.
Conclusions. The experiment shows that it is possible to perform traffic analysis on the Tor network and acquire significant data. It shows that the Tor network is widely used in the world but with the US and Europe accounting for most of the traffic. As expected there have been a shift from HTTP to HTTPS traffic when comparing to previous research. However, there is still unencrypted traffic on the network, where some of the traffic could be explained by automated tools like web crawlers. Tor users' needs to increase their awareness in what traffic they are sending through the network, as a user with malicious intent can perform the same experiment and potentially acquire unencrypted sensitive data.

Keywords: Deep packet inspection, Traffic analysis, Tor network, The Onion router

Sammanfattning

Bakgrund. Med ökat Internet användande i världen så har journalister, dissidenter och kriminella flyttat sina operationer till Internet. Som följd har regeringar och myndighetsorganisationer ökat sitt övervakande av deras lands nätverk. Detta har ökat populariteten av program som maskerar användarens identitet som till exempel Tor projektet. Genom att kryptera och skicka trafiken genom flera noder är användaren skyddad från att bli identifierad. Men vad skickar Tor användare för trafik genom Tor nätverket, och är en del av trafiken i klartext trots riskerna? Hur har Tor användningen ändrats jämfört med 11 år sedan?
Syfte. Målet med avhandlingen är att analysera Tor trafik. Detta kommer att ge värdefull information om Tors användning. Den insamlade datan kommer att jämföras med tidigare Tor analyser.
Metod. Tor exit noder kommer att drivas i USA, Tyskland och Japan med avlyssningsprogram som gör en djup paketinspektion på varje trafikflöde för att detektera attributer såsom applikationsprotokollet, hur många byte skickas i ett flöde och vad dess innehållstyp är om trafiken skickas i klartext. All sparad data anonymiseras.
Resultat. Resultaten visar att 100,35 miljoner trafikflöden samlades in, där 32.47% skickade 4 eller mindre paket totalt. Det mest använda protokollet var TLS med 55,03%, följt av HTTP med 15,91% och okända protokoll på 16%. Länderna som tog emot mest trafik var USA med över 45% av all trafik, följd av Nederländerna, Storbritannien och Tyskland med under 10% av all trafik vardera. Dom mest frekvent använda destinationsportarna var 443 med 49,5%, 5222 med 12,7%, 80 med 11,9% och 25 med 9,3%.
Slutsat. Experimentet visar att det är möjligt att göra en trafik analys av Tor nätverket som ger värdefull data. Tor nätverket används i de flesta delar av världen, men USA och Europa står för de största delarna av trafiken. Som förväntat såg vi ett skifte från HTTP till HTTPS trafik jämfört med tidigare forskning. Dock finns det fortfarande okrypterad trafik på nätverket, där delar av den kan förklaras av automatiska verktyg som web crawlers. Tor användare måste bli mer medvetna om vad för trafik de skickar igenom nätverket, då en användare med illvilliga mål kan återskapa denna studien och potentiellt se okrypterad trafik.

Nyckelord: Djup paketinspektion, trafikanalys, Tor nätverk

Acknowledgments

We would like to thank Nurul Momen from Blekinge Institute of Technology and Davide Girardi from Truesec AB for their invaluable help in the development of this master thesis. We would also like to thank City Network AB for contributing with infrastructure and technical support.

Additional notes

- This paper is written in the authors' second language and the paper have not been reviewed by a professional reviewer. The paper might contain grammatical or spelling errors.
- Repeating Author names in the bibliography is dashed.
- The study is done during the Covid-19 pandemic.

Contents

Chapter 1

Introduction

The Internet is the way of communication today. People, businesses and entire countries rely on the Internet for their jobs and past-time. According to Our World in Data, there were 413 million users' of the Internet in the year 2000 but had grown to over 3.4 billion by 2016 [40]. The Internet's popularity have led factions to move their operations online such as journalists, dissidents in authoritarian countries and criminals all using the Internet for their own goals.

This in turn has led governments and law enforcement to increase their surveillance of the Internet despite more internet traffic being sent through the encrypted HTTPS (HyperText Transfer Protocol Secure) protocol instead of plain-text traffic using HTTP (Hypertext Transfer Protocol) as Cisco [33] and Google [18] reports. This is thanks to other methods of identification such as knowing what websites you visit, cookies and ads that track your browser and fingerprinting, which collects information based on your browser and device. These methods give dictatorships the ability to silence critics, protesters and restrict certain websites such as Facebook or Amnesty. This surveillance increased the popularity of several programs which mask its users' traffic and prevents user identification. The most popular of them is the Tor browser.

Tor is a free, open-source software that allows a user to conceal his or her location on the Internet and add extra encryption to your traffic. Since the increased usage of the Internet in the 1990s, scientists have hypothesized if there is a way to communicate on the Internet safely, even with the traffic being monitored and tracked. The result was Tor, short for The Onion Routing project. Developed by US researchers in 1995 [46] at the U.S Naval Research Laboratory, the Tor project was released in October 2002 as a free and open-source client, along with the code to operate a node, to allow full transparency of the anonymizing process and encourage different entities to use the network to conceal U.S intelligence communications online. In 2006 a non-profit organization called the Tor Project was created by its founders to maintain Tor. Today the Tor Project stands for preserving the right to privacy on the Internet and protecting the freedom of speech. It allows users' to hide their identities online and access websites blocked by their countries government, among other usages, as described in The Tor Project website [44].

Despite Tor's popularity and importance, there is not a lot of studies on its users' behavior due to ethical and privacy concerns. What protocols are sent through the nodes? How have they changed over time? Which countries are the biggest destinations of Tor traffic and why? In this thesis, we will collect Tor exit node traffic leaving the network in a way that preserves security and personal data integrity, before subjecting it to analysis and comparing it with previous studies.

Terminology

Relay / Node - This is the router that forwards traffic. These exist in several forms.

Circuit - The collection of nodes that creates the path through the network. Explained by Dingledine et al. [14].

Exit node - The last node in the path where outgoing traffic is sent to its intended destination.

Guard node / Entry node - A guard node is the first relay in the circuit. This is the entry point into the Tor network. Because of this, it is also sometimes referred to as an *entry node.*

Middle node - The node between the entry and exit node. Depending on the number of hops several middle nodes can exist in a circuit.

Bridge node - These nodes are used to obfuscate Tor traffic so that the traffic is harder to detect and block.

These nodes are described by the Australian Cyber Security Center [3] and on the Tor project website [55].

Traffic flow - Organizes packets into flows based on shared identifiers such as port, protocol, source and destination IP addresses [34].

Network port / Port - A port is a number that identifies specific applications and services. Applications can use ports to listen to connection requests [12].

1.1 Problem description and research gap

The main problem is how to capture and analyze Tor users' traffic while preserving their integrity. The Tor network, despite its popularity, have not seen much research into user behavior and what protocols are sent through the network, due to the ethical and legal issues in operating Tor nodes and analyzing user data. The latest Tor traffic analysis was made by Mani et al. in 2018 [31]. However, this research does not provide the information that we are interested in. Therefore our research is more comparable to Chaabane et al. [11]. Several years of technological development and increased Internet usage have likely changed the composition of Tor traffic. This is a research gap that we wish to address.

1.2 Aim, objectives and research questions

The thesis objective is to capture outgoing traffic from the Tor exit nodes, analyze it and draw conclusions based on the collected data and compare it with previous research. The nodes will give us insight into how the Tor network is being used today compared to previous experiments done in the past.

We will create a framework that collects, parses and anonymizes the Tor traffic data. A discussion regarding the ethical and legal concerns in the setup and running of Tor exit nodes will take place as we want to provide guidelines and inform about the potential consequences of operating Tor nodes. The ethical concerns are something Chaabane et al. [11] discusses. The Tor Research Safety Board has guidelines on how to study the Tor network that will be considered [56].

Goal

- Perform an analysis of outgoing Tor exit node traffic.

Sub-goals

- Develop a method that gathers, collects and parses data from a Tor exit node.
- Compare the analyzed data with previous Tor traffic studies.
- Discuss ethical and legal concerns when operating a Tor node.

We consider the legal aspects both to know that the experiment is legal to perform and also to gain knowledge about the laws regarding hosting Tor exit nodes and gather network data.

Research Questions (RQ)

- **RQ1**: How can Tor exit node traffic be systematically collect, classify and analyze in an efficient and secure way?

  Clarification: The collection, classification and analysis of Tor exit node traffic are vital for the thesis results. The data must be collected systematically. The classification of what data flows through the node must be determined by a method with a high enough accuracy to generate valid results. The results can be used to perform an analysis and draw conclusions. This must be done efficiently in the way that the method does not need extreme amounts of hardware capacity and can be used on a traditional PC. The system must also be secure in the sense that when using the method the node owner and the users' using the exit node will not be vulnerable to attacks or data loss.

- **RQ2**: How does the collected data compare to previous Tor network analysis?

  Clarification: To understand how the Tor network has changed over time we compare the data with an analysis made by Chaabane et al. [11]. Their study was done in 2010 and during the years the Tor network and how it is used has changed. The comparison will highlight these changes.

1.3 Scope and limitations

Three exit nodes will be created and operated. One in North America, one in Europe, and one in Asia. The limited number of nodes and geographical constraints might affect the composition of data and accuracy when compared to the real usage of Tor worldwide. The number was determined by economical factors, and the nodes spread out across the world will give us as accurate data as possible.

Chabaane et al. used 6 nodes running for a total of 23 days with a 100 KB/s bandwidth, and also created entry nodes to record users' geographical entry location [11]. Our exit nodes will operate for 7 days due to time constraints, but with the bandwidth of 2 MB/s to allow as much data to be collected as possible, and still be comparable to other studies. Our study will not operate entry nodes due to economical constraints.

Due to ethical limitations on data collected, plain text traffic will not be subjected to complex analysis such as determining if passwords, email, hacking attempts, etc. are sent in plain text.

1.4 Ethical and societal aspects

Collecting data traffic brings many ethical issues, most notably there is no way to get consent from users' before collecting the data in this case. The Tor project's stated goal is the preserving of privacy on the internet, which makes it more difficult to justify collecting data. However, we believe this study on Tor usage will help shine a light on how the Tor network is being used. This will provide information to both scientists and users' of Tor. The results will show whether Tor users' understand the need for encrypted traffic even outside of the Tor circuit, and will help increase the security of the Tor network. Scientists may use our results and our framework as a basis for future studies.

The thesis experiment involves collecting internet traffic, which is classified as personal data according to the General Data Protection Regulation (GDPR) law created by the European Union [36], which is explained in the *General guide to GDPR compliance* in [38]. Due to this legislation, extra measures will be taken to preserve the integrity of Tors users'. These are:

- Automate the parsing process to avoid viewing private data.
- Parsing and anonymizing data before long-term storage.
- Deleting unnecessary data after parsing.
- Receiving non-binding ethical advice from an ethical committee.
- Limiting access to collected data.

An opinion request regarding our work was sent to the Ethical Advisory Board in South East Sweden [29], jointly run by Lund University and Blekinge Institute of Technology. In the request, we described the goal of analyzing user data and our steps to anonymize and secure the data. The advice received from the board was to prepare for the handling of non-encrypted data since we had indicated that plain-text traffic might be captured. Besides this, they detected no difficulties in conducting the study as it was more a technical study than one handling personal data.

We hope by discussing ethical and legal issues with running a Tor node will help the development of new exit nodes. This will help the legitimate users' of Tor who are using it for its intended purpose, preserving privacy online and supporting the freedom of speech. By performing analysis on Tor traffic, we wish to enlighten users' to what user data can be accessed despite being routed through Tor.

1.5 Thesis outline

The report is divided into the following chapters. Chapter 2: *Background* describes the technical research needed for the experiment. Chapter 3: *Related work* discusses related studies and how we address a research gap. Chapter 4: *Method* provides our thesis sub-objectives and the method for the literature review and experiment. Chapter 5: *Results and Analysis* contains our results, its analysis and comparison to previous research. Chapter 6: *Discussion* discusses the results and possible validity threats. Chapter 7: *Conclusions and Future work* describe our conclusions and ideas for future studies based on our work.

Chapter 2

Background

In this chapter, we explain the knowledge needed to understand the thesis. This includes what the Tor network is, how it works, why it is useful and what purpose it has. Furthermore, we explain potential risks regarding using the Tor network and contributing to the network with nodes. Lastly, we talk about the tools used to extract and parse traffic data.

2.1 The Tor network fundamentals

The Tor network was created to give users' the possibility to use the internet privately without being surveilled. Nodes in the Tor network together create a circuit. This is the path through the network.

When a packet goes through the Tor network it will be encrypted in layers. The browser generates separate encryption keys for each hop in the path. The entry node knows the origin of the traffic and decrypts the first header to know what middle node to send the traffic to. The middle node only knows the node the traffic is coming from, decrypts the next header and sends it to the exit node, which is the last node in the path. The exit node decrypts the final header and sends the traffic to the intended destination. No node knows the complete path the traffic is taking. The traffics destination only knows what exit node the traffic is coming from, and thus the users' location is concealed, according to The Tor Project's website [44]. Figure 2.1 shows a simple flow of Tor traffic traveling through the network.

When establishing the communication. The exit node is the first node to be chosen. The selection process follows some principles. It checks if the client has specified anything in its *torrc* file. This file is the client's configuration file for the Tor network and it can for example specify what nodes not to use. The exit node that is chosen needs to let your traffic exit the network i.e it needs to allow that specific traffic. Some exit nodes block ports of the node e.g if you want to send mail you can not use an exit node that has blocked port 25 where SMTP (Simple Mail Transfer Protocol) traffic goes through.

Tor tries to find an exit node with enough traffic capacity to handle the throughput necessary for that instance. Additionally, all circuits have some constraints. First,

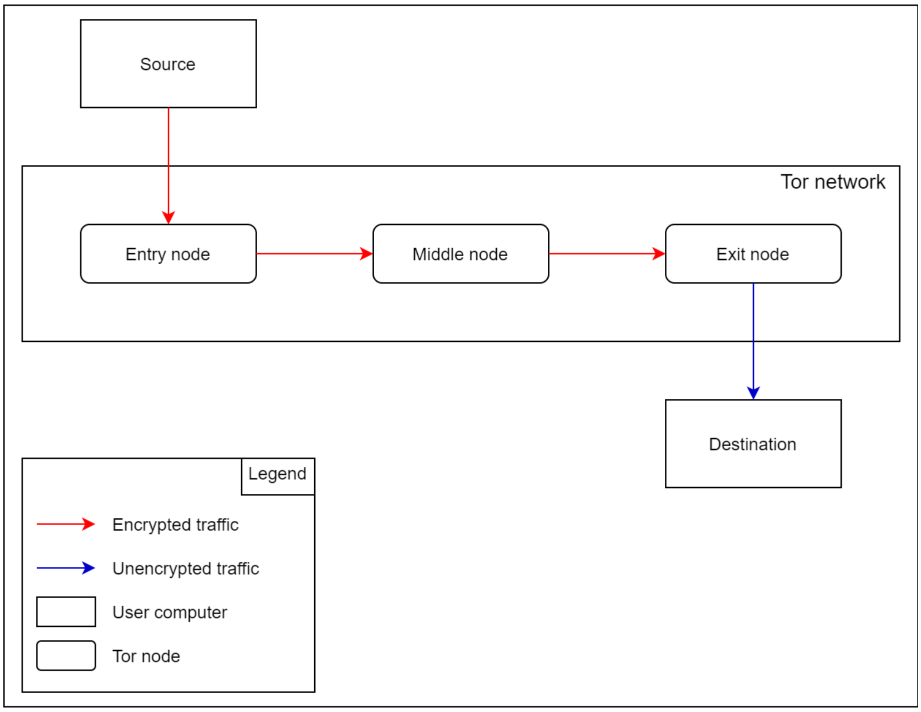

Figure 2.1: Diagram of how Tor works. The arrows indicates the flow of the traffic.

the same node can not be used two times in the same circuit. Two nodes from the same family can not be part of the same circuit. If someone operates multiple nodes close together they can define them as a family. The nodes do not share the same /16 subnet. Non-valid and non-running nodes will not be selected. Non-valid nodes are nodes that are not configured correctly. The first node must be a guard node. These nodes are privileged entry nodes that get selected to be the start of a circuit [43] [14].

