Investigation on the presence of third-parties on EU websites before and after GDPR

Master thesis

Michal Krištofik

Innovative Communication Technologies and Entrepreneurship
Aalborg University Copenhagen, Denmark
September 2020 – March 2021

Front page image: © cpomagazine.com

AALBORG UNIVERSITY
COPENHAGEN

Aalborg University Copenhagen
A.C. Meyers Vænge 15
2450 København SV

Semester Coordinator: Henning Olesen

Secretary: Charlotte Høeg

Semester: Fourth, service development (4SER)

Title: Investigation on the presence of third-parties on EU websites before and after GDPR

Project Period: September 2020 – March 2021

Semester Theme:
Master Thesis

Supervisor(s):
Sokol Kosta
Jannick Kirk Sørensen

Project group no.: n/a

Members
(do not write CPR.nr.):

Michal Krištofik - 20156728

Abstract:

The implementation of General Data Protection Regulation (GDPR) in May 2018 has caused many discussions about compliance and its effect on the web. In this project we explore if the commencement of GDPR had an effect on the presence of third-parties on sites with EU/EEA origin. The data for analysis are collected in period between February 2018 and June 2020 with total of 59 successful harvests. In each harvest 12 778 sites are visited, with 10 089 having the EU/EEA origin. We analysed the development of TPs, HTTP responses, and status of visited sites and found that there was an initial small drop in the number of TPs after the GDPR commencement followed by the slow gradual decrease over harvests. Due to the data-harvesting methodology where the same set of sites is visited every harvest, we observe that in every harvest fewer sites return 200 – OK HTTP response status. This, in combination with the lack of strong evidence in favour of GDPR, leads to a conclusion that we cannot attribute the decrease in number of TPs to the commencement of GDPR. Secondly, we analyse the purpose and maliciousness of TPs.

Pages: 71
Finished: 31.03.2021

Contents

List of acronyms and abbreviations

AAU Aalborg University
API Application Programming Interface
CPU Central processing unit
CRISP-DM Cross-industry standard process for data mining
CSV Comma-separated values
DF Dataframe
DNS Domain name system
DPD Data Protection Directive
EEA European Economic Area
EU European Union
FP First-party
GDPR General Data Protection Regulation
GSB Google Safe Browsing
HTTP Hypertext transfer protocol
IAB Interactive Advertising Bureau
ICANN Internet Corporation for Assigned Names and Numbers
IP Internet protocol
ISP Internet service provider
JSON JavaScript object notation
OSEMN Obtain, Scrub, Explore, Model, and iNterpret
RAM Random-access memory
RD Root domain
RFC Request for Comments
TDSP Team Data Science Process
TP Third-party
TXT Text
URL Uniform resource locator
VM Virtual machine
VPN Virtual private network

1 Introduction

Online services of today are often a combination of original content and third-party resources which are embedded regularly and online services often depend on them. Third-parties allow services to be deployed faster and easier by using external libraries to implement features, to obtain analytics of user behavior, provide revenue stream by employing ad networks, or have the content loaded faster by using content delivery networks. As a result that leads to higher dynamics on the web, dozens of extra requests, and data sharing. This is especially true these days, when providers need to monetise the content in order to provide it for free, by dedicating parts of websites to display ads by ad networks and generate revenue.

This inclusion of third-parties (TPs) comes at the cost of service providers losing control over what is being loaded on their sites, especially in the case of ad networks and sometimes libraries. The inclusion of a single TP, can result in requests to additional TPs, creating third-party trees, posing a privacy and security risk [1]. A popular way of sharing information among TPs are cookies, which can also be considered personal data under certain circumstances according to General Data Protection Regulation (GDPR) and violate it [2].

Among the aims of the GDPR, is to enhance protection and give more control to users over their data [3]. Since it came into force in May 2018, many studies have focused on the GDPR and its effects on the web, such as the presence of trackers [4, 5] and third-parties [1, 6, 7], or data sharing [8, 9]. The common assumption was that since providers need to review their cookie policies, data collection and its purpose, data sharing, or embedded TPs to achieve the compliance, the presence of third-parties, trackers or cookies would decrease.

We focus on TPs and analyse their presence on the web before and after the commencement of GDPR to find out if it had any substantial effect. The study by Sørensen and Kosta [6] is interesting in particular since our research is based on the same, but even wider-reaching, data-set. As part of the research, we will be therefore extending their study and comparing some metrics, to see if any changes can be observed but also coming up with our own analysis.

In this thesis a longitudinal study analysing the presence of TPs on the sites with EU/EEA origin before and after GDPR is presented. The data were harvested in the period from February 2018 to June 2020, where, from the total of 1 250 sites, 12 778 sub-pages are visited every harvest. The analysis focuses on the 10 089 visited sites each harvest that have the EU/EEA origin, where 6 868 unique TPs were found in total. We have analysed the development of TP levels across harvest while supporting the findings with an analysis on the number of responses across harvests and 'health' status of visited sites followed by finding trends for each of 11 categories of visited sites. Based on results from all these analyses, we can conclude on general level that the number of TPs have decreased slightly through the harvesting period, but it cannot be directly attributed to the effect of GDPR due to its insignificance and study design. Secondary, we have analysed the properties of TPs such as their maliciousness and purpose.

1.1 Background

The following section provides the explanation of regulation and technical terms, understanding of which is beneficial for better comprehension of the topic and problem in subsequent sections.

As already mentioned, this thesis aims to find whether the GDPR influences the presence of TPs on European websites. Therefore to begin with, what GDPR is, and its focus is described. Afterwards, multiple techniques employed on the web for tracking of users are elucidated. Firstly, the concept of third-parties and trackers is covered, followed by cookies and cookie syncing and closing with browser fingerprinting.

Even though the aim of this thesis is not to analyse the presence of tracking techniques, we believe that it is crucial to explain them to obtain a bigger picture, since they influence the presence of TPs.

1.1.1 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (EU) 2016/679 or GDPR[1] in short, is an EU law that replaces Data Protection Directive (DPD)[2] (officially Directive 95/46/EC) which came into force in December 1995. The proposal for GDPR was first adopted by the European Commission in January 2012 [10], then four years later in April 2016 it was approved and came into force on 25 May 2018 [3]. It applies and is legally binding to all EU as well EEA member states.

One of the differences between DPD and GDPR is already the document type, as the directive sets an objective which each member state has to translate to their own laws, which results in different legislation in member states, and the regulation is already a binding legislative act which applies the same way to all member states. This results in a unified approach to data and their protection, and a single digital market. As DPD came into force in 1995 and the technology advancement since then was enormous, user behavior changed, and many more people have access to internet and services, it became weak in defining what are the personal data, and rights which an individual has to their data became deficient.

The GDPR aims to *"enhance data protection rights of individuals and to improve business opportunities by facilitating the free flow of personal data in the digital single market"*[10]. GDPR can therefore be seen as an evolution of the DPD which gives the rights and power to an individual over their personal data and concretely define terms, such as personal data, mirroring the technological advancements.

It does not only apply to all entities gathering, processing, or storing data on the grounds of EEA member states, but also to all entities outside EEA gathering, processing, or storing data of individuals of member states. Therefore, its impact reaches beyond EEA borders, which is known as Brussels's effect[11].

Privacy by design is established and organisations always must know who is responsible

[1] `https://eur-lex.europa.eu/eli/reg/2016/679/oj`, accessed 03 March 2021

[2] `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A31995L0046`, accessed 03 March 2021

and how they process personal data in the "supply chain". Among motivations for complying with the GDPR is the penalty of up to 20 million EUR or up to 4% of the global turnover of the preceding financial year, depending on the infringement.[3]

1.1.2 Users tracking on the web

There are many ways how a user can be tracked online. These days, popular tracking techniques are fingerprinting, trackers, and cookies which are explained further in the section.[12, 13] All of the above mentioned techniques influence the presence of TPs on given website, therefore it is important to understand the concepts behind them.

Third-parties A Third-party (TP) is an entity that is not directly and intentionally visited by a user but is requested while loading the first-party website. Often they provide crucial services for the proper functioning of the website since they can load libraries, fonts, or resources used on the website. On the other hand, TPs can also *"pose risks to users, which is obviously unintended by the service provider. For example, third parties can create security problems (e.g., malvertising), might have negative privacy implications (e.g., trackers), or they can include content that might impact users in other negative ways (e.g., crypto-miners)."* [1] Nonetheless, it is important to point out that loading a TP does not automatically mean it is unwanted or a threat.

Sometimes, as already mentioned, third-parties function as trackers, tracking users' behavior, and browsing history to create a user's profile. They use cookies to track users and build those profiles, and cookie syncing to share this information with each other (see further in the section). There are (i) Single-website trackers, and (ii) Cross-website trackers.

Single-website trackers They are usually third-party trackers which are run by a different company and a request is made to servers other than the first-party's. They use first-party cookie which is not shared across websites and stay only within the first-party domain. Cookies can be created by using a tracking pixel which is a small JavaScript code that creates a transparent 1x1 pixel. Every time the website is loaded, the browser also makes a request for the pixel to the tracker which tracks the user's activity based on the unique user ID from the cookie and therefore provide analytics. An example of a TP tracker is Google Analytics[3] which tracks users on thousands of websites but the information is not shared among them since the first-party cookie is set.

Cross-website trackers On the other hand, even though these are TP trackers as well, they make use of Third-party cookies which are attached to the tracker. This means, that the same user, will have the very same unique ID set by the tracker passed along in cookie on all sites the tracker is deployed at, allowing it to build an extensive user profile of a given user and, for example, target ads much better.

[3] `https://analytics.google.com/`, accessed 03 March 2021

Cookies HTTP Cookie *"is a small piece of data that a server sends to the user's web browser. It remembers stateful information for the stateless HTTP protocol"*[14]. An important part of it is a unique ID that distinguishes a given user from others. The cookie is saved by the browser and upon every following request, it is sent to the server which can identify the user by that unique ID.

Generally cookies are used for: (i) Session management (e.g. session cookie), (ii) Personalisation (e.g. first-party cookie), (iii) Tracking (e.g. third-party cookie) [15]. Session and personalisation cookies are generally very useful as they can be used to remember the users' session so they don't have to login every time they access the page or remember the settings they have on a webpage. On the other hand, third-party cookies are usually not of big benefit to a user, since such cookies let advertising companies track users among websites, building their user-profiles and serving ads.

Cookie syncing A cookie is domain-specific and as every advertiser is publishing ads only on a portion of websites, one can only track a portion of the user's behavior online. The challenge is to track users on as many websites as possible, to have as much data about a user as possible to build detailed profiles, and serve personalised ads. Therefore, advertisers partner up and share user's data (sync cookies) with each other. As described in Figure 1, when a user visits a website, an ad (or third-party tracker) sends a request to the server which creates a unique ID and assigns it to a user, if it does not yet exist, and stores it in a cookie (C1). As part of the request, the user is then redirected to the partner server as well (C2) which contains the unique ID of the user (C3). The partner server checks if it already has a user corresponding to the received ID for the site, if not, it creates it, and stores it in the cookie-matching table (C4). If the syncing is bidirectional, the partner server can then pass its unique user ID as part of the redirect back to the advertiser, and then both can sync user data (C5).

Such behavior raises concerns about users' privacy and data sharing. In 2020, Urban et al. looked into how GDPR impacted data sharing in ad networks[8]. They found out that *"GDPR has a statistically significant impact on cookie syncing, which is reduced by around 40%"*[8] and that even though the number of cookie syncing dropped, the number of third-parties have not changed much.

Browser fingerprinting Fingerprinting is a tracking technique that identifies a user based on hardware and software properties and settings of their device. The obtained information is then usually transformed into an identifier. Browser fingerprinting uses available browser and device information such as language, screen size, or location for identification.[16] With the use of cross-browser fingerprinting, it is possible to identify up to 99% of users as demonstrated by Eckersley [17]. In [18], Hupperich et al. described how fingerprinting can be used to achieve price discrimination based on the location. This technique is also resilient to *"multiple recommended privacy precautions (masking your IP address through a VPN and deleting or blocking cookies) trackers can still use your digital fingerprint to re-identify and re-cookie your device when you visit a website"*[16].

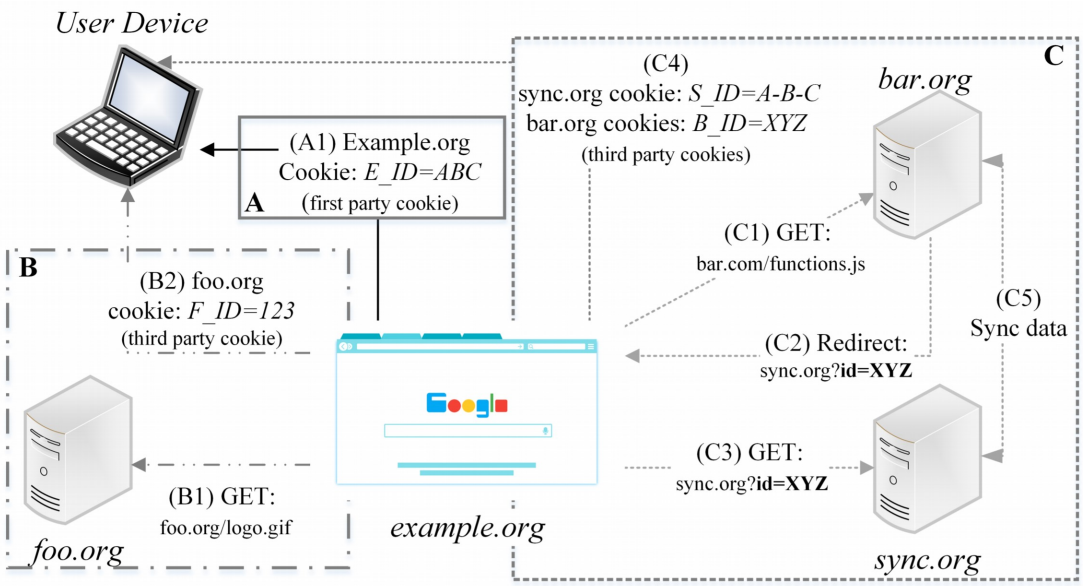

Figure 1: Cookie types: (A) first-party cookie, (B) third-party cookie, and (C) synchronised cookie. Taken from [9].

The problem of user tracking on the web and users' privacy are topics on their own, are outside of the thesis's scope, and this section only scratched their surface. The aim though, was to briefly explain what GDPR is and that it can have an influence on how a user is tracked online. Also various ways of how it can be done and that they influence the presence of TP is outlined.

As already mentioned, third-parties are often crucial for first-party functionality it is often not an issue that they are loaded. Some TPs are however used for tracking users' activities on the web, making use of cookies, which can also be synced among TPs, or fingerprinting. All of this contributes to the higher number of third-party records when requesting a first-party. The distinction between tracking techniques and third-party types is outside of the scope of this thesis and only TPs, in general, are analysed as also stated by the problem formulation in the following section.

1.2 Problem formulation

Multiple studies have been conducted over the years looking into the presence of third-parties before and after GDPR, but usually over the span of a single or a few harvest. Therefore, doing the longitudinal study on harvests over almost 2.5 years can reveal interesting results, possibly also outside the scope of GDPR. This brings us to the problem formulation which we will answer by this thesis:

"Can any changes in the presence of third-parties on European websites be observed and attributed to the commencement of General Data Protection Regulation (GDPR)?"

Sub-questions To support the problem formulation and aid in keeping on the right track during the process, the problem formulation is partitioned to more manageable focus areas: data obtaining, data scrubbing, data exploration, and data interpretation. These can be transformed into a set of sub-questions that can be used as guidance during the process:

- Which enrichment data needs to be obtained for more detailed third-party analysis?
- What is the structure of data-sets?
- How to prepare and process harvest data-sets for the analysis?
- How to transform processed data to visualised results?
- How can the results be interpreted?

All sub-questions are further broken down, answered, and steps taken are documented in the respective sections of Chapters 4 and 5, while the problem formulation as a whole is answered in Chapter 7.

1.3 Report structure

We begin this thesis by introducing the problem area, background, and setting up the problem in Chapter 1. We then explain the framework followed throughout the thesis, its processes and all phases we underwent during the work with data in Chapter 2. Relater works and technologies used for implementation are covered in Chapter 3. The biggest chapter is Chapter 4, which is structured according to the framework followed and each process is described there. It begins with how data are obtained - Section 4.1 and what is their structure - Section 4.2. Then the longest process covering how data are cleaned, transformed, enriched, or reduced is explained in Section 4.3 followed by data exploration and visualisation in Section 4.4. The output of previous chapters is then processed and results are presented in Chapter 5. Reflection on the work and future plans are outlined in Chapter 6 and the thesis is concluded with the problem formulation answer in Chapter 7.

2 Methodology

To answer the problem formulation defined in the previous section, we need to obtain, process, and interpret vast amounts of data. One of the fields that extract knowledge and insights from data is Data science. It is an interdisciplinary field that focuses on *"the study of the generalisable extraction of knowledge from data"*[19]. Techniques and theories from multiple fields such as mathematics, computer science, information science, or statistics are utilised to obtain the knowledge from data. Data scientists therefore must have a vast skill set spanning multiple fields and make use of various methods to extract knowledge and insights from data. Since we need to extract the knowledge from data to answer the problem formulated, we consider this thesis to be a data science project.

This chapter covers data science models, in particular OSEMN framework and Gantt chart - the project schedule tracking method.

2.1 Data science process

Every field of study has its specifics, best practices, and models to follow. Data science is no exception. There are many models one can follow, such as such as Cross-industry standard process for data mining (CRISP-DM)[20], Team Data Science Process (TDSP)[21], or Foundational Methodology for Data Science[22].

There is also an OSEMN framework, which was first published in 2010 by Hilary Mason and Chris Wiggins in an article on *A Taxonomy of Data Science*. They coined the term OSEMN which describes steps that most data scientists follow: Obtain, Scrub, Explore, Model, and iNterpret the data. According to them: *:"Different data scientists have different levels of expertise with each of these 5 areas, but ideally, a data scientist should be at home with them all"*[23]. Since then, this term has gained popularity in the circles and is often referred to when describing tasks that one should be comfortable working on when working with data [24][25]. The OSEMN framework processes are briefly explained in Figure 2, which shows all phases in chronological order, but one can move across phases as needed and eventually go through all of them.

The above-mentioned methodologies can all be applied in the course of working on this thesis; however, the most suitable one should be chosen and followed throughout. Since there is a single person working on the thesis, and not a team that needs to collaborate and know who is working on what, and what is the progress, a more simple methodology can be chosen. There is also no need for customer acceptance, or machine learning application. Because of that, the OSEMN framework is followed.

The OSEMN framework begins with the *obtain* process to obtain the data sets. At the same time however, before obtaining the data to do the 'data science' there is a need to know what that data is needed for, to set a research goal and frame a problem to solve. That is the fundamental step and the right questions need to be asked in order to produce actionable output, which is only as good as the problem itself: *"Far better an approximate answer to*

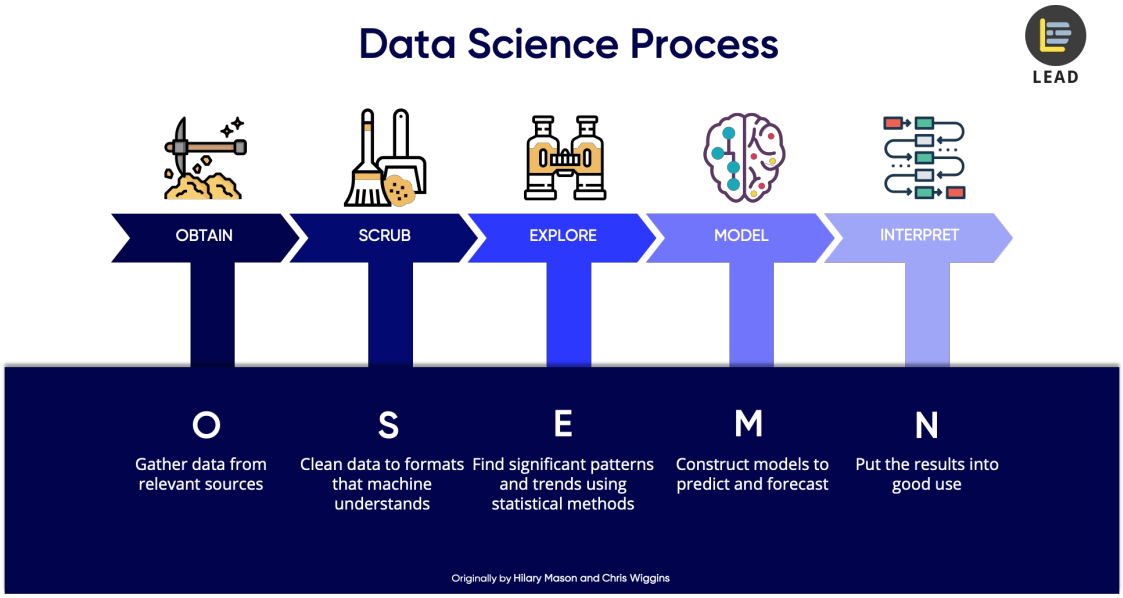

Figure 2: All phases in OSEMN in chronological order with a short explanation. The data scientist does not have to follow the chronological order of processes, they can go back and forth between the phases. The main idea though is that one will go through all phases eventually. Taken from [26].

the right question, which is often vague, than the exact answer to the wrong question, which can always be made precise"[27] which only confirms the importance of this step.

2.1.1 Obtain

It is rather rare, that one has access to all data-sets required. The skill of obtaining the data is therefore a fundamental part of the skill-set of every data scientist.

The process of obtaining necessary data can be divided into: (i) finding and curating suitable data, and (ii) creating your own. In the first case, it is often about querying data from databases or APIs, downloading them from different locations, or extracting them from other files. The latter is about automated data retrieval (e.g. web scraping) or scripting. In both cases, multiple data sources are often required.

2.1.2 Scrub

Once the data are obtained, they need to be scrubbed – cleaned, as, due to inconsistencies, it is relatively rare that they would be ready to be used right away. This step is rather time-consuming as *"80% of the work in any data project is in cleaning the data"*[28], but the time invested is worth it as cleaned data are easier to work with, and generally produce better results.