Tor works by operating several thousand nodes across the world. When a user connects to the Tor browser, a list of active nodes is downloaded from a Tor directory server. A path between these nodes are semi-randomly generated with a default of n nodes needed to create a path (with a default of n=3 nodes which is the minimum amount). These nodes are selected using the aforementioned principles, and the exit node's location is weighted towards the traffic's destination.

2.2 Weakness of the Tor network

Over the years the Tor network have been tested in many different ways. Partly by people trying to break the anonymity but also by researchers trying to understand the network and its limitations better. We further discuss this in the related work Chapter 3. The Tor project organization is constantly working to make the Tor network better, but is important to understand that it is not a perfect system and has weaknesses. Understanding these weaknesses is important because with the knowledge of its limitations comes security.

Hayes [21] describes two different types of traffic confirmation attacks, active and passive. Active traffic confirmation attacks are when an attacker sets up multiple Tor nodes, both exit and entry nodes. They modify the entry node headers to include some additional information and look for that information on the exit nodes. If the exit nodes find information from the entry node they can see where the packet comes from and where it was going even without knowledge about the middle nodes. A passive confirmation attack eavesdrops on exit and entry nodes to try to link them together without changing the traffic.

Some countries try to block the Tor network by blocking the IP addresses of the relays run by volunteers. These are publicly available and provided by the Tor organization. To bypass this the Tor project have created special relays that users' can connect to. These relays are called bridges and do not have publicly available addresses. This does not however stop these countries from monitoring traffic to identify it as Tor traffic. This can be done with deep packet inspection (DPI). DPI allows these states to recognize the traffic even if it is from an unknown IP address or if it is encrypted. Other methods include stochastic packet inspection, flow level inspection, circuit-level inspection, and different machine learning methods. This is described by Khalid Shahbar and A. Nur Zincir-Heywood [41].

To mitigate this the Tor project organization have created Pluggable Transports. Pluggable Transports changes the traffic so it can not be identified as Tor traffic. Shahbar and Zincir-Heywood describe several ways to do this [41]. Flashproxy is one and it uses JavaScript to continuously change IP addresses. These IP addresses then help users' access the Flashproxy supported websites. These websites include a Flashproxy Javascript code snippet. Fifield et al. [15] further explain how Flashproxy works. ScrambleSuits is another one that protects against follow-up probing attacks and it can change the network's fingerprint. More examples exist on the Tor projects website [45]. All of these different methods try to obfuscate or hide the fact a connection is using the Tor network, to make it harder for organizations and countries to identify and consequently block the Tor network.

Sometimes it's not the Tor network that is the problem, instead, it is the user that is the reason for the loss of privacy. Operational security or OpSec is a risk management process that gives an understanding of what your actions may have on the impact on security and anonymity. This is explained by Zhang [58]. If a user wants to remain anonymous they need to have a good practice of how to use the Tor network and the internet overall. This includes identifying sensitive information and not sharing this information. The Tor browser has three different levels of security [49], standard, safer, and safest. The standard level includes all Tor browser and website features. The safer level disables javascript for non-HTTPS sites and website features that are often dangerous. The safest option disables javascript by default on all sites, this level also only allows website features for static sites with very limited services. So if your main goal is to be anonymous to the greatest extent these are things to understand and consider.

It is also important to understand that Tor only encrypts the traffic through the Tor network. That means if the traffic is not encrypted on the application layer e.g HTTP, the traffic can be captured before the traffic enters and after it leaves the Tor network. This is explained on the Tor project FAQ site [43].

2.3 Considerations before setting up an exit node

Before setting up a tor node consider the following questions provided by The Tor Project[47].

Do you want to run a Tor exit or non-exit (bridge/guard/middle) relay? Running an exit node has difficulties because when the traffic goes through the network it is encrypted until it reaches the exit node, after that point the traffic is not encrypted by Tor. This results in it looking like traffic coming through the Tor network comes from the exit node. This can be problematic because the Tor node owner will be seen as responsible and receive abuse complaints. Some internet service providers and law enforcement may be provoked and thinking that the Tor node owner is carrying out malicious or illegal acts, even though they are not responsible for the traffic. The exit nodes are for this reason more needed as exit nodes are generally in short supply due to the reasons mentioned, although all nodes contribute to the network. It is not recommended to run an exit node on your private internet connection.

If you want to run an exit relay, *Which ports do you want to allow in your exit policy?* (More allowed ports usually means potentially more abuse complaints). When you set up an exit node you can choose what traffic flows through by blocking ports. For example, if you block port 25 you block traffic that relies on port 25. The Tor organization provides more information on Reduced exit policy good practice [16].

What external TCP (Transmission Control Protocol) port do you want to use for incoming Tor connections? "ORPort" configuration: We recommend port 443 if it is not used by another daemon on your server. ORPort 443 is recommended because it is often one of the few open ports on public WIFI networks. Port 9001 is another commonly used ORPort.

What email address will you use in the Contact info field of your relay(s)? This information will be made public. This email address will receive the abuse complaints and be the contact information for the node. As such an anonymous email address is highly recommended.

How much bandwidth/monthly traffic do you want to allow for Tor traffic? When configuring the node you can choose how much bandwidth the node will have available. Setting this slightly lower than the max bandwidth will make the node more reliable.

Does the server have an IPv6 (Internet Protocol version 6) address? This is to simply know if you should enable IPv6 on the node. For this experiment, IPv6 will not be used.

2.4 Traffic analysis methods

This section describes tools and techniques used to perform traffic analysis on networks.

What is Deep Packet Inspection?

Deep Packet Inspection is a technique to analyze network packets for different purposes such as filtering, blocking or redirecting traffic. Deep packet inspection looks at headers and protocol fields to gain information about the package, even if the package's application data is encrypted. Unlike shallow packet inspection that only performs an inspection on the top-level layers, DPI uses all layers to gain enough information about the package to determine its application protocol. DPI was previously not available due to computers having too little computing power to allow DPI in real-time as discussed by Knapp and Langill in [27]. DPI is quickly gaining popularity in cybersecurity modules as explained by Rodrigues et al. in [39].

Network interface data dump

A network interface is the point between the computer and a private or public network and works as a middleman between them, translating signals back and forth to allow communication between the two entities. The interface can be a physical network interface card (NIC), or it can be software such as the loopback address *127.0.0.1.* To extract data from a network interface you use a packet analyzer, which is a program designed to intercept and log both incoming and outgoing traffic. The traffic is stored in either *pcap* or *pcap-ng* file format. Modern computers can have many interfaces, such as an ethernet cable connection and a WIFI connection at the same time. By allowing the packet analyzer access to one or several of these interfaces, you can capture the traffic coming in and out of your computer. The traffic capturing can be done both by printing or dumping the traffic into files as the open-source program *tcpdump* does. It can also be captured directly by a program handling the data, with no raw traffic data being stored on the computer.

Chapter 3

Related Work

This chapter describes the process of finding and collecting related work through a literature mapping study, and a summary of our findings.

3.1 Literature mapping study

The literature mapping study was needed to gain knowledge about the current research that existed on the subject. A mapping study was chosen as the best alternative given the time constraints on the project. An alternative would have been a systematic literature review, but was deemed too resource demanding. We constructed a research query based on the three keywords *tor*, *traffic* and *analysis*. Originally we used BTH Summon [7] which consists of several databases such as IEEE, O'Reilly, Springer, etc. but after the mapping was completed we expanded to other databases to make sure we did not miss relevant articles. We constructed a new search query and applied it to the IEEE [24] and ACM Digital Library [2] databases. The results of the mapping study can be viewed in figure 3.1. The results gave a total of 29 relevant articles.

The background review started by identifying the necessary technical techniques needed. These where:

- Setting up a Tor exit node.
- Capture all incoming and outgoing traffic.
- Perform DPI parsing on the captured traffic and store the data.
- Anonymizing data.
- Analyze the data after experiment completion.

We searched the Internet for open-source programs and modules that could perform these operations in a Linux environment.

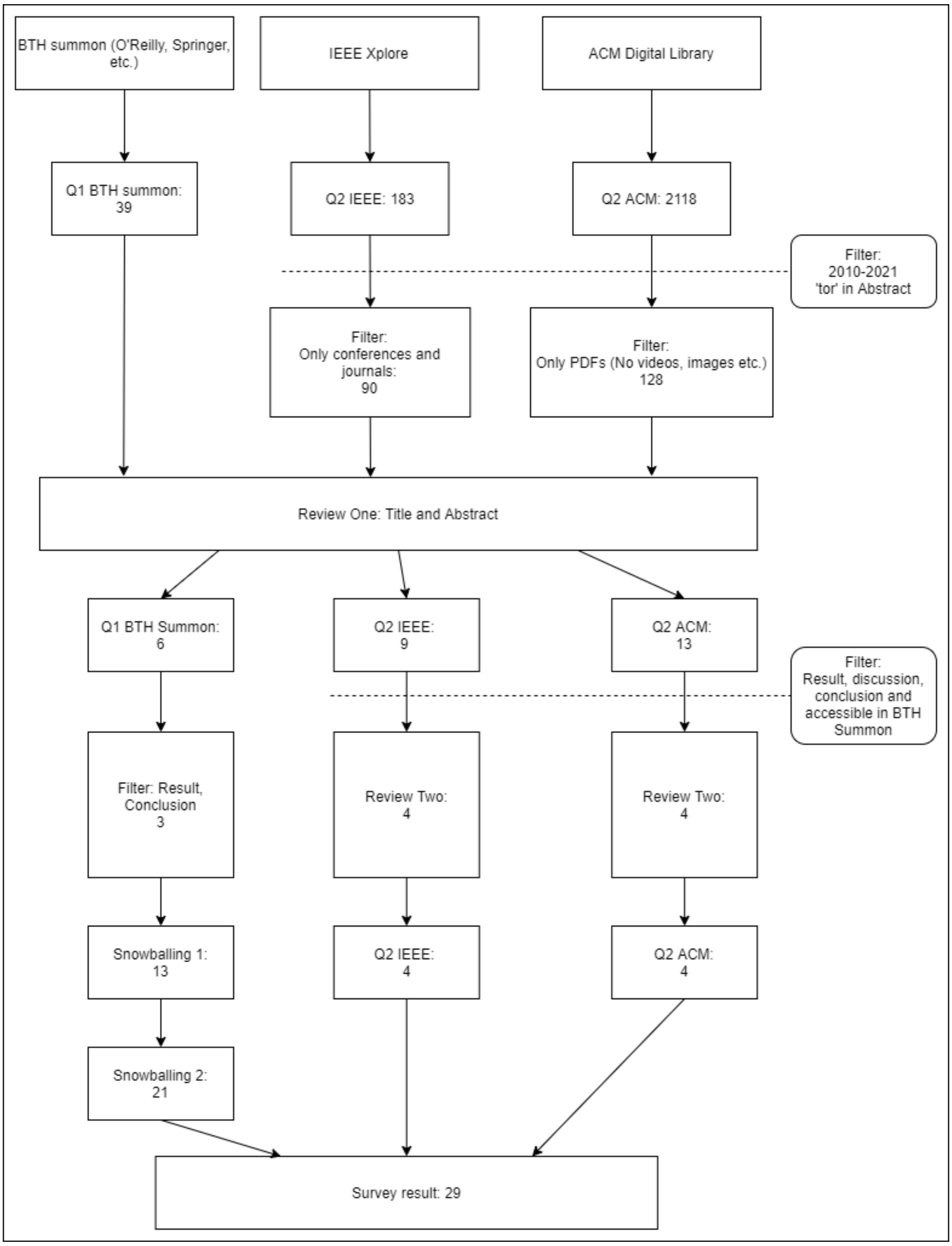
BTH summon (O'Reilly, Springer, etc.)
IEEE Xplore
ACM Digital Library
Q1 BTH summon: 39
Q2 IEEE: 183
Q2 ACM: 2118
Filter: 2010-2021 'tor' in Abstract
Filter: Only conferences and journals: 90
Filter: Only PDFs (No videos, images etc.) 128
Review One: Title and Abstract
Q1 BTH Summon: 6
Q2 IEEE: 9
Q2 ACM: 13
Filter: Result, discussion, conclusion and accessible in BTH Summon
Filter: Result, Conclusion 3
Review Two: 4
Review Two: 4
Snowballing 1: 13
Q2 IEEE: 4
Q2 ACM: 4
Snowballing 2: 21
Survey result: 29

Figure 3.1: Diagram of the mapping study process.

3.2 Literature mapping study results

Table 3.1 shows an overview of the studies that are the most relevant to this study.

Author	Title	Year	Comment
Dingledine et al. [14]	Tor: The Second-Generation Onion Router	2004	Explains the Tor network.
A. Chaabane et al. [11]	Digging into Anonymous Traffic:a deep analysis of the Tor anonymizing network	2010	Tor analysis by monitoring Tor nodes.
Groš et al. [20]	Protecting TOR exit nodes from abuse	2010	Propose a honeywall method to increase the security.
Jensen et al. [25]	Recruiting New Tor Relays with BRAIDS	2010	Discusses the tor networks community-driven aspect and its problems.
Barker et al. [5]	Using traffic analysis to identify TheSecond Generation Onion Router	2011	Explores the identification of Tor traffic.
Song et al. [42]	A De-anonymize attack methodbased on traffic analysis	2013	Tests the Tor networks unlinkability between sender and reciver.
Ling et al. [28]	TorWard: Discovery, Blocking, and Traceback of Malicious Traffic Over Tor	2015	Uses TorWard to detect and block malicious traffic in the Tor network.
Jansen and Johnson [26]	Safely Measuring Tor	2016	Builds a tool that safely measures Tor.
Huang and Bashir [23]	The Onion Router: Understanding a PrivacyEnhancing Technology Community	2016	Conduct a survey that explores the motivation of Tor relay volunteers and their opinion on anonymous networks.
Mani et al. [31]	Understand-ing Tor Usage with Privacy-Preserving Measurement	2018	Uses previous tools to understand how much the tor network is used.
Basyoni et al. [6]	Traffic Analysis Attackson Tor: A Survey	2020	Analyzes different attacks on the Tor network and evaluate how practical they would be.

Table 3.1: Overview of mapping study results.