There are several reasons that may cause those inconsistencies: missing values, inconsistent labeling, or wrong data format. On the other hand, sometimes only certain data

from the data set are needed; therefore filtering may be applied, or certain columns of data set may be extracted. After this step, data should be uniform and consistent even though additional post-processing may sometimes be needed.

2.1.3 Explore

Having the data cleaned, they can be explored – by doing exploratory data analysis because no predictions are made in this step and the hypothesis is not being tested. Data are rather looked into, zooming in and out, to understand the information contained within and getting to know them even better. There are two main objectives: (i) to find patterns by visualising, (ii) to extract features to identify and test significant variables (statistics).[29]

2.1.4 Model

Throughout this process, data scientists apply their knowledge of mathematics and statistics, and utilise various data science tools. The main objectives here are the creation of predictive models and their subsequent evaluation and refinement. The aforementioned models are then employed for predictions and interpretation of the data set.

This step begins with reduction of the dimensionality of the data set as some features are not crucial for creating the model; therefore only relevant ones should be chosen. Afterward, one of the approaches is to split the data set into training and validation sets. The model is trained on the training set and techniques as clustering, classification, or associations are used on it. Once the model is in place, validation data are applied to it and loss functions of both models are compared – so-called cross-validation. Even after the validation, the model is not automatically correct, as George Box said *"all models are wrong, but some are useful"*[30]. The goal is to create a more accurate model which is often also the most predictive.

2.1.5 Interpret

In the end, the result of all the above-mentioned steps produces actionable insights which need to be interpreted and presented. The main objectives of this step are visualisation of findings, evaluation of results, reaching a conclusion, and communication of outcome. The results need to be driven by the question or problem laid down in the beginning. They often need to be presented to an audience with a non-technical background and in such a manner that their value can be seen and understood by them. As Albert Einstein said: *"If you can't explain it simply, you don't understand it well enough"*. Thus, non-technical skills are of great importance.

Application of the OSEMN framework This framework is to be followed in this thesis, as already mentioned at the beginning of this chapter. The framework has five processes, and all except one - *Model*, are applied. The reasoning behind not having this process is the scope of this thesis, where the creation of predictive models, its training, and validation are

not part of the process. As the problem formulation states, the aim is to analyse historical data and interpret the results.

While working on the thesis and following the framework, seven individual phases can be identified spanning one or more processes:

1. Obtain harvest data
2. Process and generate unique TPs for all responses
3. Obtain and scrub enrichment data
4. Enrich all responses with enrichment data
5. Process and generate data for visualisation
6. Explore and visualise data
7. Results and conclusion

The phases' jumps or iterations among processes were not planned beforehand but evolved on the go. They are the results of becoming more familiar with data the more one works with them, and realisation of what can further be reported by them. This often leads to finding that the data need to be processed further, in a different way, or more data need to be obtained. The OSEMN framework allows such iterations and jumping. Even though it depicts the processes in chronological order, one does not have to follow them in a waterfall fashion. This can be seen in Figure 3, where the seven phases are presented with respect to the OSEMN processes.

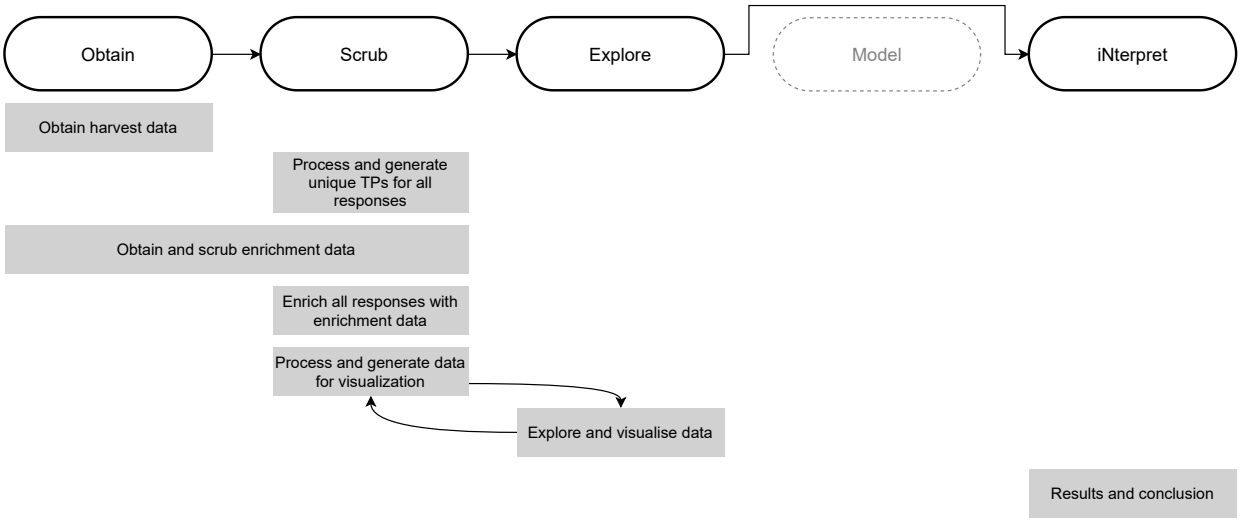

Figure 3: All phases connected with data work in this thesis with respect to the OSEMN framework processes.

Every phase is further explained in Chapter 4, where all steps and work carried out are presented following the OSEMN structure, except the last phase – *Results and conclusion* which is covered in Chapter 5 and 7.

2.2 Gantt chart

There exist various tools which ease project management, planning, and scheduling. The available time for this thesis is limited, therefore a tool that can illustrate a project status is needed. We chose the Gantt chart as a tool, since already being familiar with the it.

A Gantt chart can be described as a bar chart, with a horizontal axis depicting time and a vertical axis containing tasks, offering a visual representation of those scheduled tasks over time; therefore it is a useful tool for tracking the project schedule. It further shows the start and finish date of tasks, and the width on the horizontal axis represents the duration of the task. It can also present tasks' dependencies, relationships among them, who is responsible for a given task, their overlapping and finish date of the project.[31]

Application of the Gantt chart The whole period of this thesis can be divided into four high-level task groups: (i) general, (ii) single harvest, (iii) implementation, and (iv) documentation. The Gantt chart is created using an online tool[4], and the latest version of it is presented in Figure 4. Each task group contains multiple general tasks with a weekly schedule throughout the thesis period.

Tasks in the *general* group lay down the foundations on which the rest of the thesis is built. The thesis topic is defined and researched there, and necessary skills for data processing sharpened. Afterwards, a VM is set up and harvest data-set is obtained. Then, a single harvest data are loaded and its structure is explored to see what can be done with data and what is missing, and initial scripts written in *single harvest* group. Then the scripts are combined in the *implementation* to work on all harvests and are further improved. Enrichment data are obtained and all data are scrubbed and explored in this part to produce visualisations and results that can be interpreted. Once this is done, everything is documented and the whole thesis written in *documentation* group.

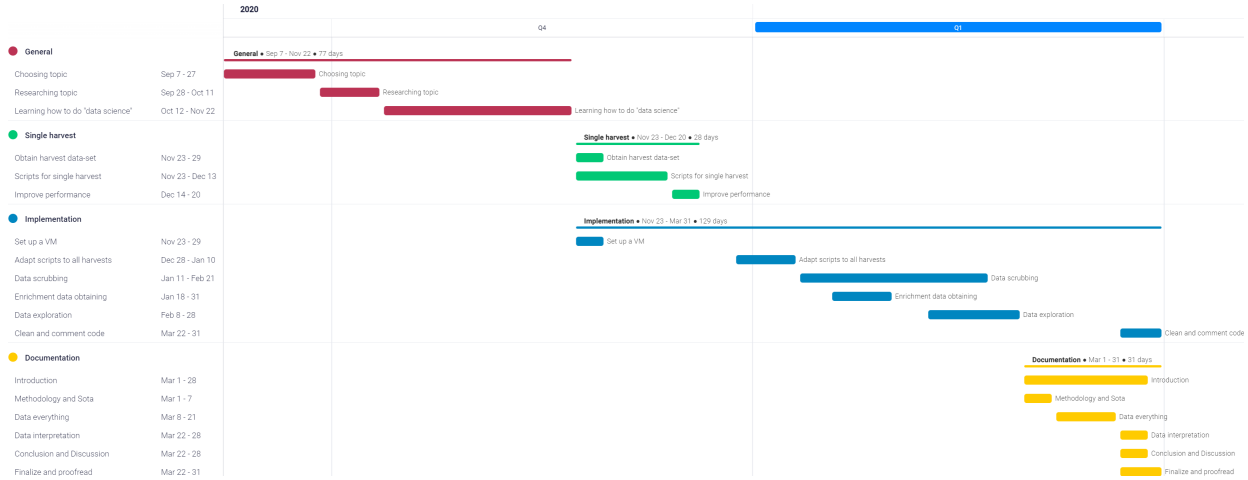

Figure 4: Latest version of the Gantt chart depicting the schedule of the whole thesis.

[4] `https://monday.com/`, accessed 06 March 2021

The above-mentioned framework and project schedule tracking method allow us to structure and plan work in a systematic way while keeping track of tasks' fulfillment and making sure to not fall behind.

3 State of the art

This chapter begins by investigating studies of other researchers to see what results they achieve, how they design their studies, and learn from their findings to make this analysis better. Then all sources of enrichment data together with reasoning of why we chose the given resource are explained. The chapter is closed with a brief explanation of all programming languages, libraries, and tools used for scripting, data processing, and visualisation.

3.1 Related works

The presence, involvement, and development of TPs and trackers on the web have attracted many researchers who dedicated their studies to this [1, 5–7, 13, 32–38].

In 2016 a study analysing the top one million Alexa sites was published by Englehardt et al. In total over 81 000 TPs distributed over a long-tail are identified.[32] Only 123 of them are present on more than 1% of first-parties. On top of that, *"Google, Facebook, Twitter, and AdNexus are the only third-party entities present on more than 10% of sites"*[32]. Cookie syncing is also very common where 157 out of 200, and 460 out of 1 000 top trackers utilised this. They also introduce OpenWPM tool which we have also used to crawl the web (covered in Section 3.2.1 and 4.1). In [39], they show how tracking techniques can be used to surveil users on the web. Since 56% of trackers transfer some personal information unencrypted, this can be used to perform targeted attacks on a user, even localise them, or use a browsing history to discredit an individual.

The study done by Sørensen and Kosta, which this thesis is partially based on, is a longitudinal study analysing the presence of TPs on the web before and after GDPR.[6] They crawled the same list of 12 778 sites in the period of February 2018 to September 2018 totaling 21 harvests. Visited sites are categorised to 11 categories and most have the origin in EU/EEA region with some extra non-European sites (e.g. US, Ukraine) to see if the GDPR also had an effect outside the EU. On average 1 431 526 HTTP responses per harvest are recorded and total number of 3 128 unique TPs are identified. They find that even though the number of TPs decrease slightly over time, it cannot be attributed specifically to GDPR since no strong evidence is found.

Helles et al.[7] analyse the top 150 sites (total 2150 sub-pages) in 28 EU countries from the Alexa database. These sites were crawled in June 2018. In total they identify 9 077 TPs which produce long-tail with the biggest companies present on most sites: Google - 89%, Facebook - 48%, Adform - 24%, AppNexus - 23%, Gemius - 19%, and Amazon - 18%. The results are thus similar to findings in [32]. Further, they find a pattern of regionalisation within the EU which they believe is due to regulations, language, and traditions in online businesses.

In [1], Urban et al. collects data about 10 000 sites and analyses relationships among them. The concept of *Third-party trees* is introduced in the study, which *"reflects an approximation of the loading dependencies of all third parties embedded into a given website"*[1]. An interesting finding is that up to eight successive requests to various services can be made by including a single TP. The number and destination of subsequent requests are non-

deterministic which may present legal problems and pose a security risk since site owners do not have a reach over what is loaded in the end on the site and where their users' data may end up. From all analysed sites only 7% of all sites do not include a TP and landing pages of sites contain 45% fewer cookies compared to sub-sites.

In a different study [13], Roesner et. al. investigate the presence of TP trackers on the web. They analyse the top 500 and 500 less popular domains according to Alexa ranking as of September 2011. A wide variety of trackers is recorded with several having a presence across many sites. In total over 500 unique trackers are identified with some being able to track more than 20% of one's online behavior by estimation. According to them, the presence of all trackers, except social media widgets, can be hindered by forcing popup and TP cookie blocking as well as the Do Not Track header. To change that, the ShareMeNot, Firefox browser extension is introduced which significantly reduces the tracking ability of social media trackers. They also introduce a new tracking taxonomy to better differentiate between different tracker types.

Others, such as Merzdovnik et. al. focus on the effectiveness of various blockers of TP trackers. They analyse the performance of these blockers on more than 100 000 websites to see which of them perform the best under which conditions. Among findings are the fact that the learning-based tools perform worse than rule-based ones and that worryingly, *"over 60% of tracking information is exchanged over unencrypted HTTP connections"* [36].

Third-parties and trackers have been present on the Internet for a long time. In [33], Lerner et al. analyse trackers from 1996 to 2016. The number of TPs grows over time and as of 2016, it was at an all-time high. The top 5 and top 20 trackers are present on 65% and 70% of all visited sites respectively. The power of individual trackers has grown over time and individual companies know more about ones browsing history than they did ever before. Therefore decisions made by individual companies' can have a big effect to users' privacy. Further, they show that choices of web browser manufacturers impact the way how trackers behave, as in 2004 when the internet explorer started with pop-up blocking by default and the number of forced trackers dropped significantly.

Similarly, Wambach et. al. studied the evolution of TP web tracking.[38] They found that web tracking has grown significantly over the years, where the top three trackers' presence grew from 10% in 2005 to 73% in 2015 on analysed sites. On top of that, one company accomplished the coverage of 80% of all analysed sites.

Some studies research the effects of web tracking directly on real users like the one by Iordanou et. al., where 350 real users used their browser extension over the period of more than four months.[35] On top of that, they generalise the results and use data of 60 millions mobile users from four European internet service providers (ISPs) finding that around 90% of trackers are hosted within the EU and around 3% of the traffic are personal data according to GDPR.

3.2 Data obtaining

This section covers all tools and sources used in the process of obtaining the data needed to answer the problem this thesis is focusing on.

3.2.1 OpenWPM

OpenWPM[5] is an open-source framework introduced in [32], built on top of the Firefox browser, and now maintained by the Mozilla Foundation. Its intended use is the measurement of web privacy, hence the abbreviation WPM - web privacy measurement in the name. *"Crawling with a real browser is important for two reasons: (1) it's less likely to be detected as a bot, meaning we're less likely to receive different treatment from a normal user, and (2) a real browser supports all the modern web features"*[40]. It is flexible, scalable and allows crawling millions of websites easily. It supports stateful measurements, the use of extensions, simulation of browsing history, proxies, and observation recording.

3.2.2 Categorisation data

They are needed for categorisation of all TPs to know their purpose, and subsequent analysis based on these categories. There are many APIs providing categorisation services but after a short analysis based on a random sample of 30 TPs, and comparing results from multiple APIs, the Webshrinker was chosen for the best performance to price ratio, as well as the simplicity.

Websrinker It[6] is a paid API that provides a categorisation of URLs, IP addresses, and websites. It returns JSON object with categories based on IAB content taxonomy standard[7]. It is claimed that results are always up-to-date by continuously crawling the web, and advanced machine learning is utilised to provide categorisation.

3.2.3 Malicious data

As part of the analysis, it is looked into whether or not the given TP may be a threat for a user. Similar to categorisation, there are plenty of services, therefore a small research is conducted to find which are the most popular ones.

Data on whether a website may be a threat to a user can be obtained from various public or private blacklists where each may have a different focus e.g. malicious, phishing, bots, etc. In [41] and [42] they analyse the list of 22 different public blacklists looking into the performance and uniqueness of sites contained in the blacklists respectively. In [43], Thao et al. analyse the list of 14 popular private and public blacklists and found that Google Safe Browsing (GSB) v.4 performs well and provides much better results in detecting younger domains. GSB is also used by the biggest browsers on the market (by share) to check for potential threats on a page. VirusTotal is a popular service to check if a web page may pose a threat to a user as described by Peng et al. in [44], listing dozens of works that have used this service.

This thesis's focus is not to find the best way of discovering that a given web-site can be a threat, but it is a small part of it. Because of that, we decide not to use any

[5] `https://github.com/mozilla/OpenWPM`, accessed 04 March 2021

[6] `https://www.webshrinker.com/apis/`, accessed 05 March 2021

[7] `https://iabtechlab.com/standards/content-taxonomy/`, accessed 05 March 2021

public blacklists since it would require a lot of time to access, process, and analyse data. Rather we chose GSB because it is widely used by browsers and offers good performance, and VirusTotal, because it is trusted by many researchers to obtain the data.

Google Safe Browsing GSB[8] is used by Chrome, Safari, Firefox, and Chromium [45], as mentioned previously, which totals almost 89% market share of desktop and 91% of mobile users as of October 2020.[9] This service has been provided by Google since 2007. Its latest version – 4 is used to obtain data, specifically Update API (v4). The API is available for free and is designed to be privacy-friendly [46].

VirusTotal It[10] aggregates results from various scanning engines and antivirus products[47] to scan files and URLs for viruses. It launched in 2004 and provides a paid service. It is very popular among researchers, as mentioned in the previous section. However, Peng et al. found in their study[44] that *"if a researcher only scans a URL once and queries the database afterward, she cannot get the updated labels"* and therefore should make another query to trigger an update of the database. Also, the results from VirusTotal and vendors directly were not always consistent. The following API endpoint is used to obtain the data[11].

3.2.4 Registrant data

Knowing who the owner of a first-party, and all associated third-parties are, is useful for finding out how many TP responses are actually *false* since they are made within the organization and do not have to be considered as TP responses. For finding the information associated with the ownership of a website, the WHOIS[12] protocol is used.

WHOIS WHOIS is a protocol defined in RFC 3912[48]. It is a query and response protocol that provides information services by querying registrar databases storing information about registrant of a resource, e.g IP address or domain name, and delivering the information in a format understandable to humans. A python library called python-whois[13] is used for making queries.

3.3 Data scrubbing and exploring

This section covers all programming tools used in the process of working with data - cleaning, processing, enrichment, and analysis.

[8] `https://developers.google.com/safe-browsing/v4`, accessed 05 March 2021
[9] `https://netmarketshare.com/browser-market-share.aspx`, accessed 05 March 2021
[10] `https://www.virustotal.com/gui/,accessed05March2021`
[11] `https://developers.virustotal.com/v3.0/reference#url-info`, accessed 07 March 2021
[12] `https://tools.ietf.org/html/rfc3912`, accessed 05 March 2021
[13] `https://pypi.org/project/python-whois/`, accessed 06 March 2021

3.3.1 Jupyter notebook

Jupyter notebook is an open-sourced web-based environment by non-profit organization Project Jupyter[14]. It is very popular in the fields of machine learning and data science. It is made up of three elements:

(i) notebook web application - allows interactive code editing, writing, and running;
(ii) kernel - execution environment which runs code of every notebook separately and returns results to the notebook;
(iii) notebook documents - is *"a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots, and rich media, usually ending with the ".ipynb" extension"*[49].

3.3.2 Python

It first appeared 30 years ago, but today's Python[15] can be described as a *"general-purpose programming language that blends procedural, functional, and object-oriented paradigms"*[50]. It is very popular in the field of data science, used regularly by 87% of data scientists due to relatively short learning curve, a vast selection of libraries, and flexibility.[51]

On top of that, many popular libraries exist to ease the work of data scientists, such as Pandas[16], Scipy[17], or Numpy[18].

Pandas It is an open-source Python library initially released in 2008. Pandas is primarily used for data manipulation (such as cleaning, reshaping, merging, etc.) and analysis of its key data-structures: (i) *Series* - one-dimensional labeled array, and (ii) *DataFrame* - two-dimensional labeled data structure which can hold different data types across columns.

Numpy It is an open-source Python library initially released in 2005. Numpy provides a vast selection of math functions that are used on multi-dimensional array objects. It is often a library of choice for large arrays as it is much faster than Python lists with its *ndarray* at the core. To list a few differences between Python lists and Numpy arrays: the data type of all elements in the Numpy array must be the same and the Numpy array's size is fixed since the creation.

Plotly It is an open-source plotting library[19] initially released in 2013. Plotly allows creation of over 40 types interactive plots which are highly customizable. It offers web-based visualisation, offline mode and display of all plots in the Jupyter notebook. At its core it

[14] `https://jupyter.org/`, accessed: 03 March 2021
[15] `https://www.python.org/`, accessed 03 March 2021
[16] `https://pandas.pydata.org/`, accessed 04 March 2021
[17] `https://www.scipy.org/`, accessed 04 March 2021
[18] `https://numpy.org/`, accessed 04 March 2021
[19] `https://plotly.com/python/`, accessed 04 March 2021

has the Plotly JavaScript library on which it is built and support multiple languages. On top of that, it integrates well with Dash to build pure Python web applications.

The presented knowledge and data sources, along with technological overview provide an outline of what will be used in the process of working with data, which will be documented in the following chapters.

4 Data obtaining, scrubbing, exploring and visualisation

The technologies and methods described previously are used in this chapter to obtain and process the data to the extent that they can be used to provide answers to the problem formulation. This chapter is the most extensive and probably the most important one, since it spans first three OSEMN processes and covers the whole journey from obtaining the data to having results that need interpretation in the next chapter. To break down the structure, all sections are created according to the OSEMN framework, with the addition of the data structure explanation:

1. Data obtaining
2. Data structure
3. Data scrubbing
4. Data exploration

Every section closely interprets all phases that are part of it, what are the inputs, outcomes and steps taken to accomplish the results. All scripts covered in this chapter can be found in the respective folder of the code base attached, or in the GitHub repository[20].

4.1 Data obtaining

The first step, based on OSEMN framework (Section 2.1), is the process of obtaining data-sets. In this case, two groups of data-sets can be identified: (i) harvested data - provided data-set accessible to AAU students containing all information recorded during harvests, and (ii) enrichment data - data which will enrich the harvested data to deliver more detailed analysis on TPs.

4.1.1 Obtain harvest data

This data-set includes requests, responses, cookies and other information recorded while accessing all first-parties from the list of sites to visit. Obtaining this data set is part of *Obtain harvest data* phase, described in paragraph *Application of the OSEMN framework* of Section 2.1. As already mentioned, this work builds on research by Sørensen and Kosta and uses extended data-set from [6] that is available to AAU students. The following section refers to their work.

In section 3.1 the authors explain the *"Selection and classification of websites"*. Based on previous works, they decide to focus on sites with a low or high expected presence of TPs. In total, 11 site categories are established, with news and weather category distinguishing between publicly and privately owned ones: (i) Entertainment, (ii) Government, (iii) Legal Services, (iv) News Private, (v) News Public, (vi) Postal Services, (vii) Public Transport, (viii) Shopping and Travel, (ix) University, (x) Weather Private, and (xi) Weather Public.

Their objective is also to have at least 5 sites for categories ii), iii), iv), and ix). Altogether, from 1 363 websites, 12 778 sub-sites are visited and all requests and responses

[20] `https://github.com/miso581/Third-party-presence-analysis`, accessed 30 March 2021

are recorded. Exact number of sites per country and category can be found in Figure 1 of [6].

The harvesting setup is described in section 3.2 *"Data collection and cleaning"*. A VM deployed within the EU with Ubuntu, 5 CPUs, and 8GB of RAM running four vanilla Firefox browsers and OpenWPM tool for data harvest is used. Cookies were not deleted after harvests and cookie consent was never given.

The extended data-set we are provided consists of 9 harvests before GDPR and 52 harvests after, with data collected over the period of almost 2,5 years between February 2018 and June 2020.

4.1.2 Obtain enrichment data

To carry out a more detailed analysis, harvested data needs to be augmented with additional – 'enrichment' data. The process of obtaining these data is part of the third phase - *Obtain and scrub enrichment data*, mentioned in paragraph *Application of the OSEMN framework* of Section 2.1. They can be put into three categories: (i) categorisation, (ii) malicious, and (iii) registrant data.

Categorisation data consists of two data-sets which have been obtained differently. First-party categorisation data-set is created by manually finding and entering the information about each FP to the table. It is provided as part of data-set from AAU. In contrast, third-party categorisation data-set is created by using an API, as explained in the next paragraph.

The process of obtaining the data for enrichment is similar across all categories, except the already mentioned first-party categorisation data-set:

1. *Choose a data source* - reasoning on which data source is chosen, and why, is covered in Section 3.2.
2. *Read the documentation* - to successfully obtain the data from a data source, documentation on how each API can be queried or library implemented is read. Also, in case of an API, obtain an API key for authentication.
3. *Write a script* - since Jupyter notebook and Python are used for programming, see Section 3.3 for more information, separate scripts implementing each API or library, according to the respective documentation, are written. All scripts have the list of unique TPs at the input and have the following structure:
   - 3.1 *Iterate through all TPs*
   - 3.2 *Do a query* - a query to obtain information about each TP is made individually.
   - 3.3 *Process the returned data* - data are returned as a JSON object. Depending on the export format, they are either directly appended to the list of JSON objects, or processed by only taking some values needed for further analysis, and appended as a list to the list of lists.
   - 3.4 *Export the processed data* - the lists are exported in multiple formats: (i) `JSON` file - the list of JSON objects is used as the source, (ii) `CSV` or `TXT` files - the processed list of lists is used as the source.

The point *3.3 Process the returned data* is part of the scrubbing process of the OSEMN framework and is therefore explained in detail in the respective section - 4.3.2, where each enrichment category has its own subsection.

Table 1 contains an overview of which data source returns data in what format, what are the exported data formats, and under what name can the script be found in the code base. All scripts are contained in the *"2) OBTAIN data for enrichment"* folder of the attached code base or on Github [52].

Data source	***Returned data format***	***Exported data format***	***Script name***
Webshrinker	JSON object	CSV, TXT, JSON	CATEGORIZATION - WebShrinker.ipynb
Google Safe Browsing	JSON object	CSV, TXT	MALICIOUS - GSB.ipynb
VirusTotal	JSON object	CSV, TXT, JSON	MALICIOUS - virusTotal.ipynb
WHOIS	JSON object	CSV, TXT	REGISTRANT - WHOIS.ipynb

Table 1: Overview of returned and exported data formats of each data source along with a name of every corresponding script. All scripts are located in the "2) OBTAIN data for enrichment" folder which can be found on GitHub [52] or in the code base attached.

After this process, all necessary data are collected and are ready to be scrubbed and explored in the following phases. But before that, their structure is described.

4.2 Data structure

Before proceeding to the next process of the OSEMN framework, it is vital to describe the structure of all obtained data-sets. Knowing each data-set's structure and content enable us to do a faster and better scrubbing, therefore in this section data structure is examined and explained.

4.2.1 Harvested data-sets

Over the harvesting period of 2,5 years, 61 harvests are made. Each harvest contains an SQLite database file and a harvest log. The SQLite file incorporates 12 tables in total: `CrawlHistory`, `crawl`, `flash_cookies`, `http_redirects`, `http_requests`, `http_responses`, `javascript`, `localStorage`, `profile_cookies`, `site_visits`, `task`, and `xpath`. A quick examination of tables uncovers that some tables contain no data at all. The purpose and sctructure of each table is not covered, since only `http_responses` and `site_visits` tables are used further. However, OpenWPM GitHub repository[21] contains the schema documentation for most of the tables.

[21] `https://github.com/mozilla/OpenWPM/blob/master/docs/Schema-Documentation.md`, accessed 06 March 2021

HTTP responses This table consists of all responses recorded during a single harvest. It has the following columns: `id`, `crawl_id`, `visit_id`, `url`, `method`, `referrer`, `response_status`, `response_status_text`, `is_cached`, `headers`, `channel_id`, `location`, `time_stamp`, and `content_hash`. The exact schema of this table can be found in the OpenWPM GitHub repository[22].

From all of the listed columns, only the following three are used further in the analysis:

(i) `visit_id` - ID of the first-party originally used for the request.
(ii) `url` - site from which the response is received.
(iii) `response_status` - HTTP response status code of the response.

No specific cleaning is performed on this table because values in any of the columns cannot be null, except the `contant_hash` column, so there should not be any missing values. Also, none of the selected columns seems to have corrupted values. Therefore, the columns are only extracted and processed, and no further cleaning operations are performed. The extraction and processing operations on this table are covered in Section 4.3.1. A sample of `http_responses` loaded as a dataframe (DF) with the three used columns is in Figure 5.

	visit_id	url	response_status
237061	2030	https://securepubads.g.doubleclick.net/pagead/...	200
417531	3493	https://www.eigenhuisentuin.nl/images/icons/se...	200
434307	3643	https://securepubads.g.doubleclick.net/gpt/pub...	200
683308	5800	https://fonts.gstatic.com/s/lato/v14/S6u9w4BMU...	200
716360	6080	http://pagead2.googlesyndication.com/pagead/sh...	200

Figure 5: A sample from `HTTP_responses` table loaded as a dataframe showing all columns used further in the analysis.

Site visited This table consists of all sites visited during a single harvest. It has the following columns: `visit_id`, `crawl_id`, and `site_url`. The exact schema of this table can be found in the OpenWPM GitHub repository[23].

In the further analysis, only two columns are used:

(i) `visit_id` - unique ID of the first-party to be crawled.
(ii) `site_url` - URL of the FP to be crawled.

The `crawl_id` is not used since it only contains information about which of the four browsers is used for crawling the given FP. Similarly to the `HTTP_responses` table, columns cannot be `null` and no inconsistencies or corrupted values are detected. The `site_url` value is based on the list of FPs to be harvested, which does not change during the harvesting period,

[22] `https://github.com/mozilla/OpenWPM/blob/master/docs/Schema-Documentation.md#http_responses`, accessed 06 March 2021

[23] `https://github.com/mozilla/OpenWPM/blob/master/docs/Schema-Documentation.md#site_visits`, accessed 06 March 2021

neither is the list order changed, and therefore, the `visit_id` and `site_url` associations are consistent across all harvests. Because of that, no scrubbing is needed on this table. The Figure 6 depicts a sample of `site_visits` table loaded as a DF and the operations performed with this table are covered in Section 4.3.1.

	visit_id	crawl_id	site_url
1252	1253	2	http://www.rts.ch/video/plus7/series/inspecteu...
5914	5915	1	https://www.cuni.cz/#event-1-2
5920	5921	3	http://aci.md/our-expertise/
6293	6294	4	http://www.delfi.lv
6327	6328	1	http://www.ilaw.legal/Investicijos#Privataus-i...

Figure 6: A sample from `site_visits` table loaded as a dataframe showing all columns.

4.2.2 Site categories

There are two data-sets that serve the purpose of categorisation, the first-party categories provided by AAU, and the third-party categories which are obtained using Webshrinker's API.

First-party categories Categorisation data for first-parties are delivered in a `CSV` file. It contains all FPs (1 363 in total) with the following information about them: `Found`, `Country`, `Europe`, `PublicPrivate`, `SiteCategory`, `URLtype`, `AdvertisingSeen`, `TopLevelDomainLookUp`, `Note`, `Visited`, and `typeOfPage`.

For further analysis, only selected columns are used:

(i) `Country` - the country of origin,
(ii) `Europe` - if a country is in EU, EEA, or outside,
(iii) `PublicPrivate` - if owned by a private or public entity,
(iv) `SiteCategory` - high-level category,
(v) `URLtype` - low-level category,
(vi) `TopLevelDomainLookUp` - root domain of the FP.

The selected six columns and a sample data of this data-set are illustrated in Figure 7.

Third-party categories Data for TP categorisation are obtained via Webshrinker API. The purpose of this data is to categorise all unique TPs so that the analysis can be performed on TPs per category as well. The data are returned as a JSON object which contains: `id` - ID of the IAB category – based on the standard, `label` - textual representation of the IAB ID, `parent` - parent's category ID, `score` - how certain the Webshrinker is about a label, and `confident` - confidence verdict, as illustrated in the JSON object snippet in Listing 1. The returned object can have multiple tiers of categories as the full returned object in Appendix A - Listing 6 shows.

	TopLevelDomainLookUp	Country	Europe	PublicPrivate	SiteCategory	URLtype
1096	sme.sk	Slovakia	EU	Private	News	PrivateMedia EU
862	dps.me	Montenegro	NonEU	Public	Education	University
246	aktualne.cz	Czech Republic	EU	Private	News	PrivateMedia EU
167	lasexta.com	Canada	NotEurope	Private	News	NewsUser
1040	stirileprotv.ro	Romania	EU	Private	News	PrivateMedia EU

Figure 7: A sample from firs-party categories loaded as a dataframe showing all six used columns.

```
...
    {"categories":
        [{"id": "IAB5",
        "label": "Education",
        "parent": "IAB5",
        "score": "0.119095739636984957",
        "confident": false }],
    "url":"https://www.aau.dk/"}
...
```

Listing 1: Returned JSON object snippet containing an example of Tier-1 categorisation from Webshrinker.

As already mentioned in Section 4.1.2, the script exports the data in `JSON`, `CSV`, and `TXT` file formats. The `TXT` file is not used in any step as it is identical to `CSV` which contains only the highest level category details of each TP. On the other hand, the `JSON` file comprises all information received about all TPs.

4.2.3 Malicious sites

Part of the TP analysis is to determine if any of TPs are malicious by nature. The data obtained from GSB and VirusTotal are used for this purpose.

Google Safe Browsing API Data returned from the API are in the JSON object which has a different structure for malicious and non-malicious sites:

(i) *Non-malicious site* - contains only the TP *url* and `False` value for the `malicious` key. Listing 2 is an example of such an object for the *aau.dk* site.

(ii) *Malicious site* - besides the information contained in the non-malicious site object, it also specifies the `platform` at which the threat is present, `threats` types, and the `cache` value. Listing 3 is an example of such an object for Google's testing site.

```
{"aau.dk": {"malicious": False}}
```

Listing 2: Example of returned JSON object from GSB for a site which is non-malicious.

```
{"http://malware.testing.google.test/testing/malware/":
    {"malicious": True,
    "platforms": ["ANY_PLATFORM"],
    "threats": ["MALWARE"],
    "cache": "300s"}}
```

Listing 3: Example of returned JSON object from GSB for a site which is malicious. It also carries the information about platforms on which the threat is present and type of threat.

The script used for obtaining the data, as already mentioned in Section 4.1.2, exported the data in `CSV` and `TXT` file formats. Both files contain the very same data, therefore the `TXT` file is not further used in any step. The header of the `CSV` file consists of `TP URL`, `malicious`, `platforms`, and `threats`. In the case of non-malicious TP, the values for `platforms` and `threats` are set to `nan` and the `cache` value is not exported in case of malicious TP.

Virus total API This API returns loads of information about each site in a JSON object[24]. For the analysis, however, only some information is needed, specifically `last_analysis_stats` containing `harmless`, `malicious`, `suspicious`, `timeout`, and `undetected` keys which values describe the number of scanning engines and antiviruses that report that the given TP belongs to the respective category, and `last_http_response_code` describing the response code of the TP.

The data are exported by the script for further use to `CSV`, `TXT`, and `JSON` files (see Section 4.1.2), where the `JSON` file contains all data contained in the returned JSON object, while the `CSV` file only contains `TP URL`, `harmless`, `malicious`, `suspicious`, `timeout`, `undetected`, and `last_http_response_code`. Both, the `CSV` and `TXT` files have identical content, therefore the `TXT` file is not used further.

4.2.4 Registrant information

Section 4.1.2 covers how and from which source the registrant data are obtained. These data are then exported to `CSV` and `TXT` file formats that have identical content. For that reason, the `TXT` file is not further used in the process.

The data returned by the WHOIS query are in the JSON object, see an example in Listing 4. As it can be observed, there are multiple *None* values in the object which is due to the chosen library and GDPR. This issue is scrutinized in Section 4.3.2 in paragraph WHOIS.

4.3 Data scrubbing

Once all data-sets essential to the analysis are obtained, and their structure understood, they can be further processed. Based on the OSEMN framework, the next process is scrubbing. During this process the obtained data are cleaned, enriched, transformed, extracted,

[24] `https://developers.virustotal.com/v3.0/reference#url-object`, accessed 07 March 2021

```
{"domain_name": "aau.dk",
 "creation_date": datetime.datetime(1997, 10, 31, 0, 0),
 "expiration_date": datetime.datetime(2023, 12, 31, 0, 0),
 "dnssec": "Unsigned delegation, no records",
 "status": "Active",
 "registrant_handle": None,
 "registrant_name": None,
 "registrant_address": None,
 "registrant_zip_code": None,
 "registrant_city": None,
 "registrant_country": None,
 "name_servers": ["auaw.aua.auc.dk", "noc.aua.auc.dk"]}
```

Listing 4: Example of returned JSON object for a aau.dk from WHOIS.

combined, calculations performed, and in the end exported. In total four phases are part of this process:

1. Process and generate unique third-parties for all responses
2. Scrub enrichment data
3. Enrich all responses with enrichment data
4. Process and generate data for visualisation

In the following sections, all phases are clarified along with scripts created to perform tasks and their input and output data described.

4.3.1 Process and generate unique third-parties for all responses

This is the second phase in the process, as described in paragraph *Application of the OSEMN framework* in Section 2.1. In this phase, the data from the provided harvest data-set are extracted and processed for use in the following phases. This is done in five steps:

1. Extract data
2. Find root domain of all FPs and response domains for every harvest's responses
3. Enrich response data with FP categorisation data
4. Produce unique TPs for every harvest
5. Obtain a list of globally unique TPs from all harvests

Each of the above mentioned steps has its own script performing the step. Scripts can be found in the *1) GENERATE list of unique TPs & process RES* folder of the attached code base or GitHub repository [52]. They are explained from the high-level in the following paragraphs.

Extract data The purpose of this step is to make data easily readable and accessible from a folder. Therefore, they first need to be extracted from the archives. The script is simple and has the following characteristics:

- *script name*: `1 - Extract files.ipynb`
- *input*: `.txz` archive file of each harvest
- *output*: separate folder with content of the archive for each harvest
- *script steps*:
    1. Import libraries
    2. Iterate through all archives and extract them to separate folders at a specified location

Find root domain of all FPs and response domains for every harvest's responses Each extracted harvest folder contains an `.sqlite` database file and a harvest log. Only the database file is used further. As explained in Section 4.2.1, it has 12 tables from which `http_responses` and `site_visits` are loaded to produce the processed responses file with root-domains of responses and FPs for each harvest.

- *script name*: `2 - Exporting CSVs of processed DFs and TPs.ipynb`
- *input*: `.sqlite` database file of each harvest
- *output*: enriched `http_responses` with a root domain of each visited site and response site, and columns indicating if response is FP-FP or FP-TP – `CSV` format
- *script steps*:
    1. Import libraries
    2. Iterate through extracted folders of all harvests:
        (a) From an `.sqlite` database file load `http_responses` and `site_visits` tables as separate Dataframes (DFs)
        (b) Merge responses and visited sites into a single DF based on `visit_id` column of both DFs
        (c) In the merged DF find a Root domain (RD) of every originally requested site – FP (`site_url` column) and all response sites (`url` column) and assign root domains to new columns - `RD_site_url`, `RD_url` respectively
        (d) Compare the `RD_url` and `RD_site_url` columns to find if the response RD matches the originally requested FP RD. If so, assign `True` to the new DF column – `first_party`, else assign `False`
        (e) Export `http_responses` enriched with a root domain of each visited site and response site, and columns indicating if response is FP-FP or FP-TP – `CSV` format

The step 2 c) of the script steps above, finds the root domain of `site_url` and `url` columns of merged DF. The function doing this, takes the DF and a column name as parameters and returns the list of RDs of given columns. Inside of the function it first checks if it is possible to obtain an RD from a URL and if so, appends it to the list. If not, it checks if the given URL is an IP address and if not, appends the `url` value to RD list, if yes it performs a reverse DNS lookup to find a root domain associated with the IP. If that is not possible, it appends the IP address to the list of root domains.

Enrich response data with FP categorisation data In this step, all previously processed `http_responses` tables are further enriched with the first-party categorisation data. This enables the filtration of all responses by country, category, etc. as well as production of unique TPs for responses to FP request from EU/EEA originated sites.

- *script name*: `3 - Enrich each RES with FP categories.ipynb`
- *input*: `.csv` files of processed `http_responses` tables and first-party categorisation data (`.csv`)
- *output*: every harvest's `http_responses` data enriched with first-party categorisation data (`.csv`)
- *script steps*:
  1. Import libraries
  2. Load first-party categorisation data
  3. Iterate through all processed `http_responses` tables:
     (a) Load `http_responses` as a DF
     (b) Create a new dataframe by merging first-party categorisation data with responses' DF on the corresponding FP root-domain column
     (c) Export as the enriched responses dataframe as a `CSV` file

Produce unique TPs for every harvest Each enriched `http_responses` table contains tens of thousands of responses from thousands of servers. In order to carry out the analysis on TPs, they first need to be extracted from the responses. Since the focus of the thesis is the EU/EEA area, only TP responses of request from visited sites with origins in this area are considered to produce the list of unique TPs.

- *script name*: `4 - Generate unique TPs per harvest for EU sites.ipynb`
- *input*: `.csv` files of enriched `http_responses` tables
- *output*: list of unique TPs for each harvest's responses - `CSV` format
- *script steps*:
  1. Import libraries
  2. Iterate through all enriched `http_responses` tables:
     (a) Load `http_responses` as a DF
     (b) Filter out all responses from non EU/EEA FP requests
     (c) Filter out all FP-to-FP communication
     (d) Obtain a DF of unique TPs by filtering out all duplicates
     (e) Export the DF with unique TPs per harvest in a `CSV` format

Obtain a list of globally unique TPs from all harvests There is a need to know unique third-parties from all harvests' responses in order to be able to obtain enrichment data for TPs, know how many unique TPs are present in total, and to aid analysis of TPs later in the process. The script in this step produces such a list:

- *script name*: `5 - Obtaining global list of unique RES TPs EU ONLY.ipynb`

- *input*: list of unique TPs for each harvest responses - `CSV` format
- *output*: list of unique TPs for all harvests responses - `CSV` format
- *script steps*:
    1. Import libraries
    2. Iterate through all files with unique TPs for a harvest:
        (a) Append TPs to the list
    3. Create a dataframe of globally unique TPs from the list
    4. Export: unique TPs of all harvest responses - `CSV` format

Phase summary After this second phase, all harvest response data are processed, enriched and transformed for use in the subsequent phases. The responses of each harvest are enriched with root domains of response sites and originally requested sites, classification of whether a response is coming from a TP or FP domain, and with first-party categorisation data. Also, a list of globally unique TPs (of EU/EEA origin requested sites) for all harvest responses is created. Table 2 provides an overview of all steps, corresponding script name, input and output data.

Step	***Script name***	***Data input***	***Data output***
Extract data	1 - Extract files	`.txz` archive file of each harvest	separate folder with content of the archive for each harvest
Find root domain of all FPs and response domains for every harvest's responses	2 - Exporting CSVs of processed DFs and TPs	`.sqlite` database file of each harvest	`http_responses` enriched with a root domain of each visited site and response site, and columns indicating if response is FP-FP or FP-TP communication (`.csv`)
Enrich response data with FP categorisation data	3 - Enrich each RES with FP categories	`.csv` files of processed `http_responses` tables and first-party categorisation	every harvest's `http_responses` data enriched with first-party categorisation data (`.csv`)
Produce unique TPs for every harvest	4 - Generate unique TPs per harvest for EU sites	`.csv` files of enriched `http_responses` tables	each harvests' list of unique TPs (`.csv`)
Obtain a list of globally unique TPs from all harvests	5 - Obtaining global list of unique RES TPs EU ONLY	list of unique TPs for each harvest responses (`.csv`)	globally unique list of TPs of all harvest responses (`.csv`)

Table 2: Overview of steps, respective script name, data used as input to the script and script output data. All scripts are located in the "1) GENERATE list of unique TPs & process RES" folder which can be found on GitHub [52] or in the code base attached.