The research on the Tor network and similar anonymous networks is limited, despite its relevance and continued growth. Dingledine et al. [14] published a paper explaining the Tor network. They described the limitations and trade-offs the Tor network had at the time. They created a threat model and gave examples of how it could be exploited.

Groš et al. [20] propose a honeywall method to increase the security of Tor exit nodes and to stop abuse issues. Chaabane et al. [11] describe a method to increase the security of a Tor node by detecting connections that exploit the exit node as a Tor tunnel. TorWard is a technique created by Ling et al. [28] that uses intrusion detection systems (IDS) to detect and block malicious traffic. They evaluate their system both theoretical and with real-world experiments to validate the feasibility and the effectiveness of their system.

Chaabane et al. [11] studied how the Tor network was used in 2010. In that paper, they showed that a lot of the traffic through Tor was BitTorrent traffic. They showed how to abuse the exit nodes to work like 1-hop proxies, meaning that they used tunneling to only use the exit nodes bypassing the Tor fundamentals. Exit nodes were vulnerable to crawling techniques. Geopolitical data on what countries the relays and the top clients came from could be shown amongst other data points by using deep packet inspection to analyze the traffic. Basyoni et al. [6] analyzes different attacks on the Tor network and evaluate how practical they would be.

Barker et al. [5] explore the identification of Tor traffic, with a focus on the encrypted traffic. They show in an experiment that even encrypted or obfuscated Tor traffic can be identified. Their experiment is conducted in a simulated environment and they discuss that real Tor traffic can be vulnerable to this kind of analysis. Song et al. [42] look to identify the flow through the network to deanonymize the traffic, testing the fundamentals of the network. The focus is on testing the unlinkability between the sender and the receiver. They utilized machine learning to perform the data analysis.

Huang and Bashir [23] talk about the increasing surveillance that countries and organizations conduct on people, and how this is used against them in different ways. The paper conducts a survey that explores the motivation of Tor relay volunteers and their opinion on anonymous networks. They show that there are both extrinsic and intrinsic motivational reasons for why they operate relays. Examples of this are that they enjoy privacy problem solving and want to enhance the network so it works better. Jensen et al. [25] express the problem with Tor that all relays are run by the community and the importance of them for the functionality of the network. They present a method to encourage more users' to run a relay by implementing a reward system.

Jansen and Johnson [26] created a tool that safely measures Tor. Their purpose was to build, test and evaluate a tool that will help understand how the Tor network is used. The study was done in 2016 and at that time daily usage of Tor was 1.75 million users' [26]. Two years later Mani et al. [31] did a more comprehensive study that included Jansen and Johnson's tool [26]. They wanted to know who and how many users' are using the network, how many Tor onion services exist, which are popular and how these are used. They discussed the challenges when measuring Tor as well as the ethics regarding it. They use proof of concept prototypes that measure Tor and improve them. They show that the Tor network is used to a much greater extent than previously estimated.

These articles show the increase in popularity the Tor network has seen in later years. They also show different ways the network can be vulnerable. It is important to understand the security and possible vulnerabilities both as an operator of a Tor node and as a user of the Tor network. With this knowledge, they can understand in what way they might be vulnerable and how they can mitigate these vulnerabilities. Our study is important to anyone interested in understanding the Tor network and wants to analyze its traffic.

Our research will differ from Mani et al. [31] in the way we collect data and what attributes we collect. For this reason, our study will be more similar to Chabaane et al. [11]. Their research was done in 2010 and much has changed since then. New tools to collect and analyze data have been developed and the Tor network has been continuously updated. This opens a research gap that we will try to fill. We wonder how this method works 11 years later, how the results differ, what changes have been made to the Tor network, how have the capturing tools changed and what is needed to conduct this kind of study.

Chapter 4

Method

This chapter will describe how we conducted our research to answer our research questions. All required steps, except for the literature mapping study in Chapter 3: Related works, are described in this chapter. We explain the motivations behind the chosen methods and their alternatives. Figure 4.1 visualizes these steps.

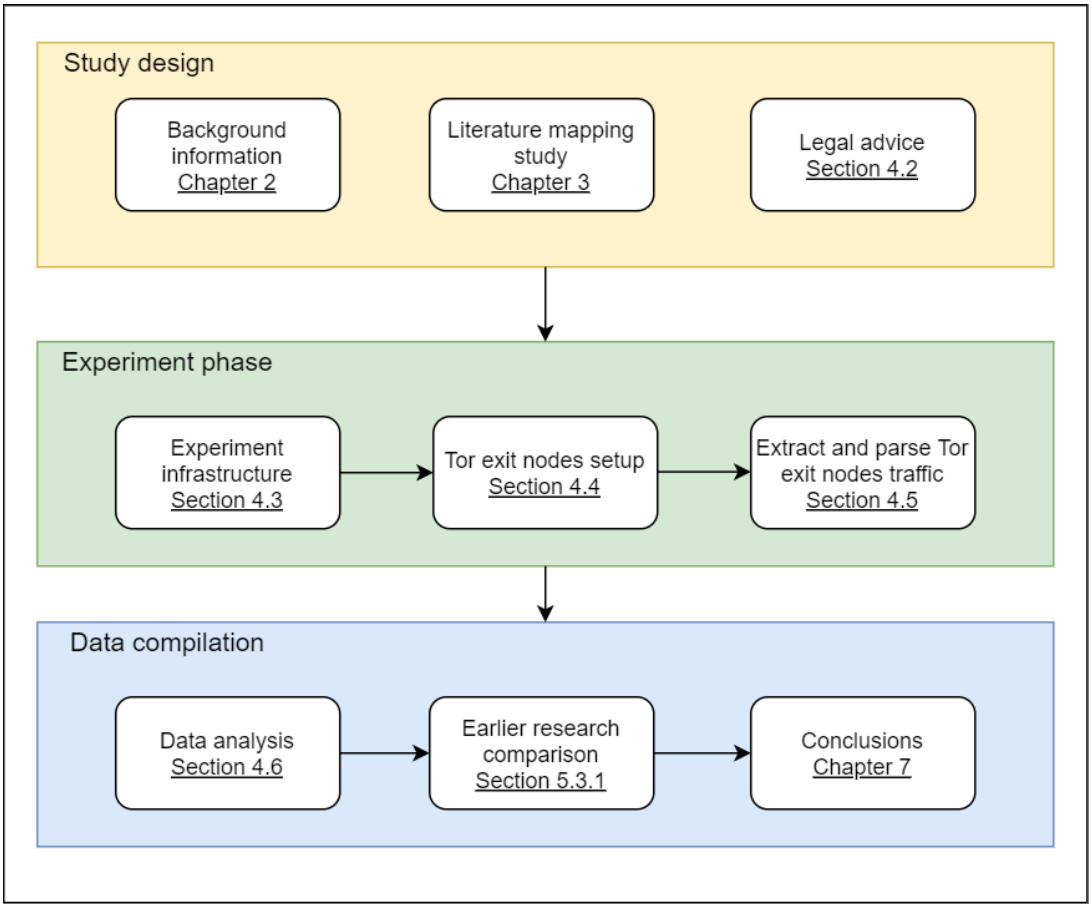
Study design
Background information Chapter 2
Literature mapping study Chapter 3
Legal advice Section 4.2
Experiment phase
Experiment infrastructure Section 4.3
Tor exit nodes setup Section 4.4
Extract and parse Tor exit nodes traffic Section 4.5
Data compilation
Data analysis Section 4.6
Earlier research comparison Section 5.3.1
Conclusions Chapter 7

Figure 4.1: Method overview over the different phases and parts of the study. The arrows indicate the order in which the phases and parts was completed.

4.1 Methodology to answer the research questions.

RQ1: How can we systematically collect, classify and analyze Tor exit node traffic in an efficient and secure way?

To answer RQ1 we decided that it was necessary to host Tor exit nodes. This was our only feasible option to give us full control over the experiment. Another possibility could have been contacting someone with existing Tor exit nodes. However, this would have limited the research and make it harder to recreate the experiment. We decided that we needed to build a script that collects network data and anonymizes it. This gave us full control regarding collecting, classification, and analyzing. Another possibility would have been to find a tool that adequately performed these actions, but the problem with this would have been that we wouldn't have full control over the collection method and possibly create workarounds to avoid collecting personal data. To conduct the research securely and lawfully we researched laws and legislation regarding data protection and Tor hosting for each country or organization where a Tor exit node would be placed. With total control over the experiment and with knowledge regarding the laws in each country, we were confident that we could securely conduct our research.

RQ2: How does the collected data compare to previous Tor network analysis?

To answer RQ2 we collected other research that did similar experiments. We did a literature mapping study to find previous research as discussed in Chapter 3: Related Works. We found and used a study by Chaabane et al. [11] as guidelines to conduct our research. We chose similar data points to make the comparisons more precise while adding more attributes that might be of scientific value.

4.2 Acquiring legal advice

This study handles two gray zones: operating a Tor exit node and capturing user traffic. Therefore we took the following measures to make sure what we are doing is legally allowed in Sweden, Germany, the USA, Japan and the European Union (EU) where we will operate Tor exit nodes:

- Contacting hosting companies to see if they allow Tor exit nodes.
- Ask for non-binding legal advice from Swedish Police Region South.
- Perform research regarding laws affecting the experiment in each mentioned country and the EU.

The Swedish Police Region South was sent a message from us asking for their view on the experiment we were going to perform. We informed them of our intent to operate exit nodes and analyze the traffic flowing through. The legal research was conducted through the internet by studying legal sources from the authorities of Sweden, Germany, the USA, Japan and the EU. The results can be seen in Chapter 5: Results.

4.3 Experiment infrastructure

31 server hosting companies were surveyed if they allowed us to operate a Tor exit node for one week. The hosting companies were collected from the Tor Projects' list of hosting companies that had a positive attitude to hosting Tor nodes on their servers. We focused on companies located in Europe and North America, but a few were added such as City Network due to their popularity in Sweden and Europe. Companies with servers located outside of Europe, preferably Asia and North America was prioritized. This is because Europe, North America and Asia have the most developed Internet usage in the world, and therefore should generate more Tor traffic if our nodes are placed on these continents. As mentioned in Chapter 2: Background, a Tor browsers choice of exit node is weighted towards its destination, and since the mentioned continents generates and receives the most traffic, nodes in those locations will gather the largest amount of data. We informed on the possibility to run the exit node with a reduced exit policy, though we preferred to allow as much traffic as possible for more accurate data. The results of this survey can be seen in table 4.1.

Number of companies	31
No Tor exit nodes allowed	16
Yes, with a reduced exit policy	6
Yes, with no exit policy	2
Did not reply to email	4

Table 4.1: Companies answers regarding hosting Tor exit nodes.

The Tor project recommends that when hosting an exit node the best way is by a network separate from your own. This can be done either by personally hosting a server or by using a hosting company. We chose to use a hosting company as this would not prevent us from hosting servers in more than one location, and replication of the experiment would be simpler with modifiable hardware.

We used low-end servers with 2 cores, 8 GB RAM, 50 GB hard drive capacity running Ubuntu 20.04 Focal. In our script we used NFStream version 6.2.3 [35] and GeoIP database version 20210323 [32]. We have an SSH proxy server with 1 core and 4 GB of ram.

4.4 Installing and configuring Tor exit nodes

The hosting company that we chose was City Network [13] due to them offering us free access to servers in the US, Japan, and Germany through a student program. We configured an Ubuntu server on each of the locations and used SSH for communication between the servers and our workstations. We configured them as Tor nodes, see appendix A.1 for configuration details, and uploaded our parser script to the servers. To avoid others from trying to connect to our servers via SSH we created an SSH proxy jump host and blocked all SSH traffic not coming from the jump host. The SSH traffic taints the data because NFstream picks it up. By using an SSH proxy we limit this traffic to one IP and that allows us to remove it from our data

Direction	**Ether type**	**IP Protocol**	**Remote**	**Port range**
ingress	IPv4	tcp	*	443
egress	IPv4	Any	*	*
egress	IPv6	Any	*	*
ingress	IPv4	tcp	*	80
ingress	IPv4	tcp	<IP adress for SSH Proxy>	22

Table 4.2: Group security rules for the servers. Egress means outgoing traffic, and ingress is incoming traffic.

in an easy way. We configured the Tor nodes to use port 443 as its inbound traffic port and port 80 to allow connections to our hosted web page displaying information about the exit node and how to contact us by using a template provided by the Tor project [51]. We rejected all other ingress traffic to our nodes. For the egress traffic, we allowed all ports to allow all traffic to pass through the node. Table 4.2 shows the security group configuration.

Roadblocks

The Tor project provides good explanations on how to set up Tor nodes. They provide detailed guides for the most essential parts of the system. However, we found that there were still questions that arose when following their guides. The first dilemma was to find a hosting company that allowed Tor exit nodes. We discussed this previously in section 4.3.

The Tor project recommends setting up reverse DNS and a WHOIS record. Reverse DNS takes an IP address and connects it to a domain name. The WHOIS record identifies the owner of a domain name or IP address. If a network admin finds the IP address of the nodes in their logs they can use reverse DNS and the WHOIS record to understand where the traffic coming from. To configure the reverse DNS we acquired a domain name and asked City Network to connect it to the IP addresses of our exit nodes servers.

4.5 Extract and parse Tor exit node traffic

We used modules compatible with the programming language Python as it is a large language with an extensive open-source community and the author's strongest programming language. Several candidates for the DPI were tested such as *ndpi*, *opendpi*, *netlify*, *scapy* and *pyshark*. However these programs either did not perform the operations we wanted, such as a simple way to receive information about a flows protocol usage, was not compatible with Python3, or were not satisfactory in a user-friendly perspective. This led to our decision to use NFStream [35], which gathers the packets into their respective flow and performs a DPI that, in some cases, provides a guess to the protocol used. This gives us the advantage of only recording flows, as the individual packets use the same protocol. A downside is that NFStream does not give access to the individual packets of flows. If NFStream makes a mistake we cannot detect or correct it.

The attributes that will be measured in each traffic flow is:

- Application protocol
- Timestamp
- Traffic category (Social media, Web, VPN, etc.)
- Source country
- Is the source IP a proxy (Yes/No)
- Destination country
- Is the destination IP a proxy (Yes/No)
- Destination port
- Duration of traffic flow (Milliseconds)
- Is the application protocol guessed by the parser (Yes/No)
- User-agent if sent in clear text
- Content-type if sent in clear text

Statistics for each node will be stored with the following attributes:

- Number of flows
- Total number of bytes for all flows (Megabytes)
- Total number of packets for all flows
- Number of flows who sent four or fewer packets (Classifies as a failed flow)

The attributes chosen are similar to the attributes collected in the study made by Chaabane et al. [11] but we have added attributes that are interesting to receive data on. Another alternative would have been to gather more attributes such as average size of packets and the client and server fingerprints. However the study focused on general attributes, and specific attributes in each flow was not in the scope of this thesis.

The attributes are measured by performing a DPI on each incoming or outgoing traffic flow, with the used DPI tool having one of the highest accuracy rates for open-source programs [8]. The source and destination countries are determined by an offline database named GeoLite2 [32] which allows us to input an IP and receive information on its location. An alternative method would be to send a request to an online database that is regularly updated and have a higher accuracy rating. However the amount of data processed means that unlimited IP lookup requests would be needed which would require financing, and sending a request and waiting for a reply for each IP would be a massive performance penalty. We used the GeoLite2 database as it is an open-source offline database with relatively high accuracy for resolving the IP correctly. This varies from country to country, with Tanzania and Nigeria having the highest incorrectly resolved IPs at 30 and 36 percent respectively.

Most countries have an incorrectly resolved rate from 0 to 10 percent, which we deem acceptable.

The application protocol gives us an understanding of what traffic flows through the Tor network. The source country indicates what country the middle node is in. This does not reveal the origin of the traffic but is still an interesting metric. An alternative would have been to create entry nodes to determine where traffic is coming from, but running entry nodes takes more time to become fully used, and due to our time constraints this was not feasible. The destination country is determined by the IP address and provides insight into what countries are the destination for Tor traffic. If the source or destination IPs are proxies helps us determine how many users' take extra precautions when sending Tor traffic. The duration of each traffic flow gives us an average in how long a traffic flow actively sends traffic. The content-type and user-agent gives insights into what data is being sent through the requests and what browsers are the most popular when using Tor. The node statistics give information about how much traffic each node have handled, making comparisons between the nodes possible. The number of flows that failed shows how many connections were attempted but not successfully. This means the flow failed to establish a TCP handshake and send at least one more packet.

The parsed data was stored in a *JSON* format first, as the format is easy to handle and analyze. During tests the format proved unable to handle a large amount of data, as we used a list that needed to be loaded into memory, append the latest data and then written back to the file. Each file parsed costed an extra second for the program to complete, and loading a large file into memory may provide performance issues. As a result, we chose the *CSV* file format. It uses one line per entry and allowed us to append data without loading its content. Comparisons showed that CSV files were considerably smaller in size than JSON with the same amount of information. CSV format also works in Excel which simplifies visualizing statistics. After the CSV file size exceeds 1 GB it is compressed and stored, and a new CSV file takes its place.

The biggest concern with the program was its performance in regards to the incoming data. If the program parsed the traffic too slow there would be a buildup with a high RAM consumption, and with a period of 7 days, we could potentially have storage issues. Therefore we attempted to get maximum performance in the program by improving the slowest function which was the file appending mentioned in the paragraph above.

4.6 Data analysis

The CSV files were compressed to reduce memory usage, transferred to our workstations and uploaded to Google Drive for backup storage. We wished to load this data into Excel for analysis, but the size of the data was well over Excel's maximum amount of rows which is one million, compared to our data of roughly 67 million rows. Instead, we used Pandas, an open-source Python library designed to handle and analyze large amounts of data [37]. Analyzing the data still required more RAM than was available, which required us to use the Dask module that is based on Pandas but uses lower memory consumption [1]. This gave us the ability to summarize, count and sort the data we needed for our analysis.

When attempting to use Dask it encountered a fault in our CSV data that was due to the CSV formatting. We used comma signs as separators, and thus any comma signs in the data itself will cause the row to have extra columns than usual, resulting in a crash. To correct this we found that the issue was in the User-agent column, and a Python program to change all comma signs in the user-agent to another sign was developed. This was all applied to each respective node's file.

After timestamps and file size had been taken from each file, they were concatenated into one file where we performed the rest of the analysis to gather the attributes mentioned in the section above. A new issue arose with the source and destination countries as well as the content-type column. The countries issue was the same fault as the user-agent column where one specific country had comma signs in its official name. As the country did not have much traffic we manually replaced the comma signs to allow Dask to complete the analysis. The content-type column was an issue since if a flow was not HTTP traffic, our program did not assign it a value. We manually extracted the content-types through a Python program that identified HTTP traffic and extracted the content-type.

The analyzed results were visualized using Excel, which performs the visualization of statistics with high quality. Other visualization programs were not considered, as Excel performs these actions in a satisfactory manner.

Chapter 5

Results and Analysis

In this chapter we show our findings regarding laws and legal advice. We present the results from the Tor traffic analysis.

5.1 Legal information regarding surveillance and exit nodes

Disclaimer: The legal statements listed below should not solely be used to decide whether you can or cannot operate a Tor exit node in the countries mentioned, or recreate the study.

Swedish laws

The Swedish Police gave oral advice by phone. Since Tor exit nodes are both a technical and legal issue it took time to be handled by the correct department. They informed us that this is non-binding, non-official legal advice, and this was not a go-ahead to do whatever we wanted, but not an outright ban on our experiment either. Summarized, the police informed us that this is a grey area and the Swedish law is not clear if it is a crime to listen to Tor traffic, as internet users' give their consent by sending traffic through a router they do not own themselves. Due to the police's regional organization, one region can track malicious traffic to our exit nodes and take further measures, unknowing of our scientific intentions. If we researched the law regarding illegal eavesdropping, used our judgment and implemented a DNS policy that stated our intentions to any law enforcement that might trace malicious traffic to our nodes, the experiment can be performed.

Swedish law states that listening or recording conversations online or in real life using technical equipment where the public have no access is illegal and can be subjected to a fine or prison for up to two years [30]. However, Tor nodes are accessible to anyone with a Tor browser, making it a public area. This, along with our scientific purpose and anonymizing all data makes the experiment okay to conduct.

Several Tor exit nodes are operated in Sweden, which gives credibility to the fact that running a Tor exit node is not illegal. However, if the DNS is not properly configured to provide information that this is a Tor exit node and your node keeps routing malicious and illegal traffic, you can be prosecuted by Swedish authorities.

EU and German laws

EU laws regulate data protection regarding the storage, collection and use of personal information [57] [38]. Personal information is classified as data that directly or indirectly may identify an individual. Data collection is allowed by businesses and organizations under some circumstances, such as when you have a contract with them, for legal purposes and when legitimate interest allows it. Our scientific reason to conduct this study, and our collected data is not stored anywhere before being anonymized gives us ground to run the experiment. Tor exit node operators who do not collect any information regarding their users' are not affected by these laws and are allowed to operate exit nodes in the European Union.

In May 2018 GDPR came into force in Germany, replacing their domestic data protection law. In Germany, it is allowed to gather data for scientific or statistical purposes without user consent according to section 27 in the Federal Data Protection Act [10]. This makes it okay for us to gather anonymized data for our study. Running a Tor exit node is legal in Germany, but due to their strong copyright laws [9], it is possible to come under the scope of German authorities if you are not blocking BitTorrent traffic.

US and Japanese laws

The US is unique in the way that they both have federal laws which affect the entire country and state laws which vary from state to state. The federal laws handle personal information [19], which works in the same way as the EU, that any information that can directly or indirectly identify a person is classified as personal information. As we anonymize all data on the fly, we are not affected by these laws. Our server is in the New York state, and it have laws regarding personal information that is similar to the federal law and thus the same argument for us works on a state level as well [17].

Japan have data protection laws similar to the EU's GDPR legislation [22]. The Act on the Protection of Personal Information and the Act on the User of Numbers to Identify a Specific Individual in the Administrative Procedure sets out rules for different business sectors such as telecommunication, financial or medical. They provide laws on what to do in case of a data breach. As we anonymize all information saved, and with no way to recreate the data it is acceptable to monitor the data due to the scientific purpose.

5.2 Tor traffic results

During the collection phase, the three exit nodes gave in total 8,53 GB of CSV formatted data, with 100,35 million flows going through the nodes. This corresponds to 5594.87 GB in total traffic. Note that the data is not distributed evenly between nodes as their usage varied. NFStream that performs the DPI on the flows made a calculated guess of the protocol based on its port in 32.61% of all flows while the rest was detected by packet dissection. See table 5.1 for further details.

Node location	Flows (Millions)	Failed flows (Millions)	Traffic flowed (GB)	Dataset size (GB)
US	47.2	14.32 (30.33%)	2152.61	4.14 (48,53%)
Germany	41.49	12.46 (30.03%)	1774.40	3.67 (43.02%)
Japan	11.66	5.71 (48.97%)	1667.86	0.72 (8.44%)
Total	100.35	32.49 (32.37%)	5594.87	8.53 (100%)

Table 5.1: Flow and data distribution from the nodes.

An interesting feature in table 5.1 is the ratio of failed flows in the Japan node, where a failed flow is classified as sending 4 packets or less in total, versus its total number of flows. It is significantly higher than the other nodes that have the same failed flow percentage. The US and German nodes had a higher ratio of failed flows early in the experiment, but after 36 hours of running the nodes, it was reduced.

Figures 5.1 and 5.2 shows the distributions of packets and flows. These are not equal because the number of packets that are being transferred in one flow can be any amount. Note here that the Japan node have more packets per flow than the others.

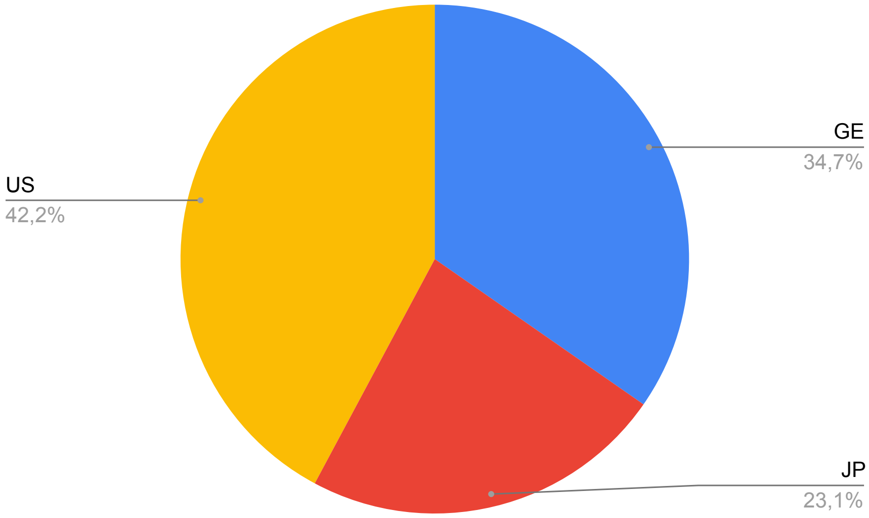

Figure 5.1: Packet distribution on the nodes.

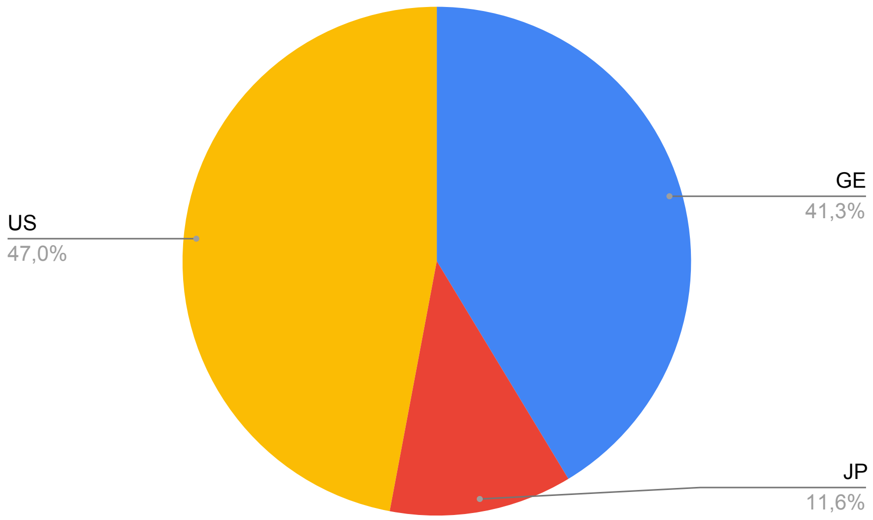

Figure 5.2: Flow distribution on the nodes.

Figure 5.3 shows the distribution of the top 8 application protocols. The Unknown section is traffic that NFStream could not detect the application protocol, meaning it can consist of several protocols. The "Other" section represents the rest of the protocols that were detected. Note that different HTTP and TLS protocol exists in the "Other" section. The total TLS protocol usage was 55.03% and the total HTTP protocol usage was 15.91%. Labels such as *TLS.Amazon* and *TLS.Cloudflare* is traffic using TLS certificates from these companies to encrypt their communication.

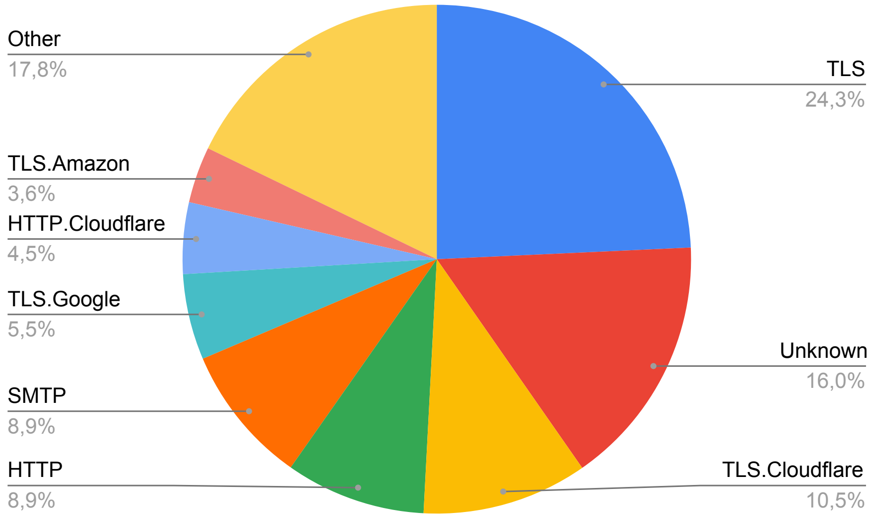

Figure 5.3: Top application protocol distribution. Unknown represents protocols not identified and Other is the collection of remaining protocols with less than 3.6% usage.

Figure 5.4 is the distribution of categories the traffic belongs to. The figure shows the top 8 most frequent categories. Web traffic is the dominant usage in the network.

The Unspecified section is traffic NFStream could not categorize.

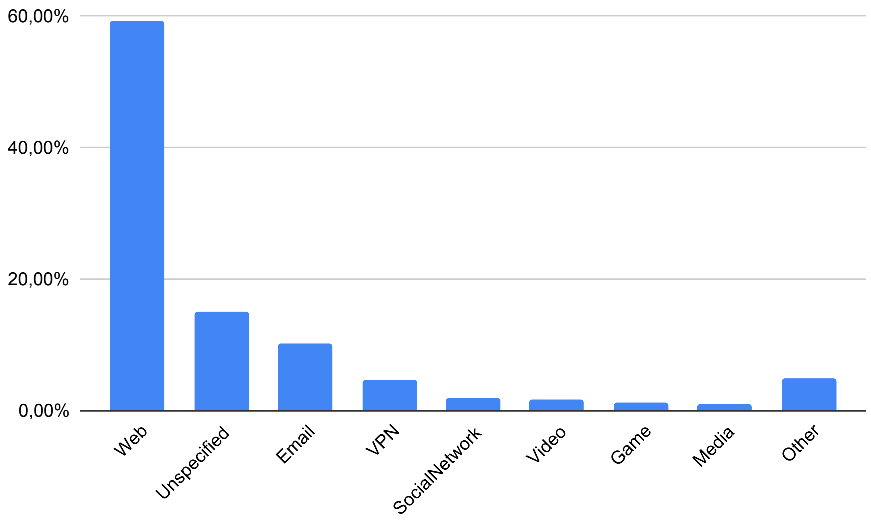

Figure 5.4: Most frequent usage categories. The unspecified column is flows that NFStream could not categorize.