4.3.2 Scrub enrichment data

The phase three - *Obtain and scrub enrichment data*, spans two OSEMN processes. The Section 4.1.2 examines how enrichment data are obtained and describes the scripts jointly from a high-level. Every script not only obtains the respective enrichment data, but also scrubs and formats them - point *3.3 - Process the returned data* of the script explanation of the mentioned section. Since this is part of scrubbing process, this section further explains how respective enrichment data are processed and scrubbed before being exported.

Site categories

Third-party categories The data from Webshrinker API are returned as the nested JSON object and processed depending on the export format. For the `.json` export, the returned object is assigned to the dictionary of returned objects in case the response is successful. If the returned object contains an error, a predefined data are assigned and in the end the dictionary is exported. In case of `.csv` export, the nested JSON object is flattened and data are appended to the list. Errors are processed similarly to the `.json` export, by appending a predefined data to the list. The exported file is not further scrubbed in this step.

First-party categories This data-set is scrubbed manually. Firstly the unique values of `Country`, `Europe`, `PublicPrivate`, `SiteCategory`, and `URLtype` columns are scrutinized. Multiple inconsistencies in labeling, due to typos, are discovered and corrected. Each column is also verified to be completely populated and no missing data are found within columns. However, the total number of all FP contained in the file when obtained is 1 329, therefore 34 first-party entries are missing completely and are manually added.

Malicious site

Google Safe Browsing Malicious data obtained from GSB API are also returned as a JSON object. When the JSON object is received, it is checked whether a TP is malicious or not. If the TP is not malicious, the object only contains the `False` value in `malicious` key and nothing else, see Listing 2. Since it is desirable to have all exported data in the same format, the remaining keys are defined as `nan` values and all information extends the list with responses. If the TP is malicious, all information are obtained from the object and the list is extended.

No cleaning of data is carried out since all sites were labeled as not malicious which means that all remaining keys' values are defined as `nan`. The script was also tested with the testing malicious site from Google to dismiss any doubts about the script functionality, as shown in Listing 3, to make sure that it is not labeling all responses as non-malicious.

VirusTotal The response from VirusTotal API contains lots of information out of which only some attributes are interesting for the further analysis, see paragraph *Virus total API* in Section 4.2.3. The returned JSON object is checked for whether all keys, which contain values necessary for later analysis, exist in the object and only then are assigned to a variable. It happens sometimes that not all keys are present in the response, therefore a `nan` value is assigned instead, or its response is an error that is caught in a try block and all variables are then assigned `nan` values. Afterward, the variables are appended to the list which is exported after the obtaining information about all TPs. Data are not further cleaned after the export in this phase.

Registrant information

WHOIS Registrant information is obtained using WHOIS protocol implemented by a `python-whois` library. As already mentioned in Section 4.2.4 and shown in Listing 4 many fields of the returned JSON object contains `None` values. This issue was found during the cleaning of data-set.

When choosing a library to implement the WHOIS query, the `python-whois` seemed like a good choice. It performed well and it was easy to implement. However, once the data were obtained and random results checked against an online WHOIS lookup tool[25], the results differentiated. Taking an example from Listing 4, the online tool also returned the registrant's name, full address, and phone number. Therefore, there is a chance that some empty or `None` values recorded are because of the library.

The bigger problem though is the GDPR. When cleaning the data-set, many TPs' WHOIS information contained "REDACTED FOR PRIVACY" or similar. Because of that, the research on why it is so, is conducted.

In May 2018, ICANN published the *Temporary Specification for gTLD Registration Data* which contains guidelines on how and which data should be kept publicly available by WHOIS to assure compliance.[53] Its Appendix A, Section 2. contains the *"Requirements for Processing Personal Data in Public RDDS Where Processing is Subject to the GDPR"*. Based on that, most fields must be treated as redacted unless permission to publish the information is given.

Some countries, on the other hand, have a legal basis for the publication of some data. In Denmark, for example, the administrator for danish domain names *"will continue to publish the name, address, and telephone number of registrants"*[54] because of the *Danish Act on Internet Domains*[55].

Research by Lu et. al published this year (2021) discovered that a GDPR has a significant impact on WHOIS *with over 85% of surveyed large WHOIS providers redacting EEA records at scale* and *over 60% large WHOIS data providers also redact non-EEA records*[56].

This clearly shows that the data obtained by this method are missing most of the information needed for the intended analysis. Because of that, the links between FPs, TPs, and their owners will not be analysed and this data-set is not investigated further.

[25] `https://whois.domaintools.com/`, accessed 07 March 2021

This phase spans *Obtain* and *Scrub* processes, however this section focuses only on the scrubbing. After this whole phase however, all enrichment data are obtained, processed, and ready to be used for analysis and enrichment of other data.

4.3.3 Enrich all responses with enrichment data

In this phase – number four, all previously processed `http_responses` tables are further enriched with the categorisation and malicious information. That makes it possible to omit the import of enrichment data and merging them with responses data every time they are being processed in phase five to generate data suitable for visualisation. The script is located in *3) ENRICH data-sets with categories, malicious, location, etc.* folder of the code base or GitHub repository [52].

- *script name*: `ENRICH each RES with categories and malicious data.ipynb`
- *input*: (i) `.csv` files with previously enriched `http_responses` tables, (ii) third-party categorisation data (`.json`), (iii) GSB and (iv) VirusTotal malicious data (`.csv`)
- *output*: every harvest's `http_responses` data enriched with categories and malicious information in separate `.csv`
- *script steps*:
  1. Import libraries
  2. Load third-party categorisation data, GSB and VirusTotal malicious data as separate dataframes
  3. Iterate through files with all enriched `http_responses`:
     (a) Load `http_responses` as a DF
     (b) Create a new dataframe by merging third-party categorisation, GSB and VirusTotal malicious data with responses DF on the corresponding TP root-domain column
     (c) Export the enriched responses dataframe as a `CSV` file

When all responses data were enriched, it was discovered that the size of two exported files is significantly smaller than of the rest - the average 1,6GB vs less than 1GB. These files were investigated and it was found that these harvests are corrupted and not all sites were visited (9 317 and 3 353 out of 12 778). This was also confirmed by looking into the original, not enriched, data. Therefore, the harvests from 2019-08-20 and 2019-12-09 are discarded and no longer used for the analysis, reducing the total number of harvests from 61 to 59.

4.3.4 Process and generate data for visualisation

The purpose of this fifth phase is to manipulate, filter, transform, further clean the data, and export them to be used in the following *Explore and visualise data* phase. This offloads the otherwise needed pre-processing of data during the exploration and visualisation refraining from making scripts too complicated and data usable only for one purpose. On top of that, this allows the reuse of data between scripts since all data are always exported.

Totally, six scripts are produced in this phase, with scripts 1) and 2) created to produce more general data and no specific visualisation in mind, while scripts 3) through 6) are produced with a vision for a specific use.

Throughout this phase, multiple iteration between this and next, *Explore and visualise data*, phase are performed to either further process the data or produce a completely new one.

First batch of data for visualisation This is the first script created in this phase. It served the purpose of blindly generating different kinds of data which can be used as input for plots. Initially, it produced 8 different outputs, which were then used in the next phase to make basic plots as a starting point. Several iterations between this and the following phase were made and in the end, this script produced 15 different `.json` files. However, as the progress and more specific plots were made, the versatility of this script ceased and new, more specific ones were created.

This script still contains the code for producing all 15 data outputs; however, only 3 of them are used in the next phase and therefore the code for the remaining 12 is commented out and not contained in the explanation below.

- *script name*: `1) First batch of data for visualization.ipynb`
- *input*: all enriched responses `.csv` files
- *output*: (i) `TPs_per_site.json` - number of unique TPs per site (ii) `responses_total_combined.json` - total number of a) all responses, b) all responses for EU/EEA, c) all responses for EU/EEA with FP-TP communication, and (iii) `visited_sites_total_combined.json` - total number of visited sites per harvest a) with responses, b) from EU/EEA, c) with FP-TP communication from EU/EEA
- *script steps*:
  1. Import libraries
  2. Iterate through all enriched responses files:
     (a) Load a +response file as a dataframe
     (b) Filter out all responses to non EU/EEA FP requests
     (c) Filter out all FP-to-FP communication
     (d) Calculate the number of unique TPs per site and append the results as a `JSON` object to the corresponding list
     (e) Calculate the number of responses: (i) all, (ii) EU/EEA only, and (iii) EU/EEA with FP-TP communication, and append the results as a `JSON` object to the corresponding list
     (f) Calculate the number of visited sites: (i) with responses, (ii) with EU/EEA origin, and (iii) with with EU/EEA origin with FP-TP communication, and append the results as a `JSON` object to the corresponding list
  3. Export each corresponding list as a separate `.json` file: (i) `TPs_per_site`, (ii) `responses_total_combined`, and (iii) `TPs_total_combined`

Enrich visited sites The main purpose of this script is to produce a list of visited sites enriched with first-party categorisation data. The exported data are then used in some plots, and also as an input for other scripts. It is a simple script but it reduces the otherwise repetitive work of importing and merging data every-time such data are needed.

- *script name*: `2) Enrich visited sites.ipynb`
- *input*: (i) an `.sqlite` database file of a single harvest, and (ii) first-party categorisation data (`.csv`)
- *output*: `visitedSitesCat.csv` - list of visited sites enriched with first-party categorisation data
- *script steps*:
    1. Import libraries
    2. Load the `site_visits` table from the database file as a DF
    3. Load first-party categorisation data as a DF
    4. Obtain a root-domain of all visited sites and append it as a new column of visited sites DF
    5. Merge FP categorisation DF and visited sites DF based on respective first-party root domain column
    6. Export the merged dataframe as `visitedSitesCat.csv`

A sample from the dataframe used as source for the export of `visitedSitesCat.csv` file depicting all columns can be found in Figure 8.

visit_id	crawl_id	site_url	url_TLD	Country	Europe	PublicPrivate	SiteCategory	URLtype	TopLevelDomainLookUp
1273	3	http://www.infolex.lv/portal/start.asp?act=pam...	infolex.lv	Latvia	EU	Private	LegalService	LawFirm	infolex.lv
5139	1	https://www.stjornarradid.is/default.aspx?Page...	stjornarradid.is	Iceland	EEA	Public	Government	NaN	stjornarradid.is
9162	1	http://www.elgiganten.dk	elgiganten.dk	Denmark	EU	Private	Consumption	ShoppingUser	elgiganten.dk
9452	2	http://www.hotnews.ro	hotnews.ro	Romania	EU	Private	News	PrivateMedia EU	hotnews.ro
10285	1	https://makler.md/ru/household-products/stitch...	makler.md	Moldova	NonEU	Private	Consumption	Shopping	makler.md

Figure 8: Sample data from the dataframe of exported `visitedSitesCat.csv` file.

HTTP responses status codes One of the factors that can influence the presence of TPs over the time, is the 'health' status of visited sites during harvests. Interesting things to consider in the analysis and to find out are how many of the visited sites are still active at the end of the harvesting period compared to the beginning, how many URLs of visited sites still point to the same resource, or what is the general trend over the harvests. This script uses the `response_status` column from every harvest's enriched response file to acquire the number of HTTP response status codes per code category, per harvest.

- *script name*: `3) HTTP responses status codes.ipynb`
- *input*: all enriched responses `.csv` files

- *output*: `FP_status_codes_EU.json` - number of visited sites per each HTTP response status code per harvest
- *script steps*:
    1. Import libraries
    2. Iterate through all enriched response files:
        (a) Load response file as a dataframe
        (b) Filter out all responses from non EU/EEA FP requests
        (c) Obtain the number of visited sites per HTTP response status code based on initiating request of every visited site
        (d) Add the obtained data to the dictionary
    3. Export the dictionary containing the number of visited sites per each HTTP status code response category per harvest - `FP_status_codes_EU.json`

Presence of TPs over harvests The general presence of TPs over harvests is an interesting topic to investigade. It enables finding TPs that are most prevalent at visited sites. The script produces a dataframe which has as its index all visited sites, and all TPs as columns and contains only `nan` or `1` values in cells, where `1` indicates that the given TP is present at least once on given visited sites throughout the harvesting period. This dataframe is then exported in a CSV format.

- *script name*: `4) Presence of TPs over harvests.ipynb`
- *input*: (i) all enriched responses `.csv` files, and (ii) `EU-UNIQUE-RES-TPs.csv` - list of unique TPs
- *output*: `EU_TP_total_occurence.csv` - presence of each unique TP at every visited site over harvesting period
- *script steps*:
    1. Import libraries
    2. Load all unique TPs as a DF
    3. Create an empty dataframe (`df_distribution`) with visited site IDs as index (1 – 12 778) and unique TPs as columns
    4. Iterate through all enriched responses:
        (a) Load response file as a DF
        (b) Filter out all responses to non EU/EEA origin visited sites requests
        (c) Filter out all FP-to-FP communication
        (d) Group the response DF by visited site IDs (`visit_ID`) and TP root domains (`RD_url`), get the size, and create a new DF out of it
        (e) Iterate through the grouped DF: (i) insert `1` to the `df_distribution` at position `[visit_ID, RD_url]` every time the given TP appears in the given visited site responses
    5. Export the `df_distribution` containing the occurrence of each unique TP at every visited site, during all harvests - `EU_TP_total_occurence.csv`

Counted presence of TPs over harvests Similarly to the analysis topic mentioned in the previous paragraph, it is also desirable to know how many times each third-party appeared at the given visited site in total across harvests. It is achievable then to, for example, find the relationships between each visited site category, TP category and the total number of TPs per given categories.

The script is almost the same as the `Presence of TPs over harvests.ipynb`. The only differences are the initiating values of the `df_distribution` being `0`, instead of `nan` and the step 4 e), where this script instead of inserting `1` to the cell, it adds up 1 to the current value of the cell.

- *script name*: `5) Counted presence of TPs over harvests.ipynb`
- *input*: (i) all enriched responses `.csv` files, and (ii) `EU-UNIQUE-RES-TPs.csv` - list of unique TPs
- *output*: `EU_TP_total_occurence_count.csv` - counted presence of each unique TP at every visited site over harvests

TPs categories data EU One of the analyses to be performed is distribution of TPs among categories. As already mentioned in Section 3.2.2, third-party categorisation data are obtained from Webshrinker API which provides 26 Tier-1 categories and hundreds of Tier-2 categories for a site categorisation based on IAB's *Content Taxonomy*. The problem with so many category options is that there is a handful of categories with a big representation of TPs and then a long tail of categories with only a few TPs. With the total of 6 868 TPs recorded and using the Tier-2 category where existing and Tier-1 where not, 228 different categories of TPs are present. The percentage of TPs represented in top categories is as follows:

- top 20 categories - 80%
- top 30 categories - 85%
- top 40 categories - 89%
- top 50 categories - 91.5%

This confirms the initial claim about the long tail, where top 20 categories contain 80% of TPs. This would make the analysis more difficult and if for example only top 20 categories were chosen for further analysis, 20% of TPs would be left out. Because of that the categories are further processed and unified.

All Tier-2 categories containing more than 1.5% of TPs (103+) are kept and the remaining categories are merged to their parent Tier-1 category. This way the size of categories is reduced to 33, which significantly ease the analysis of data and changes the distribution of TPs in top categories:

- top 10 categories - 78%
- top 15 categories - 88%
- top 20 categories - 93.9%

- *script name*: `6) TPs categories data EU.ipynb`

- *input*: (i) `EU-UNIQUE-RES-TPs.csv` - list of globally unique TPs (ii) third-party categorisation data
- *output*: (i) `EU_TPs_categorization_processed.csv` - all TPs enriched with processed categorisation, (ii) `EU_TPs_categorization_processed_TOP_15.csv` - TPs from top 15 TP categories enriched with processed categorisation, and (iii) `EU_TPs_categories_size.csv` - number of TPs per TP category
- *script steps*:
  1. Import libraries
  2. Load file with all unique TPs as a DF
  3. Load third-party categorisation data as a dictionary
  4. Define a TP category unification dictionary
  5. Define Tier-2 category IDs with more than 1.5% of TPs
  6. Iterate over the categorisation dictionary and for each TP:
     (a) Check if the TP has a Tier-2 category: (i) if `True` - use Tier-2 category for categorisation, except for TPs with category `id == IAB25-WS1` (ii) else use Tier-1 category
     (b) Create a dictionary with data from given Tier, and append it to the category data list
  7. Create a DF from category data list
  8. Merge TPs DF with categorisation DF on respective TP root domain column and create a new DF
  9. Create a new empty column (`new_cat`) in the merged DF
  10. Iterate through all TPs in the merged DF and check their value in `cat_id` column:
      (a) If a `400` value is present: (i) assign an `uncategorized_IP` value to the `new_cat` column
      (b) Else if a value is in the list of top Tier-2 categories: (i) assign current category to the `new_cat` column
      (c) Else: (i) assign a value based on the unification dictionary to the `new_cat` column
  11. Iterate through all TPs in the merged DF and check if the TP RD contains an IP address:
      (a) if `True`: (i) assign a value based on the uncategorised sites dictionary to the `new_cat` column
  12. Create a new DF containing the number of TPs in each category (based on `new_cat` column)
  13. Export: (i) the whole merged DF as `EU_TPs_categorization_processed.csv`, (ii) merged DF with TPs of the top 15 most populous categories as `EU_TPs_categorization_processed_TOP_15.csv`, and (iii) the DF containing the number of TPs per TP category as `EU_TPs_categories_size.csv`

The distribution of TPs across categories can be found in Figure 9 which depicts a dataframe with all categories and TPs count ranked.

	new_cat	count		new_cat	count		new_cat	count		new_cat	count
1	Uncategorized	1059	9	Under Construction	240	17	Education	88	25	Health & Fitness	44
2	Content Server	1038	10	Business	219	18	Non-Standard Content	74	26	Careers	42
3	Technology & Computing	871	11	Hobbies & Interests	176	19	Personal Finance	70	27	Home & Garden	41
4	Advertising	634	12	Arts & Entertainment	163	20	Uncategorized_IP	60	28	Automotive	40
5	Marketing	461	13	Travel	123	21	Style & Fashion	54	29	Science	21
6	News	318	14	Shopping	120	22	Food & Drink	53	30	Illegal Content	18
7	Cloud storage and hosting	276	15	Society	94	23	Sports	52	31	Family & Parenting	14
8	Adult Content	262	16	Law, Gov't & Politics	89	24	Real Estate	51	32	Pets	2
									33	Religion & Spirituality	1

Figure 9: The whole dataframe containing ranked list of all categories and the number of TPs per category.

Phase summary In this fifth phase - *Process and generate data for visualisation* data are filtrated, combined from multiple sources, further cleaned, and manipulated to export different sets of data which can be used in the following phase for creation of plots. In total, six scripts are created with scripts 1) and 2) producing general data for visualisation and scripts 3) - 6) producing data for creation of specific plots. This way, the plots can be created just by loading the data without the need to combine multiple data sources or cleaning.

The Table 3 provides an overview of all scripts (name of the step is the same as script name), data used as an input and produced output data. Every script can be found in the code base or in the GitHub repository [52] in the *"4) GENERATE data for analysis"* folder.

4.4 Data exploration

The third process of the OSEMN framework is the exploration. The data used in this process are already cleaned extracted, combined, and reduced from the previous one, so here the data are mostly just loaded to to produce visualisations, such as plots, which can then be interpreted as results, since the *model* process is not employed in this thesis.

Only one phase is identified in this process during the work - Section 4.4.1 *Explore and visualise the data* and it will be explained in the rest of this section.

4.4.1 Explore and visualise the data

In this sixth phase, the data are being further explored and visualised to understand them better and see what value they can bring. In total 10 scripts are created which produce 11 diagrams.

These scripts could be grouped by the amount of steps needed to further prepare data for a specific diagram as: (i) load & plot, and (ii) load, calculate or transform, then plot.

All scripts contained in *load & plot* group, follow almost identical steps: load data, set up the diagram parameters, and create the diagram. No extra processing of loaded data is needed, since the diagram depicts more general information and the fact that the data were processed in the previous phase. The scripts included in this group are: (i) `1)`

Script name	*Data input*	*Data output*
1) First batch of data for visualization.ipynb	all enriched responses `.csv` files	(i) `TPs_per_site.json` - number of unique TPs per site, (ii) `responses_total_combined.json` - number of responses per different criteria, (iii) `TPs_total_combined.json` - total number of visited sites per harvest per different criteria
2) Enrich visited sites.ipynb	(i) an `.sqlite` database file of a single harvest, and (ii) first-party categorisation data (`.csv`)	`visitedSitesCat.csv` - enriched list of visited sites
3) HTTP responses status codes.ipynb	all enriched responses `.csv` files	`FP_status_codes_EU.json` - number of FPs per status code
4) Presence of TPs over harvests.ipynb	(i) all enriched responses `.csv` files, and (ii) `EU-UNIQUE-RES-TPs.csv` - list of unique TPs	`EU_TP_total_occurence.csv` - occurrence of each TP at every visited site
5) Counted presence of TPs over harvests.ipynb	(i) all enriched responses `.csv` files, and (ii) `EU-UNIQUE-RES-TPs.csv` - list of unique TPs	*output*: `EU_TP_total_occurence_count.csv` - counted occurrence of each TP at every visited site
6) TPs categories data EU.ipynb	(i) `EU-UNIQUE-RES-TPs.csv` - list of globally unique TPs (ii) third-party categorisation data	(i) `EU_TPs_categorization_processed.csv` - TPs with processed categorisation, (ii) `EU_-TPs_categorization_processed_TOP_15.csv` - TPs from top 15 TP categories, and (iii) `EU_-TPs_categories_size.csv` - number of TPs per TP category

Table 3: Overview of script name (each step has the same name), data used as input to the script and script output data. All scripts are located in the "4) GENERATE data for analysis" folder which can be found on GitHub [52] or in the code base attached.

`Responses total, EU, EU TP.ipynb`, (ii) `2) Visited sited total, EU, EU TP.ipynb`, (iii) `4) EU TP responses in relation to visited sites.ipynb`, and (iv) `7) Presence of top 1% TPs.ipynb`.

The other group of scripts - *load, calculate or transform then plot*, contain all the remaining ones. These scripts, after loading the data, need to perform additional steps to either transform the data into the format accepted by the library (such as transposing of dataframes), delete not-needed columns, set different indexes, order data, or perform calculations (such as sums, or means over a rows or columns) to produce the needed data in an expected format.

This section will further look briefly into each of the scripts and describe its input data and output diagram, reference to where the diagram is located in the thesis, and the purpose of the script. Since steps in all scripts are similar, they will not be described in detail. All scripts are located in the *5) EXPLORE and VISUALIZE* folder of the attached code base or in the GitHub repository [52].

- `1) Responses total, EU, EU TP.ipynb`:
  - *input data*: `responses_total_combined.json` - contains the number of total, EU/EEA, and EU/EEA with TP communication responses per harvest
  - *output plot*: bar chart with harvest date on x-axis and number of responses on y-axis
  - *reference to figure*: Figure 10
  - *purpose*: Visualise the distribution of number of responses per harvest based on three criteria: (i) total number of responses, (ii) all responses to EU/EEA first-party requests, and (iii) only TP responses to EU/EEA first-party requests. This offers a more granular overview of responses, and comparison if e.g. the decrease in total number of responses influences also the EU/EEA or only non-EU, or if the decrease in number of TP responses to EU/EEA origin first-party requests influence the overall number of EU/EEA responses.

- `2) Visited sited total, EU, EU TP.ipynb`:
  - *input data*: `visited_sites_total_combined.json` - contains the number of total, EU/EEA, and EU/EEA with TP communication visited sites per harvest
  - *output plot*: bar chart with harvest date on x-axis and number of visited sites on y-axis
  - *reference to figure*: Figure 11
  - *purpose*: Visualise the number of visited sites per harvest based on three criteria: (i) total number of visited sited, (ii) all visited sites with EU/EEA origin, and (iii) only visited sites with EU/EEA origin with a TP response. The number of requested visited sites is consistent across the harvest - 12 778. However, the number of visited sites with at least one response fluctuates slightly over the time. This plot does not directly offer a reason on why is that happening but depicts the trend and opens a door for further analysis.

- `3) Development of TPs over harvest.ipynb`:
  - *input data*: all unique responses per harvest (`.csv`)
  - *output plot*: combined bar and line chart with number of unique TPs on y-axis and harvest date on x-axis
  - *reference to figure*: Figure 13
  - *purpose*: Visualise the development of the number of TPs over harvests including the TPs active in all previous harvests, and TPs not anymore active. This offers a view on how much the total number of TPs fluctuates and if it is mostly the same TPs present across harvests or they die out and new ones appear.

- `4) EU TP responses in relation to visited sites.ipynb`:
  - *input data*: (i) `responses_total_combined.json` - contains the number of total, EU/EEA, and EU/EEA with TP communication responses per harvest, and (ii) `visited_sites_total_combined.json` - contains the number of total, EU/EEA, and EU/EEA with TP communication visited sites per harvest

  - *output plot*: combined bar chart with number of responses on primary y-axis and line chart with the number of visited sites on secondary y-axis with harvest dates on x-axis
  - *reference to figure*: Figure 12
  - *purpose*: Visualise and put in a perspective if the amount of all responses to EU/EEA first-party requests is influenced by the number of all visited sites with EU/EEA origin, or by visited sites with EU/EEA origin with a TP response.

- `5) + 6) Status codes EU visit_sites.ipynb`:
  - *input data*: `FP_status_codes_EU.json` - HTTP response status codes of all first-parties with EU/EEA origin
  - *output plot*:
    1. stacked bar chart with number of visited sites per grouped HTTP response status code on y-axis and harvest date on x-axis
    2. stacked bar chart with number of visited sites per specific HTTP response status code on y-axis and harvest date on x-axis
  - *reference to figure*:
    1. Figure 15
    2. Figure 16
  - *purpose*: Visualise the 'health' of visited sites over the harvesting period. The harvesting period is almost 2.5 years and the same sites are being visited over and over, therefore, there is a high chance that some visited sites will become inactive or will redirect to another site.
    1. Investigate the grouped HTTP response codes.
    2. Investigate specific most populous HTTP response codes, besides `200 - OK`.

- `7) Presence of top 1% TPs.ipynb`:
  - *input data*: `EU_TP_total_occurence.csv` - presence of each TP on every visited site
  - *output plot*: bar chart with individual TPs on x-axis and percentage presence of given TP on all visited sites
  - *reference to figure*: Figure 14
  - *purpose*: Visualise the presence of top 1% most present TPs to see if the TPs are equally distributed or produce e.g. long tail and to compare the distribution with previous study.

- `8) Number of TPs per website category with respect to the harvest date.ipynb`:
  - *input data*: `TP_occurence_per_visit.json` - total presence of each TP on every visited site
  - *output plot*: box-and-whisker plot with FP harvest dates grouped by FP categories on x-axis and number of unique TPs on y-axis
  - *reference to figure*: Figure 18

- *purpose*: Visualise and compare the individual FP categories among each other in terms of the number of TPs, as well as compare the development of TPs within each category over the harvesting period.

- `9) Average number of TPs per category.ipynb`:
  - *input data*: `TP_occurence_per_visit.json` - total presence of each TP on every visited site
  - *output plot*: scatter plot with harvest dates on x-axis and average number of unique TPs by category on y-axis depicted with markers and by linear regression line
  - *reference to figure*: Figure 17
  - *purpose*: Visualise the average number of TPs for each FP category over the harvesting period as well as the linear regression line. This enables to better analyse the TPs trends over the time, more closely compare average number of TPs between categories, and see trends.

- `10) FP - TP categories relations.ipynb`:
  - *input data*: (i) `EU_TP_total_occurence_count.csv` - counted presence of each TP on every visited site and (ii) `visitedSitesCat.csv` - list of visited sites enriched with first-party categorisation data
  - *output plot*: heatmap with top 15 TP categories on x-axis and all visited sites' categories on y-axis
  - *reference to figure*: Figure 19
  - *purpose*: Visualise the presence of each TP category in each of the visited sites categories to find if any TP category is predominantly present on certain visited sites.

- `11) Relationship between manual and automatic categorization.ipynb`:
  - *input data*: (i) `EU_TPs_categorization_processed.csv` - all TPs enriched with processed categorisation and (ii) `TPidentitiesMay2019.csv` - all manually categorised TPs from [6]
  - *output plot*: heatmap with automatic categorisation categories on x-axis and manual categorisation categories on y-axis
  - *reference to figure*: Figure 20
  - *purpose*: Visualise the relationships between manual and automatic categorisation of TPs, to assess if how well does the automatic categorisation performs.

After this sixth phase, various types of plots, such as bar or scatter chart, box-and-whiskers plot, or heatmap, are produced from all previously processed data. These plots will be interpreted and used to draw a conclusion in the last process of OSEMN framework in the following chapters.

To summarise, this chapter contains all processes, phases, and steps to produce results which can be interpreted in the subsequent chapters. It begins with a description of how the

data are obtained, followed by an explanation of their structure, how they are cleaned and processed to produce statistics and visualisations which can be used to draw conclusions.

5 Results

Providing the results and their interpretation is part of the last process of the OSEMN framework as well as the last phase of the work carried out during the thesis, therefore the main findings are presented in this chapter. The data preparation and processing in the previous chapter allow us to study the results from multiple angles.

This chapter is divided into six sections, based on the area investigated, to provide the results in a more structured manner; beginning with comparison of pre- and post-GDPR number of responses and visited sites in each harvest. This is followed by the investigation of TP's presence development over harvests, and the health status of visited sites. Lastly, the average presence of TPs on visited site categories is investigated. The results from these sections support the creation of a comprehensive understanding of how and if the GDPR affects the TP presence. In the last two sections we analyse the properties of third-parties such as the category of TP or its maliciousness.

In each section, the given area is analysed, multiple diagrams are presented and assumptions are proposed. The colour of each metric across diagrams in a single section is unique to avoid any misunderstandings that can occur from using the same colour scheme for different metrics across diagrams. All assumptions and partial conclusions are combined to draw a conclusion and provide the answer to the problem formulation in the Section 7 - Conclusion.

As already mentioned in the previous chapters, the harvesting period is almost 2.5 years (February 2018 - June 2020), during which 61 harvests were made, 9 before the GDPR came into force and 52 after. During the enrichment of the harvested data-sets it was found that two after General Data Protection Regulation (GDPR) harvests were corrupted (9 317 and 3 353 sites were visited out of 12 778) and therefore, excluded from the further analysis, leaving 59 harvests in total.

Throughout the following sections, the focus is on 10 089 sites with EU/EEA origin, since the goal is to analyse changes in behaviour on sites of EU/EEA origin, as problem formulation states (Section 1.2), while some general statistics are also provided for all visited sites in the first two sections.

5.1 Responses and visited sites

A general overview of how many sites are visited during each harvest, and how many responses are returned to requests made, is crucial for understanding how the number of third-parties develop over time and are influenced by this.

On average 1 386 060 responses are recorded over all harvests. Figure 10 represents the development of responses with respect to the harvest date and origin of the requested first-party. Table 4 details the statistics on the average, min, and max number of responses per all harvests, before and after GDPR, and their change. Both will be now used to analyse the responses. Notice, that the plot is not a stacked bar chart and all bars starts from 0 on y-axis.

It can be observed from the diagram, that a month before the GDPR came into force, the number of responses suddenly rose slightly and just before and after the GDPR coming into force the number dropped. By looking at the TP responses during the same period, it seems that the rise and drop was caused by the presence of TPs in the responses. In general though, a trend seems to be that the number of responses decrease over time at all metrics similarly.

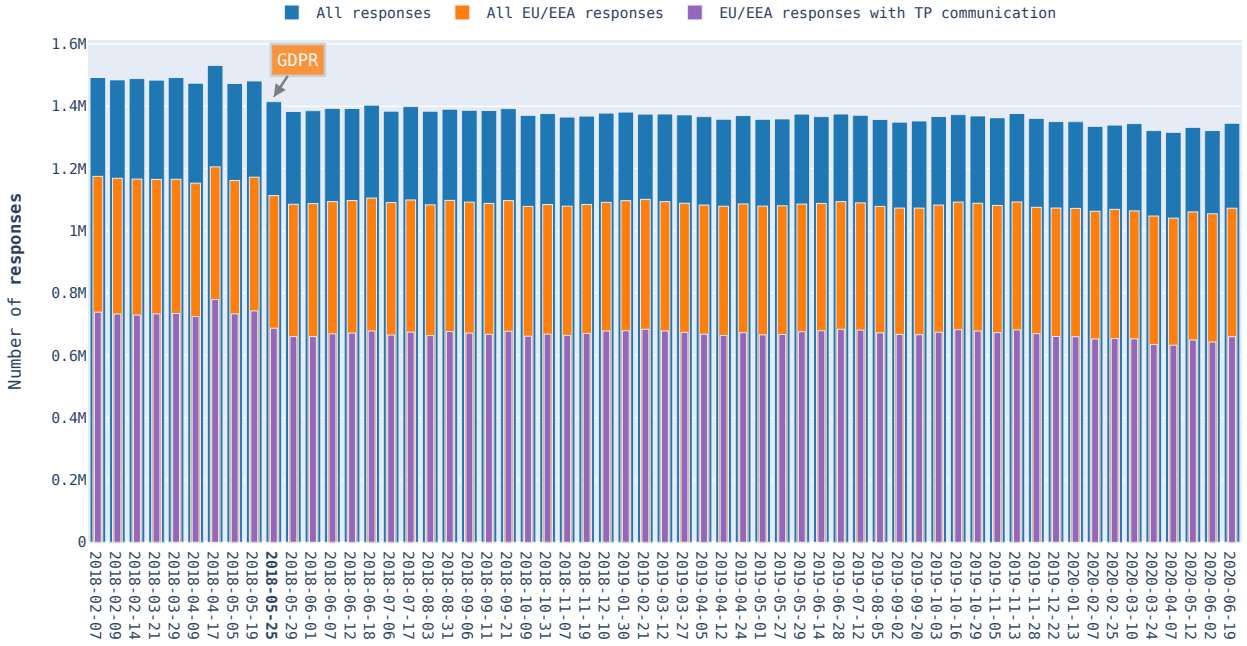

Figure 10: Number of total, EU/EEA origin requested sites, and EU/EEA origin requested sites with TP responses with respect to harvest date. Notice, that it is not a stacked bar chart and all bars starts from 0 on y-axis.

Having another look at the table (4), the number of responses after the GDPR is on average lower by 8.1% totally, and by 9.6% lower for TP responses compared to before. Comparing the after GDPR metrics with all harvests measurements, the numbers decreases approximately by 1.4% for all metrics. Therefore, if looking just at the number of TP responses over the harvesting period, we could say that the GDPR seems to have had an effect since there was an initial decrease after its commencement strengthened by a decreasing trend of TP responses.

As already mentioned, the number of visited sites is 12 778 each harvest. One would assume that as 12 778 sites are requested, the same number of sites would have at least one response, being, OK, Redirect, Not found, etc. Figure 11, however, shows that it is not the case and the number of sites that returns at least a single response to the request fluctuates over time and the maximum number of visited sites with a response is 12 733 and minimum 12 625, as Table 5 shows. Notice that the plot is not a stacked bar chart and all bars start from 9 000 on y-axis.

Responses per harvest	***Average***	***Min***	***Max***
All harvests			
Total	1 386 060	1 316 053	1 530 985
EU/EEA origin	1 096 155	1 040 497	1 205 310
EU/EEA origin with TP communication	678 908	632 643	779 121
Before GDPR			
Total	1 488 871	1 472 908	1 530 985
EU/EEA origin	1 170 141	1 152 924	1 205 310
EU/EEA origin with TP communication	738 763	724 834	779 121
After GDPR			
Total	1 367 554	1 316 053	1 414 568
EU/EEA origin	1 082 837	1 040 497	1 112 887
EU/EEA origin with TP communication	668 134	632 643	687 232
Change (after - before GDPR)			
Total	-121 317 / -8.1%	-156 855	-116 417
EU/EEA origin	- 87 304 / -7.5%	- 112 427	-92 423
EU/EEA origin with TP communication	-70 629 / -9.6%	-92 191	-91 889

Table 4: Statistics on the average, min, and max number of responses during all harvest, before GDPR, after GDPR and thier change.

Comparing Table 4 and Table 5 we can see that the number of responses after the GDPR decrease on average between 7.5-9.6% compared to the 0.33-1.04% decrease in visited sites. Even if calculating the decrease from total number of requested sites, the decrease is 0.64-0.79%. On top of that, after the GDPR, the number of visited sites with a TP response decrease by more than 1%, which is not much in fact, but it is 3-4x more than in the case of other metrics. This suggests, that the decreasing number of responses is not directly caused by the decreasing number of visited sites with a response. On top of that, the GDPR seems to indeed have an effect on the amount of visited sites with a TP response and responses in general.

This claim is further supported by the plot in Figure 12 which combines certain metrics from the previous two plots, namely EU/EEA: responses with a TP communication, visited sites with a TP response and all visited sites with given origin. Notice, that the bar chart uses primary (left) y-axis and line charts use secondary (right) y-axis and start from values 9 000. The development of the number of responses over the time does not mirror the number of visited sites with a TP response, which suggests that these metrics are not directly related and the number of TP responses is not significantly influenced by a decreasing number of visited sites with or without a TP response.

Coming back to the problem with a different number of requested sites and sites with

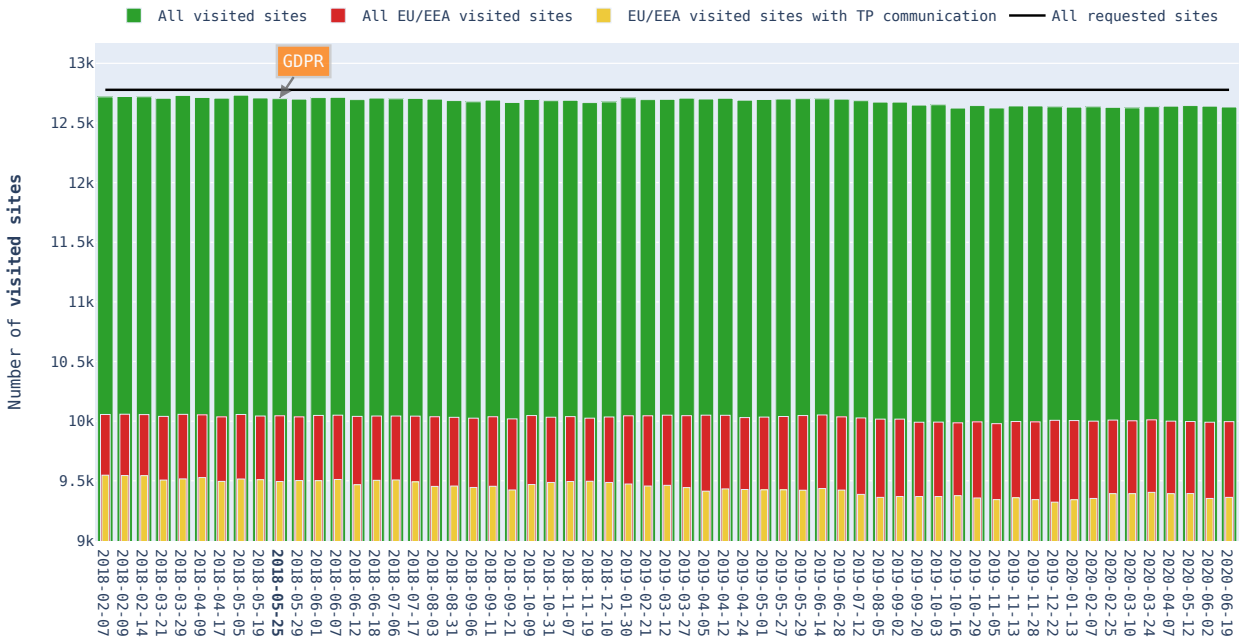

Figure 11: Number of total, EU/EEA origin, and EU/EEA origin with TP responses visited sites per harvest. Notice, that it is not a stacked bar chart and y-axis starts at value 9 000.

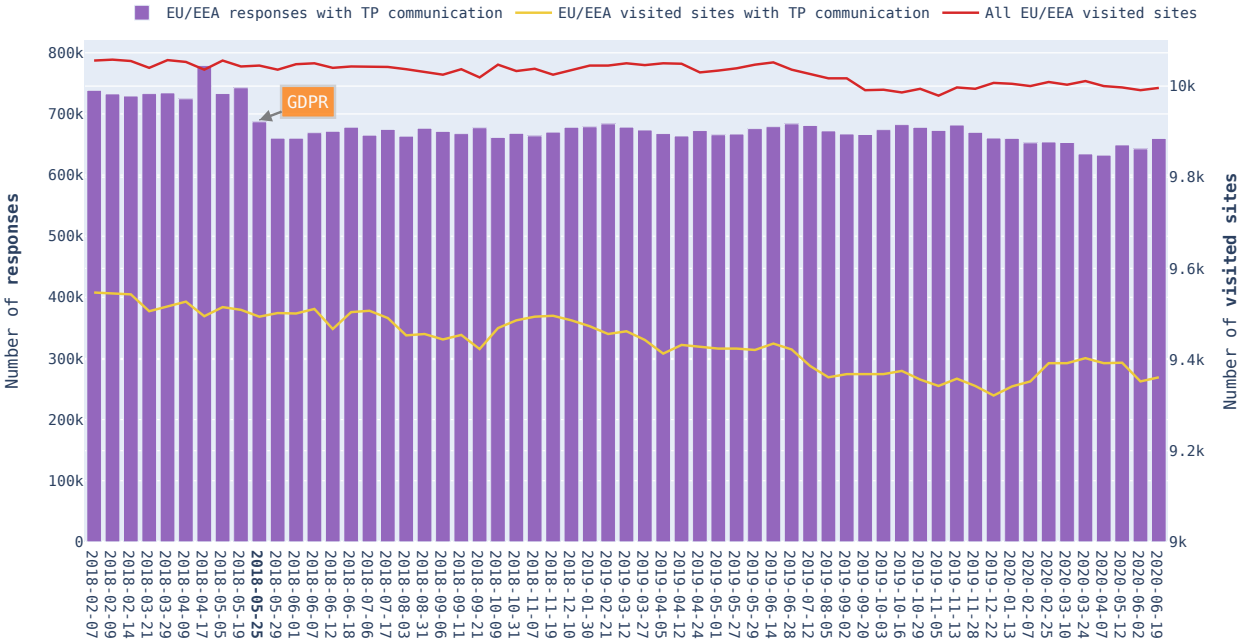

Figure 12: Number of responses in relation to visited sites with respect to harvest date. Notice, that the bar chart uses primary (left) y-axis and line charts use secondary (right) y-axis which starts at value 9 000.

Visited sites per harvest	***Average***	***Min***	***Max***
All harvests			
Total	12 682	12 625	12 733
EU/EEA origin	10 028	9 979	10 058
EU/EEA origin with TP communication	9 439	9 321	9 547
Before GDPR			
Total	12 718	12 707	12 733
EU/EEA origin	10 050	10 036	10 058
EU/EEA origin with TP communication	9 522	9 495	9 547
After GDPR			
Total	12 676	12 625	12 715
EU/EEA origin	10 024	9 979	10 052
EU/EEA origin with TP communication	9 423	9 321	9 511
Change (after - before GDPR)			
Total	-39 / -0.33%	-82	-18
EU/EEA origin	-26 / -0.26%	-57	-6
EU/EEA origin with TP communication	-99 / -1.04%	-147	-36

Table 5: Statistics on the average, min, and max number of visited sites with responses during all harvest, before GDPR, after GDPR and change.

responses. This was investigated further by looking into the logs of multiple harvests. It was found that the requested sites that do not return any response are in fact requested, but the browser used for crawling experiences an error: `Received failure status while executing command: GET`, as snippets in Listing 5 show. After the analysis of multiple log files and attempts to request sites with no response, it is observed that most of such sites are not active anymore, and probably were not active during the harvest either. This for some reason caused the browser to crash and not record any data. Therefore, visited sites with missing responses can be considered as not active anymore.

The preliminary results of this section suggest that the GDPR has a slight effect on the decreasing number of TP responses, since the number decreased by 9.6% compared to the pre-GDPR period and the number of visited sites with a TP communication decreased by 1%, which is 3-4x more compared to other metrics of visited sites. On top of that, there seems to be no direct link between the number of visited sites with a TP communication and total number of TP responses.

```
...
2020-01-13 10:34:51,522 - MainProcess[MainThread] - BrowserManager - INFO : BROWSER 2:
    EXECUTING COMMAND: ('GET', 'http://ass.tvnet.lv/lietotajs/cool-vecis', 15, 209)
2020-01-13 10:34:51,560 - MainProcess[MainThread] - FirefoxExtension - DEBUG : Visit Id: 209
2020-01-13 10:34:51,749 - MainProcess[Thread-210] - TaskManager - INFO : BROWSER 2:
    Received failure status while executing command: GET
2020-01-13 10:34:51,750 - MainProcess[Thread-210] - TaskManager - DEBUG : BROWSER 2: Browser
    restart required
...
2020-01-14 07:00:02,134 - MainProcess[Thread-12496] - BrowserManager - INFO : BROWSER 4:
    EXECUTING COMMAND: ('GET', 'http://joq.al/artikull/315986.html', 15, 12586)
2020-01-14 07:00:02,169 - MainProcess[MainThread] - FirefoxExtension - DEBUG : Visit Id: 12586
2020-01-14 07:00:02,296 - MainProcess[Thread-12587] - TaskManager - INFO : BROWSER 4:
    Received failure status while executing command: GET
2020-01-14 07:00:02,296 - MainProcess[Thread-12587] - TaskManager - DEBUG : BROWSER 4:
    Browser restart required
...
2020-01-14 06:53:04,275 - MainProcess[Thread-12178] - BrowserManager - INFO : BROWSER 1:
    EXECUTING COMMAND: ('GET', 'http://www.da.nametests.com', 15, 12509)
2020-01-14 06:53:04,310 - MainProcess[MainThread] - FirefoxExtension - DEBUG : Visit Id: 12509
2020-01-14 06:53:04,490 - MainProcess[Thread-12510] - TaskManager - INFO : BROWSER 1:
    Received failure status while executing command: GET
2020-01-14 06:53:04,491 - MainProcess[Thread-12510] - TaskManager - DEBUG : BROWSER 1:
    Browser restart required
...
```

Listing 5: Three snippets from the log file of the harvest from 2020-01-13. These snippets shows the error messages for the visited sites that returned no response to the request. This pattern is repeating throughout the harvests. It seems, that when a browser accounts a failure on response, it does not record anything to the ***http_responses*** *table of the harvest. Because of that, the number of visited sites fluctuates over the time.*

5.2 Third-parties

One of the aims of this thesis is to look into third-parties, their presence, development, categories, or maliciousness. In this section we will look into the development of TP presence over the harvesting period, as well as the omnipresence of the top TPs.

There exists many interesting questions in regards to TPs that one can come up with, try to analyse, and provide the answer to. To begin with, the development of TPs is depicted by Figure 13. The presence of TPs seems to have a slightly decreasing trend without any substantial spikes or drops over the time. On average, 2 900 unique TPs are present in each harvest, as Table 6 shows. The assumption about the decreasing trend is also supported by a comparison of pre- and post-GDPR averages: 3 056 vs. 2 872 TPs which is a decrease of 6%. It can also be observed that at harvests just around the date of GDPR coming to force (harvests 05-19, 05-25, 06-01, and 06-07 of 2018) the number of TPs drops in total by 184. In addition the total number of TPs present at a single harvest never exceeds 3 000 after GDPR, compared to the before period where it was constantly above. Generally though,

there are no signs of any significant increase or decrease in the presence of third-parties.

Taking a look from another angle, 6 868 unique TPs are present in total throughout the harvests from which only 1 265 are present at least once in every single harvest. This indicates that TPs are quite volatile and often appear only a handful of times before becoming inactive. One of the reasons might be the presence of advertising TPs which only serve the ad during a short period of a campaign or show various ads. This trend of volatility, however, diminishes a bit after the GDPR when the decreasing trend of the TPs appearing at all previous harvests is slightly flattened.

Based purely on the data contained in the figure and table, some premature conclusions can be made that the GDPR seemingly has influence on the number, and volatility, of TPs presented over the harvested period. Since it came into force, the number of TPs decreased on average by 6% and the TPs are less volatile.

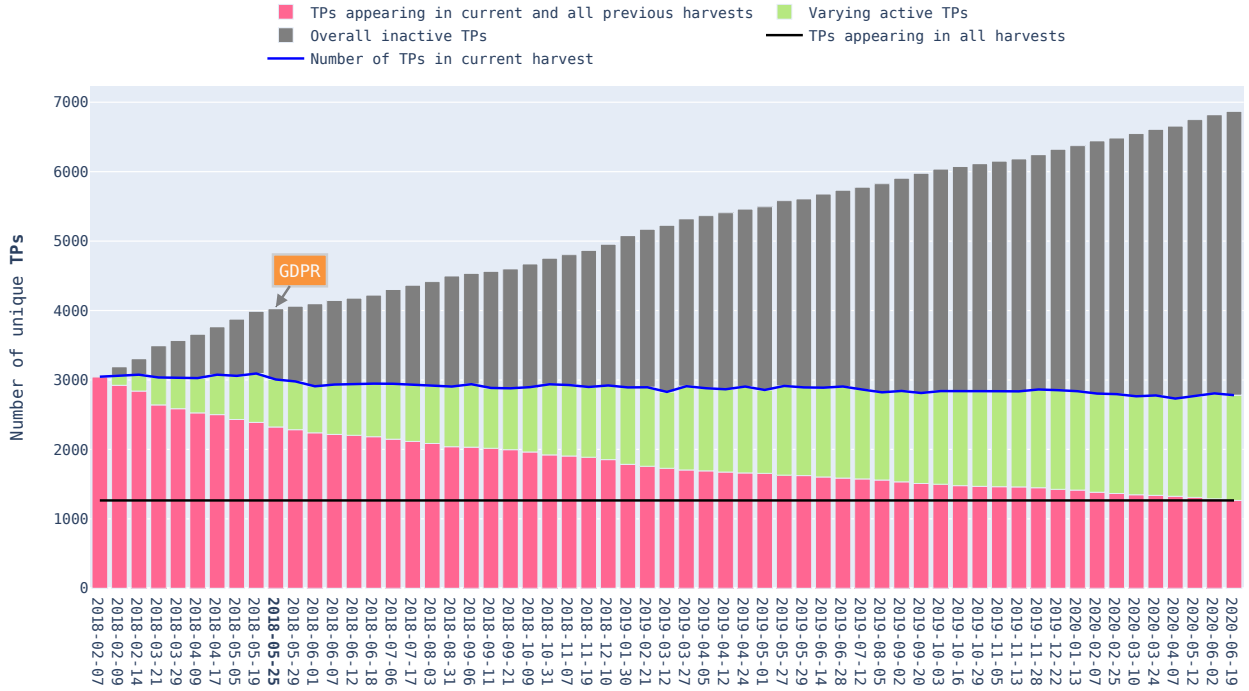

Figure 13: Development of third-parties over harvests. The diagram depicts the overall development of third-parties along with total number of inactive TPs and TPs present on every harvest.

Furthermore, we can look at the distribution of the most common third-parties. Out of 12 778 visited sites, 10 089 are of EU/EEA origin, meaning that a TP must be present on more than 100 visited sites to be included in Figure 14. It depicts all TPs present on more than 1% of visited sites. This totals 545 TPs out of 6 868 which produces a long tail with only a handful of TPs present on most of visited sites.

These results can be compared with a previous study which used the same data-set containing the first 21 harvests (compared to the extended data-set, used in this work, with 59 harvests) - Figure 2 of [6]. Both plots have a similar distribution of TPs producing a

Third-parties per harvest	*Average*	*Min*	*Max*
All harvests	2 900	2 732	3 093
Before GDPR	3 056	3 027	3 093
After GDPR	2 872	2 732	3 007
Change (after - before GDPR)	-184 / -6%	-295	-86

Table 6: Statistics on the average, min, and max number of third-parties during all harvest, before GDPR, after GDPR and change

long tail. However, the study found only 3 128 TPs with 181 present at more than 1% of all visited sites, compared with 6 868 and 545 TPs respectively in this analysis. This may imply and partially confirms the previous claim, that the TPs are less volatile post-GDPR since the number of TPs present on more than 1% of visited sites more than triples, while the total number of TPs only doubles.

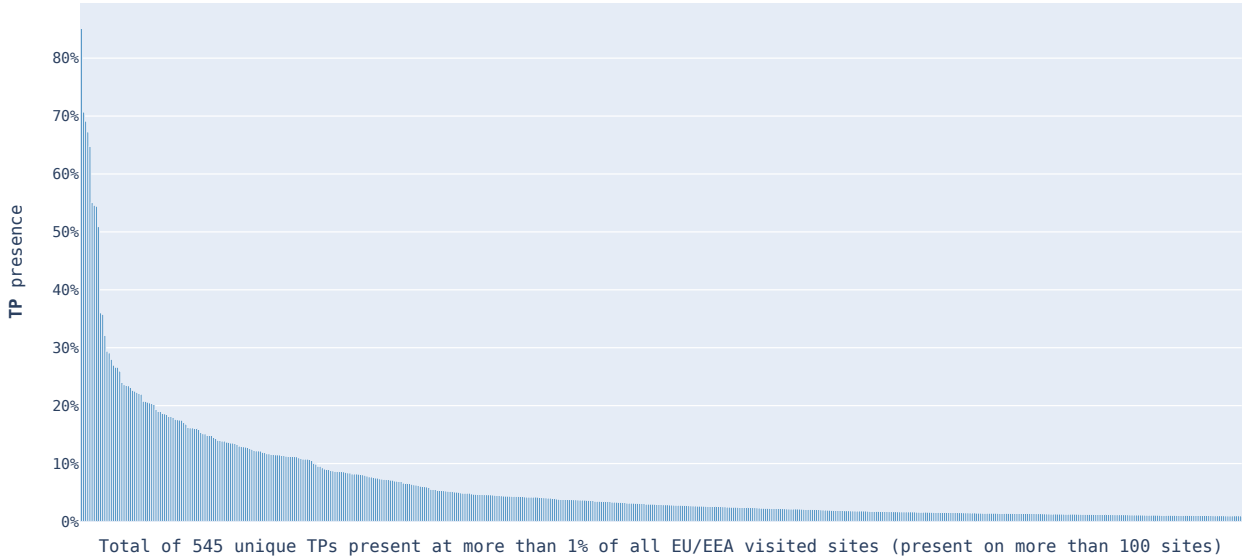

Figure 14: Third-parties present on more than 1% of all visited sites. In total of 545 of such TPs are identified producing a long tail with a handful of TPs present on most sites.

If we look at the top 20 TPs, it can be observed, that 10 TPs are owned by Google, 2 by Facebook, and 1 by Amazon which means that 65% of all TPs in the top 20 list are owned by three large corporations, see Table 6. There are several studies which invetsigate the presence of TPs on the web. In 2012, Roesner et. al. in [13], presented the results of the top trackers on top 500 domains worldwide as of September 2011. Even-though this research was done almost ten years ago and focuses on the trackers, 6 of their top 20 trackers are present in our top 20 TP list and 11 are present in our top 40. Another study by Englehardt et. al

[32] analyse the internet traffic data from top one million sites worldwide as of January 2016. Their results on the presence of the top third-parties on the web are similar to ours with 13 of their top 20 TPs present in our top 20 list and 15 in our top 40 list. This indicates that there are several TPs and trackers that are always present in the top lists even before and after GDPR as well as worldwide or just at EU/EEA area.

When comparing the results of the top 20 TPs with the previous study of Sørensen and Kosta [6], which is based on the same data-set but with shorter harvesting period, very few changes can be observed in the results, with 16 of their top 20 TPs present in our top 20 and 19 present in our top 40. On top of that, almost all TPs owned by the mentioned corporations (besides googleadservices.com) are present in both top 20 lists. This indicates that the big players have kept their positions and there has been not much of a shift on the market.

To summarise this, the commencement of GDPR, seems not to have a big impact on the presence of the top TPs, which are repeating among studies done before and after its commencements, with global or only EU/EEA focus. On the other hand, it seems that TPs are present less after GDPR, are less volatile, and the same ones are present on more visited sites.

Rank	***Third-party***	***Presence***	***Rank***	***Third-party***	***Presence***
1	google-analytics.com	85%	21	criteo.net	24%
2	googleapis.com	71%	22	scorecardresearch.com	24%
3	doubleclick.net	69%	23	bidswitch.net	23%
4	gstatic.com	67%	24	openx.net	23%
5	google.com	65%	25	mathtag.com	23%
6	facebook.com	55%	26	adsrvr.org	23%
7	googletagmanager.com	55%	27	moatads.com	22%
8	google.dk	54%	28	addthis.com	22%
9	facebook.net	51%	29	hotjar.com	22%
10	adform.net	36%	30	rlcdn.com	21%
11	cloudflare.com	36%	31	bluekai.com	21%
12	adnxs.com	32%	32	casalemedia.com	21%
13	googlesyndication.com	29%	33	advertising.com	20%
14	googletagservices.com	29%	34	adsafeprotected.com	20%
15	cloudfront.net	28%	35	krxd.net	20%
16	2mdn.net	27%	36	googleadservices.com	19%
17	rubiconproject.com	27%	37	dotomi.com	19%
18	pubmatic.com	27%	38	twitter.com	19%
19	criteo.com	26%	39	serving-sys.com	19%
20	yahoo.com	24%	40	smartadserver.com	19%

Table 7: Top 40 most present TPs across all harvests with the presence in percents. In total, 12 778 sites are visited.

5.3 Visited sites' 'health' status

The 'health' status of visited sites is an interesting and indeed important feature to look into because it can greatly effect our previous assumptions. As it has been explained previously, the number of visited sites with a response fluctuates among harvests anywhere between 12 625 to 12 733 out of 12 778 requests made to those sites, with a decreasing tendency throughout the harvests. That also means, that roughly one percent of all visited sites does not have any response recorded on average. It was also shown that this does not have a noticeable impact on the number of TP responses. However, it is important to look into the status of visited sites that returned a response.

The `http_response` table of every harvest contains a column with a HTTP response status code of every response. By investigating the status code of the response to initiating request for every visited site, we can obtain an overview of the 'health' of all sites during the harvesting period, which is presented in Figure 15. It can be observed that the number of `2xx - Success` response codes is steadily decreasing from 8 315 at the first to 3 891 at the last harvest. On the other hand, the trend is opposite for the remaining status code groups, especially for `3xx - Redirection`, where the number of sites increased from 1 678 to 5 574. The statistic overview of the remaining codes is in Table 8. If we look to a more detailed diagram in Figure 16, it depicts the most populous response codes which together with `200 - OK` make approximately 98% of all response codes during harvests. We also notice that the `301 - Moved Permanently` makes up roughly 70% of all contained responses.

All of the the above mentioned, can have a major effect on the results coherence and can signify a unforeseen deviation in measurements. The increasing number of `4xx - Client error`, `5xx - Server error`, and sites with missing responses means that fewer and fewer responses are received for site requests, which may result in lower number of TPs and responses. On top of that, the growing number of `3xx - Redirect` means that fewer and fewer sites which are originally requested are visited in the end, since the request is redirected to a different URL, which distorts the consistency of results. This causes the results to be skewed, in particular the slow and steady decrease of the number of responses (of all types) as well as the constant small decrease of the number of TPs over harvests, may be caused by this.

Generally speaking, the chosen approach for crawling the web using the static list of sites to visit and obtaining harvested data-set, seems to not be the correct one for harvests that are longitudinal. To clarify, the longer the period of harvest, the more visited sites are responding with some kind of error, redirect or no response at all, the less data can be obtained about all visited sites and the more uncertainty is then introduced in the analysis.

5.4 Average number of third-parties per first-party categories

The investigation continues with the analysis of the distribution of third-parties over harvests grouped by the visited sites categories. The distribution is presented in Figure 18. There are 11 categories of visited sites in total, which can be grouped into (i) public and (ii) private sector websites. The assumption is that public websites have fewer TPs present and their

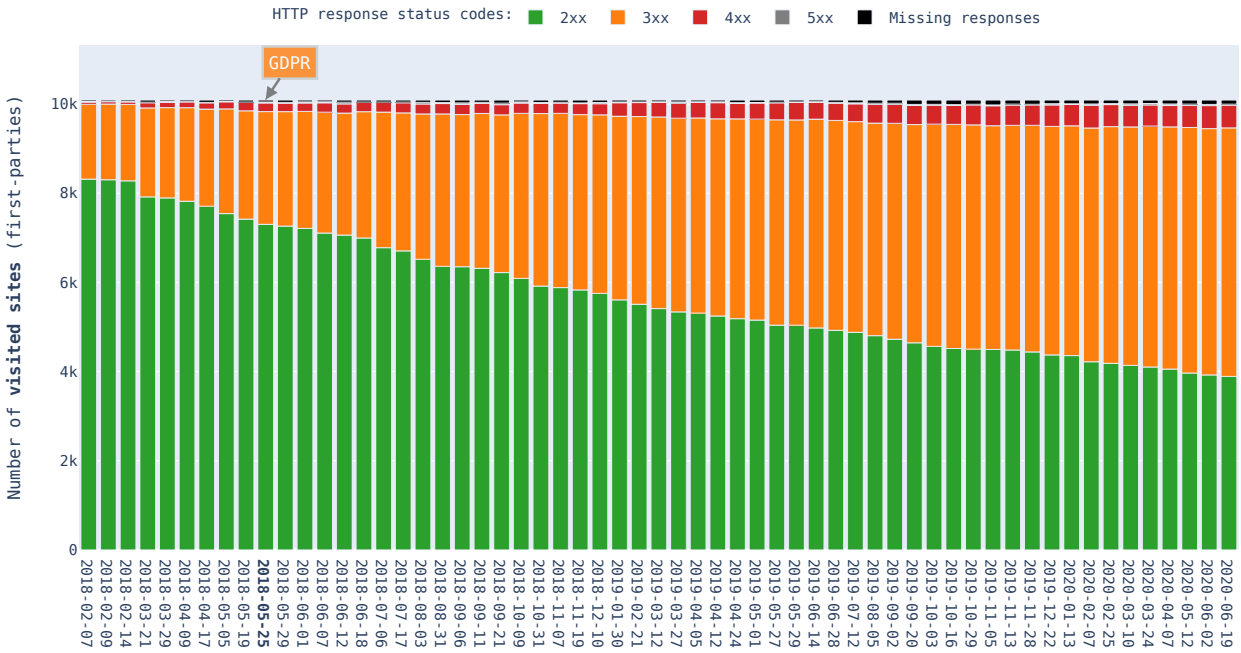

Figure 15: Categorised HTTP response status codes obtained from a response to an initiating request for every visited site of EU/EEA origin over harvesting period.

Status code	***First harvest***	***Last harvest***	***Change (last - first)***
2xx - Success	8 315	3 891	-4 424 / -53.2%
3xx - Redirection	1 678	5 574	+3 896 / +332.2%
4xx - Client error	50	510	+460 / +1020%
5xx - Server error	13	21	+8 / +161.5%
Missing response	33	93	+60 / +281.8%

Table 8: Statistics on number of first-parties with a HTTP response status code of the given category per first and last harvest, and their difference.

presence is stable, and vice versa for private websites. The idea is that private websites need to make more income from displaying personalised ads, which also means more tracking TPs, while the public ones' main purpose is to inform instead of generating revenue by having a website.

The public websites group contains six categories, namely (i) University, (ii) Government, (iii) Public news, (iv) Postal services, (v) Public transport, and (vi) Public weather. The University sites have a steady number of TPs appearing in them with the exception of some outliers. The Government sites have a slight increase of TP presence in mid 2019 and then it stabilises. In addition, the Public transport sites encounter an increase of TP presence followed by a decrease, back to previous levels, from the end of 2018 to beginning

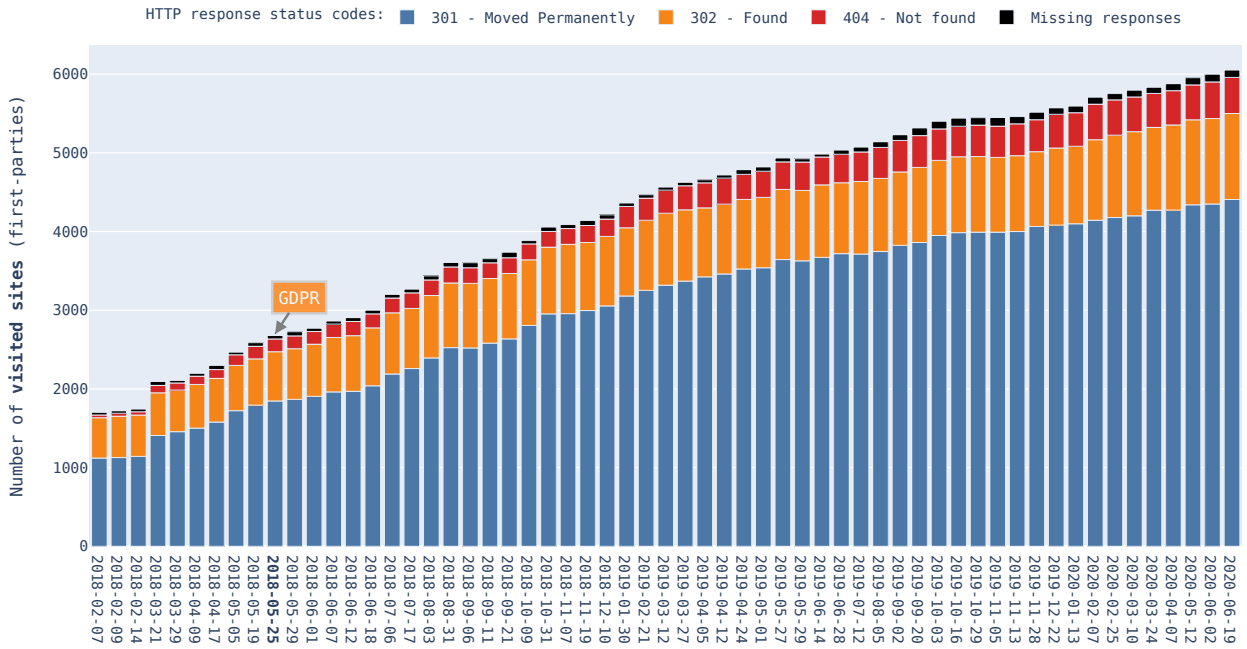

Figure 16: Specific HTTP response status codes obtained from a responses to an initiating request for every visited site of EU/EEA origin over harvesting period.

of 2020. Then again, the number of TPs present on Public weather sites increase over all harvests since the end of 2018 and similarly for Postal services which have an increasing number of TPs from mid 2019. On the other hand, the Public news sites seems to be the only category with a small decrease of number of TPs over the time.

Another group is private websites containing five categories: (i) Entertainment, (ii) Legal services, (iii) Private news, (iv) Shopping and travel, and (v) Private weather. Taking a closer look at all categories, we can detect that all of them, besides Legal services, have encountered some level of decrease in TPs number. The Legal services, similarly to Government, experience a sudden spike of number of TPs in the beginning of 2019 which is immediately leveled and kept stable until the end, but generally have low presence of TPs, such as public ones. On the contrary, the Entertainment, Shopping and travel, and Private news have the highest number of third-parties and their fluctuation, with a decreasing tendency of TP presence. On top of that, Private weather sites experience a noticeable decrease of TPs presence and highest fluctuations over the time.

To confirm the assumptions about the increasing or decreasing TP trends, a scatter plot containing the linear regression with an average number of TPs per category with respect to harvest date, is constructed, see Figure 17. The Private weather and Private news sites have a noticeable decline in the number of TPs, while the Public weather and Postal services have a noticeable increase. We can therefore identify a general trend that most private sites, except the Legal services, have a downwards sloping regression line and most public sites, except the Public news, have an upward sloping regression line, as Table 9 shows.

Study by Sørensen et. al., have analysed the same metrics on the smaller data-set of

first 21 harvests, Figure 4, Figure 5, and Table 2 in [6]. A subtle difference in trends can be observed when comparing the results. More site categories follow a trend of decreasing number of TPs over the time and more coherent distribution of TPs can be observed among the public sites.

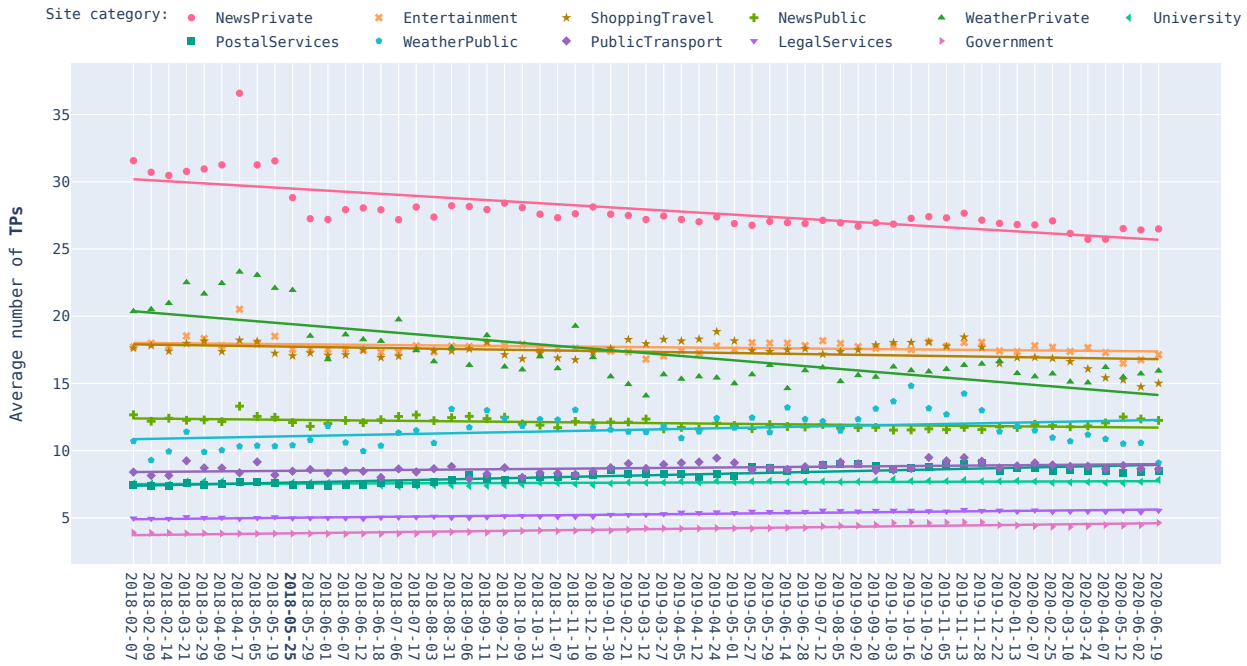

Figure 17: Average number of third-parties per each category of visited sites with respect to the harvest date. The linear regression is represented by the line, which depicts the trend of TP presence. Slope coefficients are in Table 9.

Category	***Slope***	***Private***	***Public***
WeatherPrivate	-0.108	X	
NewsPrivate	-0.078	X	
ShoppingTravel	-0.019	X	
NewsPublic	-0.012		X
Entertainment	-0.011	X	
University	+0.004		X
PublicTransport	+0.010		X
LegalServices	+0.012	X	
Government	+0.015		X
WeatherPublic	+0.025		X
PostalServices	+0.027		X

Table 9: Slope coefficients per site category for regression lines in Figure 17.

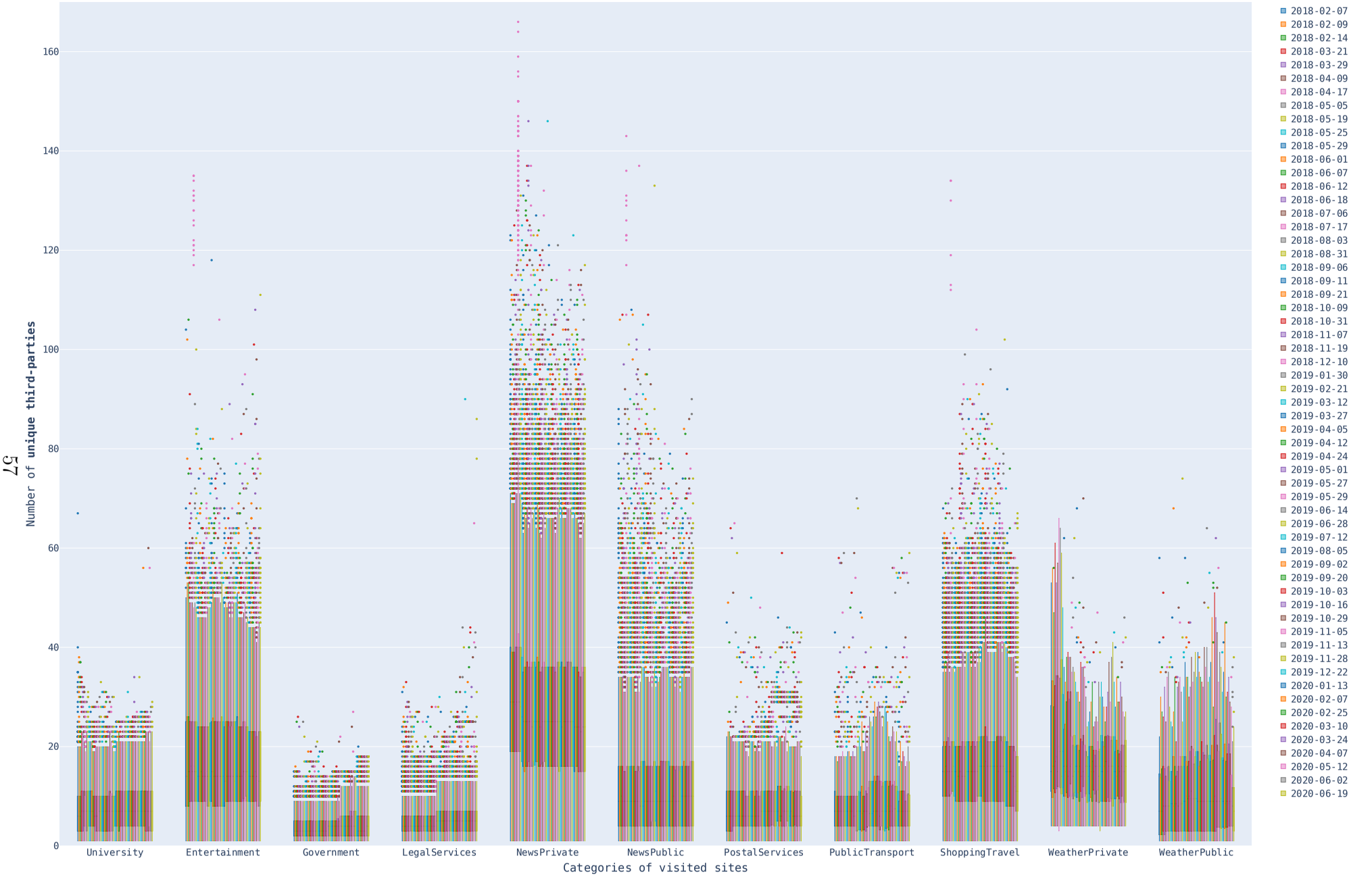

Figure 18: Number of unique third-parties per each category of visited sites with respect to harvest date.

5.5 Third-party categorisation

It is interesting to know what is the purpose of TPs present in the harvests, knowing already which ones are present on visited sites. To obtain the category of each TP, a Webshrinker API is used. After processing the results, 33 categories of TPs are defined. Figure 19 presents a heatmap with visited site categories and top 15 most populous third-party categories, with TP distribution among them. Top 15 categories represent 93.7% of all TPs. We can see that only seven TP categories are present substantially across visited site categories. Unsurprisingly, many Marketing, Advertising and Technology TPs are present on Private news, Shopping and travel, and Entertainment sites, suggesting a high presence of advertisement.

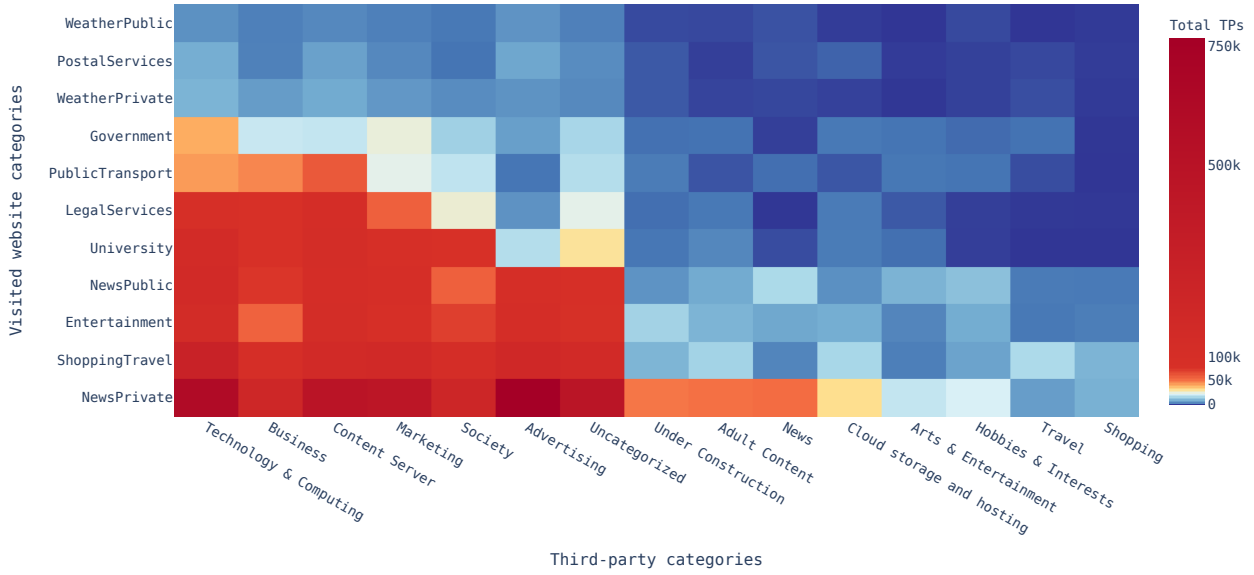

Figure 19: Heatmap with the number of third-party occurrence with respect to the TP category and visited site category

Before we investigate this topic further however, it is vital to find out how well the chosen categorisation tool performs. Sørensen et. al. in their study categorised most of the TPs that appeared through the shorter harvesting period by hand [6]. Therefore, this data are used to compare how the categorisation tool categorise TPs from their list, and see if there is any correlation among the results.

Unfortunately, it seems that there is quite low correlation among corresponding categories, as presented in Figure 20. It is difficult to say why it may be so. To begin with, we do not know what are the exact criteria on which the categorisation engine bases the decision. Most TPs from their categorisation fall into the *Technology & Computing*, or *Uncategorized*. In case of *Technology & Computing* it seem that the automatic categorisation of TPs is too broad. Some manual categories such as *Advertising*, *Analytics*, *Distribution technology*, or *Social media* are categorised partially correctly, on average 30% match, but others do not perform that well.

On the other hand, the manual categorisation provides only 15 categories which might

be a reason for some 'zero' spots and it would be beneficial if more categories were included. Also, a manual mapping from the granular automatic categorisation to the manual categories may help to produce better results. Nonetheless, it can be observed from the figure, that if the manual categorisation is considered as a ground truth, further analysis based on obtained categorisation data, would be inaccurate and therefore we are not further proceeding with it.

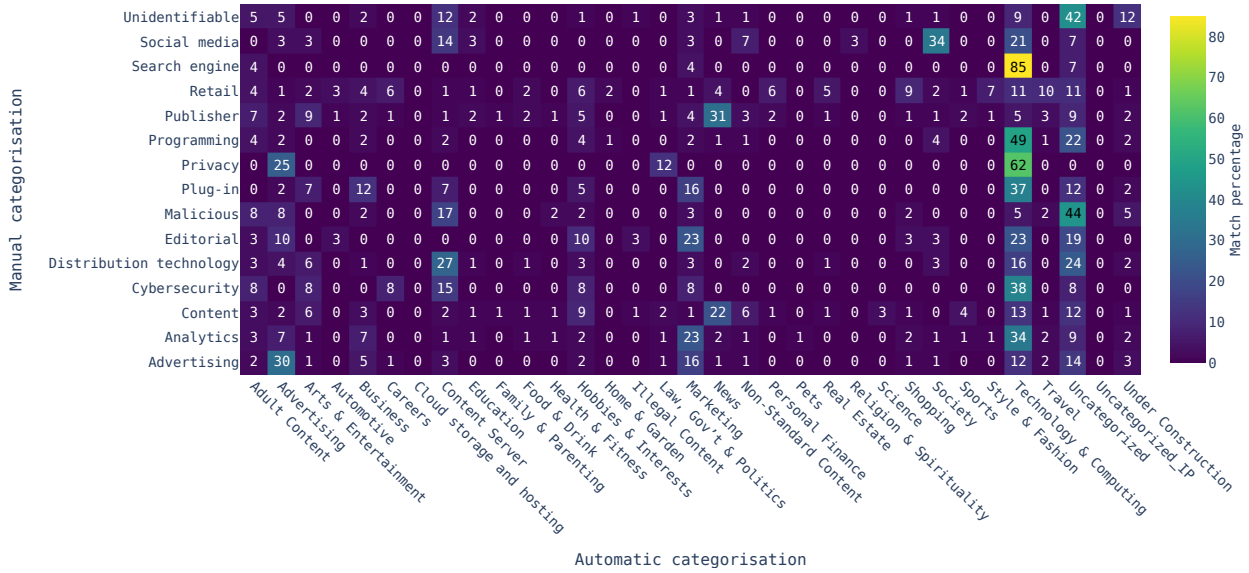

Figure 20: Heatmap with the percentage match between the manual third-party categorisation and automatic TP categorisation.

5.6 Maliciousness of third-parties

Lastly, we investigate the maliciousness of third-parties to find out if any of the TPs have sinister intentions. Two sources are used to obtain this information, one very popular among researchers - VirusTotal [44], and the other one implemented by the most popular web-browsers - Google Safe Browsing [45], and compare the results.

VirusTotal aggregates results from dozens of sources and provides an overview on how many of them considers the given TP as malicious. Successful results are returned for 6 621 TPs of 6 868. The results are as follows:

- 193 TPs are labeled as malicious by 1 source
- 38 TPs by 2 sources
- 16 TPs by 3 sources
- 12 TPs by 4 sources
- 2 TPs by 5 and 6 sources
- 1 TP by 7, 8, and 9 sources respectively

Most of the TPs that are labeled as *malicious* are only marked by a single source and given that there are around 70+ aggregated sources, this is not a high number. As Peng et. al. in their study [44] explain, the problem with this service is that the sources often do not agree with each other, as in our case, and therefore it is important to select a good enough threshold from which we considered the site to be malicious as well. They argue that most of the researchers set the threshold to 1 with a few choosing 2 or 3. If the approach of most researchers is followed and a threshold of 1 is chosen the final results by VirusTotal are 266 malicious TPs being present during harvests.

GSB on the other hand, returned successful results for all TPs. Since all TPs are labeled as non-malicious, we can only conclude that based on this API, all TPs are safe.

To summarise, both services' results differentiate from each other due to a different approach and underlying labeling techniques. We can conclude that no TP is considered as malicious by GSB, while 266 are considered as malicious by a VirusTotal. To say with certainty how many of the TPs are actually malicious from the ones reported, the results from the VirusTotal can be used as a starting point and further analysis needs to be performed by investigating each of the 266 TPs in details.

In the preceding sections, six areas of interest were defined, investigated, and analysed producing assumptions and premature partial conclusions which need to be put into perspective and combined to draw a conclusion and answer the problem formulation which is done in Section 7.

6 Discussion and future work

This chapter consists of discussion, where we look back, reflect on certain topics and discuss approaches taken, and future work, where we propose how the current analysis could be improved in the future and what are the next steps.

6.1 Discussion

Decisions taken throughout the process might not always been the right one, therefore we will look into selected topics and discuss what went wrong and how it could be improved.

Harvest data-set We would like to reflect on the process of obtaining the harvest data-set. In Section 5.3, we have shown that throughout the harvesting period the number of visited sites returning a `2xx - OK` response code have decreased by 53% and at the same time the number of redirects increased by 332%, and errors or no responses increased by 650%, when comparing first and last harvests' HTTP response status codes. To put it differently, the data obtained are less and less consistent and accurate over the time.

This shows, that the approach chosen by Sørensen et. al. in [6] is not well suited for a longitudinal study such as this one and a different one should be chosen in the future, should the harvesting continue. One of possible approaches could be the use of a dynamic list of sites to visit instead of the static one. The dynamic list could be based on a predefined list of landing pages (such as 1 363 in this data-set) where for each landing page n-subpages would be randomly generated and visited every harvest.

The proposed approach can by verified, if feasible, by using only visited sites which return `2xx - OK` from the current date-set to find out how much the number of TPs vary among sub-pages of given site to visit. If the number of TPs is similar across all sub-pages for given visited sites, then the suggested approach can be followed and should produce much less error and redirect response codes.

Another drawback of the selected crawling design is the single location from which all requests are made. This does not allow an analysis on whether there is varying behavior of TPs on visited sites across different countries.

Third-party categorisation The existing manual categorisation procedure has been automated as covered in Section 5.5, but the level of accuracy achieved was not satisfactory when comparing the categories obtained using an API and manually. Since we do not know how the underlying technology decides on the label for each TP, it is difficult to assess why the correlation is low. For the future we suggest manual categorisation with a broader range of categories, and use of crowd sourcing platforms to perform the categorisation. This way more precise and faster categorisation can be achieved.

Remove "false" third-party servers Many TPs were found throughout harvests, 6 868 in total. Therefore, there is a chance that some of visited sites deploy their own servers, e.g.

content server, which have a different domain from the visited site and thus it is counted as a TP. By obtaining a list of servers operated by the first-party, we could remove "false" TP servers, that are owned or controlled by the FP. We tried to find a registrant of every TP using WHOIS, but since the commencement of GDPR, most information is *Redacted for privacy* which made the data unusable. It is therefore necessary to find a different source to obtain this information to make the analysis even more precise.

Third-party cookies blocking by browsers by default During the harvesting period, there have been some changes in the privacy landscape and approaches that browsers took. In particular, third-party cookies are blocked by several of the biggest browsers by share by default, such as Firefox from September 2019[26] or Safari from March 2020[27]. Chrome will join others by 2022[28]. The browser used by OpenWPM is Firefox, however, the version used throughout all harvests is 52.4.1 and TP blocking by default is present since version 69.0, therefore the results are not affected by this change. On the other hand, it would be interesting to do further harvests using the old and new version of Firefox to determine if decisions taken by browser manufacturers influence the presence of TPs more than GDPR.

6.2 Future work

We did our best to analyse and provide results to successfully answer the problem formulation, but there is always room for improvement though. In this section we present how analysis could be improved in the future and what are the areas and topics which can be further looked into. We also introduce next planned steps after finishing this thesis.

Include more metrics Throughout analysis, only two tables from `.sqlite` database files of every harvest were used, responses (`http_responses`) and visited sites (`site_visits`). Other tables such as those containing information about requests (`http_requests`) or redirects (`http_redirects`) could be analysed in the future to obtain a more precise picture of the web traffic. This would allow for better understanding of where the increasing number of visited sites requests are being redirected to, as the number of redirections increased by 332% when comparing first and last harvests. It would also provide more accurate analysis of TP presence and possibly introducing new metrics as well, such as average number of redirects per visited site or list of top sites being redirected to.

On top of that, including subdomains of TP URLs when obtaining the root-domain would allow a better categorisation of TP purposes: for example instead of just `example.com` from `https://abcd.example.com` the `abcd.example.com` would be the outcome, which would allow a more detailed analysis of purpose of each third-party, if such detailed analysis would be beneficial.

[26] `https://www.mozilla.org/en-US/firefox/69.0/releasenotes/`,accessed25March2021

[27] `https://developer.apple.com/documentation/safari-release-notes/safari-13_1-release_notes#overview`, accessed 25 March 2021

[28] `https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html`, accessed 25 March 2021

Write a journal paper and make data-sets public A previous study by Sørensen and Kosta used the same data-set with fewer harvests to analyse the effects of GDPR.[6] This thesis has built on these findings and data-sets from the study, and aimed to extend the analysis from an 8 months period (21 harvests) to 2.5 years (59 harvests) with a vision to come up with results and new findings which can be used as a base for writing a paper to be published in the journal in collaboration with authors of the original study. That is still the vision, which we will hopefully accomplish in the near future. Once the paper is published, all data-sets will also be made publicly available.

Publish the results online with interactive diagrams Visualisation of results is very important since it allows better understanding and interpretation of them. Multiple diagrams have been produced and introduced in this report. However, making it possible to interact with presented results, filter them, or change diagram types is very practical. Therefore, along with publishing the data-sets, an improved visualisation of results, allowing the above-mentioned interaction, will be presented and published. The production of such diagrams is already in progress. In Figure 21 an example implementation on TP development is shown. It is possible to select a time period which automatically updates the upper diagram's timeline along with the table with statistics. On top of that, when hovered over the top diagram's bars, the bottom one gets updated and shows the top 1% of TPs for given harvest.

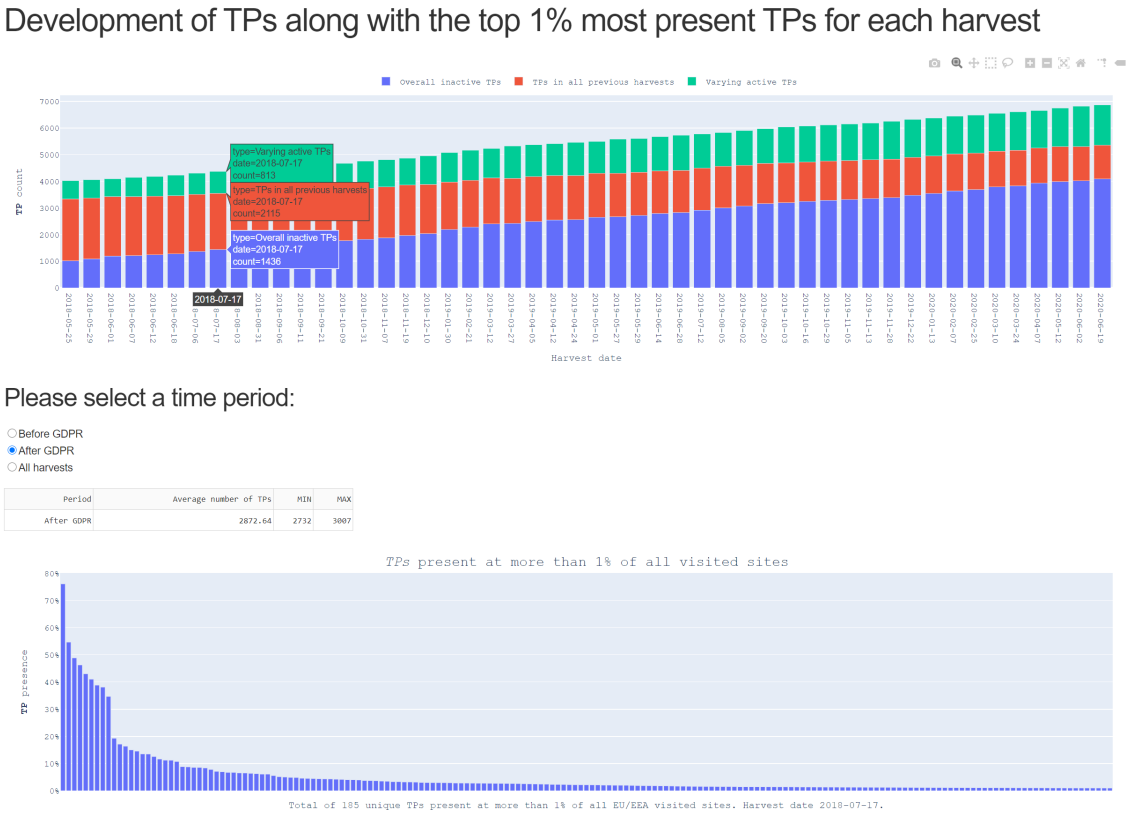

Figure 21: Example from in progress implementation of interactive diagrams.

7 Conclusion

The aim of this whole report is to answer the problem formulated in the beginning and to document all processes throughout. Multiple studies focusing on the GDPR and its influence on the third-party or tracker presence on the web have been already published [1, 6, 7], but we wanted to contribute with our work to this.

The Obtain, Scrub, Explore, Model, and iNterpret (OSEMN) framework was followed through the thesis and all work we have done closely mirrored phases defined by the framework. The exception is the *Model* process which was skipped, since making predictions was not the aim of this thesis.

Different kinds of data were first obtained, with some being accessible to all AAU students and some needing to be gathered. The harvest data-set was among the ones which was obtained from AAU while data used for enrichment had to be obtained from various sources.

Afterwards, once all data have been obtained, they were cleaned, transformed, combined, extracted and exported in different formats all as part of the scrub process. There were many challenges in this process which can be well summarised by a quote by John Tukey: *"The data may not contain the answer. The combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data."*[57] Many steps had to be repeated multiple times due to errors in the data found later in the process. This process was also the most laborious and time consuming process, which was expected.

When all necessary data have been exported and were ready to be explored and visualised, another process began. Here, the fun part with playing, visualising, transforming, and performing calculations on data took place. Multiple diagrams were produced in this process which became the backbone during the results' interpretation and conclusion. These are also part of the last process.

During the interpretation process, all previously produced diagrams and statistics are presented and analysed, in the Results chapter, producing multiple assumptions and partial conclusions which need to be put into perspective and combined to finally answer the problem formulation, which is the very last part of this thesis and will be presented now.

As already mentioned previously, the harvesting period is almost 2.5 years from February 2018 to June 2020. In total, data from 59 harvests are used and total number of visited sites during each harvest is 12 778, out of which 10 089 heve the EU/EEA origin.

Throughout the analysis of responses and third-parties, we can observe that there is a tendency of both metrics decreasing throughout the harvesting period, A noticeable, though only slight drop in both metrics is present right before and after the commencement of GDPR and afterwards the decreasing trend is steady. There is on average a 7.5% decrease in number of responses to requested sites with EU/EEA origin and 9.6% decrease in number of responses to requested sites with EU/EEA origin with a TP communication when comparing averages of the pre- and post-GDPR period. On top of that, the number of visited sites with a TP response decrease slightly – by 1.04%

On average, there are 2 900 unique TPs present at each harvest with 3 056 and 2 872

TPs present on average in pre- and post-GDPR period which is a 6% decrease of TP presence. In total, 6 868 unique TPs are identified. Third-parties are also less volatile and appear on more visited sites after the commencement of GDPR with 545 TPs appearing at more than 1% of visited sites compared to 181 TPs in the previous study done on first 21 harvests [6].

When we analyse the average presence of TPs on visited sites by sites' category and group them by sector to public and private, we identified an interesting trend. Private sites follow a decreasing trend in number of TPs over harvests (4 out of 5 categories) while in contrast, public sites follow an increasing trend (5 out of 6 categories). Among other observations are the generally higher number of TPs present on all private sites and higher fluctuation in numbers of TPs across harvests in comparison to public sites, with the exception of Legal service.

The analysis on the 'health' status of visited sites is based on the HTTP response code of the initiating request for each visited site. In total 10 089 visited sites have an EU/EEA origin and while in the first harvest 8 315 returned `2xx - OK` status code, in the last one it was only 3 891 which is a 53.2% decrease. On top of that, other status codes have a substantial increase of their presence, up to 10x: from 1 678 to 5 574 – 332% increase for `3xx - Redirect`, from 50 to 510 – 1020% increase for `4xx - Client error`, from 13 to 21 – 161% increase for `5xx - Server error`, and from 33 to 93 – 281% increase for those that have no recorded response, when comparing first and last harvests.

That challenges all of the above analysis with its results, since the number of visited sites returning an OK status is decreasing over harvests, more requests are being redirected to a different URL and more and more sites return an error code or no response at all. This explains to a great extent why we can observe a trend of slowly decreasing number of responses and unique TPs throughout the analysis.

Third-party categorisation analysis was not fruitful, since we realised that the TP categories obtained using the Webshrinker does not correlate with the manual categorisation of TPs from the previous study based on same, but 21 harvests long data-set. Therefore, any further analysis on this topic would produce inaccurate results.

Maliciousness of TPs is not proved as Google Safe Browsing have not labeled any TP as malicious and most of the 266 (3.8%) reported TPs by VirusTotal are labeled by one or two sources out of more than 70. In order to make a confident conclusion on this topic, further research is needed.

To summarise, the number of third-parties have a tendency to decrease through the harvesting period, However, based on this research we cannot prove that it is due to the effect of GDPR since the number of successfully requested sites to visit is also decreasing, producing a smaller number of results which have contributed to the decreasing number of TPs. Only a slight sudden decrease of TPs right before and after the commencement of GDPR is observed, which is not significant enough and might also be the result of certain providers temporarily stopping their services to become compliant with the regulation and slowly resuming them over time. Therefore, there is not enough evidence to prove that the GDPR had a noticeable effect on the presence of TPs during the harvesting period.

References

[1] Tobias Urban et al. "Beyond the Front Page: Measuring Third Party Dynamics in the Field". In: *Proceedings of The Web Conference 2020 (WWW '20)*. Apr. 2020. ISBN: 9781450370233. DOI: 10.1145/3366423.3380203.

[2] Richie Koch. *Cookies, the GDPR, and the ePrivacy Directive*. URL: https://gdpr.eu/cookies/.

[3] European Parliament. *Regulation (EU) 2016/679 - General Data Protection Regulation (GDPR)*. Apr. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[4] Jannick Sorensen, Hilde Van den Bulck, and Sokol Kosta. "Privacy Policies Caught Between the Legal and the Ethical: European Media and Third Party Trackers Before and After GDPR". In: *SSRN Electronic Journal* (Aug. 2019). ISSN: 1556-5068. DOI: 10.2139/ssrn.3427207.

[5] Xuehui Hu and Nishanth Sastry. "Characterising Third Party Cookie Usage in the EU after GDPR". In: (2019). DOI: 10.1145/3292522.3326039.

[6] Jannick Sørensen and Sokol Kosta. "Before and after GDPR: The changes in third party presence at public and private European websites". In: *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*. Vol. 11. New York, New York, USA: Association for Computing Machinery, Inc, May 2019, pp. 1590–1600. ISBN: 9781450366748. DOI: 10.1145/3308558.3313524.

[7] Rasmus Helles, Stine Lomborg, and Signe Sophus Lai. "Infrastructures of tracking: Mapping the ecology of third-party services across top sites in the EU". In: *New Media & Society* 22.11 (Nov. 2020), pp. 1957–1975. ISSN: 1461-4448. DOI: 10.1177/1461444820932868.

[8] Tobias Urban et al. "Measuring the Impact of the GDPR on Data Sharing in Ad Networks". In: *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20)*. ACM, Oct. 2020, pp. 222–235. ISBN: 9781450367509. DOI: 10.1145/3320269.3372194.

[9] Tobias Urban et al. *The Unwanted Sharing Economy: An Analysis of Cookie Syncing and User Transparency under GDPR*. Tech. rep. Nov. 2018.

[10] *Proposal for a Regulation (GDPR)*. Tech. rep. Brussels: Presidency of the Council of the European Union, June 2015. URL: https://data.consilium.europa.eu/doc/document/ST-9565-2015-INIT/en/pdf.

[11] Anu Bradford. "The Brussels Effect". In: *Northwestern University Law Review* 107.1 (2012). URL: https://ssrn.com/abstract=2770634.

[12] Iskander Sanchez-Rola et al. "The web is watching you: A comprehensive review of web-tracking techniques and countermeasures". In: *Logic Journal of the IGPL* 25.1 (Feb. 2017), pp. 18–29. ISSN: 1367-0751. DOI: 10.1093/jigpal/jzw041.

[13] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. "Detecting and Defending Against Third-Party Tracking on the Web". In: *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*. San Jose, CA: USENIX Association, 2012. DOI: 10.5555/2228298.2228315.

[14] Mozilla Developer Network contributors. *Using HTTP cookies - HTTP*. Feb. 2021. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies.

[15] Kaspersky. *What is a Cookie? How it works and ways to stay safe*. 2021. URL: https://www.kaspersky.com/resource-center/definitions/cookies.

[16] Nick Briz. *This is Your Digital Fingerprint*. July 2018. URL: https://blog.mozilla.org/internetcitizen/2018/07/26/this-is-your-digital-fingerprint/.

[17] Peter Eckersley. "How Unique Is Your Web Browser?" In: *Privacy Enhancing Technologies*. Ed. by Mikhail J. Atallah and Nicholas J Hopper. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–18. ISBN: 978-3-642-14527-8.

[18] Thomas Hupperich et al. "An Empirical Study on Online Price Differentiation". In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. CODASPY '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 76–83. ISBN: 9781450356329. DOI: 10.1145/3176258.3176338.

[19] Vasant Dhar. "Data Science and Prediction". In: *Commun. ACM* 56.12 (Dec. 2013), pp. 64–73. ISSN: 0001-0782. DOI: 10.1145/2500499.

[20] Pete Chapman et al. *CRISP-DM 1.0 Step-by-step data mining guide*. Tech. rep. 1999.

[21] Roald Bradley Severtson. *What is the Team Data Science Process?* 2017. URL: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview.

[22] IBM Analytics. *Foundational Methodology for Data Science*. Tech. rep. IBM, June 2015. URL: https://tdwi.org/~/media/64511A895D86457E964174EDC5C4C7B1.PDF.

[23] Hilary Mason and Chris Wiggins. *A Taxonomy of Data Science*. Sept. 2010. URL: http://www.dataists.com/2010/09/a-taxonomy-of-data-science/.

[24] Jeroen Janssens. *Data Science at the Command Line*. First edition. O'Reilly Media, Inc., 2014. ISBN: 9781491947852.

[25] Davy Cielen, Arno Meysman, and Mohamed Ali. *Introducing Data Science: Big Data, Machine Learning, and more, using Python tools*. First. Manning Publications, 2016. ISBN: 9781633430037.

[26] Cher Han Lau. *5 Steps of a Data Science Project Lifecycle | by Dr. Cher Han Lau | Towards Data Science*. Jan. 2019. URL: https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492.

[27] John Tukey. "The Future of Data Analysis". In: *The Annals of Mathematical Statistics* 33.1 (1962), pp. 1–67. URL: https://www.jstor.org/stable/2237638.

[28] Dhanurjay "DJ" Patil. *Data Jujitsu: The Art of Turning Data into Product*. O'Reilly Media, Inc., Aug. 2012. ISBN: 9781449341152.

[29] Randy Lao. *Life of Data | Data Science is OSEMN*. Nov. 2017. URL: https://medium.com/breathe-publication/life-of-data-data-science-is-osemn-f453e1febc10.

[30] George Box. "Robustness in the Strategy of Scientific Model Building". In: *Robustness in Statistics*. Elsevier, 1979, pp. 201–236. DOI: 10.1016/b978-0-12-438150-6.50018-2.

[31] Ian Sommerville. *Software engineering*. 9th. 2011. ISBN: 9780137035151.

[32] Steven Englehardt and Arvind Narayanan. "Online tracking: A 1-million-site measurement and analysis". In: *Proceedings of the ACM Conference on Computer and Communications Security*. Vol. 24-28-October-2016. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 1388–1401. ISBN: 9781450341394. DOI: 10.1145/2976749.2978313.

[33] Adam Lerner et al. "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016". In: *Proceedings of the 25th USENIX Security Symposium*. Austin, TX, Aug. 2016, pp. 997–1013. ISBN: 978-1-931971-32-4. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/lerner.

[34] Adrian Dabrowski et al. "Measuring Cookies and Web Privacy in a Post-GDPR World". In: *Passive and Active Measurement*. Cham: Springer International Publishing, 2019, pp. 258–270. ISBN: 978-3-030-15986-3.

[35] Costas Iordanou et al. "Tracing Cross Border Web Tracking". In: *Internet Measurement Conference (IMC '18)*. ACM, 2018. ISBN: 9781450356190. DOI: 10.1145/3278532.3278561.

[36] Georg Merzdovnik et al. "Block Me if You Can: A Large-Scale Study of Tracker-Blocking Tools". In: *Proceedings - 2nd IEEE European Symposium on Security and Privacy, EuroS and P 2017*. Institute of Electrical and Electronics Engineers Inc., June 2017, pp. 319–333. ISBN: 9781509057610. DOI: 10.1109/EuroSP.2017.26.

[37] Timothy Libert, Lucas Graves, and Rasmus Kleis Nielsen. *Changes in Third-Party Content on European News Websites after GDPR*. Tech. rep. 2018. URL: https://reutersinstitute.politics.ox.ac.uk/our-research/changes-third-party-content-european-news-websites-after-gdpr.

[38] Tim Wambach and Katharina Bräunlich. "The Evolution of Third-Party Web Tracking". In: *Information Systems Security and Privacy*. Cham: Springer International Publishing, 2017, pp. 130–147. ISBN: 978-3-319-54433-5.

[39] Steven Englehardt et al. "Cookies That Give You Away: The Surveillance Implications of Web Tracking". In: (). DOI: 10.1145/2736277.2741679.

[40] Steven Englehardt. *The Web Privacy Problem is a Transparency Problem: Introducing the OpenWPM measurement tool*. Jan. 2016. URL: https://freedom-to-tinker.com/2016/01/14/the-web-privacy-problem-is-a-transparency-problem-introducing-the-openwpm-measurement-tool/.

[41] Benjamin Zi Hao Zhao et al. "A Decade of Mal-Activity Reporting: A Retrospective Analysis of Internet Malicious Activity Blacklists A Decade of Mal-Activity Reporting: A Retrospective Analysis of Internet". In: *Asia CCS '19: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. ACM, 2019, pp. 193–205. ISBN: 9781450367523. DOI: 10.1145/3321705.3329834.

[42] Jesse Van Der Velden. "Blacklist, do you copy? Characterizing information flow in public domain blacklists". In: *32th Twente Student Conference on IT*. Jan. 2020. URL: https://essay.utwente.nl/80567/1/Velden_BA_EEMCS.pdf.

[43] Tran Phuong Thao et al. "Large-Scale Analysis of Domain Blacklists". In: *The Eleventh International Conference on Emerging Security Information, Systems and Technologies*. Sept. 2017, pp. 161–167. ISBN: 978-1-61208-582-1.

[44] Peng Peng et al. "Opening the blackbox of virustotal: Analyzing online phishing scan engines". In: *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 478–485. ISBN: 9781450369480. DOI: 10.1145/3355369.3355585.

[45] Adam Oest et al. "PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists PhishTime: Continuous Longitudinal Measurement of the EEectiveness of Anti-phishing Blacklists". In: *Proceedings of the 29th USENIX Security Symposium*. Aug. 2020, pp. 379–396. ISBN: 9781939133175. URL: https://www.usenix.org/conference/usenixsecurity20/presentation/oest-phishtime.

[46] Google. *Chrome Browser Privacy Policy - Google Chrome*. Jan. 2021. URL: https://www.google.com/intl/en/chrome/privacy/#safe-browsing-practices.

[47] VirusTotal. *Contributors – VirusTotal*. URL: https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors.

[48] Leslie Daigle. *WHOIS Protocol Specification*. Tech. rep. The Internet Society, Sept. 2004. URL: https://tools.ietf.org/html/rfc3912.

[49] Bharath K. *Everything You Need To Know About Jupyter Notebooks*. Dec. 2020. URL: https://towardsdatascience.com/everything-you-need-to-know-about-jupyter-notebooks-10770719952b.

[50] Mark Lutz. *Learning Python*. 5th. O'Reilly Media, Inc., 2013. ISBN: 9781449355739.

[51] Bob Hayes. *Usage of Programming Languages by Data Scientists: Python Grows while R Weakens*. June 2020. URL: http://businessoverbroadway.com/2020/06/29/usage-of-programming-languages-by-data-scientists-python-grows-while-r-weakens/.

[52] Michal Krištofik. *GitHub repository with the code base*. 2021. URL: https://github.com/miso581/Third-party-presence-analysis.

[53] *Temporary Specification for gTLD Registration Data*. Tech. rep. ICANN, May 2018. URL: https://www.icann.org/resources/pages/gtld-registration-data-specs-en.

[54] *Whois and GDPR*. 2018. URL: https://www.dk-hostmaster.dk/en/gdpr.

[55] *Lov om internetdomæner*. Feb. 2014. URL: https://www.retsinformation.dk/eli/lta/2014/164.

[56] Chaoyi Lu et al. "From WHOIS to WHOWAS: A Large-Scale Measurement Study of Domain Registration Privacy under the GDPR". In: Mar. 2021. DOI: 10.14722/ndss.2021.23134.

[57] John Tukey. "Sunset Salvo". In: *The American Statistician* 40.1 (1986), p. 76. ISSN: 00031305. DOI: 10.2307/2683137.

A Returned JSON object from Webshinker


```
{
    data: [
        {
            categories: [
                {
                    id: "IAB5",
                    label: "Education",
                    parent: "IAB5",
                    score: "0.119095739636984957",
                    confident: false
                },
                {
                    id: "IAB5-9",
                    label: "College / Graduate School",
                    parent: "IAB5",
                    score: "0.026850971755581390",
                    confident: false
                }
            ],
            url: "https://www.aau.dk/"
        }
    ]
}
```


Listing 6: Whole returned JSON object containing an example of Tier-1 and 2 categorization from Webshrinker.