Figure 5.5 shows the distribution of countries that was the destination for an outgoing flow originating from one of our exit nodes. 244 unique countries and their territories received traffic, with the United States receiving the most connections through the Tor exit nodes followed by the Netherlands. The section marked as "Other" is all other countries that received less traffic than the top 10.

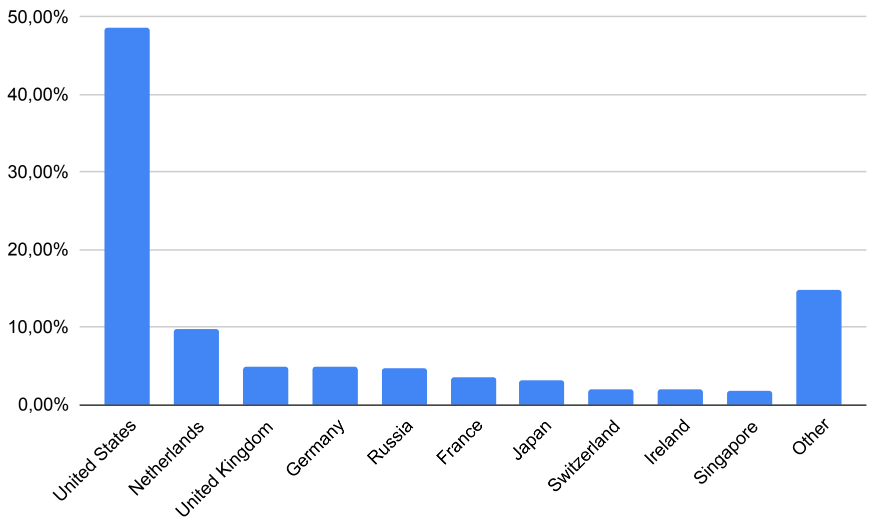

Figure 5.5: Destination of outgoing traffic by country.

Figure 5.6 shows the 4 most used destination ports for outgoing traffic. Port 443 is the port used by HTTPS and HTTP over TLS/SSL, i.e encrypted web traffic. Port 5222 is used for XMPP (Extensible Messaging and Presence Protocol) and jabber clients. These are used for instant messaging. Port 80 is used for HTTP (HyperText Transfer Protocol) and port 25 is used for SMTP (Simple Mail Transfer Protocol). Extracting what countries were sent data using port 5222 showed the Netherlands receiving the most traffic, followed by the UK and Switzerland. The three combined received 99.78% of all traffic using the destination port 5222.

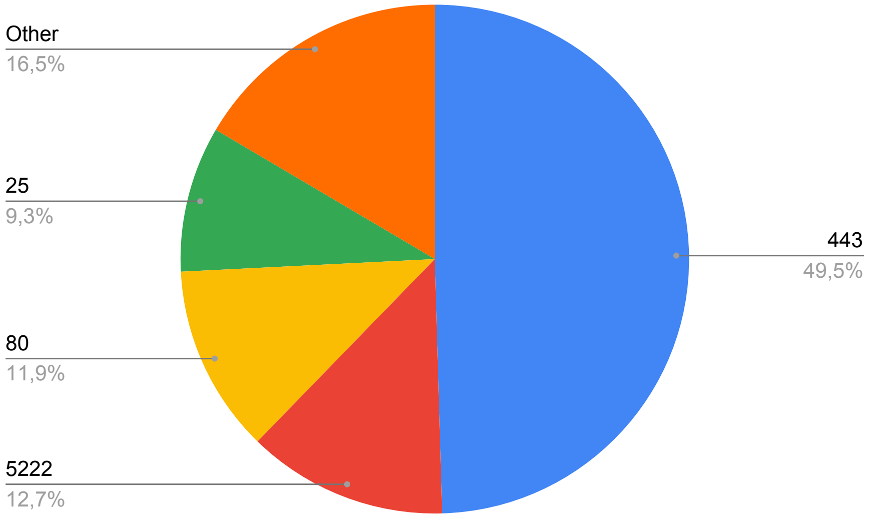

Figure 5.6: Most used destination ports. The Other section represents all ports with less than 9.3% of usage.

Table 5.2 shows the most used user-agents. The user-agent gives information to network peers about what operating system and what browser the web server is using. Baiduspider/2.0 is Baidu's web crawling spider, belonging to the Chinese search engine [4].

User-agent	Percentage
Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)	7.54%
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0) Gecko/20100101 Firefox/83.0	3.20%
Mozlila/5.0 (Linux; Android 7.0; SM-G892A Bulid/NRD90M; wv) AppleWebKit/537.36 (KHTML; like Gecko) Version/4.0 Chrome/60.0.3112	2.70%

Table 5.2: Top 3 most popular user-agents in HTTP flows. The low percentage means there were not many identical user agents in the captured flows.

Of all HTTP traffic detected, 70.46% had an unknown content-type, meaning either NFStream failed to detect it or it was encrypted. Table 5.3 shows the distribution of the identified content-types. Text content-type is usually used for text in human-readable formats. Application is used for transmission of application or binary data, while image, audio and video are used to send their respective types of data formats.

content-type	Percentage
Text	60.65%
Application	37.25%
Image	1.25%
Audio	0.06%
Video	0.05%
Other	0.74%

Table 5.3: Top 5 content-types detected in HTTP traffic. The Other section represents all content-types not in the top 5 types.

In figures 5.7, 5.8 and 5.9 the intensity of the flows during a 24 hour day period is shown, starting from 00.00 to 23.59. Figure 5.10 shows the flows per hour of all nodes calculated to Greenwich Mean Time (GMT).

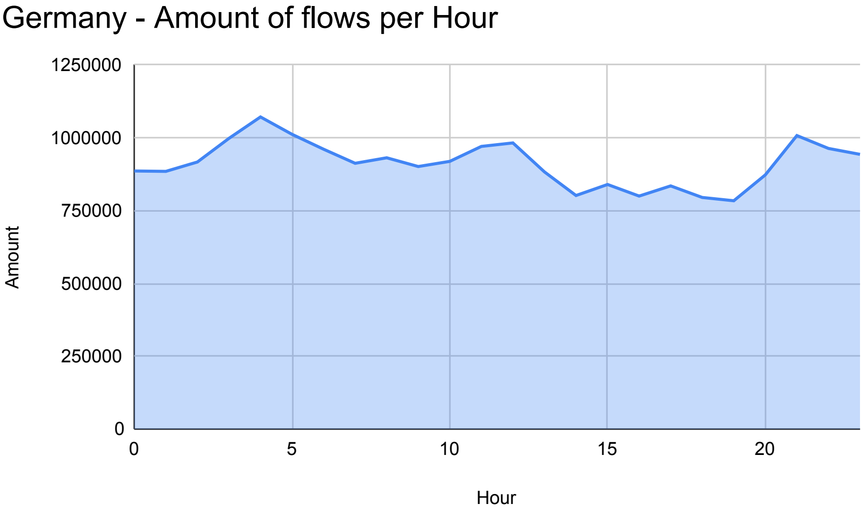

Figure 5.7: Number of flows per hour of the day in Germany.

Figure 5.11 shows the origin country of traffic whose destination was one of our exit nodes. This includes incoming traffic from middle nodes and request responses to traffic sent from the exit nodes. A total of 176 unique countries or their territories was detected to have sent traffic to our exit nodes.

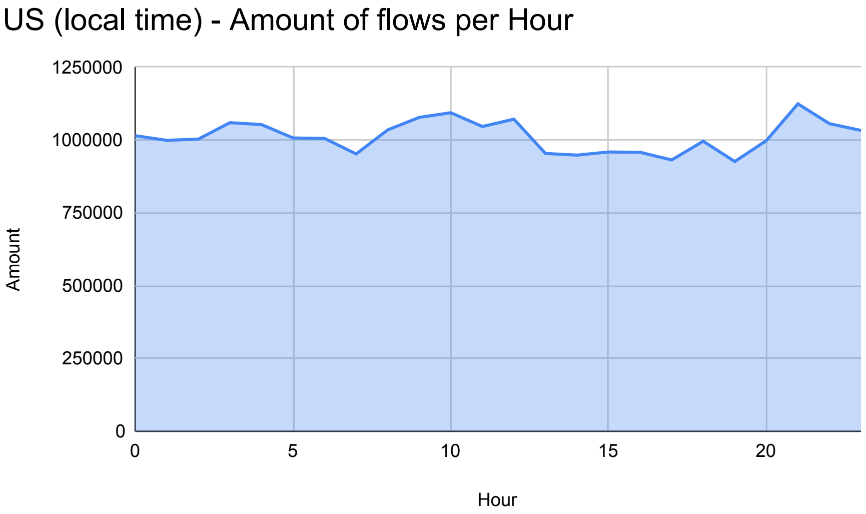

Figure 5.8: Number of flows per hour of the day in the US.

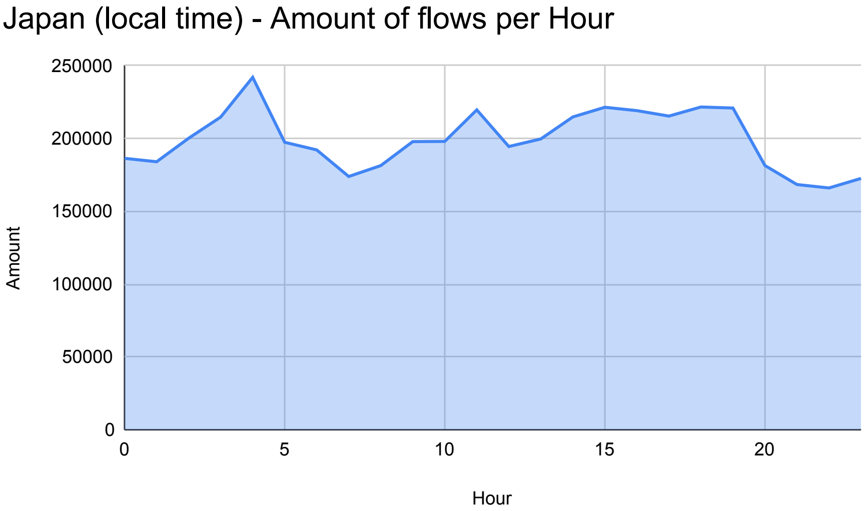

Figure 5.9: Number of flows per hour of the day in Japan.

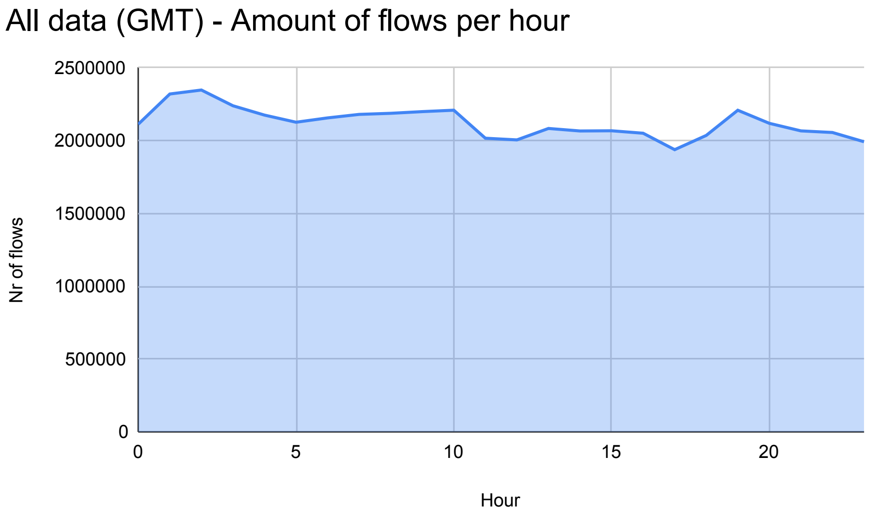

Figure 5.10: Number of flows per hour of the day for all nodes. Calculated by converting local time to GMT and combining them.

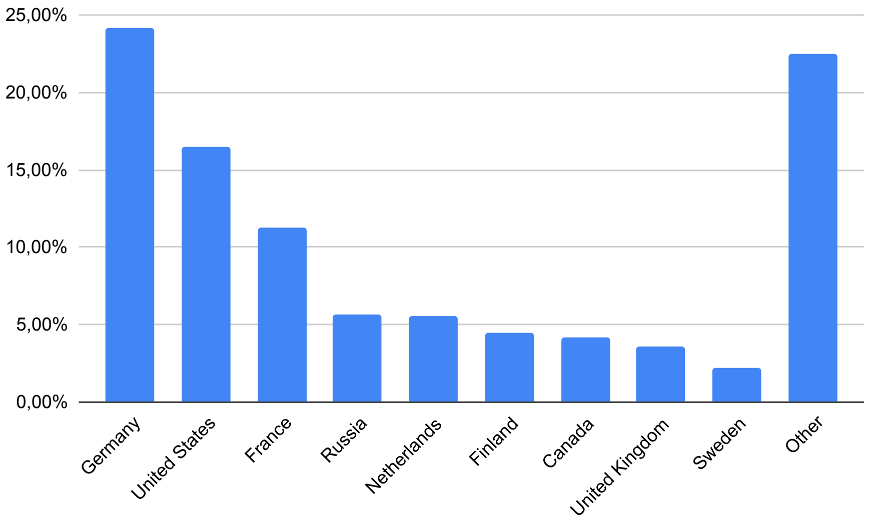

Figure 5.11: Sources of incoming and outgoing traffic by country to our exit nodes. This includes middle node traffic and request responses from our exit nodes. The Other section represents remaining countries outside of the top 9.

Another result was the GeoIP module detection of IPs that were proxies. Of all source IPs, none were detected as a proxy, versus the destination IPs where 0.00034% of all flows was detected as a proxy. This amounts to roughly 34 119 flows out of 100.35 million. The average flow time was 25.1 seconds.

5.3 Analysis

The results support the trend of Internet traffic moving from HTTP traffic to HTTPS or otherwise encrypted traffic as data from Cisco [33] and Google [18] both confirm. This can be seen with the TLS protocol usage at 55.03% and the destination port 443, explicitly used for encrypted communication, used by 49.5% of all flows. Despite this, it is still a significant amount of HTTP traffic flowing through the Tor network that could be effortlessly captured by us and potentially reveal personal information. This is supported by the usage of port 80 that is used for HTTP traffic being at 11.9%. HTTP traffic containing images, video or audio is very low in regards to the content-types detected, but images are still being sent at 1.25%, raising the question as to what images are being sent over the Tor network in plain text. This is a surprise as the Tor project's main usage is to preserve privacy and prevent identification, meaning that it is generally used by people who do not wish to be identified. Some HTTP traffic used the Baidu spider, a crawler for the Chinese search engine [4] as user-agent, suggesting that at least a part of the HTTP traffic was not human-generated.

16% of the protocols could not be detected by NFStream, which is a very large amount. The study made by Chaabane et al. eleven years ago [11] suggests that this could be BitTorrent traffic as it has increased its attempts to avoid surveillance. However, this can not be verified or supported by us due to our stored and anonymized data not providing enough support to reach any conclusion.

The Japan node generating less data was also to be expected, as Tor is widely used in Europe and the US. This is supported by the data from the top source and destination countries as all countries, except for Japan and Singapore in the destination chart are countries located in North America or Europe. However it is important to notice that 244 unique locations, meaning countries and their overseas territories, received traffic and 176 of those sent traffic to our exit nodes, showing that Tor traffic exists in almost every country in the world. This does not mean that it is used in every country in the world, but is at the very least routed through it. Less data from the Japan node also means the traffic destination is weighted towards Europe and North America.

The number of flows per hour, as depicted in figures 5.7, 5.8, 5.9 and 5.10, had smaller peaks than expected, with those peaks not being in the expected locations as people come home during the evening and uses the internet, which should see an increased usage during hours 20-24, before dropping off during the night. The Japan chart, however, sees its biggest spike during the hours 03-05, compared to the US and Germany which have a small peak around 20-23 which is more expected. This also suggests that each Tor node is used by a large number of countries with different time zones, using the node to its near maximum usage all hours of the day, thus limiting the peaks not just to the country's time zone where the node is located.

5.3.1 Earlier research comparison

We are comparing our results with a study made by Chaabane et al. [11] eleven years ago that uses similar data gathering and traffic classification techniques as we do, giving us the chance to directly compare overlapping results. Other studies that we found did not provide comparable data and we perform our comparison primarily with Chaabane et al. study. The differences in our experiments are that their study ran 6 Tor exit nodes in the same geographical locations as us (US, Germany, France, Taiwan and Japan) while we had one node in the US, Germany and Japan each. They allocated 100 KB/s bandwidth for the nodes compared to our 2 MB/s. Chaabane et al. captured traffic during two time periods, 7 days and another later in the year during 10 days to avoid any time-correlated results, which we could not perform due to time constraints. They set up an entry node, allowing them to record the users' country of origin while we can only capture traffic coming from middle nodes.

Chaabane et al. [11] found that 68.57% of flows were HTTP traffic. Our analysis showed that only 15.91% of the traffic flows were HTTP. They found that 4.64% was BitTorrent traffic where we could not identify BitTorrent in our findings. However, we had a section with "Unknown" traffic corresponding to 16% of the traffic flows. This partition may include BitTorrent traffic that could not be detected, as BitTorrent was detected in 174511 flows as can be viewed in the appendix. Considering the large size of BitTorrent traffic in their study, it raises the question if BitTorrent traffic is obfuscated or has simply shrunk in size. We also saw a significant increase regarding TLS and encrypted traffic where Chaabane et al. [11] found that 1.83% of the flows was SSL encrypted compared to our findings where 55.03% was TLS encrypted. The "Unknown" partition also grew from 5.94% to 16%. When looking at the content-type we had a small sample size due to the amount of HTTP traffic we collected where large parts of the data had no content-type. However, the data that we found showed that 60.65% was text compared to Chaabane et al. who found that 27.9% of the content-types was text. They showed that 18% was application, we found this to be 37.25% of our data. All other categories were a small portion of our data. Where Chaabane et al. saw a bigger representation of other categories such as images, video and audio.

Chapter 6

Discussion

In this chapter, we discuss the answers to our research questions, the results and validity threats against the data.

6.1 Revisiting the research questions

RQ1: How can we systematically collect, classify and analyze Tor exit node traffic in an efficient and secure way?

Setting up a server with a Linux OS, 8 GB RAM and 50 GB of hard drive storage gave us the infrastructure necessary to run a Tor exit node without a reduced exit policy and collect traffic passing through the node. The software used was capturing traffic directly into the parser program. NFStream divided the traffic into its respective flow and used DPI to classify each flow. This flow division allowed us to collect and analyze the traffic in a systematic way. The components mentioned gave us the ability to collect data that is of scientific value. Please view the appendix for the code that answered RQ1. This answers our first research question.

RQ2: How does the collected data compare to previous Tor network analysis?

There is a clear difference in the results compared to the analysis made eleven years ago. HTTP traffic have had a huge drop, with a large share of the remaining HTTP traffic using encryption as well. Encrypted traffic have increased from 1.83% of all flows to 55.03%. Some similarities remain of the usage of Tor divided into categories where web usage and social networking are still in the top categories.

The content-type distribution in plain text traffic have shifted with images dropping from 31.7% to 1.25%, applications and text having been increased from 18% and 27.9% to 37.25% and 60.65% respectively.

6.2 Reflection on the results

Discussion regarding the two major data results in our study: the legal data acquired and the analyzed Tor traffic data.

Legal data

Gathering the correct legal data was harder than expected. Finding the correct legislation on internet surveillance for each country and processing it was difficult, as legislative texts are defined in a complex language, making reading them more difficult. Online sources explain the laws more simply but must be verified to make sure they are stating the correct facts. Another factor was to determine what laws applied to us. As mentioned in Chapter 5: Results, the Japanese laws have legislation for specific businesses such as the financial sector, so determining what laws applied to us was also a part of the process. We focused on the laws regarding internet surveillance and handling of sensitive data. Germany had a law that specifically stated that internet surveillance was accepted for scientific or statistical purposes, which made it simpler to determine whether our experiment was allowed or not. Other country's legislative texts were not as specific.

Operating a Tor exit node is possible if it is operated in an informed and transparent way. Meaning that for example the ISP that hosts the IP address knows and understands that a Tor node is operating and using a reverse DNS. However, the capturing of traffic on Tor exit nodes is a gray zone. The Tor project distinctly states that snooping on exit traffic is not something they advise [50]. This does not mean that it is not being done by users' with illegal or scientific intentions. We believe that it is important to do this kind of research to understand both the network better and to understand what information can be collected. If users' understand if they use a deprecated or old protocol through Tor they can still be identified, they might be induced to update their protocols and be safer from surveillance.

Tor traffic data results

The results fit our view of the Tor network to a great extent. With traffic mostly in Europe and the US as it have historically been, was of no surprise as Asia and the rest of the world are still working on bringing the Internet to everyone. TLS accounting for over half of the recorded flows protocol is of no surprise either as there is a push towards HTTPS usage. Still, we were surprised to see HTTP usage at 15.91% even though some of it is non-human generated traffic. This is because the authors believed the shift from HTTP to HTTPS would be even greater in the Tor network. Its users' wish to remain anonymous and retain their privacy, which is not possible when sending non-encrypted traffic. Legacy traffic such as using old protocols or sending HTTP traffic always exists, but not in this scale as indicated by the data. There is also small amounts of audio, video and images being sent through HTTP which makes us authors wonder what the purpose of those transmissions are. This could be legal transmissions, but with the Tor network used by people who do not wish to be identified and tracked over the Internet, makes us wonder what is inside the traffic. The top user-agent being a web crawler shows that automatic tools are using HTTP. This points towards that automatic tools can be a significant part of the HTTP traffic and requires further studies.

The number of flows per hours data also surprised us, as we believed the peaks would follow the hours of the day for each nodes respective country but revealed little and had possibly erratic peaks that forced us to reassess how each node is used, and that the majority of node traffic is not generated in the country where the node is located.

The reason why the Japan node has a failed flow rate of 48.97% compared to the US and German nodes of 30% is also unknown. All nodes were configured the same way and with the same server infrastructure. We speculate that the reason is for its low amount of regular flows, making the failed connections stand out more compared to the other nodes.

GeoIP detected 34 119 flows that used a destination proxy. The purpose of these is unclear. Using a proxy on top of Tor will undoubtedly slow down the already slow speed of the Tor network. It is an interesting finding that can be worth exploring further.

6.3 Contribution and fulfillment of the objectives

This research shows how to perform an analysis on Tor exit nodes effectively and securely. It updates the previous knowledge about the traffic going through Tor exit nodes and contributes to the understanding of the Tor network. It shows comparisons to previous research and highlights the biggest changes. It gives insight and tools to perform similar experiments. Including a parser script and detailed instruction on how to configure exit nodes. It discusses the legal and ethical aspects of perform this sort of research. The research does not however show how much the Tor network is used in total. It provides a sample size of data that indicates how the network is used.

6.4 Encountered problems and solutions

This section describes the problems we encountered during our study. We describe how we handled these problems that include academic problems with the writing and structuring of the thesis, and technical problems we encountered during the designing and construction of the experiment.

Data collection limitations

As we described in the scope and limitations section 1.3, we decided that three exit nodes on three different continents for 7 days was sufficient enough to yield a good result that would be statistically significant and accurately compared with the study by Chaabane et al. [11]. On the other hand, a greater set of nodes with a broader geographical spread, as well as running the nodes for a longer time frame would yield a better result. We had to limit our amount of nodes due to the time constraints of the thesis. City Networks sponsored the experiment that allowed us to set up the three nodes without economical costs. As City Network had servers in the US, Germany and Japan we decided that this was sufficient compared to the scope of the thesis.

To collect the network data we first used tcpdump, a tool that can listen to a network interface and capture internet traffic from that interface into a pcap file. This meant saving sensitive data to long-term storage, meaning we had to comply with each countries laws regarding sensitive data storage. To bypass this issue we instead sent the traffic directly into the parser, a feature supported by NFStream. The parser could then anonymize all data before saved to the disk.

Hosting servers

A limiting factor was the reliance on server hosting companies to provide the hardware for our project. This forced us to take into consideration where their servers were located, their willingness in running a Tor exit node, and the economical cost for it. The best scenario would have been a company with servers on each populated continent and allowing Tor exit nodes with no restricted exit policy. This would have given the most accurate data for the experiment. In the end, we found City Network allowed us to set up Tor exit nodes in Germany, Japan, and the US without any policies in place. This proved to be a very good option where they also gave us technical support regarding the servers when needed.

Attacks on our servers

After making the server public we immediately began receiving SSH brute force attacks from different sources. To combat this problem we created and added SSH keys with RSA 4096 bit keys, disabled logging in as root and logging in using a password and username. However, the attacks were still visible in the authenticating logs, meaning that the parser would still record and store the SSH flows. This tainted the data as we explicitly wanted only Tor user traffic, and not malicious traffic that comes with hosting a public server. To resolve this we added a jump host server where we added our SSH key, and added security group rules to the servers. The rules allowed incoming Tor traffic from port *443* and SSH traffic on port *22* from our jump host IP. All other incoming traffic was blocked while all outgoing traffic was allowed.

During the week of gathering results the server hosting the node in Germany was subjected to a suspected SYN attack, which sent requests to our HTTP website that displayed that we are a Tor exit node. This made the parser handling incoming flows, to increase its RAM usage from 1.6 GB to 7.6 GB in a short time, making the Linux OS kill the process as the server's maximum memory was 8 GB. With the parser restarting, it "lost" all the current flows it had in memory. The parser restarted automatically and we disabled port 80 for a short time before resuming normal operations. As the flows consisted of attacks against our servers, we do not believe the gathering of Tor traffic was significantly impacted.

Compilation when analyzing the results

As we explained in section 4.6 we had some complications when we tried to analyze the results. We used the CSV format on the output data. This gave us problems where entries had commas in them, confusing the Dask CSV analyzer. We had to address these problems before we could start the analysis. If had used the JSON format, these problems might have been avoided. We also found that our data set was too big to be used directly in Excel. Being able to use our data directly in Excel was one of the arguments for using CSV format over JSON. However, the CSV format still provided a smaller file size for the same amount of data, and CSV files are compatible with the Pandas module used for the analysis, making it a great choice for the experiment.

Points of improvement

During the experiment, we learned things that could have been done better or yield better results. This experience includes a more extensive data collection e.g collect the sizes of the flows. This gives information about what traffic takes up most of the bandwidth inside the Tor network. Another is changing from port 443 as the listening port on the Tor nodes to separate the incoming traffic to gain more control over the data. Another is to use NFStream more as it is a versatile tool that can be used to gather more information than we used it for. For example, it could be used to save data regarding port scans and hacker attacks. It is also possible to use NFStream to map connections. But if that is done the ethical aspects must be reconsidered.

6.5 Validity threats

Here we summarize all the threats that can affect the validity of the collected data.

Internal validity

- The literature mapping study we conducted only used three databases: *IEEE*, *ACM Digital Library* and *BTHSummon*. BTHSummon consists of several databases, but despite this our mapping study might have missed papers relevant to our study.

- Configuring hosting nodes identically - To ensure that it is no difference between the nodes, we configured them in identical ways and used equally powerful machines.

- Data collection is software-dependent - For our research, we created software that used the tool NFStream. This makes us completely dependent on that the data produced by NFStream is accurate, with a low false positive rating. Although nDPI, which NFStream is built on, was among the programs with the highest accuracy using deep packet inspection [8], it has no official false-positive rate or accuracy. A more accurate result could be done by either paid commercial software or deploying machine learning models to reduce the false-positive rate. However, this is also difficult due to the scarcity of training sets for this type of analysis.

- While compiling the source and destination countries, a small, but not an insignificant number of IP addresses could not be resolved to a specific country. Using an IP database with better accuracy could minimize this issue with less impact on the data validity. Either way, this is software-dependent.

- When compiling the data there is a possibility that we count and or incorrectly express the data. To minimize these errors we counted the sorted attribute with the number of lines in the CSV file. Since they matched, no line had been missed. This was repeated for each attribute in the collected data.

External validity

- Similar research methodology - To make our research relatable and useful we conducted it in a similar way to a previous study by Chaabane et al. [11]. We used newer tools that were not available or not as developed as previous studies. The results are not negatively affected by this.

- Study repeatability - To make the repeatability of our study as easy as possible we provide details of how we conducted our research including the source code to our parser program and instructions on how we installed and configured the exit nodes.

- Result representation - We present the result as they were collected, without modifications. This allows the reader to draw their own conclusions.

- Extraordinary circumstances - The experiment was conducted during the Covid-19 pandemic. During this time the internet usage increased. It is possible that this changes the usage of the Tor network and subsequently affects the results of this experiment. We could not do anything about this and it is not certain that this would affect the results.

Chapter 7

Conclusions and Future Work

This chapter includes a conclusion of our experiment where we mention the most important points and takeaways from our study, and provide some ideas for future work.

7.1 Conclusions

We have demonstrated that it is still possible to perform a traffic analysis on the Tor network by using open-source tools available to anyone with low-end server infrastructure.

The results show the TLS protocol is used in 55.03% of all flows, which is supported by the usage of destination port 443 at 49.5%. The HTTP protocol usage is at 15.91%, though not all HTTP traffic is human-generated due to a web crawler user-agent being detected. Port 5222 that normally is used for XMPP (Extensible Messaging and Presence Protocol) client connections were used in 12.7% of the flows, with the Netherlands, UK and Switzerland combined receiving 99.78% of all traffic that used port 5222 as its destination.

This shows that despite people's assumption using Tor is risk-free, with little to no chance to be identified, the possibility exists to perform an analysis and capture plain text traffic that might contain personal information. Due to all exit nodes being run by volunteers, a user with malicious intent could set up a server, recreate this experiment and collect plain text information that potentially could identify Tor users'. This poses a threat to users' using the network for visiting sites blocked by their country or dissidents organizing against authoritarian regimes. Tor users' should increase their awareness of what traffic is sent in the Tor network, and always use an up-to-date encryption method for their traffic regardless of its importance. Another suggestion would be to warn Tor users' when sending HTTP traffic through the Tor browser to improve their awareness about sending non-encrypted traffic.

The US and Europe still account for most traffic and are the majority of the traffics source and destination countries, wherein the destination country the US accounts for over 45% of all traffic, suggesting that the Tor network is not as extensively developed and used in Asia as in the West.

Comparing to a similar analysis made eleven years ago, the TLS usage has increased from 1.83% to 55% which is supported by the growth in HTTPS websites on the clear internet. Its counterpart HTTP traffic has seen a significant drop, with the content-types for images, audio and video being detected in 1% of all content-types, while the text and application content-types now make up 97.90% of all detected content-types. The content-type was unknown in 70.46% of the HTTP flows, suggesting that the traffics content was encrypted.

Operating a Tor exit node for other than scientific purposes is a grey area, and each country or organization uses different laws that may or may not describe the areas of *traffic capturing*, *Tor exit node hosting* and *scientific data usage* in a satisfactory manner. Therefore, before performing a traffic analysis experiment on the Tor network or hosting a Tor exit node, one should always contact legal sources such as a government department or a lawyer with knowledge regarding these issues.

7.2 Future work

As future work, we think that there is still room for exploring what information can be collected by performing a traffic analysis on the Tor network. Either by using NFStream or similar tools more extensively or by machine learning as an alternative to DPI to gather more information about the traffic. There is also a possibility that flows could be combined to gain a deeper understanding of the size and type of traffic that is being sent.

We believe that the framework we developed will help do similar analyses, both to verify or disprove our results and continue to gain knowledge about the Tor network. Our parser script can be improved as discussed in section 6.4: Encountered problems and solutions and the issues mentioned in chapter 4: Method. Hosting entry nodes as part of the experiment can yield interesting results about the origin of Tor traffic.

7.3 Final statements

In a broader sense a discussion about anonymity, freedom of speech, and the right to be anonymous on the internet are needed. This is something that is being discussed more and more in many parts of the world, but oppression is still a major problem. Tor has great value where it provides tools to avoid oppression. In recent years the world has seen an increase in surveillance where big corporations and countries collect data about users' on the internet. For anyone that wants to avoid this the Tor project provides tools to gain anonymity. It is therefore important to understand Tor, how it is used and highlight its weaknesses, to preserve the choice to be anonymous.

When we look at the ethical part of Tor we found difficult questions to answer, such as the pros and cons of allowing Tor. Tor provides anonymity and freedom of speech, but it also enables criminals to communicate and facilitates illegal activities. Tor exit nodes are in this sense exposed and proven problematic to run. Countries and organizations that promote freedom of speech and the right to be anonymous should maybe discuss facilitating the administrations of Tor exit nodes.

Bibliography

[1] Anaconda, Inc. (2021) *Dask*. Accessed 8th Mar. 2021. [Online]. Available: https://docs.dask.org/en/latest/dataframe.html

[2] Association for Computer Machinery. (2021) *ACM Digital Library*. Accessed 22nd Feb. 2021. [Online]. Available: https://dl.acm.org/

[3] Australian Cyber Security Center. (Oct. 2020) *Defending Against the Malicious Use of the Tor Network*. Accessed 21st Jan. 2021. [Online]. Available: https://www.cyber.gov.au/acsc/view-all-content/publications/defending-against-malicious-use-tor-network

[4] Baidu. (2021) *Baidu search engine*. Accessed 15th Apr. 2021. [Online]. Available: https://https://ir.baidu.com/

[5] J. Barker, P. Hannay, and P. Szewczyk, "*Using traffic analysis to identify The Second Generation Onion Router*," 2011.

[6] L. Basyoni, N. Fetais, A. Mohamed, and M. Guizani, "*Traffic Analysis Attacks on Tor: A Survey*," 2020.

[7] Blekinge Institute of Technology. (2021) *The Library Database*. Accessed 22nd Feb. 2021. [Online]. Available: https://bibliotek.bth.se/databases

[8] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "*Independent comparison of popular DPI tools for traffic classification*," 2014.

[9] Bundestag. (2018) *Act on Copyright and Related Rights*. Accessed 16th Mar. 2021. [Online]. Available: https://www.gesetze-im-internet.de/englisch_urhg/englisch_urhg.html

[10] ——. (2019) *Federal Data Protection Act*. Accessed 16th Mar. 2021. [Online]. Available: http://www.gesetze-im-internet.de/englisch_bdsg/englisch_bdsg.html#p0239

[11] A. Chaabane, P. Manils, and M. A. Kaafar, "*Digging into Anonymous Traffic: a deep analysis of the Tor anonymizing network*," 2010.

[12] Cisco. (2021) *Ports explained*. Accessed 10th Mar. 2021. [Online]. Available: https://study-ccna.com/ports-explained/

[13] City Network. (2021) *It-infrastruktur som tjänst med inbyggd regelefterlevnad*. Accessed 22nd April 2021. [Online]. Available: https://citynetwork.se/

[14] R. Dingledine, N. Mathewson, and P. Syverson, "*Tor: The Second-Generation Onion Router*," 2004.

[15] D. Fifield, N. Hardison, J. Ellithorpe, E. Stark, D. Boneh, R. Dingledine, and P. Porras, "*Evading Censorship with Browser-Based Proxies*," 2012.

[16] A. Færøy. (2021) *Reduced Exit Policy.* Accessed 12th Feb. 2021. [Online]. Available: https://gitlab.torproject.org/legacy/trac/-/wikis/doc/ReducedExitPolicy

[17] K. Gold and R. Kantrowitz. (2020) *New York - Data Protection Overview.* Accessed 16th Mar. 2021. [Online]. Available: https://www.dataguidance.com/notes/new-york-data-protection-overview

[18] Google. (2020) *HTTPS encryption on the web.* Accessed 22nd Januari 2021. [Online]. Available: https://transparencyreport.google.com/https/overview?hl=en

[19] A. Green. (2020) *New York - Data Protection Overview.* Accessed 16th Mar. 2021. [Online]. Available: https://www.varonis.com/blog/us-privacy-laws/

[20] S. Groš, M. Salkic, and I. Šipka, "*Protecting TOR exit nodes from abuse*," 2010.

[21] J. Hayes, "*Traffic Confirmation Attacks Despite Noise*," 2016.

[22] D. Hounslow and R. Nozaki. (2020) *Japan - Data Protection Overview.* Accessed 16th Mar. 2021. [Online]. Available: https://www.dataguidance.com/notes/japan-data-protection-overview

[23] H.-Y. Huang and M. Bashir, "*The Onion Router: Understanding a Privacy Enhancing Technology Community*," 2016.

[24] IEEE. (2021) *Advancing Technology for Humanity.* Accessed 22nd Feb. 2021. [Online]. Available: https://ieeexplore-ieee-org.miman.bib.bth.se/Xplore/home.jsp

[25] R. Jansen, N. Hopper, and Y. Kim, "*Recruiting New Tor Relays with BRAIDS*," 2010.

[26] R. Jansen and A. Johnson, "*Safely Measuring Tor*," 2016.

[27] E. D. Knapp and J. T. Langill. (2021) *Deep Packet Inspection.* Accessed 3rd Mar. 2021. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/deep-packet-inspection

[28] Z. Ling, J. Luo, K. Wu, W. Yu, and X. Fu, "*TorWard: Discovery, Blocking, and Traceback of Malicious Traffic Over Tor*," 2015.

[29] Linnaeus University. (2021) *Ethical Advisory Board in South East Sweden.* Accessed 1st Mar. 2021. [Online]. Available: https://lnu.se/mot-linneuniversitetet/samarbeta-med-oss/Projekt-och-natverk/etikkommitten-sydost/

[30] S. Malmberg. (2021) *Olovlig avlyssning.* Accessed 23rd Feb. 2021. [Online]. Available: https://lagen.nu/begrepp/Olovlig_avlyssning

[31] A. Mani, T. Wilson-Brown, R. Jansen, A. Johnson, and M. Sherr, "*Understanding Tor Usage with Privacy-Preserving Measurement*," 2018.

[32] MaxMind, Inc. (2021) *GeoIP2 City Accuracy.* Accessed 10th Mar. 2021. [Online]. Available: https://www.maxmind.com/en/geoip2-city-accuracy-comparison?country=&resolution=250&cellular=all

[33] B. Nahorney. (2019) *Threats in encrypted traffic.* Accessed 22nd

Januari 2021. [Online]. Available: https://blogs.cisco.com/security/threats-in-encrypted-traffic

[34] NFStream Developers. (2021) *APIs Documentation.* Accessed 4th Mar. 2021. [Online]. Available: https://www.nfstream.org/docs/api

[35] ——. (2021) *NFStream: Flexible Network Data Analysis Framework.* Accessed 16th Feb. 2021. [Online]. Available: https://www.nfstream.org/

[36] Official Journal of the European Union. (2021) *REGULATIONS, REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016.* Accessed 1st Mar. 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN

[37] Panda Developers. (2021) *Pandas.* Accessed 7th Mar. 2021. [Online]. Available: https://github.com/pandas-dev/pandas

[38] Proton Technologies AG. (2021) *Complete guide to GDPR compliance.* Accessed 8th February 2021. [Online]. Available: https://gdpr.eu/

[39] G. A. P. Rodrigues, R. de Oliveira Albuquerque, F. E. G. de Deus, R. T. de Sousa Júnior, G. A. de Oliveira Júnior, L. J. G. Villalba, and T.-H. Kim, "*Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection*," 2018.

[40] M. Roser, H. Ritchie, and E. Ortiz-Ospina. (2021) *Internet.* Accessed 8th February 2021. [Online]. Available: https://ourworldindata.org/internet#internet-access

[41] K. Shahbar and A. N. Zincir-Heywood, "*Traffic flow analysis of tor pluggable transports*," 2015.

[42] M. Song, G. Xiong, Z. Li, J. Peng, and L. Guo, "*A De-anonymize attack method based on traffic analysis*," 2013.

[43] Tor Project. (2019) *Tor: FAQ.* Accessed 9th Feb. 2021. [Online]. Available: https://2019.www.torproject.org/docs/faq.html.en

[44] ——. (2019) *Tor: Overview.* Accessed 21st Jan. 2021. [Online]. Available: https://2019.www.torproject.org/about/overview.html.en

[45] ——. (2019) *Tor: Pluggable Transports.* Accessed 9th Feb. 2021. [Online]. Available: https://2019.www.torproject.org/docs/pluggable-transports.html.en

[46] ——. (2020) *The Tor Project.* Accessed 10th Dec. 2020. [Online]. Available: https://www.torproject.org/about/history/

[47] ——. (2021) *Technical Setup.* Accessed 11th Feb. 2021. [Online]. Available: https://community.torproject.org/relay/setup/

[48] ——. (2021) *Tor Automatic Software Updates.* Accessed 15th Feb. 2021. [Online]. Available: https://community.torproject.org/relay/setup/guard/debianubuntu/updates/

[49] ——. (2021) *Tor Browser Security Settings.* Accessed 10th Feb. 2021. [Online]. Available: https://tb-manual.torproject.org/security-settings/

[50] ——. (2021) *Tor Eff legal faq.* Accessed 12th Mar. 2021. [On-

line]. Available: https://community.torproject.org/relay/community-resources/eff-tor-legal-faq/

[51] ——. (2021) *Tor exit page.* Accessed 29th Mar. 2021. [Online]. Available: https://community.torproject.org/relay/setup/exit/

[52] ——. (2021) *Tor Nyx.* Accessed 15th Feb. 2021. [Online]. Available: https://nyx.torproject.org/

[53] ——. (2021) *Tor Package Repository for Debian.* Accessed 15th Feb. 2021. [Online]. Available: https://support.torproject.org/apt/tor-deb-repo/

[54] ——. (2021) *Tor relay search.* Accessed 1st Apr. 2021. [Online]. Available: https://metrics.torproject.org/rs.html

[55] ——. (2021) *Types of tor relays.* Accessed 10th Mar. 2021. [Online]. Available: https://community.torproject.org/relay/types-of-relays/

[56] Tor Research Safety Board. (2021) *Tor Research Safety Board.* Accessed 21st January 2021. [Online]. Available: https://research.torproject.org/safetyboard/

[57] Your Europe. (2020) *Data protection and online privacy.* Accessed 15th Mar. 2021. [Online]. Available: https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm

[58] E. Zhang. (2020) *What is Operational Security? The Five-Step Process, Best Practices, and More.* Accessed 9th Feb. 2021. [Online]. Available: https://digitalguardian.com/blog/what-operational-security-five-step-process-best-practices-and-more

Appendix A

Appendix

A.1 Experiment infrastructure, Tor node setup

This section will walk through the installation of a Tor node. An exit node and other nodes have the same software but are different in the configuration. It is easy to start with a non-exit node and then change it to be an exit node from that point. This walk-through will use Ubuntu (20.04 Focal) Linux. The installation process will be similar to other operating systems as well. We chose the Linux operating system because it is well known, reliable and we were familiar with it. The Linux terminal makes it easy to make run programs, configure settings and work with the software.

The first thing to do is to enable automatic software updates. It is important to install security updates as quickly as possible. The Tor project describes how to do this in detail [48]. Second, configure the Tor package repository. The Tor project maintains its own package repository for Debian. They do this because the Debian repository i.e apt, apt-get, etc provides the LTS version of Tor. This means that the version might miss important updates. For this reason, they highly recommend using their package repository. The tor project provides a guide on how to do this [53]. Then update the system and install the tor software. This is easily done in the terminal with the following command. This will now use our new Tor repository.

```
$ sudo apt update && apt install tor
```

When the installation is finished it's time to configure. Open the torrc file located in /etc/tor/torrc. This configuration file includes all the configurations on how the node will operate. The following configuration is used in this experiment.

```
SocksPort 0
ControlPort 9051
CookieAuthentication 1
ORPort 443
IPv6Exit 0
Nickname <nickname>
RelayBandwidthRate 2000KB
RelayBandwidthBurst 4000KB
ContactInfo <yourmail@mail.com>
```

```
DirPort 80
DirPortFrontPage /etc/tor/tor-exit-notice.html
ExitPolicy accept *:*
ExitRelay 1
```

In the torrc configurations file, there are explanations of what the different settings do. *SocksPort 0*, is if you wish to connect to Tor from the local computer. Setting this to 0 will make it work only as a relay. *ControlPort 9051* and *CookieAuthentication 1* are for the use of nyx and the possibility to monitor the node. *ORPort 443* is the port for incoming tor connection, port 443 is the default. *Nickname <nickname>*, this is the name of the node. It is necessary, will be public and seen on the Tor project relay search [54]. *RelayBandwidthRate 2000 KB* and *RelayBandwidthBurst 4000 KB* is the way to limits the bandwidth the node has access to. *ContactInfo yourmail@mail.com*, contact information is necessary. The *DirPort 80* and *DirPortFrontPage /etc/tor/tor-exit-notice.html* uses port 80 to host a web page that informs users' about Tor. We use a template provided by the Tor project [51] that we downloaded to the servers. *ExitPolicy reject *:** is where we change the node to an exit node. *:* means no exits allowed i.e a middle node. This is where we regulate the exit policy and change it to an exit node. We use ExitPolicy accept *:* to allow all traffic. ExitRelay 1 states that it is an exit relay.

When the configuration is finished, restart the Service and it will start running with the right configuration.

```
$ sudo systemctl restart tor@default.service
```

Lastly, we can monitor the node with the program nyx. This is a monitoring software for Tor and it can be downloaded in the following way:

```
$ sudo apt-get install nyx
```

Then just run it in the terminal with:

```
$ sudo nyx
```

See the nyx website for more information [52].

Specifics for exit nodes

- Reverse DNS lookup (PTR)
- DNS resolver

To do reverse DNS lookup we contacted the hosting company we used. They can then connect the IP addresses of the server with a domain name we own.

Exit nodes do DNS resolution for Tor clients. Therefore we need to enable this. We do this by the following:

```
$ apt install unbound
$ cp /etc/resolv.conf /etc/resolv.conf.backup
$ echo nameserver 127.0.0.1 > /etc/resolv.conf
$ chattr +i -f /etc/resolv.conf
```

A.2 Parser script

This is the script that analyzed and parsed the traffic, before compressing and storing it. A service was created to continuously run the program, in case it was forced to restart for any reason. The config.py file below is the configuration file used by the parser.

```
# Standard imports
from os import remove, path
from zipfile import ZipFile, ZIP_DEFLATED
from datetime import datetime, timezone
import logging
import maxminddb
import json

# Extra imports
from nfstream import NFStreamer

# Local files
import static_files.config as conf


def convert_timestamp(timestamp):
    """
    Gets a 13-digit UNIX timestamp and converts it to a readable string format.
    :return: Timestamp (Str)
    """
    digits = len(str(timestamp))

    # Remove 3 numbers to get under 10 characters
    if digits > 10:
        timestamp = float(timestamp / 1000)

    utc_time = datetime.fromtimestamp(timestamp, timezone.utc)
    local_time = utc_time.astimezone()

    return local_time.strftime("%Y-%m-%d %H:%M:%S.%f")


def check_files():
    """
    Checks the file size. If too big the file gets rotated and zipped.
    """
    csv_file = f"{conf.folder_path}/output/{conf.csv_file}.csv"

    try:
        if path.getsize(csv_file) >= conf.max_file_size:
            conf.numb_zip_files += 1
            mode = "w"
            if path.exists(f"{conf.folder_path}/output/zipped_files/"
                           f"{conf.zip_file_name}"):
                mode = "a"

            # If zipfile exists append to it, else create it
            with ZipFile(f"{conf.folder_path}/output/zipped_files/"
                         f"{conf.zip_file_name}", mode, ZIP_DEFLATED) as zipper:
                zipper.write(csv_file, f"flow_{conf.numb_zip_files}.csv")

            logging.info(f"Rotated and zipped current CSV file. "
                         f"Zipped filename: flow_{conf.numb_zip_files}.csv")
            remove(csv_file)

    except Exception as e:
        logging.error(f"Failed to rotate and zip file. Error: {e}")


def get_country(ip):
    """
    Gets country based on IP.
    :param ip: IP-address (Str).
    :return: Name of country (Str) and if IP is anonymous proxy (Boolean).
    """
    # Initialize connection to database
    try:
        reader = maxminddb.open_database(f"{conf.folder_path}/static_files"
                                         f"/GeoLite2-Country.mmdb")

        # Get country from IP
        results = reader.get(str(ip))
        reader.close()

        country = "Unknown"
        if results is None:
            return f"Local IP:{ip}, False"

        if "country" in results:
            country_data = results["country"]  # Standard country dict
        elif "registered_country" in results:
            country_data = results["registered_country"]  # Anonymous proxy
        elif "continent" in results:
            country_data = results["continent"]  # In case the country is not
            # found
        else:
            logging.error(f"No key was found in the data: {results}. IP: {ip}.")
            return f"Unknown (Error), False"

        country = country_data["names"]["en"]

        is_proxy = False
        if "traits" in results:
            if "is_anonymous_proxy" in results["traits"]:
                is_proxy = results["traits"]["is_anonymous_proxy"]

    except Exception as e:
        logging.error(f"Error when getting country from IP. "
                      f"IP: {ip}\nError: {e}")
        return f"Error: {ip}, False"

    return f"{country}, {is_proxy}"


def write_global_stats(data, file_name):
    """
    Loads global data file, appends it and write it back.
    :param data: Data to be appended (Dict).
    :param file_name: Filename (Str).
    :return: None.
    """
    # Check if file needs to be created
    if path.isfile(file_name):
        with open(file_name, "r+") as f:
            data_from_file = json.load(f)
            data_from_file["Flows"] += data["Flows"]
            data_from_file["Failed_flows"] += data["Failed_flows"]
            data_from_file["Megabytes"] += data["Megabytes"]
            data_from_file["Packets"] += data["Packets"]

            # Reset file pointer to beginning
            f.seek(0)
            json.dump(data_from_file, f, indent=4)
            # Removes all of the file content after the specified
            # number of bytes (i.e extra stuff)
            f.truncate()
    else:
        with open(file_name, "x") as f:
            json.dump({"Flows": data["Flows"],
                       "Failed_flows": data["Failed_flows"],
                       "Megabytes": data["Megabytes"],
                       "Packets": data["Packets"]}, f, indent=4)


def write_to_flows_csv(data, file_name):
    """
    Create data string for csv file.
    :param data: Data to be appended (Str).
    :param file_name: Filename (Str).
    :return: None.
    """
    if not path.isfile(file_name):
        # If no csv file exist, set first line
        with open(file_name, "w+") as f:
            f.write("App_name, Timestamp, Category, Src_Country, Src_is_proxy,"
                    " Dst_Country, Dst_is_proxy, Port,"
                    "Bidirectional_duration_ms, If guessed, If HTTP, "
                    "content-type \n")

    with open(file_name, "a+") as f:
        f.write(data)


def capture_traffic(network_flows):
    """
    Captures data from interface through NFStreamer and collects flow data.
    """
    csv_data = ""
    flows_stored = 0
    global_stats = {"Flows": 0,
                    "Failed_flows": 0,
                    "Megabytes": 0,
                    "Packets": 0}

    logging.info("Acquired NFStream object. Starting flow data collection.")

    for flow in network_flows:
        try:
            # Skip all flows coming from SSH jumphost
            if "SSH" in flow.application_name and conf.proxyhost in flow.src_ip:
                continue

            # Add one to total flows
            flows_stored += 1

            # Classify flow as failed if packet is less
            # than TCP handshake + 1
            if flow.bidirectional_packets <= 4:
                global_stats["Failed_flows"] += 1
                continue

            # Get global statistics
            global_stats["Flows"] += 1
            global_stats["Packets"] += flow.bidirectional_packets
            global_stats["Megabytes"] += flow.bidirectional_bytes / 1000000

            # Append data from current flow
            csv_data += f"{flow.application_name}, "
            csv_data += f"{convert_timestamp(flow.bidirectional_first_seen_ms)}, "
            csv_data += f"{flow.application_category_name}, "
            csv_data += f"{get_country(flow.src_ip)}, "
            csv_data += f"{get_country(flow.dst_ip)}, "
            csv_data += f"{flow.dst_port}, "
            csv_data += f"{flow.bidirectional_duration_ms}, "

            if flow.application_is_guessed == 1:
                csv_data += "True, "
            else:
                csv_data += "False, "

            # Add extra dict keys if HTTP content
            if "HTTP" in flow.application_name:
                csv_data += f"{flow.user_agent}, "
                if flow.content_type:
                    csv_data += f"{flow.content_type}\n"
                else:
                    csv_data += f"No content-type\n"
            else:
                csv_data += "Not HTTP\n"

            if flows_stored >= 100:
                # Write to file and restore values
                write_to_flows_csv(csv_data, f"{conf.folder_path}/output/"
                                             f"{conf.csv_file}.csv")
                write_global_stats(global_stats, f"{conf.folder_path}/output/"
                                                 f"{conf.global_file}.json")

                csv_data = ""
                global_stats = {"Flows": 0,
                                "Failed_flows": 0,
                                "Megabytes": 0,
                                "Packets": 0}
                flows_stored = 0

                # Check if file rotation is needed
                check_files()
                conf.written_flows += 100

                if conf.written_flows >= 5000:
                    logging.info("Parser has successfully stored 5000 flows.")
                    conf.written_flows = 0

        except Exception as e:
            logging.error(f"Unable to save flow data. Current flow: {flow}"
                          f"\nError: {e}")
            global_stats["Failed_flows"] += 1
            continue


def get_streamer():
    """
    Get's an NFStreamer object. Exits the program if it is unable to
    acquire the object.
    :return: NFStreamer object (Class Obj)
    """
    attempts = 0
    while attempts < 3:
        try:
            network_flow = NFStreamer(source=conf.interface_name,
                                      active_timeout=3600)
            return network_flow

        except Exception as e:
            logging.error(f"Failed to get NFStream object on attempt nr "
                          f"{attempts}. Error: {e}")
            attempts += 1

    logging.critical(f"Failed to establish NFStream object on 3 attempts. "
                     f"Exiting program.")
    exit(0)


if __name__ == '__main__':
    # Initialize log file
    logging.basicConfig(filename=f'{conf.folder_path}/main.log',
                        format='%(levelname)s:%(asctime)s: %(message)s',
                        level=logging.DEBUG)
    logging.info("Starting program run...")

    # Start NFStreamer
    nfstream_obj = get_streamer()
    capture_traffic(nfstream_obj)
```

config.py

Configuration file for the parser script.

```
csv_file = "flow_data"
global_file = "global_data"
zip_file_name = "flow_data.zip"
folder_path = "<path to parser script folder>"
proxyhost = "<Proxy host IP>"

numb_zip_files = 0
written_flows = 0
max_file_size = 1073741824  # 1 GB

interface_name = "<Tor network interface name>"
```

A.3 Additional results

This section shows more of the top used protocols, ports, and countries that were not visualized in the results sections. The number is a counter on how many flows had this attribute.

Top 40 most used protocols

Protocol	Flows
TLS	17025756
Unknown	10242244
TLS.Cloudflare	6725942
HTTP	5900572
SMTP	5650151
TLS.Google	3599623
TLS.Tor	3080808
HTTP.Cloudflare	2897119
TLS.Amazon	2311471
TLS.Twitch	843340
TLS.YouTube	732434
SSH	691776
TLS.Facebook	545249
TLS.Steam	416943
TLS.Yahoo	360228
TLS.SMTPS	358780
TLS.Microsoft	275485
TLS.NetFlix	239990
HTTP_Proxy	212593
HTTP.Amazon	208567
Whois-DAS	188041
TLS.GoogleServices	184592
HTTP.Microsoft	180802
BitTorrent	174511
HTTP.MS_OneDrive	173150
TLS.Twitter	170501
TLS.POPS	147154
HTTP.Google	129470
Steam	114576
TLS.Telegram	105871
TLS.TikTok	104593
TLS.AppleiTunes	98361

TLS.Instagram	340618	MQTT	95140
HTTP.Steam	331866	IMAPS	94859
SMTPS	307074	TLS.Microsoft365	94816
TLS.IMAPS	290853	MongoDB	83560

Top 40 countries that sent data to the exit nodes

Country	Flows		
Germany	1006680	Austria	30345
United	684455	Spain	28665
France	468527	Ireland	25756
Russia	236240	Bulgaria	25369
Netherlands	232510	Japan	23480
Finland	184275	Moldova	21833
Canada	174411	Singapore	20645
United	147348	Indonesia	18790
Sweden	89577	Luxembourg	18362
Switzerland	75368	Hungary	18014
Lithuania	57219	Mexico	11948
Czechia	54419	Greece	11039
Norway	53052	Israel	10232
Italy	46613	Portugal	10064
Latvia	46008	Kazakhstan	8818
Ukraine	43941	Estonia	8356
India	38266	Hong Kong	8217
Poland	36255	Belgium	7073
Romania	35487	Australia	5926
Denmark	33104	Panama	5769

Top 40 countries that received traffic from the exit nodes

Country	Flows		
United States	31001644	India	242656
Netherlands	6221182	Australia	233595
United	3128345	Austria	21014
Germany	3112332	Czechia	211955
Russia	3007574	Taiwan	201197
France	2264933	Spain	197359
Japan	1980225	Latvia	169698
Switzerland	1299799	Turkey	154791
Ireland	1200437	Vietnam	123865
Singapore	1066878	Antigua and Barbuda	123751
Sweden	1039544	Iran	111727
Canada	991046	South Africa	108064
Hong Kong	791636	Romania	100173
South Korea	541454	Argentina	96648
Brazil	411278	Panama	94262
Poland	359060	Morocco	89772
Ukraine	351609	Belize	70291
Finland	322512	Denmark	69027
Italy 2	78488	Bulgaria	67015
China	246246	Norway	66845

All categories

Category	Flows		
Web	40209955	VoIP	200390
Unspecified	10242864	RPC	196278
Email	6903821	System	156030
VPN	3126581	Streaming	143265
SocialNetwork	1247621	Music	115157
Video	1125686	Database	107004
Game	916522	Shopping	58322
Media	752993	IoT-Scada	26757
RemoteAccess	740755	ConnectivityCheck	22065
Network	383520	SoftwareUpdate	20900
Cloud	382171	Mining	4075
Download-FileTransfer-Fi	300289	DataTransfer	1752
Chat	286982	VirtualAssistant	278
Collaborative	201067		

Top 40 most used destination ports

Port	Flows		
443	34941370	27017	104028
5222	8118779	53	102421
80	7937378	2000	87855
25	5907677	3389	68221
22	629224	853	64252
8118	498381	8081	61577
465	483023	30005	57143
8080	475254	2052	47789
587	437006	2222	40553
993	420429	30000	39823
25565	414228	8088	38832
4003	241627	8085	35953
8000	232830	444	35687
25461	223085	8880	33029
43	186763	2078	32987
8443	160766	0	31663
995	153063	8010	29995
50001	147035	8888	28812
5060	123225	25000	28479
7547	106085	4433	28473

Top 40 content-types

Type	Flows
text/html	1536438
application/json	688880
application/x-www-form-urlencoded	249966
text/plain	156568
text/javascript	116017
application/octet-stream	99890
application/ocsp-request	48815
text/xml	37700
application/ocsp-response	23990
multipart/form-data	20634
image/jpeg	14641
image/gif	11886
application/xml	11261
application/javascript	9521
image/png	8320
text/css	6614
html/text	2869
audio/mpeg	1915
application/binary	1561
video/mp4	1542
application/x-javascript	1535
image/webp	1347
binary/octet-stream	1016
application/rss+xml	853
image/vnd.microsoft.icon	830
image/x-icon	780
application/pdf	755
application/x-rtsp-tunnelled	644
application/vnd.apple.mpegurl	500
application/zip	458
text/json	412
image/svg+xml	377
application/x-gzip	244
/bin/sh /proc/self/fd/0	212
application/atom+xml	179
application/vnd.bestbuy.v1+json	174
text/vnd.trolltech.linguist	118
x-www-form-urlencoded	114
font/woff2	88

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden