COOKIE DIALOGS AND THEIR COMPLIANCE

THE QUEST FOR AN AUTOMATED AUDIT PROCESS TO ENHANCE PRIVACY REGULATION

by

Koen Aerts

in partial fulfillment of the requirements for the degree of

Master of Science
in Software Engineering

at the Open University, faculty of Science
Master Software Engineering
to be defended publicly on Thursday July 22, 2021 at 15:00 PM.

Student number:	852158283	
Course code:	IM9906	
Thesis committee:	Dr. Fabian van den Broek (chairman),	Open University
	Dr. ir. Hugo Jonker (supervisor),	Open University

Open Universiteit
de beste! www.ou.nl

Contents

1

ABBREVIATIONS

Abbreviation	Meaning
API	Application Programming Interface
CCPA	California Consumer Privacy Act
CMP	Consent Management Provider
DOM	Document Object Model
DPA	Data Protection Authority
ECDF	Empirical Cumulative Distribution Function
EDA	Exploratory Data Analysis
EDPB	European Data Protection Board
EDPS	European Data Protection Supervisor
GDPR	General Data Protection Regulation
IAB	Interactive Advertising Bureau
ICC	International Chamber of Commerce
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LED	Law Enforcement Directive
LGPD	Brazilian General Data Protection Law
PDF	Probability Density Function
PLD	Pay-Level Domain
SEM	Standard Error of the Mean
TCF	Transparency and Consent Framework
TLD	Top-Level Domain
URL	Uniform Resource Locator
VM	Virtual Machine
ccTLD	country code Top-Level Domain
ePD	ePrivacy Directive

2

INTRODUCTION

Today's internet users are still overwhelmed with cookie dialogs while browsing, as website publishers want to obtain approval to use online tracking. Cookie dialogs shown on websites originate from the ePrivacy Directive passed (ePD) in 2002 and have not left the internet scene since then. Website publishers need to present such a cookie dialog when they want to place cookies on the user's machine for non-essential purposes. Although the primary use of a cookie is to provide essential purposes such as maintaining a login token between the client and the server, they are also used for tracking purposes and collecting user data, e.g., to present specific advertisements. Unfortunately, website publishers do not always respect the regulation to safeguard users' privacy and their data. Trevisan et al. [TTBM19] showed that, from a collection of 35,862 websites, 49 percent set cookies for tracking purposes before users' consent. Further, even if a website presents a consent and reject choice, the visitor is often nudged towards the consent element. Such practice where design elements are used to persuade a user to click on a particular element is called a dark pattern. Earlier research performed by Soe et al. [SNGS20] showed that a variety of dark patterns is implemented in online news outlets. As a result, users' online privacy rights are violated, often without their knowledge.

Despite evolved data protection laws, many countries struggle with the actual enforcement of the ePD and GDPR concerning users' data and protecting their privacy. As there is no European control system in place that proactively audits website publishers, it is impossible to verify every website for compliance. Currently, the burden lies on the Data Protection Authority (DPA) of each EU Member State. Citizens of each Member State can lodge a complaint with their national DPA if their rights concerning personal data are possibly violated. Only when citizens are aware of certain infringements while visiting a website can they contact the DPA, which could result in an investigation. Further, big companies with a large user base are sometimes actively monitored by the DPA, as they have a lot of influence. As a result, a large pool of websites is not observed by a privacy authority. This is a major concern, as it means that the majority of websites are free from any control.

For this research, we investigate how we can support the enforcement of the data protection regulations by providing an automatic process to audit a large set of websites to

detect violations in cookie dialogs and the cookies set. Investigations conducted by a DPA are manual labor, and the number of websites is too high to perform audits proactively. The resources of DPAs are too scarce to conduct such work manually. An automatic process that could indicate whether a website is compliant or not enables auditing on a large scale, which could decline data privacy infringements or at least show to what extent the current website landscape is compliant.

2.1. Scope

There is a limitation on the amount of time and resources we can invest in this research. Therefore, it is conducted within the following scope.

- Although many legislations exist in the world that regulate personal data processing and protect users' privacy, we are only interested in the European laws for our research, i.e., the ePrivacy Directive and the GDPR. Further, we do not examine all details of these laws as we are concerned about the specifics related to cookie dialogs.
- In the same line, the datasets we use as input for our crawler are websites from European countries, namely all members from the European Union plus three nearby countries. However, our research could be extended outside the EU as European data protection laws also apply to companies worldwide that process data from EU citizens.
- We examine compliance before users' consent only on the initial website visit and do not explore possible privacy infringements after a user action, i.e., consent or reject.
- Although there is a wide range of design choices used in cookie dialogs that could be linked to a dark pattern, we focus on observing the presence of balanced consent and reject options in a cookie dialog.
- Despite the name suggesting otherwise, website publishers are also obliged to implement a cookie dialog to perform online tracking with other technologies besides cookies. Although cookies are commonly used, as browsers will prevent third-party cookies in the future, other technologies such as Web Storage could be used more. However, for our research, we focus on the use of cookies.

2.2. Contributions

We list the main contributions of our research.

- To explore automation possibilities, we developed a crawler based on the open-source project OpenWPM [1]. Our analysis [2] showed that it is possible to rely on automation to

[1] https://github.com/koenae/openwpm-crawler
[2] https://github.com/koenae/analysis

detect violations for a preliminary list of rules. Our proof of concept supports future research on automation capabilities.

- Our research provides insights into the usage of cookie dialogs within Europe. The same analysis is performed without JavaScript and with the popular uBlock extension enabled.
- We examined the usage of the IAB Transparency and Consent Framework (TCF), a community effort that should increase compliance, by Consent Management Providers (CMPs) in European websites. Website publishers can integrate CMPs to outsource a cookie dialog implementation and decrease complexity on their part. Earlier studies that use TCF in their investigation are primarily based on TCF version 1. However, for our research, we use version 2.
- We collect the cookies that are set when initially visiting a website to detect compliance violations. Our research differs in that we link cookie purposes with the open-source knowledge base Cookiepedia to perform our analysis.
- We support automation capabilities for dark pattern detection by providing an implementation to automatically detect the presence of balanced consent and reject options in a cookie dialog.

3

BACKGROUND

3.1. CRAWLERS

Crawlers are a type of bot that can be used to retrieve elements from a website automatically. Search engines also use them to index the content of websites. Although there exist several terms, such as spiders or scrapers, which can be used to distinct particular details, we use the general term crawler for this research. There is a distinction between stateful and stateless crawls. As the name suggests, stateful crawls maintain a certain state between different browser instances. E.g., this makes it possible to hold login credentials between browsers, making the crawling process more efficient as the login step needs to be executed only once. In the context of privacy measurements, cookies set in the browser will remain during the crawl execution. This makes it possible to identify trackers for which techniques are used, such as cookie syncing, cookie respawning [AEE+14], and replication of user profiles [ERE+15]. On the other hand, stateless crawls do not maintain a state by which a browser always appears to be a new user. Observation of previously mentioned techniques is therefore difficult. Currently, OpenWPM does not support the usage of stateful crawls. However, this does not thwart the purpose of our research. As we examine each website separately to observe our defined compliance level, no state has to be maintained during our crawling process. We can monitor the cookies set for tracking purposes within each browser instance.

3.2. DATA PROTECTION LAWS

As early as 2000, Peng et al. discussed the concerns around the emerging technology of cookies [PC00]. They described that if users give away personal information, e.g., through form inputs, and the data is saved into a cookie, it could be used as tracking technologies. The researchers concluded that users need to be aware of cookies and know how to refuse or delete them. A year later, in 2001, Millett et al. provided criteria for assessing online informed consent [MFF01]. At that time, some browsers provided options for users to handle cookies. E.g., Figure 3.1 depicts the cookie settings offered by Netscape Navigator 4.0,

a browser project that closed in 2008, by which users could accept only those cookies sent from the originating server. Further, in the same time area, Friedman et al. proposed a conceptual model of informed consent [FFM00]. Their model was based on five components: disclosure, comprehension, voluntariness, competence, and agreement. They stated that some websites manipulated users by bombarding a user with information about cookies. One effect was that a user would fail to notice a cookie that he or she might want to avoid.

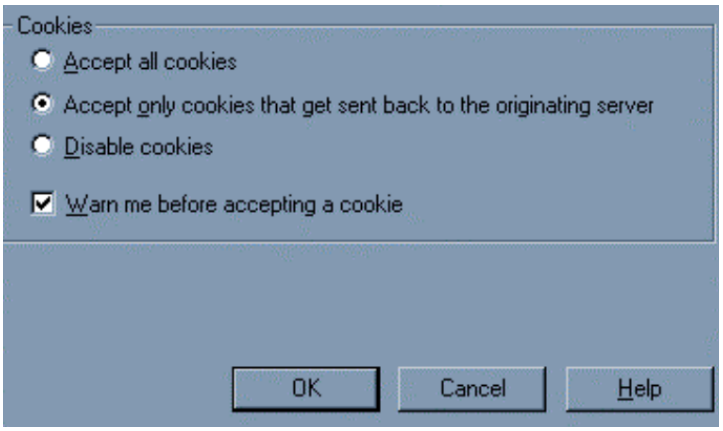


Figure 3.1: Netscape Navigator 4.0 cookies configuration options (adapted from [MFF01]).

In recent decennia, several laws have emerged to protect users' data. In 2002, the Directive on Privacy and Electronic Communications (ePD) was enforced on European grounds. It is mainly concerned about how telecom companies and Internet Service Providers (ISPs) manages users' data. Further, Article 5(3) of the directive states the following:

Member States shall ensure that the storing of information, or the gaining of access to information already stored, in the terminal equipment of a subscriber or user is only allowed on condition that the subscriber or user concerned has his or her consent, having been provided with clear and comprehensive information, in accordance with Directive 95/46/EC, inter alia, about the purposes of the processing.[1]

An important term in the description is 'consent'. By this, publishers are obliged to ask users' consent before storing information used for non-essential purposes. However, a directive does not impose how the rules have to be implemented. Each European member state implements the details of the directive into national laws. The exact implementation can therefore differ between different member states.

More recently, the General Data Protection Regulation (GDPR), applicable as of May 25th, 2018, complements the ePD. The GDPR is mainly concerned with the processing of personal data, while the ePD elaborates on the GDPR regarding electronic communications. Cookies are mentioned only once in the GDPR, but companies that use them to track users' browser activity risk significant fines. Recital 30 of the GDPR states:

Natural persons may be associated with online identifiers provided by their devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other

[1] Article 5(3) ePrivacy Directive

identifiers such as radio frequency identification tags.[2]

This means that when cookies are related to an individual, it is considered as personal data, and therefore falls under the data protection rules, as mentioned in Recital 26:

The principles of data protection should apply to any information concerning an identified or identifiable natural person.[3]

These regulations are the origin of cookie dialogs that users currently encounter daily when browsing the internet. If website publishers use cookies for non-essential purposes, i.e., cookies that are not necessary for the main functionality of the website, they have to inform the user unambiguously about which cookies will be set on the user machine. Further, users must have the option to reject or accept the use of such non-essential cookies. The distinction between cookies used for essential and non-essential purposes can be described as follows[4]:

Essential cookies:

- used solely to carry out or facilitate the transmission of communications over a network, or
- strictly necessary to provide an online service that users have requested.

Non-essential cookies:

- these are any cookies that do not fall within the definition of essential cookies, such as cookies used to analyze your behavior on a website ('analytical' cookies) or cookies used to display advertisements ('advertising' cookies).

Besides the previous distinction on the purpose of a cookie, other classifications exist. First, we can distinguish a cookie based on its duration. Cookies that only exist temporarily during a browser session, i.e., deleted after closing the browser, are called *session cookies*. On the other hand, *persistent cookies* are saved on the user machine even after the browser is closed. To erase these cookies, users have to delete them from their machine manually. Although these types of cookies also have a duration, they often exist for a long period without manual intervention. Secondly, we can categorize cookies based on their origin. *First-party cookies* are set by the website a user visits. Contrary, *third-party cookies* are set by another website and therefore have another origin than the website a user visits. This last category is often linked with cookies used for marketing and tracking purposes since the same third party can be connected to many different websites.

Still in progress is the ePrivacy Regulation (ePR), which will replace the ePD to simplify and strengthen the privacy rules. E.g., it proposes that no consent will be needed for non-privacy intrusive cookies. Also, users will be able to use browser settings to disable

[2]https://www.privacy-regulation.eu/en/recital-30-GDPR.htm
[3]https://www.privacy-regulation.eu/en/recital-26-GDPR.htm
[4]https://gdprprivacypolicy.org/cookies-policy/#toggle-id-2

online tracking. Until then, explicit consent from the user is required. A foreword by the European Data Protection Supervisor (EDPS)[5], states that although the current ePD already provides individuals with a degree of protection in the context of electronic communications, its scope and provisions no longer ensure a sufficient degree of protection. The rapid advancements in technological developments outdated the provisions of the ePD, which did aspire to be technologically neutral.

Further, a European body called the European Data Protection Board (EDPB) exists as a central EU entity to safeguard the data protection laws. Their mission is to ensure consistent application in the EU of the GDPR and the Law Enforcement Directive (LED)[6]. LED is another legislation that handles the protection of personal data that falls outside the scope of the GDPR, namely the processing of personal data in law enforcement, which is explicitly excluded from the GDPR[7]. However, Leiser et al. [Lei19] conclude that LED has limited transparency requirements and lower standards for protecting data subject rights than the GDPR. The EDPB mainly provides general guidance to support comprehension of these data protection laws and ensures correct application of it in individual cases. However, the EDPB does not enforce EU data protection laws. Their members[8] include the Data Protection Authorities (DPAs) of the EU countries and the European Data Protection Supervisor (EDPS). The EDPS is an independent EU data protection authority. They have the powers of investigation, corrective powers and sanctions, and authorization and advisory powers. In particular, in case of complaints from individuals, they have the authority to bring infringements to the attention of the Court of Justice and powers to engage in legal proceedings in accordance with the primary law[9].

3.3. Transparency and Consent Framework

The Interactive Advertising Bureau (IAB) of Europe supports advertising businesses and website publishers by developing certain standards that should help the community to be compliant with the regulations. The Transparency and Consent Framework (TCF) is such a framework the IAB introduced. This framework establishes a set of global technical standards to establish trust and transparency between all involved actors. According to their own declaration, TCF is the only GDPR consent solution built by and for the industry, creating a true industry-standard approach[10]. Figure 3.2 depicts the actors and how they relate to each other. One of the actors is a Consent Management Provider (CMP), a central unit that website publishers can use to collect users' consent (1). The user's consent, together with its selected purposes, is stored in a consent string, which the CMP further processes (2). Then, consent is further transferred to vendors that are registered with the IAB TCF (3). Lastly, an advertiser can collect the user data to display specific ads on the website (4).

CMPs that register with TCF are required to take and pass the CMP validator, developed

[5]https://edpl.lexxion.eu/data/article/11389/pdf/edpl_2017_02-006.pdf
[6]https://edpb.europa.eu/about-edpb/about-edpb/who-we-are_en
[7]https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679 - Article 2(2d)
[8]https://edpb.europa.eu/about-edpb/about-edpb/members_en
[9]https://edps.europa.eu/frequently-asked-questions_en
[10]https://iabeurope.eu/transparency-consent-framework

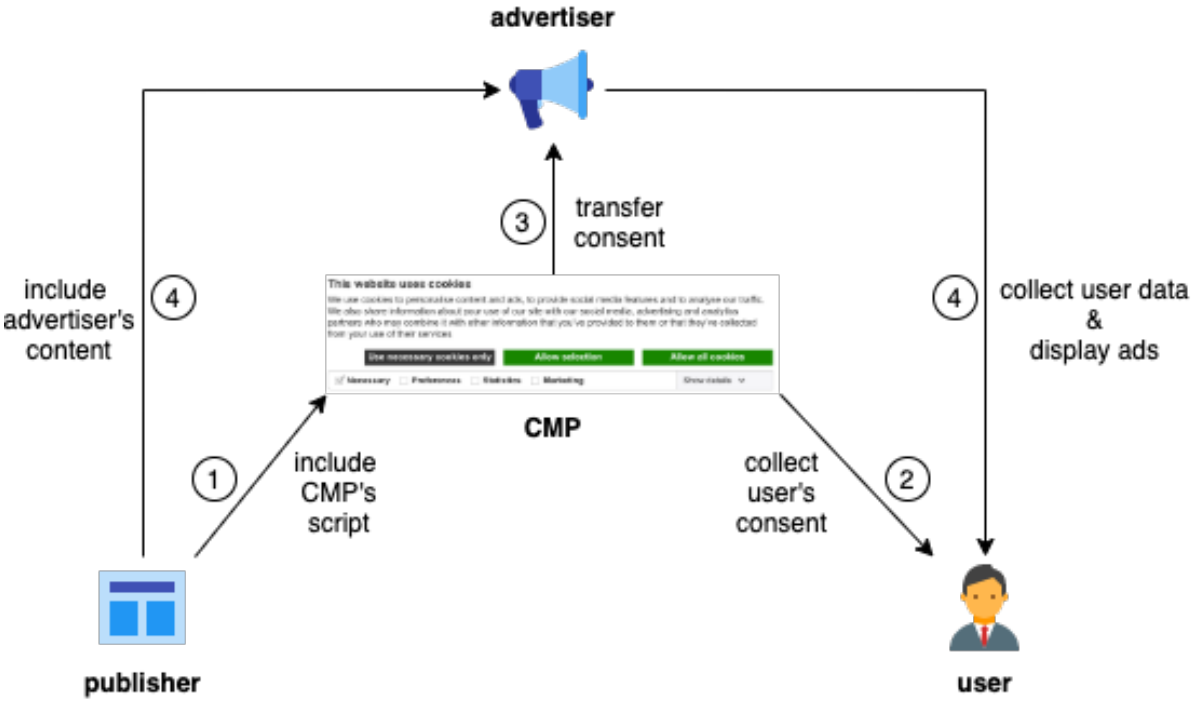


Figure 3.2: The actors and activities involved when using a Consent Management Provider (CMP) under IAB Europe's TCF (adapted from [MBS20]).

by IAB Europe, to verify that the implementation of the CMP is compliant with the technical specifications[11]. When validated and approved, a unique ID is assigned to the registered CMP and added to the IAB CMP list[12]. This validation process is repeated each year for every registered CMP. Vendors can participate in TCF by filling in a specific registration form where they need to declare the purposes for which personal data will be processed. In addition, they need to indicate on which legal basis they rely to process the collected data. Besides relying on consent, vendors can choose to rely on legitimate interest, which means they can freely use the collected personal data in a way the user would expect. It is the most flexible condition under the GDPR for processing personal data. However, the EDPB stated that the legitimate interest condition is not an appropriate lawful basis for processing personal data related to tracking, profiling, and advertising[13] [14]. Research performed by Santos et al. [MSB20] showed that hundreds of advertisers rely on legitimate interest for purposes that instead should rely on consent. Such statements and research findings do not support the IAB declaration of enhancing transparency in the community. In recent years, TCF has undergone several updates. Figure 3.3 depicts the two major versions and their related purposes[15]. As of version 2, vendors need to indicate a legal basis for 12 purposes. Most of these added purposes originate from a purpose of version 1. Two special purposes are

[11] https://iabeurope.eu/tcf-for-cmps
[12] https://iabeurope.eu/cmp-list
[13] "Opinion 03/2013 on purpose limitation (WP 203), adopted on 2/04/2013" https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2013/wp203_en.pdf
[14] "Opinion 06/2014 on the notion of legitimate interests of the data controller under Article 7 of directive 95/46/ec (WP 217)" https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp217
[15] https://iabeurope.eu/wp-content/uploads/2019/09/TCF-v2-Webinar-Publishers-webinar.pdf - slide 20

added to the new version, which vendors can use to provide technical functionalities.

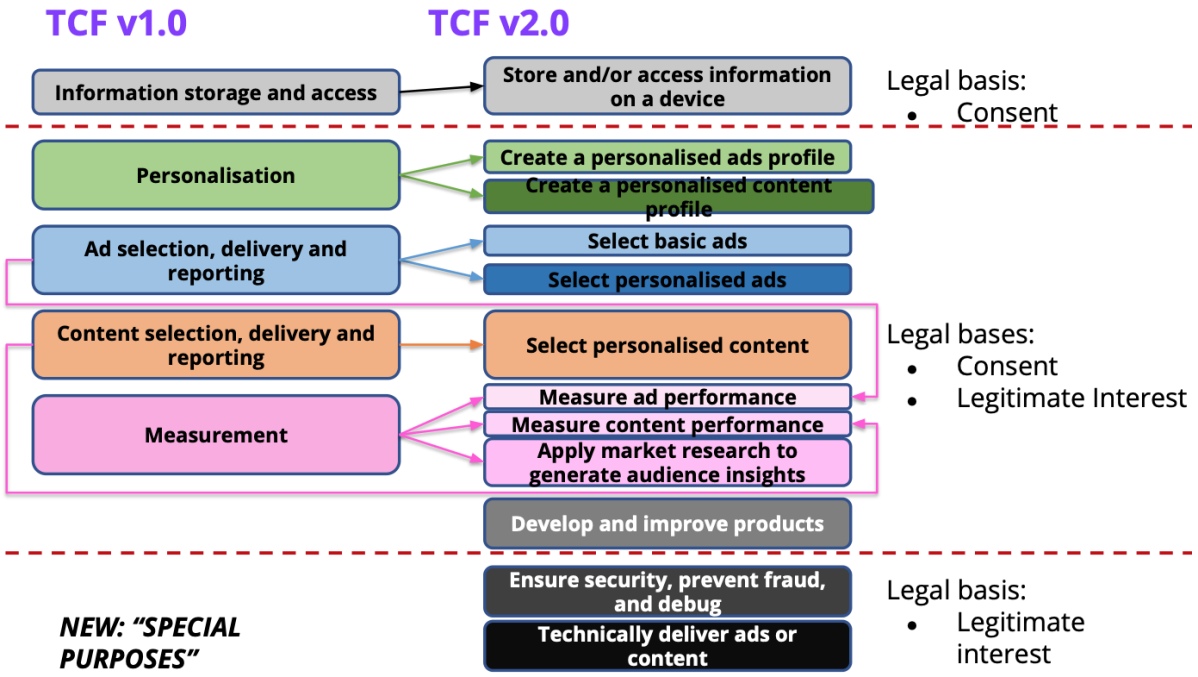


Figure 3.3: Purposes of TCF version 1 and version 2.

4

RELATED WORK

4.1. LEGAL COMPLIANCE

In recent years, numerous studies have been conducted to examine the possible infringements of cookie banners. Matte et al. investigated the use of banners from IAB Europe's Transparency and Consent Framework (TCF) [MBS20]. The authors observed the availability of an opt-out option, and if a users' choice is respected. Therefore, three computer scientists performed manual checks to confirm violations. Their findings conclude that 6.8% of their dataset does not provide a refusal option. They identified violations by also examining available options in a possible second layer of a banner. Our research does not perform such manual checks as our goal is to provide an automated process. In RQ4, we observe if consent and reject options are available in the first layer, as these should be offered in the same way [SBM19]. Further, the authors notice that some vendors associated with the IAB framework use trackers, even when a user refused consent. The authors acknowledge that their study examined IAB Europe TCF version 1. Our research is based on TCF version 2. However, we do not examine consent strings of IAB to identify for which purposes consent is given.

The same researchers, Matte et al., analyzed the purposes for data collection in more depth which advertisers involved in the TCF can employ [MSB20]. The two main legal bases for processing data are consent or legitimate interest. By analyzing the purposes declared by the registered advertisers, they reveal that hundreds of advertisers rely on legitimate interest for purposes that should rely on consent. A positive evolution is that the choice of legitimate interest is slowly decreasing. These findings mainly cover version 1 of TCF. The authors note that version 2 could become more popular as Google's CMP will adopt this version. Automating their analysis to identify when advertisers use a purpose without legal grounds is a difficult task as legitimate interest is a legal basis, and the actual purpose for data processing can be hard to uncover. Therefore, our research does not cover such purpose analysis.

In other research, Santos et al. propose 22 requirements that cookie banners must meet to be fully compliant with the GDPR and ePrivacy Directive guidelines [SBM19]. They used

a bottom-up approach to start from the legal sources to formulate the requirements. They tested each proposed requirement manually on several websites and defined possible violations. E.g., their first requirement states that consent must be obtained before a user identifier is stored. Manual tests show that even well-established companies fail to address such a basic requirement to protect users' data. Other requirements mostly face the same conclusion. Many websites seem not to apply the authors' proposals. However, some requirements are subject to interpretation, e.g., a user should have a balanced choice to either accept or reject a consent. Because there is no defined standard in cookie banner design, it is complicated to draw clear conclusions. Their study's main contribution is the link to actual legal sources, which could wake some legal auditors. Our research extends their work, as we cover some elements defined by their legal requirements and examine the integration into an automated process. Rule 13 of their defined requirements states a banner must present a fair or balanced design choice. This requirement forms the basis for our analysis of RQ4, where we identify consent and reject elements to identify a dark pattern. The authors state that it is not possible to verify this requirement automatically because of the lack of standards in cookie banner design. However, we consider that some design choices nudge users to consent and can relatively easily be identified by an automated crawler.

Another study by Trevisan et al. analyzed a large dataset of websites with an automation tool to check the compliance of cookie banners to the ePrivacy Directive (ePD) [TTBM19]. Their analysis looked for a difference in the domain of the installing cookie and the visited website. If at least one profiling cookie is found, the website is marked as violating the ePD. Results show that 49% of the tested websites do not respect the ePD and install profiling cookies before any users' consent is given. In this complex ecosystem dominated by advertisers and where web services need to monetize the content they offer, it is still difficult for legislators to regulate online privacy. Therefore, the authors conclude that the enforcement of the ePD is a failure. However, this study is conducted just after the introduction of the GDPR. So actual improvements due to these added legislations are not evaluated. As our research is performed three years after the GDPR enforcement, website publishers have had the time to adapt according to the legislation. Also, our method to detect violations differs as our implementation is based on OpenWPM. We observe the cookies set and discover their purpose by using an open knowledge base.

4.2. Dark patterns

Soe et al. and Nouwens et al. examined the use of so-called dark patterns that mislead and nudge users into giving consent [SNGS20, NLV+20]. The first study manually tested 300 websites to identify the use of dark patterns in cookie banners. Besides data collection before a users' consent, which they also identify as a dark pattern, the authors describe other new dark patterns based on existing ones examined in other studies. E.g., the use of "multiple choice panels" is identified on several websites where the reject option is given in only one smaller panel. To verify their observations, they conducted two studies. One in 2019 and the second in April 2020, in which they revisited the 300 websites. One of the results shows that a user needs an average of 10 clicks to opt-out of all cookies. Further, the use of more privacy-friendly words, such as "deny" or "reject", are not commonly used. Also, the purpose of the cookies is only given for 125 out of the 300 websites. However, as their data

set only consisted of news outlets and magazines, a more varied set of websites could yield different results. Our research uses more varied datasets, i.e., the top 500 websites of all European countries filtered from the global Tranco list. The authors urge the need for standard terminology so that users clearly understand the consent information. Also, to make an audit more feasible and reliable, regulators should rely on some automatic process to flag violations to increase compliance, which we investigate in our research by focusing on automation.

The second study by Nouwens et al. investigated interface designs of Consent Management Providers (CMPs) regarding their compliance with the EU's General Data Protection Regulation and how these affect users' consent actions [NLV+20]. Several variables from the design interface were collected, such as the CMP vendor and the presence of accept and reject buttons. They looked for an identifying HTML element to determine the presence of a particular CMP. In comparison, in RQ1, we detect a CMP by performing the ping operation. In extension, we do not only examine the presence of a CMP. In RQ2, we also identify if a cookie dialog is displayed when visiting a website. Nouwens et al. used three measurable requirements to evaluate if a certain provider is compliant or not. These requirements alone are not sufficient to be legally compliant but can be measured in a quantitive manner. Their results show that of the 680 websites, 16,8% records consent when the user visits the website, and more than 50% did not have a button to reject all purposes. We also take into account the native language of each country of our dataset to observe the presence of accept and consent buttons to expand our search field. Secondly, they examined how 40 participants interacted with consent pop-ups by using a browser extension. Interesting to notice is that by removing the "reject all" button, the probability of consent increases with more than 22 percentage points. As the participants were all computer science department members, a more general public could produce other outcomes. Further, the study by Mathur et al. investigated the use of dark patterns in shopping websites and taxonomized these patterns based on several characteristics [MAF+19].[1]

The importance of a transdisciplinary approach to improve current legal guidelines on the use of dark patterns is discussed by Gray et al. [GSB+20]. They reviewed recordings from over 50 sites and analyzed the design and users' means of interaction with the consent banners on these websites. For every phase they identified in the consent task flow, outcomes were analyzed from a designer, user, interface, and social impact perspective. E.g., the authors identify the use of a tracking-wall, which is an instance of a consent wall but with a more aggravating element. Besides blocking access to the website, the user has only one option, namely, to accept. Such walls make it difficult for a user to make a specific and informed consent. The authors mention that although these outcomes are clear from a legal perspective, it is difficult to indicate which design elements are lawfully suitable. There is a need for standardization as there are no uniform design requirements to check compliance with the GDPR. Therefore, they advocate for combined research between legal, ethics, and Human-Computer Interaction. However, detecting all design requirements automatically for compliance will be difficult as several are interpretive and need different combined methods, such as manual analysis.

[1] Overview of the findings of "Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites", WebTap Princeton University, July 17, 2019, https://webtransparency.cs.princeton.edu/dark-patterns

4.3. CRAWLING

The process of crawling to collect artifacts and specific data automatically from websites is a popular method to broaden the range of analysis and is often used in research. Matte et al. built a crawler, namely Cookinspect, based on a Selenium instrumented Chromium [MBS20]. Thereby, they could perform a fully automated scan to detect whether a website uses a TCF banner. During the scan, they also checked the storage of positive consent and the existence of third-party tracking requests before any user action. Further analysis showed that 46.5% of the websites pre-tick purpose options, and 5.3% set a consent string without respecting a users' refusal. Trevisan et al. also implemented a tool called CookieCheck to perform an automated scan [TTBM19]. Nouwens et al. developed a web scraper to collect data about the top 5 CMPs using Alexa's 10,000 most popular websites in the UK to conduct their study [NLV+20]. The authors conclude that automated tools could support data protection authorities in enforcing legislation. By employing the string methods and additional information gathered from web pages, Uzun et al. uses an approach called UzunExt [Uzu20]. This approach uses additional information to improve a crawler's execution time, which is an often ignored property in research. In our research, we use a similar method as the aforementioned studies to provide automation. Although we do not build our crawler from scratch, we use OpenWPM, an open-source automation project specifically focused on privacy studies, to perform our analysis. Using this tool gives us the advantage of focusing our implementation on our specific research needs as OpenWPM provides several built-in data collection hooks.

Another study by Jonker et al. investigated the extent to which bot detection occurs, which can severely impact the outcomes of research findings [JKV19]. The authors used fingerprinting techniques that rely on browsers' properties and the DOM model to establish a so-called fingerprint surface of a web bot. A scan of the Alexa top 1 million revealed that many of the top sites use PhantomJs-detection. These websites can distinguish a regular website visit by a user from a visit performed by a crawler that uses the PhantomJs browser. Less common is the use of client-side detection for bot detection. They also examined if sites that employ bot detection act differently to users. Four types of different behaviors were discovered, e.g., for a selection of 20 websites, 12 websites present different content to a web bot. Their main finding is that PhantomJs is highly detectable as a web bot. 12% of the top 1 million Alexa sites detect the use of PhantomJs. Our implementation of OpenWPM does not use PhantomJs for browser automation. In order to decrease bot detectability, we use a headful browser based on Firefox Nightly.

5

METHODOLOGY

In this section, we discuss general methodology topics related to all research questions that we will cover. As each of our research questions uses a specific methodology, we discuss further specifics in the relevant sections.

5.1. RULING SYSTEM

A crucial element in our research is how we determine if a website is compliant or not. Our ruling system needs to be based upon existing laws. The extent and detailed descriptions of current laws require specific knowledge and expertise in the field of jurisdiction. In our field of software engineering, we do not possess this expertise. That is why a transdisciplinary approach is necessary to tackle our research topic, an observation that is previously noted by Gray et al. [GSB+20]. However, this does not mean we cannot make valid observations about the outcomes of our scanning output or by performing an audit on a limited set of websites. We consider one rule of the ePD as fundamental. I.e., the obligation that websites must obtain users' informed consent before using any kind of tracking technology. The definition of "informed" and its exact meaning requires expertise in the field of jurisdiction, which we consider a grey area. To analyze the existence of such an informed consent, our scanner would have to support linguistic analysis to identify certain semantic properties. This is not in the scope of our current research. Therefore, we exclude the terminology "informed" from our rules. Further, there exist many tracking technologies that put users' privacy at risk. Table 5.1 shows a list of privacy threats examined by an analysis conducted by Estrada-Jiménez et al. [EPRF17]

However, our research focuses on the usage of cookie dialogs. Therefore, our main concern is when cookies are set and for what purpose. Further, Santos et al. analyzed the legal and technical requirements of consent dialogs under the GDPR and ePD. They identified 22 requirements from legal sources and both technical and legal experts to verify compliance of cookie dialogs [SBM19]. We distill one specific requirement, R13, from their research: website publishers need to offer a balanced consent choice. This means that users need to be able to reject consent as easily as giving consent. The researchers describe that

Privacy threat
First-party tracking
Third-party tracking
Cookie matching
Fingerprinting
Flash cookies
Canvas fingerprinting
HTML5 local storage

Table 5.1: List of privacy threats examined by Estrada-Jiménez et al. [EPRF17].

the validation assessment needs to be performed fully manually as currently there are no standards in cookie banner design. We think this requirement is a good starting point for our examination. I.e., to detect a limited set of design elements that violate the regulations automatically. Although Santos et al. discuss other requirements, we select R13 from their research as others can already be partially automatically assessed. We define the rules outlined in Table 5.2. Detecting a violation of one or more of these rules results in a non-compliant website.

Rule	Description	Origin
R1	Cookies for targeting/advertising purposes must not be set before user consent	ePD/GDPR
R2	Balanced consent and refuse choices	Requirements from Santos et al. [SBM19]

Table 5.2: Defined ruling system used for our research.

5.2. Extending OpenWPM

To perform our defined compliance checks in an automatic process, we use a web crawler. I.e., a bot that can navigate to specified URLs to collect data of the visited website. In 2016, Englehardt et al. implemented a tool named OpenWPM, specifically aimed at web privacy measurement research [EN16]. OpenWPM is built on top of Firefox and uses Selenium[1] to enable browser automation. To perform data collection, several hooks to instrument the automation are made available. OpenWPM is an open-source project and can therefore be easily extended[2]. We use this tool as the basis of our crawler implementation. Figure 5.1 shows the high-level components that are part of OpenWPM.

The task manager distributes commands to the browser managers, which on their terms, provide an abstraction layer for automation of each browser instance. The code snippet in Listing 5.1 appends the GET command to a sequence of commands, which are then passed on to the task manager, who will distribute the given commands to the browser instances.

[1] https://github.com/SeleniumHQ/selenium
[2] https://github.com/mozilla/OpenWPM

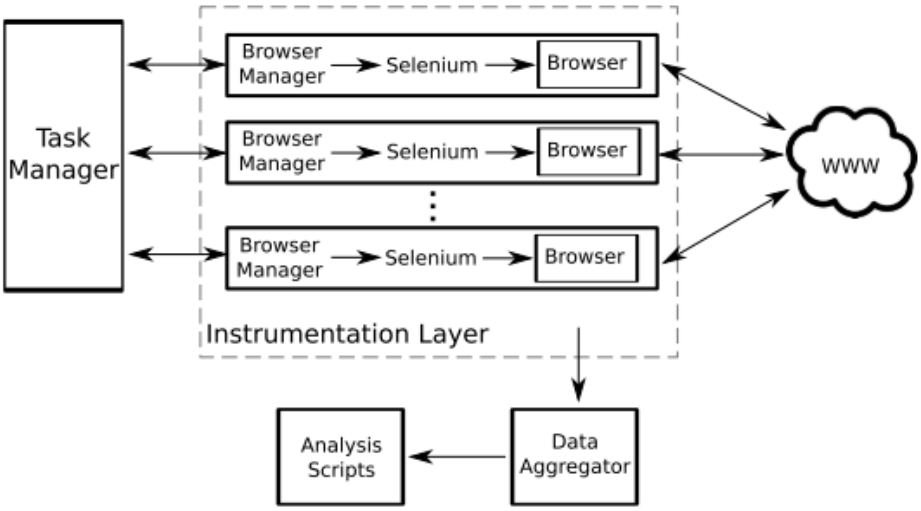


Figure 5.1: High-level components of openWPM [EN16].

Listing 5.1: GetCommand appended to the command sequence.

```
command_sequence.append_command(GetCommand(url=site, sleep=3), timeout=60)
```

After Selenium visits the provided URL, the depicted GET command will wait for 3 seconds. The timeout indicates that if Selenium can not reach the website, an error will occur. The number of browser instances that run simultaneously can be set through a config parameter. The data aggregator module forms another abstraction layer for the browser instrumentation. Our implementation extends the default OpenWPM task manager to provide extra commands to the browser managers. OpenWPM is built with Python and Python libraries. Our extended version uses OpenWPM v0.13.0[3]. At the time of writing, a new major release, v0.14.0[4], is released. This new version contains several major refactorings that could break our extended implementation. Further, currently unknown or untested side effects could output different results. Therefore, our implementation will continue to use v0.13.0.

Some websites employ bot detection mechanisms, which may result in different content being shown to the visitor, restriction of resources, or exclusion from a website[JKV19]. Such techniques can put a restraint on the results of our study if the bot does not simulate an actual website visit. OpenWPM provides an optional browser parameter that can be set to mitigate employed bot detection techniques. Table 5.3 lists the three mitigation mechanisms used by OpenWPM during the execution of the GET command. We ran our crawler against the top 100 websites of France with bot mitigation disabled and enabled. Comparing the output results showed that there is no significant impact when performing a crawl without bot mitigation.

[3]https://github.com/mozilla/OpenWPM/releases/tag/v0.13.0
[4]https://github.com/mozilla/OpenWPM/releases/tag/v0.14.0

bot mitigation 1	move the cursor randomly around a number of times
bot mitigation 2	scroll in random intervals down page
bot mitigation 3	randomly wait so page visits happen with irregularity

Table 5.3: Optional built-in bot mitigation techniques performed by OpenWPM.

5.3. DATASET

For this research, we use a list of top websites that serves as the input for our crawler to perform the analysis. There are several top website lists available. E.g., Amazon provides a Software-as-a-Service named Alexa, which is based on its Alexa Traffic Rank data[5]. Alexa gathers its data by providing a browser extension[6] which extracts web visits from the users. Their website list is based upon the number of users that install and enable the browser extension. However, Le Pochat et al. [PvGT+19] showed in their research how easily the Alexa website ranking can be manipulated through the use of the browser extension. Even a small number of page visits leads to a substantially higher ranking. This way, adversaries can manipulate the ranking with minimal effort. Their research showed that other often used top website lists such as Cisco Umbrella, Majestic, and Quantcast are also susceptible to manipulation. When such lists are used in research studies, it affects the results, and any biases may hinder correct conclusions. Also, a portion of the top website lists are unreachable or do not respond with a successful status code. Le Pochat et al. identified that only 49% of the Umbrella list websites responded with a status code 200, and 30% returned a server error. Therefore, the researchers produced a top website list named Tranco, based upon the rankings of Alexa, Umbrella, and Majestic. Tranco has several built-in manipulation countermeasures. First, domains that appear only in one or a few lists are filtered out, as this is recognized as an isolated manipulation attempt. Secondly, potential malicious domains can be removed using the Google Safe Browsing list[7]. Further, it obtains one million domains by averaging all three rankings over the past 30 days, thereby reducing the effect of a possible manipulated list. All three rankings need to be manipulated to the same extent to influence the combined Tranco list. Also, it is possible to retrieve a past version of the standard Tranco list, which enables reproducibility. Other rankings change daily, making it difficult to retrieve a list from the past to reproduce a certain result. Tranco is an open-source contribution and therefore freely available for anyone[8]. The researchers provided a webpage with many configuration options to generate a Tranco list[9].

Our research uses generated Tranco lists with a specific configuration. First, from all the available combined domains, we only retain the Pay-Level Domains (PLDs). A PLD is a subdomain of a top-level domain (TLD) by which we can identify a single user or organization that controls the website. Figure 5.2 shows the different parts of an URL and their naming.

[5] https://www.alexa.com/topsites
[6] https://chrome.google.com/webstore/detail/alexa-traffic-rank/cknebhggccemgcnbidipinkifmmegdel?hl=nl
[7] https://safebrowsing.google.com
[8] https://github.com/DistriNet/tranco-list
[9] https://tranco-list.eu

Notice that a TLD can contain another subdomain. E.g., the United Kingdom uses "co.uk" where "co" is a subdomain of the TLD uk. Therefore is not a PLD.

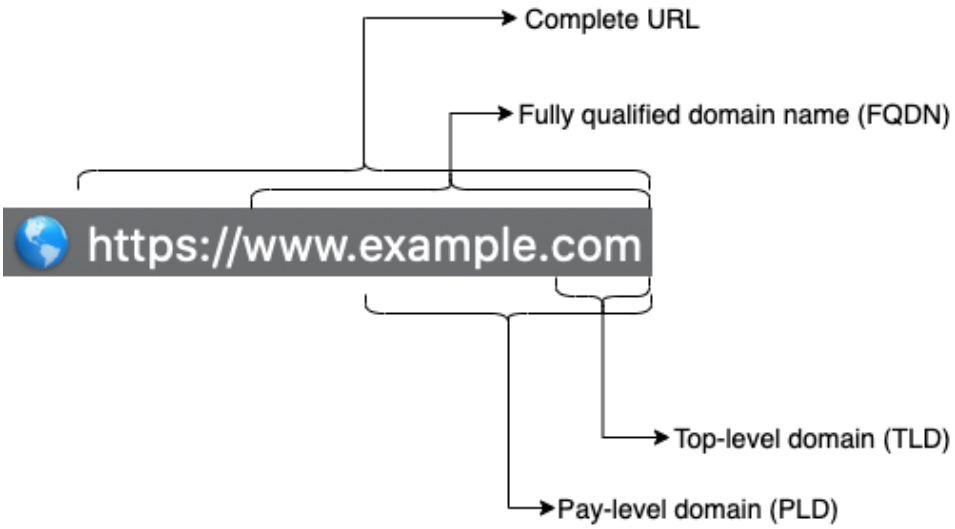


Figure 5.2: The different parts of an URL and their naming.

Next, for each country that is part of our test list, we use the country code top-level domain (ccTLD), a TLD reserved for countries. The ccTLD is used as a filter to only obtain websites of a certain country. Table 5.4 shows three ccTLDs used for our research as a filter to get our data sets.

Country	ccTLD
The Netherlands	nl
Belgium	be
France	fr

Table 5.4: Partial list of countries with their associated ccTLD used as a filter to generate datasets with Tranco.

Further, domains flagged as dangerous by Google Safe Browsing are removed from the generated data set. This decreases the chance that a website with malicious content is visited during an automatic crawl, possibly infecting the system with malware or compromising data. Although we run our crawler in an isolated environment, we want to reduce possible overhead due to an infected system. Also, our generated data sets have a unique ID. Other researchers can use our data sets for their study or to reproduce certain results we present here[10]. Therefore, we should mitigate possible threats as we have no control over the environments where these data sets might be used.

Tranco generates the combined list using the Dowdall rule. The first domain gets 1 point, the second 1/2 points, and the last 1/N points, which results in a statistical distribution of website popularity called the long-tail effect. A low number of websites have a high visitor rate. On the other hand, there are a large number of rarely-visited websites.

[10]https://github.com/koenae/openwpm-crawler/tree/master/datasets

The former is shown by the curve's steep shape, while the latter forms the long tail. Figure 5.3 shows an example of such a distribution. Kumar et al. examined website popularity by measuring a number of website requests, taking into account the location and time of the requests, which resulted in a similar long-tail distribution [KNS09].

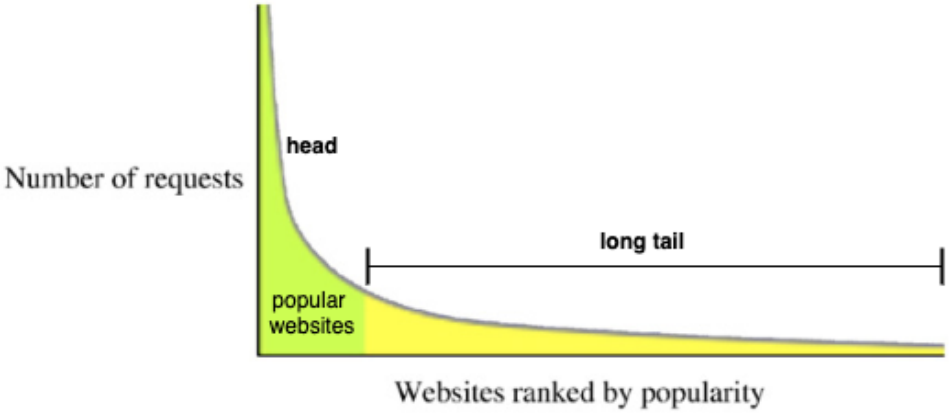


Figure 5.3: Popularity of websites depicted as a long-tail distribution (modified from [KNS09]).

Applying our filters to the aggregated Tranco list for each country, we take the top 500 domains. Alternatively, we could spread our set of 500 websites per country evenly over several segments. E.g., for each successive 1000 websites of the Tranco list, we could take 100 websites until we have a similar subset of 500 websites. This way, our subset would contain more websites that are infrequently used. However, the websites of the top 500 have the most impact on the internet landscape. In correlation, this means that our results have a higher impact, as even minor compliance violations affect a large user base.

5.4. System Setup

First, we ran our extended implementation of OpenWPM on a Virtual Machine (VM) with Ubuntu version 18.04.5 LTS. Table 5.5 lists the configuration of the VM.

Setting	Value
Memory	3.8 GB
Processor	Intel® Core™ i7-4870HQ CPU @ 2.50GHz × 2
OS type	64-bit
Virtualization	KVM

Table 5.5: Our VM configuration running on Ubuntu.

Running multiple crawlers on one VM is possible. However, the browser processes of each crawler instance sometimes conflict with each other, which increases the number of incomplete visits. Therefore, using this system setup is not scalable. We migrated our VM to Proxmox[11], an open-source virtualization platform where you can quickly scale up re-

[11] https://www.proxmox.com

sources and VMs. Our Proxmox server has 32 GB memory and 8 CPUs available. We configured and converted one VM as a template so that we can easily build a new VM by using the template without having to reconfigure all settings. Figure 5.4 shows the server view of our Proxmox installation with six crawler VMs that we can run simultaneously. The last instance depicts our created template.

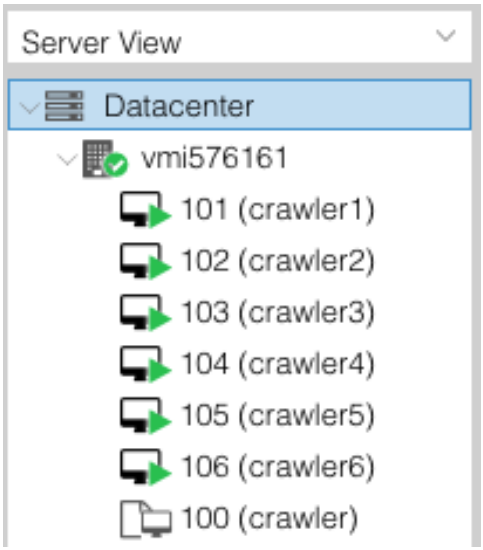


Figure 5.4: Our Proxmox crawler setup.

5.5. Ethical considerations

The use of a crawler to collect data from websites is subject to some ethical aspects. Thelwall et al. [TS06] reviewed several moral issues to provide a new set of guidelines for web crawlers. They identified types of problems that can arise for society or individuals, which we will use here for our discussion relating to our research.

Denial of service. Thelwall et al. [TS06] mention that when a server is busy responding to robot requests, it may slow down response times to other users. Such behavior would interfere with the primary purpose of a website to deliver its online content and service to users. On today's expansive internet, websites may need to process a high number of requests. We ran our crawler on a maximum of five hundred websites per European country, which is only a small portion of the available websites. Further, we only visit the homepage of a website to collect data and DOM elements of the content. Our crawler does not perform navigation steps to other URL paths within the same website. Therefore, the impact of the website visit we perform automatically is limited. Although our implementation could contain errors we are not aware of, we did not encounter errors during the execution of the crawler that indicated a denial of service. Also, we ran our crawler only a few times per research question with a significant time interval in between.

Cost. As the bandwidth of web hosts has its limitations, exceeding it can incur costs for website owners. Thelwall et al. [TS06] noticed that consequences range from automatically paying the excess cost to disabling the website. However, our crawler does not download full pages of content. As we are only interested in selected elements from a cookie dialog, our bandwidth usage is limited. However, future extensions of our implementation would have to take into account such limitations.

Privacy. The internet is a public domain. Therefore, viewing and downloading the content of a website automatically by a crawler does not invade privacy. The same actions can be manually performed, which means regular users are subject to similar privacy principles. Although a crawler could be implemented to visit specific URL paths which are not visible on the website, our implementation only detects cookie dialogs and related artifacts on the homepage. Also, we do not store most of the data we find on websites, only specific elements from a cookie dialog needed to detect violations.

Copyright. Making permanent copies of copyright material, i.e., web pages, without permission of the publisher is an illegal practice [TS06]. E.g., the Internet Archive[12] stores web pages to make them publicly freely available. They use an opt-out policy by which owners can keep their content out of the archive. We acknowledge that we do not make it possible for websites to opt out from the privacy research we perform. However, we do not make our saved data publicly available. Only our analysis results for which we filter the gathered content can be openly viewed. Further, we do not take screenshots of the cookie dialogs we detect or download full website pages.

Robots Exclusion Protocol. Publishers can use the Robots Exclusion Protocol[13] to prevent a crawler from navigating to certain paths of a website. Therefore, they need to place a file named robots.txt with instructions to prevent crawlers from scanning certain areas. Listing 5.2 shows the robots.txt file from the OU website[14], which displays instructions to disallow a crawler from scanning specific paths.

Listing 5.2: Robots.txt file from `https://www.ou.nl/robots.txt`.

```
User-Agent: *
Disallow: /web/studieaanbod/
Disallow: /web/informatica/
Disallow: /web/cultuurwetenschappen/
Disallow: /web/managementwetenschappen/
Disallow: /web/natuurwetenschappen/
Disallow: /web/psychologie/
Disallow: /web/rechtswetenschappen/
Disallow: /web/nieuws-en-agenda/
Disallow: /web/studeren/
```

[12] https://archive.org
[13] https://www.robotstxt.org/norobots-rfc.txt
[14] https://www.ou.nl/robots.txt

```
Disallow: /web/over-ons/
Disallow: /web/senior-digi-vaardig/
Disallow: /*p_p_id
Disallow: /*p_auth
Disallow: /*redirecter
Allow: /*p_p_id=UserProfileViewer_WAR_userprofileportlets_cws
Allow: /*p_p_id=UserProfileViewer_WAR_profieldienst
Sitemap: https://www.ou.nl/sitemap.xml
```

Thelwall et al. [TS06] mention that ethical crawlers will read the instructions and obey them. However, our crawler does not scan available robots.txt files. Presumably, website publishers want to ban crawlers from specific paths, as shown in the previous Listing 5.2. On the other hand, we only visit the homepage of a website. Also, our crawler is currently used to only perform specific needs for our research. However, if our crawler would be used on a wider scale, extracting the robots.txt instructions should be considered. Further, as this is an extra step in the crawler process, it would increase the running time.

6

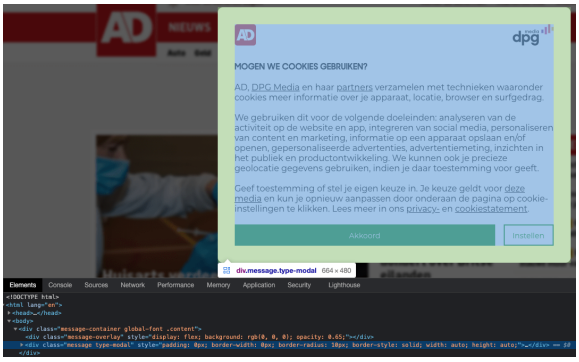
RQ1: To what extent do websites offer a cookie dialog?

The wide range of different implementations makes it a difficult task to detect whether an element on the website is a cookie dialog or not. Figure 6.1a depicts the HTML div element used for the cookie dialog of the website www.ad.nl. The element has a CSS class attribute with the name 'type-modal'. This may indicate that a modal is used to ask for consent. However, the modal could be shown for another purpose, e.g., to select the desired user language. Figure 6.1b shows another example, now from www.ah.nl. Here, the modal is related to an HTML div element with a CSS id attribute 'cookie-popup'. The name suggests that the modal is used to ask for consent, which in this case is. However, it could be that attribute names related to a cookie dialog are used for other purposes, are not visible to the user, or are deliberately used to trick privacy investigations. Although we acknowledge the shortcomings of examining HTML attribute names to detect a cookie dialog, we use this method in our research as there is no waterproof alternative. Secondly, we can integrate this method in our crawler implementation to cover a high number of websites. Taking a screenshot of the website and detecting a cookie dialog with image recognition could be an alternative as OpenWPM provides an option to save screenshots. However, the image detection analysis afterward is likely more complex.

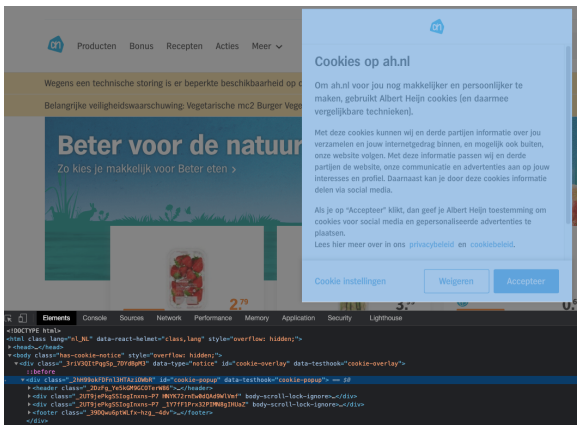
6.1. Experiment

We extended our OpenWPM implementation with a command named "detect_cookie_dialog" to automatically detect a cookie dialog when initially visiting a website during a crawl. To determine whether an HTML element is a cookie dialog, we base our decision on existing lists of HTML/CSS attributes commonly used by ad-block browser extensions. Adblock[1], a well-known browser extension to block advertisers, provides filter lists that they use for their extension. They have divided their lists into different categories for specific

[1] https://adblockplus.org



(a) Selected HTML cookie dialog element from www.ad.nl.



(b) Selected HTML cookie dialog element from www.ah.nl.

Figure 6.1: Selected cookie dialog HTML elements.

purposes[2]. We use the filter list specifically aimed at blocking cookie banners[3]. This list is an extension of EasyList, a filter list created in 2005 by Rick Petnel[4], which is widely used. When a user enables the AdBlock extension, it will look if certain HTML/CSS attribute names of the visited website match the list items. If there is a match, AdBlock will hide the element so that the user is not bothered with notification popups. We use the same technique to detect the presence of a cookie dialog. Fanboy's list not only consists of attributes to hide notifications, but it also contains advanced rules to block third-party scripts. These script blocking rules are not necessary for our cookie dialog detection. Therefore, we took over the CSS ID and Class attributes in two separate files, used as input for our XPath queries.

Besides using specific names in HTML attributes, websites can also use an inline iframe to implement a cookie dialog. An iframe embeds another webpage into the existing page. The benefit of using this method is that websites can load content from external sources without the interference of the current page. E.g., Figure 6.2 depicts the selection of an iframe of the website www.nu.nl. Its source attribute links to external content from dpg-media.nl. The iframe has its own HTML head and body tags as it is a separate inline page. Our detection command searches for iframes within the visited webpage. A webpage can contain multiple iframes, e.g., to show several advertisements. Therefore, we loop over all the available iframes to filter out the possible cookie dialog. First, we base our detection decision on the source attribute of the iframe element and analyze if the source string contains the words "cmp" or "consent". If no element was found, based on this selection criteria, we search further in the embedded webpage of the iframe. To make it possible to switch into the context of the iframe, we first navigate to it via the Selenium command switch_to.frame(frame), with the frame parameter being the iframe element. Then, we search for the existence of HTML elements with a class attribute that contains the string "banner", "consent", or "cmp". If no match is found, we conclude that there is no iframe used for a cookie dialog. Our selection criteria for the iframe source and class attribute strings are based on the manual analysis we first conducted. We examined the cookie di-

[2]https://adblockplus.org/nl/subscriptions
[3]https://secure.fanboy.co.nz/fanboy-cookiemonster.txt
[4]https://easylist.to

alogs of the top 50 Belgium and Netherlands websites to know what names are commonly used in cookie dialog attributes. We acknowledge that we did not investigate other banners from a website, e.g., those related to advertisements. However, our further mentioned validation outcomes show that our crawler does identify cookie dialogs instead of advertisements.

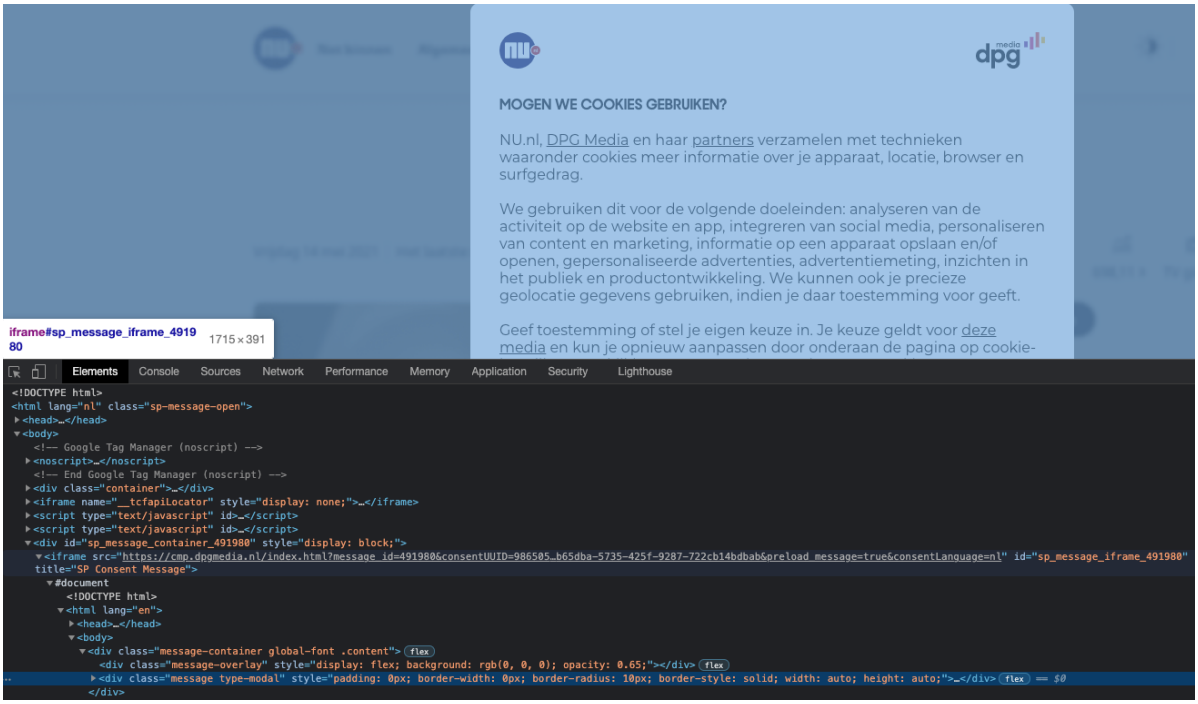
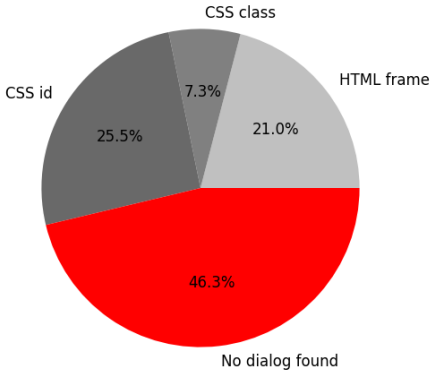


Figure 6.2: Selected iframe from www.nu.nl.

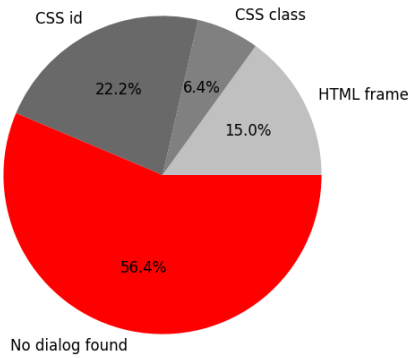
6.2. Results and analysis

We ran our crawler against the top 500 website lists of ccTLDs of all European Union members plus three non-EU members (UK, Norway and, Switzerland) filtered from Tranco's worldwide list, which we also use for the following research questions. Figures 6.3a and 6.3b show the cookie dialog detection rates of The Netherlands and Belgium respectively by performing a crawl with our command "detect_cookie_dialog" enabled. These first results show that The Netherlands has a higher percentage of cookie dialogs compared to Belgium, namely, 43.6% and 53.7%. Further, the United Kingdom has a high detection rate, depicted in Figure 6.3c. We detected that 31.9% of the crawled UK websites do not visualize any cookie dialog to the user. On the contrary, our crawl detected the highest rate of websites that do not use a cookie dialog for Estonia, with an outcome of 70.7%, as outlined in Figure 6.3d. Figures 6.3e and 6.3f show the results for Ireland and Switzerland respectively. The detection rates for Ireland are similar to the UK. Switzerland, a country known for preserving users' privacy, seems to have low detection rates. Low detection rates do not necessarily have to result in higher regulation violations. It could be that more websites do not want to set cookies for non-essential purposes and therefore do not ask for users' consent. A trend that is noticed in the detection results of all countries is that iframes are often used to visualize a cookie dialog. CSS class attributes have overall low detection rates.

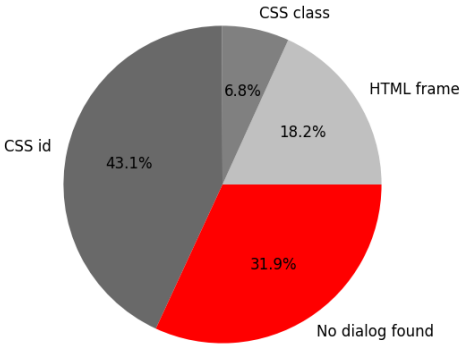
It seems that such attributes are rarely used.



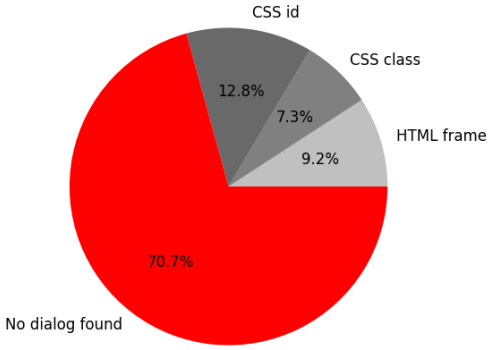
(a) The Netherlands.



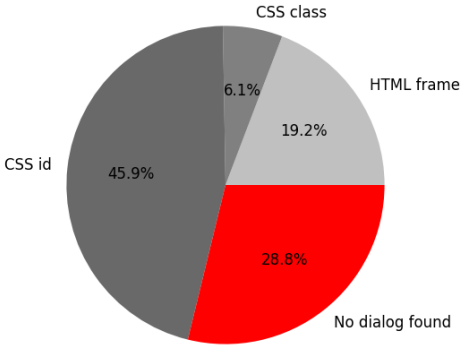
(b) Belgium.



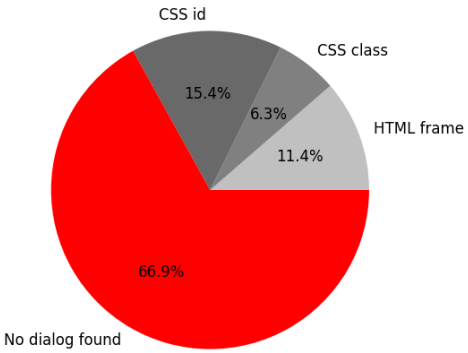
(c) UK.



(d) Estonia.



(e) Ireland.



(f) Switzerland.

Figure 6.3: Cookie dialog detection.

6.3. VALIDITY

To validate the correctness of our cookie dialog detection rates, we manually reviewed a list of websites to observe if the website visualizes a cookie dialog or not. Next, we com-

pared the outcomes of our manual examination with the results of the crawl output. We based the validation on two countries, The Netherlands and Belgium. For each country, we took a sample of 100 websites, i.e., the first and last 50 websites. This way, we have a mixed sample of both often, and less-visited websites as this could influence the presence of cookie dialogs. The manual validation results for the Belgium sample show that our crawler does not detect 14 websites that show a cookie dialog. One website is falsely reported to present a dialog. Validating the sample for The Netherlands results in similar numbers. Twelve websites that show a dialog are not detected by our crawler. Similarly, one website is falsely reported to contain a dialog. These first validation results reveal an error margin of 13 to 15 percent. This means that our measurement of "no dialog found" is an overcount. With a precision and recall of respectively 0.98 and 0.8, the F1 score, which takes both false positives and false negatives into account, results in 0.88. One reason for this is that our XPath query searches for a hard match, i.e., it looks if an item of the id and class lists equals the value of an HTML element attribute. E.g., our crawler indicates that the website www.vlaamsparlement.be does not show a cookie dialog. However, when we visit the website, a dialog is shown at the top of the page. Figure 6.4 the HTML element that is part of the dialog. The button element has a class with the value "eu-cookie-compliance-default-button".

Our detection command does not find this element because it searches for an exact match. Therefore, we should add this value manually to our lists to find this dialog in subsequent crawls. We could have chosen to use a fuzzy search and not an exact match. However, such a method would increase the chance of false positives as certain elements will be wrongly detected as a dialog. E.g., if our XPath query would perform a search for strings containing the value "cookie", it could be associated with elements like a link to the cookie policy or elements of a cookie factory website. Our validation confirms that our matching techniques limit the number of false positives. A total of 2 websites indicate the presence of a cookie dialog wrongly. The downside is that some cookie dialogs are not discovered, as mentioned earlier. An option to further enhance our dialog detection technique would be to introduce more advanced selectors. Adblock Plus also uses such a technique to spot and block advertisements. E.g., they use extended CSS selectors to detect specific advertisement content. One example of this is :-abp-has(> div > a.advertiser), which selects elements that contain, as a direct descendant, a that contains an <a> with the class advertiser[5]. However, they note that such selectors must be sparingly used as it impacts performance.

Secondly, the values of our lists used for detecting HTML elements mainly consist of English words. We chose the English language as currently available lists from Adblock and others also use the English vocabulary. Further, manually inspecting HTML elements in the browsers confirms that, although the website uses another language, the code and related HTML tags are often written in English. Front-end frameworks such as Angular or React, often used in today's websites, also use the English vocabulary, which could make website publishers more reluctant to use their native language. To further increase our detection rates, we could translate fragments of text that appear in a certain amount of dialogs. E.g., our crawl for the 500 websites of Germany did not detect the cookie dialog of

[5]https://help.eyeo.com/en/adblockplus/how-to-write-filters#elemhide-emulation

www.bundesregierung.de. The displayed dialog contains the following text:

Wir verwenden Cookies, um Ihnen die optimale Nutzung unserer Webseite zu ermöglichen. Es werden für den Betrieb der Seite notwendige Cookies gesetzt. Darüber hinaus können Sie Cookies für Statistikzwecke zulassen. Sie können die Datenschutzeinstellungen anpassen oder allen Cookies direkt zustimmen.

We could translate a part of the first sentence, i.e., "cookies to enable you to optimally use our website", to detect dialogs from all countries that contain such content.

6.4. JavaScript Disabled

Furthermore, we ran our cookie dialog detection with JavaScript disabled. This is possible by setting the option "javascript.enabled" to false in our OpenWPM implementation. Figure 6.5 depicts the results for the same countries as we showed in a previous section. The higher results of websites where no cookie dialog is detected are prominent. Even Ireland, with a high detection rate with JavaScript enabled, now only detects a dialog on roughly 30 percent of our dataset. These outcomes indicate that websites expect users to have JavaScript enabled to consent or reject. However, current regulations seem not to mention that users are obliged to enable JavaScript in their browser to make a consent option available. Therefore, it is not clear if such practice is lawful. Although most browsers have JavaScript enabled by default, users can disable it manually, resulting in technical implications regarding current regulations. As long as website publishers do not set non-essential cookies, even if no consent option is shown without JavaScript, users' privacy should be preserved.

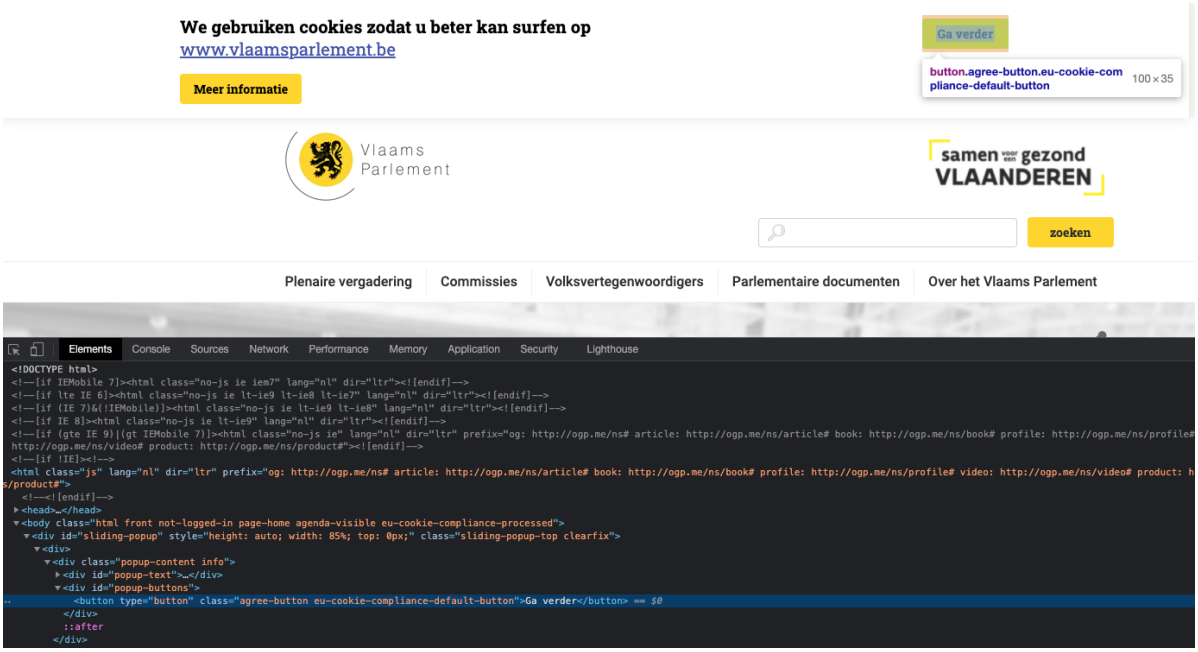
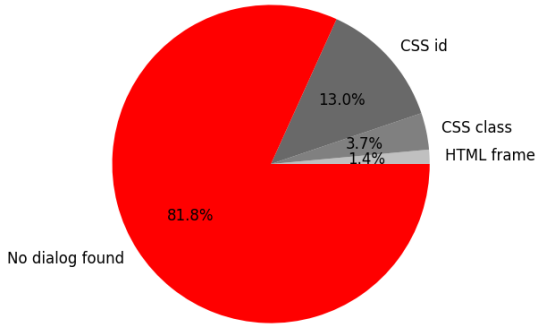
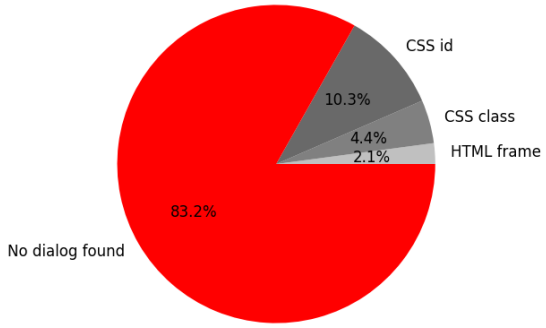


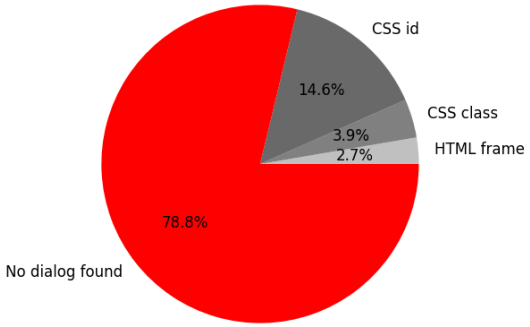
Figure 6.4: Selected cookie dialog button element from `www.vlaamsparlement.be`.



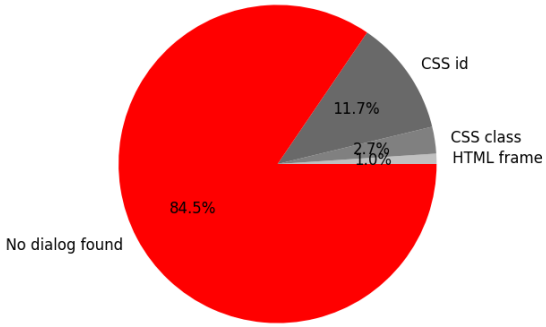
(a) The Netherlands.



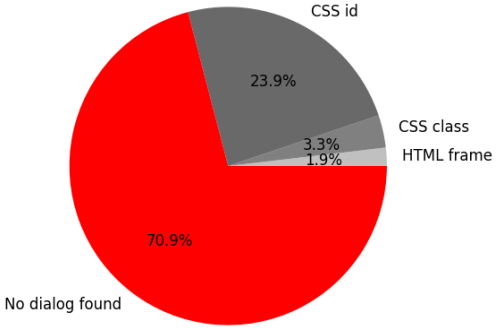
(b) Belgium.



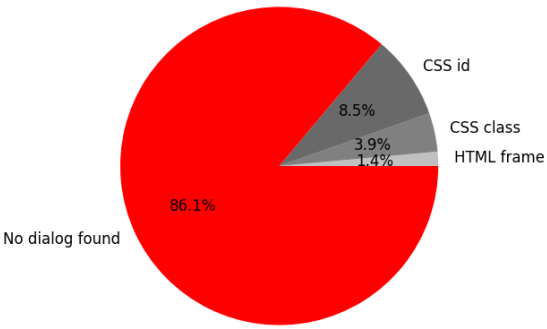
(c) UK.



(d) Estonia.



(e) Ireland.



(f) Switzerland.

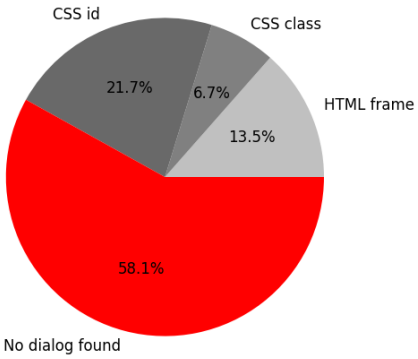
Figure 6.5: Cookie dialog detection with JavaScript disabled.

6.5. uBLOCK EXTENSION

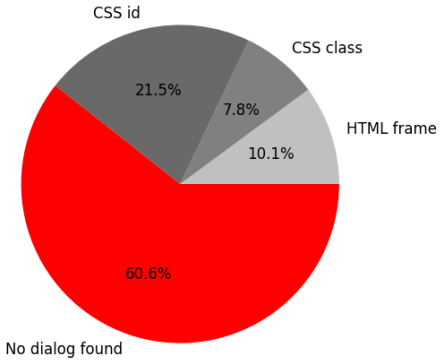
Lastly, we ran our crawler with the uBlock extension[6] enabled to examine if it blocks cookie dialogs. As we cannot manually install extensions in the OpenWPM browser, we down-

[6]https://addons.mozilla.org/nl/firefox/addon/ublock-origin/

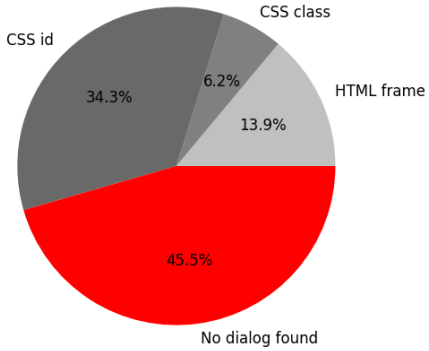
loaded the uBlock xpi file[7] and integrated it into our implementation. We ran uBlock with the default configuration. No additional settings were set. Figure 6.6 depicts the results for the same countries as in the previous sections. Overall, the results show that the uBlock extension does block a number of cookie dialogs. However, without JavaScript, detection numbers are still lower.



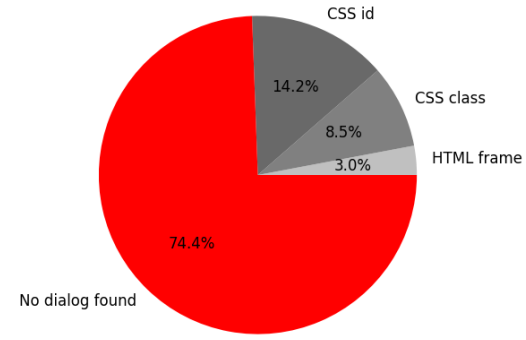
(a) The Netherlands.



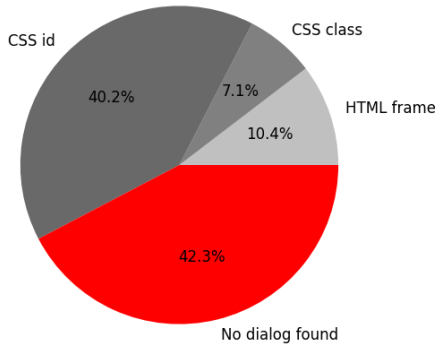
(b) Belgium.



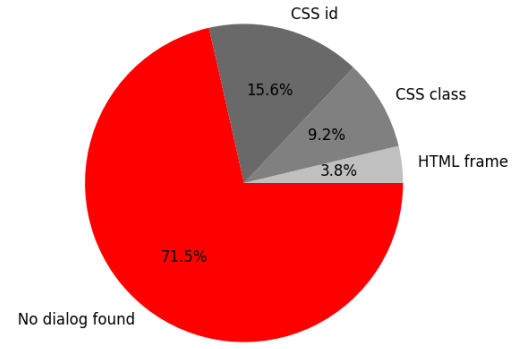
(c) UK.



(d) Estonia.



(e) Ireland.



(f) Switzerland.

Figure 6.6: Cookie dialog detection with uBlock extension installed (default configuration).

[7]https://github.com/gorhill/uBlock/releases

6.6. DISCUSSION

The presented results show a preliminary answer to our first subquestion. Although our experiment has a certain error margin, our outcomes indicate the distribution of cookie dialogs within Europe. Between several countries, there is a significant difference in the presence of dialogs. Switzerland and Norway have a high number of websites where we did not detect a dialog, with respectively 66.9 and 67.1 percentage. First indications could suggest that because these countries are no EU member, such countries would implement the laws differently, resulting in a lower detection rate. However, the United Kingdom, only recently a non-EU member state, does have a high detection rate. Higher detection rates for a country do not necessarily have to result in more violations regarding privacy laws. However, as tracking and advertisements are frequently used on websites, low detection rates for a country raise suspicion.

7

RQ2: To what extent is there a difference in the providers of third-party dialogs used in ccTLDs?

Website publishers can use third-party dialogs to ask for user's consent instead of using their own implementation. Publishers should be assured that they are compliant with the regulations when using a cookie dialog from a CMP. Therefore, they rely on the CMP's expertise. A CMP registered with TCF is considered compliant by the IAB and respects the ePD and GDPR rules. However, this is true as long as the privacy and data protection practices from the IAB are trusted. A report from the Belgian DPA conducted in October 2020 revealed infringements with the regulations. One conclusion of the DPA was that TCF allows companies to swap sensitive information without the authorization of a user. The IAB continues to work with European DPAs to improve TCF[1]. Nevertheless, such findings and public statements could impact trust around the IAB community. Also, CMPs are not required to register themselves with the framework.

Earlier reports already show some insights into the usage of CMPs across a set of websites. Each quarter, the ad tech company Kevel (formerly AdZerk) provides a report about the use of CMPs in the top 10K US sites[2]. Their Q1 2021 CMP adoption rates show that 25.2% of these websites use a CMP, which is an increase of 12% from the previous Q4 measurement. Further, there is a distinct measurement for publishers that show ads. OneTrust is the most used CMP with 870 encounters. As the analysis of Kevel is performed on a list of US websites, the results do not necessarily reflect the CMP landscape of European websites. Currently, there seems to be no analysis of the CMP usage of European country code top-level domains (ccTLDs).

[1] https://iabeurope.eu/all-news/iab-europe-comments-on-belgian-dpa-report
[2] https://www.kevel.co/cmp/

7.1. EXPERIMENT

As a first method to identify the usage of a certain CMP by a website, we saved the JavaScript content. OpenWPM makes it possible to save unstructured data to a local LevelDB database, such as the HTML content and scripts that a website uses. LevelDB is a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values[3]. This data can also be manually viewed in the developer console of a web browser. We set the browser parameter save_content to 'script' in our OpenWPM implementation to gather this data during an automatic crawling process. After the crawling process, the keys, which are arbitrary byte arrays, are used to query the database and obtain the content of the scripts. Only three basic commands are provided in LevelDB to get, put or delete keys and their values. No SQL syntax is available to query the values in the store directly. We first performed a get operation to retrieve the script contents. Next, we analyzed the scripts by searching for strings that start with __tcfapi or __cmp, indicating that the website uses a CMP associated with TCF. As mentioned in the documentation of TCF [4], these API commands should be available if a CMP is present. Performing this analysis on the output of a crawl of the top 100 Belgian websites revealed that these function calls are rarely used in the scripts' content.

CMPs registered with the IAB are required to provide four API commands: ping, getTCData, addEventListener, and removeEventListener. The getTCData command returns a TCData object that contains a list of properties of the CMP used by a publisher. Figure 7.1 shows an example of such an object and its values. An important property is the TC String,

Figure 7.1: getTCData API command properties and values.

[3] https://github.com/google/leveldb

[4] https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework/blob/master/TCFv2/IAB%20Tech%20Lab%20-%20CMP%20API%20v2.md

which captures the preferences a user submits into an encoded and HTTP-transferable string. It contains the user consent, metadata, and other related fields. The IAB provides an online tool to encode and decode such strings[5]. These values are particularly interesting during the examination after user consent. For our analysis of detecting a CMP, an action for consent or refusal is not needed. Therefore, we rely on the ping API command to retrieve basic information about a CMP, such as the CMP name and TCF version. Table 7.1 shows the returned data when executing a ping operation on the website www.telegraaf.nl.

Property	Value
apiVersion	2
cmpId	7
cmpLoaded	true
cmpStatus	loaded
cmpVersion	1
displayStatus	visible
gdprApplies	true
gvlVersion	81
tcfPolicyVersion	2

Table 7.1: Properties and their values returned when executing the ping API command on the website `www.telegraaf.nl`.

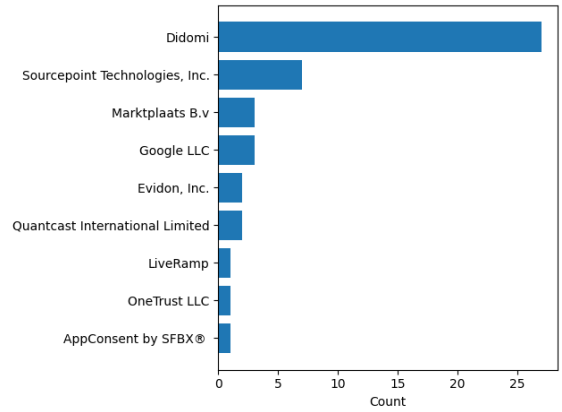
There are existing mechanisms to detect the presence of a CMP that uses TCF. For example, the browser extension "CMP Check" is developed to perform the getTCData API command on a website and show the returned data in a frame. Such extensions make it easy to perform a quick check of whether a website uses TCF. However, such manual analysis on a per-website basis will not scale for our investigation as it is our purpose to automate the analysis for a large number of websites.

We extended the default OpenWPM implementation with a new command named 'ping_cmp', which executes the ping operation on each website visit. We ran the modified OpenWPM crawler for our list of countries. For each country, we took the top 500 website list from Tranco. Then, for each website visit, a check is performed whether the __tcfapi function is available or not to make sure we can execute the ping operation. If the __tcfapi function is available, the returned CMP id is mapped with its name. Without performing this lookup, we would only be able to distinguish a CMP by its id but not know the company behind it. Every Thursday, IAB updates the list of registered CMPs with the TCF[6]. We save the CMP name with its id and the property tcfPolicyVersion to verify which version of TCF is used by the CMP. The data is saved in a new table 'ping_cmp' to make it easier to do further analysis.

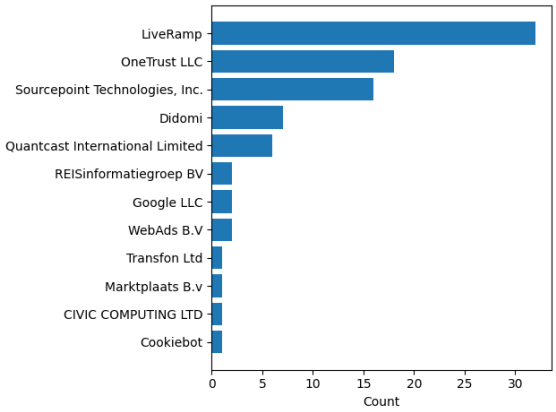
[5]https://iabtcf.com
[6]https://cmplist.consensu.org/v2/cmp-list.json

7.2. RESULTS AND ANALYSIS

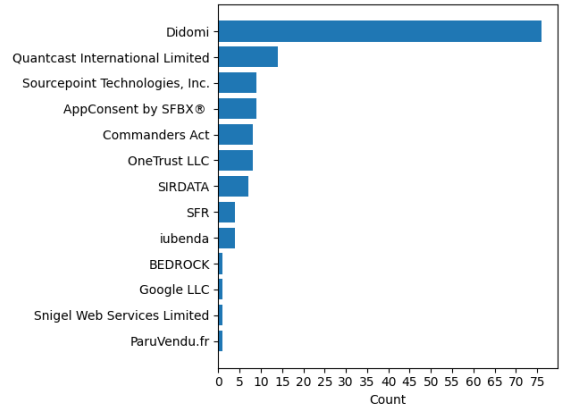
Figures 7.2a, 7.2b and 7.2c show the results of the crawls for Belgium, The Netherlands, and France, respectively. These results show a significant difference in the number of websites that use a CMP per country. Of the Belgian top 500 websites, only 38 websites are able to perform a ping operation. On the other hand, the results of France show that 139 websites return data on the ping operation and thus utilize a CMP. The most used CMP for Belgium and France is Didomi, which could be related to the fact that it is a France company based in Paris. In The Netherlands, Didomi is in place 5 with LiveRamp on top. The low number of detected CMPs could be related to the fact that although a CMP is required to provide the ping command, a website publisher seems not obligated to provide it to its visitors[7]. Secondly, it is possible that a website does not set cookies for non-essential purposes, in which case the publisher is not required to ask for consent and thus not required to use a cookie dialog. Nevertheless, these outcomes give an indication of CMP usage per country.



(a) Belgium.



(b) The Netherlands.



(c) France.

Figure 7.2: CMP usage.

Further, analyzing the countries that are not a member of the European Union from our data list show low detection of CMP usage for Switzerland and Norway. We detected that 15

[7] https://iabeurope.eu/wp-content/uploads/2019/12/2020-02-11-Webinar-CMP-technical-implementation.pdf - slide 5

websites from the top 500 websites of Switzerland use a CMP. Our measurement for Norway reveals CMP usage by 13 websites. On the contrary, for the United Kingdom, 97 websites, i.e., almost 1/5, respond to the ping operation. Although the first two low measurements could indicate a relation between EU membership and CMP usage, there seems to be no direct correlation due to the high UK measurement. Also, we did not perform a crawl for all non-EU members within European geographical boundaries, which would be needed to investigate specific correlations further.

7.3. VALIDITY

The web visits that indicate that a CMP is used can be verified. Manually performing the API command in the developer console of the browser returns the same value when performing this action in our automatic crawl. Manual analysis for a short handpicked list of websites confirms this. Therefore, we can be confident that our CMP measurement does not include false positives. However, we acknowledge that our reported numbers are an undercount. One reason for this is that during the crawling process, some website visits do not fully complete. Such occurrences are logged in the database table 'incomplete_visits'. E.g., after the crawl of the top 500 France websites, 33 visit IDs are logged as incomplete. This happens when the browser crashes or another error occurred such as a time-out. Another possibility is that at the time of crawling, a certain website in the list is unreachable. Such a scenario is less likely because our data list consists of the most used websites of a country. However, our measurements for Cyprus report 108 incomplete visits. We validated this high number in a second crawl, which resulted in the same number of incomplete visits. Secondly, we manually visited a selection of the websites that the crawler did not seem to reach, which confirmed that the websites are unreachable. At this moment, we do not know the cause for this high number of unreachable websites. Fortunately, the incomplete visits for the other countries are limited. These events lead to the existence of false negatives.

To further minimize the undercount, we could inspect the HTTP requests send during a website visit. E.g., our crawling output indicates that the website www.pole-emploi.fr does not respond to the ping command. Performing the ping manually in the browser confirms this. But in the network requests, a POST call to privacy.trustcommander.net is identified, which indicates the website uses the CMP from Commanders Act. Another example of this behavior is when we visit the website www.philadelphia.be. Performing the ping operation responds with a JavaScript ReferenceError, which means the tcfapi command is not available. However, when we click on the link "cookie settings" a popup appears with an indication in the footer "Powered by OneTrust," which indicates it is likely that this website uses a CMP from OneTrust. However, due to time constraints, we did not integrate such analysis into our crawler.

Further, we tested the ping command in the console on the four main browsers, i.e., Firefox, Chrome, Edge, and Safari. First indications show that the ping command can be executed and results in similar outputs across the four browsers. However, as our crawler is built with Firefox, we did not manually check every website ping result for all the browsers.

We have to remark that websites can include their own JavaScript functions, which means that it is possible that a fake implementation of the tcfapi commands could wrongly indicate the presence of a CMP. We tested a fake implementation of the ping operation we performed for our research to demonstrate how easily this can be achieved. We used Witchcraft[8], a Google Chrome extension, to inject custom Javascript into a website. For our test, we used the website tudelft.nl, which does not respond to the ping operation. To enable script injection, we created a file named tudelft.nl.js with the Javascript content depicted in Listing 7.1.

Listing 7.1: JavaScript injection script.

```
const injectedScript = document.createElement('script');
injectedScript.type = 'text/javascript';
injectedScript.innerHTML = "
    function __tcfapi(command, version, pingReturn) {
        var tcData = {apiVersion: '2', cmpId: '28'};
        var success = true;
        pingReturn(tcData, success);
    }";
document.body.appendChild(injectedScript);
```

Then, when we reload the website, our script is included. As a result, our fake ping operation is available, which returns a tcData object with the apiVersion and cmpId properties. Such fake implementation could affect our results as our crawler would detect the usage of a CMP when in reality, this is not the case. Websites could use this technique to influence certain privacy research examinations, like the one we perform for this research. We did not observe our crawler results for the presence of such fake implementations. As we measured a limited usage of CMPs, we assume the presence of such implementations rarely occurs. However, for complete verification, such analysis could be performed in further research.

Further, to ensure reliability, we performed the crawling process at different times to check for any differences in the results. Only a negligible difference in the number of incomplete visits is observed. The measurement of our CMP ping command is constant.

7.4. Discussion

The results of our ping operation reveal a first perspective on the CMP landscape of Europe. The popular CMP tracker from Kevel[9] indicates a CMP usage of 26% for Q2 2021 for the US top 10K websites. We detect an average of 11,6%, which is substantially lower compared to their results. However, our measurement is only based on the top 500 websites of each country. Examination of more websites could yield different results. Smaller EU countries, such as Malta and Cyprus, indicate very low CMP usage, respectively 1 and 6 websites respond to our ping operation. There exist only 305 websites with a ccTLD .mt for Malta, which could be a reason for their low rates. For the websites that use a CMP, our measurement indicates a wide variety of providers. Overall, the top 3 CMPs for each country cover

[8]https://github.com/luciopaiva/witchcraft
[9]https://www.kevel.co/cmp

the largest portion. Quantcast and OneTrust are the most used CMPs. These are respectively 10 and 6 times the number one CMP. Next is Didomi, with four occurrences as being the top 1. Again, comparing it with the report of Kevel shows some differences. According to their audit, OneTrust is the foremost used CMP. The outcomes we present here could also be valuable for further research, as detected violations by prominent CMPs could have a significant impact.

8

RQ3: What kind of cookies are set before a user's consent is given?

In order to identify whether a website sets non-essential cookies, we need to collect the cookies initially set when visiting a website. Website publishers are obliged to ask users' consent before any cookies are set for tracking/advertising, or more generally, for non-essential purposes. Therefore, publishers must show a notification, in the form of a cookie dialog or banner, to ask users' consent. This means that only necessary cookies may be set as long as the user does not give consent. Publishers that do not provide a correct implementation of this behavior violate the regulations and, in extension, R1 of our defined ruling system from section 5.1.

8.1. Experiment

The requirements listed by Santos et al. define a rule R1, which states that consent must be collected before an identifier is stored [SBM19]. The researchers use the term 'identifier' to acknowledge that other techniques exist besides cookies to store an identifier on a user's machine. Violation of this rule is a complex task. E.g., one should analyze all possible browser storages, such as web caching. Certainly, when a combination of storage techniques is used, complexity grows. They conclude that there exist no technical tools to identify the purposes of identifiers. As our research is focused on the use of cookies, for our derived rule R1, we are mainly interested in the purpose of cookies set before consent.

A crucial element in this observation is the list of purposes we use as a basis to detect a violation. We use Cookiepedia, the largest database of pre-categories cookies maintained by OneTrust[1], to identify the category of a cookie. Table 8.1 depicts the categories used by Cookiepedia, which is based on the classification developed by the UK International Chamber of Commerce (ICC)[2].

[1] https://cookiepedia.co.uk
[2] https://iccwbo.uk

Category	ICC description
Strictly Necessary	These cookies are essential in order to enable you to move around the website and use its features, such as accessing secure areas of the website. Without these cookies services you have asked for, like shopping baskets or e-billing, cannot be provided.
Performance	These cookies collect information about how visitors use a website, for instance which pages visitors go to most often, and if they get error messages from web pages. These cookies do not collect information that identifies a visitor. All information these cookies collect is aggregated and therefore anonymous. It is only used to improve how a website works.
Functionality	These cookies allow the website to remember choices you make (such as your user name, language or the region you are in) and provide enhanced, more personal features. For instance, a website may be able to provide you with local weather reports or traffic news by storing in a cookie the region in which you are currently located. These cookies can also be used to remember changes you have made to text size, fonts and other parts of web pages that you can customize. They may also be used to provide services you have asked for such as watching a video or commenting on a blog. The information these cookies collect may be anonymized and they cannot track your browsing activity on other websites.
Targeting / Advertising	These cookies are used to deliver adverts more relevant to you and your interests. They are also used to limit the number of times you see an advertisement as well as help measure the effectiveness of the advertising campaign. They are usually placed by advertising networks with the website operator's permission. They remember that you have visited a website and this information is shared with other organizations such as advertisers. Quite often targeting or advertising cookies will be linked to site functionality provided by the other organization.

Table 8.1: Cookie purposes classification by Cookiepedia.

There exist other lists that map a cookie name to its purpose. E.g., the Open Cookie Database, a project hosted on GitHub[3], is an effort to describe and categorize all major cookies. The completeness of this database depends on the community's input.

As a first examination, we measured the number of cookies set when initially visiting a website. Although this measurement is not directly related to the purpose of a cookie, it is a first indication of how much information is gathered or stored on the user machine. We enabled the browser parameter 'cookie_instrument' in our OpenWPM implementation to collect the cookies per website during a crawl. It records cookies set both by JavaScript and via HTTP responses. The collected data is persisted in the SQLite table 'javascript_cookies'.

[3]https://github.com/jkwakman/Open-Cookie-Database

OpenWPM does not only collect the cookies set but saves all related traffic. I.e., it records an element in the table when a cookie is added-or-changed or when a cookie is deleted. Therefore, we cannot rely on the number of rows per website visit to count the number of cookies. We loop over the added-or-changed records and drop those from the list encountered in the deleted records. As a result, the cookies that are actually set and thus not deleted remain. The code snippet in Listing 8.1 outlines the core loop of our analysis script to calculate the number of cookies set for each website of a crawl. Setting the param 'inplace' on True ensures that the deletion is performed in the panda DataFrame itself, so we do not have to create a new variable.

Listing 8.1: Collecting only the cookies that are added.

```
for r1 in addedOrChangedRecords.name.values:
    if r2 in deletedValues.name.values:
        deletedValues.drop([deletedValues.index[(deletedValues["name"]
            == n)][0]], inplace=True) addedOrChangedRecords.drop([
        addedOrChangedRecords.index[(addedOrChangedRecords["name"]
        == n)][0]],inplace=True)
```

Further, we extract the names of the cookies per website to a JSON file. The element from the JSON is depicted as an example in Listing 8.2, where our crawler has detected five different cookies for the website www.belgium.be.

Listing 8.2: Element from cookie extraction file.

```
"http://www.belgium.be": [
  "language",
  "\_ga",
  "\_gid",
  "\_gat",
  "TS016a4e3d"
],
```

It is possible to look up the purpose of each found cookie manually via the website of Cookiepedia. However, such a manual technique is not scalable. Unfortunately, Cookiepedia does not provide an API to request the purpose of a cookie, which would make it convenient to request a bulk of purposes. Therefore, we implemented a script based on Puppeteer[4], a Node library often used for testing functionalities and crawling websites to capture specific DOM elements. Our script concatenates each cookie of our JSON list to the Cookiepedia URL "https://cookiepedia.co.uk/cookies/" that is used to discover its purpose. During the automatic website visit, we extract the text of the HTML elements used by Cookiepedia to display the description and purpose of the cookie. We count each encountered purpose to produce a final sum of purposes for each URL of our JSON output. Listing 8.3 shows the loop we perform from our script to capture each purpose from the Cookiepedia website.

Listing 8.3: Part of our Puppeteer script to discover cookie purposes.

```
for (let cookie_name of cookie_names[url]) {
```

[4]https://pptr.dev

```
    try {
        await page.goto(base_url + cookie_name);
        const description = await page.$eval('#content-left > p', el
           => el.textContent);
            const purpose = await page.$eval('#content-left > p >
               strong', el => el.textContent);
        cookie_purposes.push({
            name: cookie_name,
                description: description,
                purpose: purpose
            });
        switch (purpose) {
            case 'Targeting/Advertising':
                target_and_ad++;
                break;
            case 'Strictly Necessary':
                necessary++;
                break;
            case 'Functionality':
                functionality++;
                break;
            case 'Performance':
                performance++;
                break;
            case 'Unknown':
                unknown++;
                break;
        }
    } catch (error) {
        continue;
    }
}
```

We did not choose to integrate this crawling process into our OpenWPM extension as this step is part of the analysis process. Secondly, it is currently not possible to perform headless crawls with OpenWPM. With Puppeteer, we can extract the data via headless crawls, which increases performance as the browser does not have to be actively opened and closed for each purpose extraction. We save the result in a separate JSON file for each country for further analysis.

8.2. RESULTS AND ANALYSIS

We ran our crawler against our Tranco datasets. Figure 8.1 depicts the number of cookies set by our top 500 websites of The Netherlands and Belgium. We use the Empirical Cumulative Distribution Function (ECDF), often used in exploratory data analysis (EDA), to

observe the number of websites that set a certain amount of cookies. It outlines a complete view of how the data is distributed. In Figure 8.1a, the plotted ECDF of The Netherlands data shows that 85.2 percent of the websites, which amounts to 426, set 2 cookies or more. Similar values are observed in Figure 8.1b. 77.4 percent of the Belgian websites set 2 cookies or more. One Belgian website, a Cameroon information website[5] is an outlier that sets 144 cookies on the initial visit. The outlier from The Netherlands, www.1limburg.nl, sets 83 cookies on the initial visit. When we visit this website, a cookie dialog appears. Inspecting the cookies indeed reveals that many cookies are set before clicking on accept. E.g., a cookie with the name "PugT" is set. According to the Cookiepedia database, this cookie is mainly used for targeting and advertising purposes, which could mean the website is in violation with the legislation as it already sets a cookie for non-essential purposes before a user gives consent.

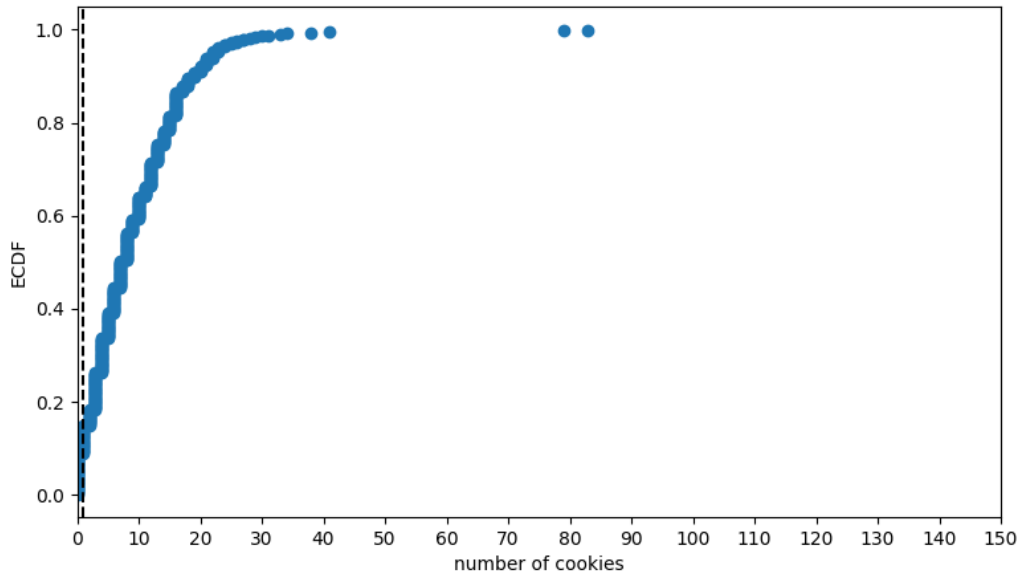
We used an analysis technique called bootstrap sampling. By resampling our collected dataset with replacement, we calculate the confidence intervals and standard error for the number of cookies set. 'With replacement' means the samples are independent of each other as the numbers taken for constructing each sample are returned to the main data set before the sample extraction. We extracted 1000 random samples, with each sample the same length as our main set. This amount of samples is needed to ensure more certainty for our calculations. The code snippet in Listing 8.4 depicts the loop we perform over our main set to calculate the mean of each sample. We used the library Numpy, abbreviated as 'np', to perform the calculations.

Listing 8.4: Calculation of mean for a list of samples.

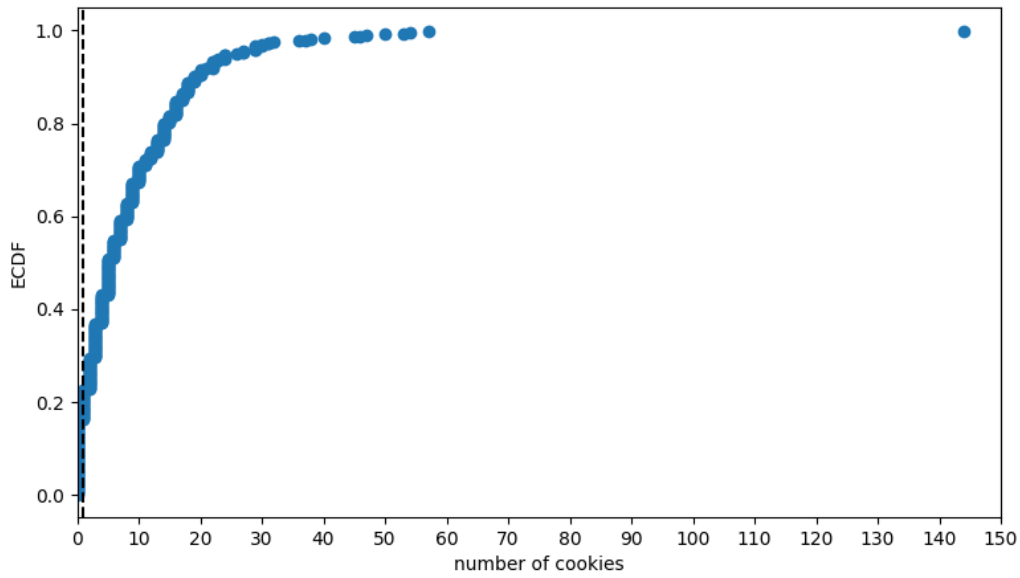
```
sample_means = []
n = len(df.amount_of_cookies)
for sample in range(0, 1000):
    sample_values = np.random.choice(a=df.amount_of_cookies.values,
        size=n)
    sample_mean = np.mean(sample_values)
    sample_means.append(sample_mean)
```

Figure 8.2 shows the results of the calculations for The Netherlands and Belgium, depicted as a probability density function (PDF). Using a PDF, we can examine the probability of other random samples. Thus, how many cookies a website will likely set. The total underlying body of the curve equals one, which is the sum of all the probabilities for the number of cookies set. The PDF for The Netherlands in Figure 8.2a depicts that websites set between 9.2 and 11.1 cookies, with a confidence of 95 percent. The results for Belgium are in the same trend in Figure 8.2b, with low and high confidence intervals of 8.3 and 10.3. Further, the standard error of the mean (SEM), which measures the distribution of sample means around the mean of the original dataset, is respectively 0.46 and 0.51. Comparing the two countries shows a slight difference in the distributions. Belgium puts on average a lower number of cookies than the Netherlands.

[5] https://www.camer.be



(a) The Netherlands - number of cookies - ECDF.

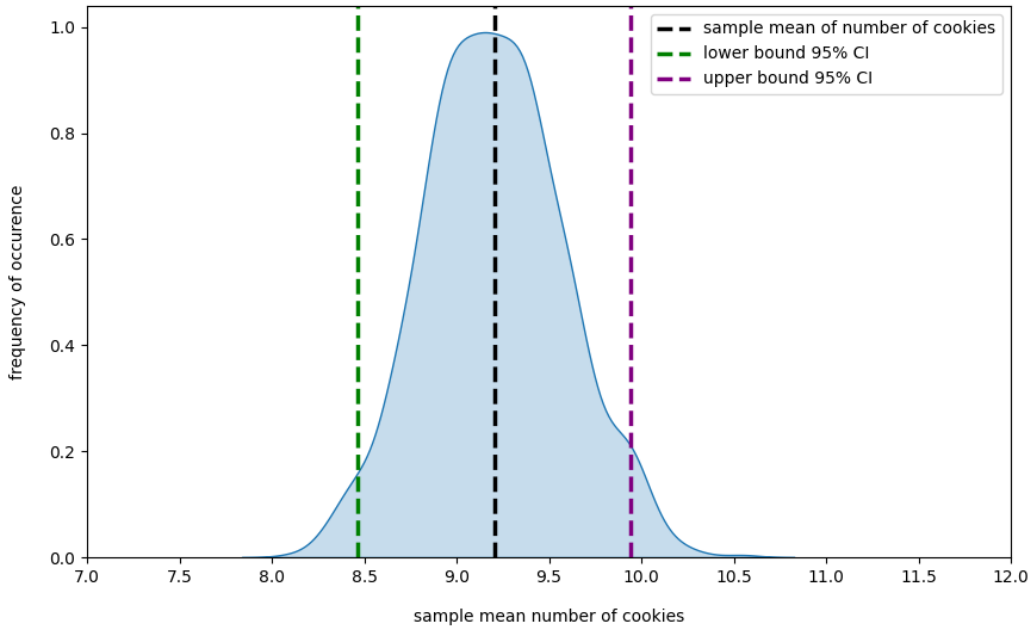


(b) Belgium - number of cookies - ECDF.

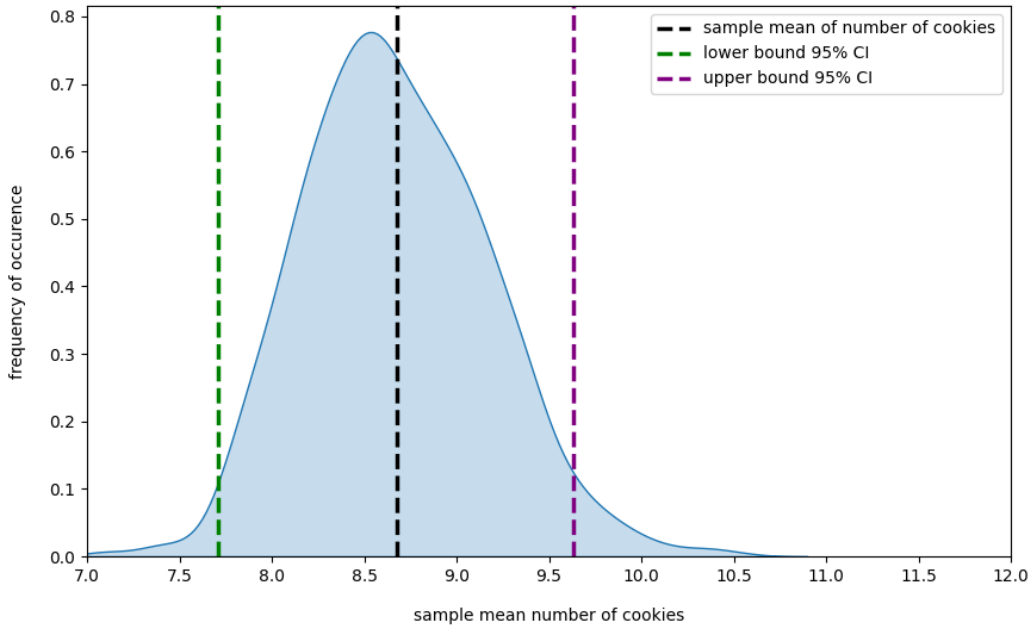
Figure 8.1: The Netherlands and Belgium - Number of cookies (empirical cumulative distribution function).

Figure 8.3 depicts the mean number of cookies for each country of our dataset displayed in the geographical map of Europe. We used the GeoJSON standard to generate the map. GeoJSON is a format for encoding a variety of geographic data structures. In August 2016, RCF 7946[6] was published as the standard specification for the GeoJSON format. Our GeoJSON file is obtained from publicly available vector maps[7], from where a specific region

[6]https://datatracker.ietf.org/doc/html/rfc7946
[7]https://geojson-maps.ash.ms



(a) The Netherlands - number of cookies - PDF.



(b) Belgium - number of cookies - PDF.

Figure 8.2: The Netherlands and Belgium - Number of cookies (probability distribution function).

can be chosen. The GeoJSON file is used as input to generate a choropleth map using the Python library Plotly[8].

[8]https://plotly.com/python/choropleth-maps/

The map shows a higher number of average cookies per website for some countries. The United Kingdom sets an average of almost 20 cookies per website. Figure 8.4a shows the plotted ECDF for the UK. Besides the one outlier, the distribution is more curved compared to other ECDF plots. It is clear that globally, UK websites set a higher amount of cookies per website. It does not necessarily have to mean that these websites violate the regulations. Further analysis is needed to observe the purposes of the cookies set. Austria, on the other hand, is almost colored white on the map, with a rounded average of 6 cookies per website. The ECDF of Austria, depicted in Figure 8.4b shows a steep curve. 68.4 percent of the websites set 2 cookies or more, with only a few outliers. For completion, we mention that the outcomes for the countries Cyprus and Malta are not shown on the map as these are not included in our GeoJSON file. Their rounded average cookies set are respectively 6 and 8, which means they fall into the lower region of the spectrum and are comparable with Austria and Belgium.

Figure 8.5 depicts the number of all collected cookies for our top 500 websites per country mapped to its purpose extracted from the Cookiepedia database. In descending order, we first encounter the United Kingdom. As previously mentioned, the UK sets the highest average amount of cookies per website. A relatively small amount of cookies is set for strictly necessary and functionality purposes, marked green and blue. Most cookies are set for performance, i.e., to record how visitors use a website, and targeting/advertising purposes. The same trend is observed for the other countries. Further, there are a high number of unknown cookies that cannot be linked to any purpose. Or at least, it is not known by the Cookiepedia database. Because of the high number of unknowns, we further examined the content of these cookies. Figure 8.6 depicts the top 10 cookies of the Netherlands, Belgium, the United Kingdom, and the Czech Republic, of which the purpose is unknown. The cookie name "fr" recurs several times. For the United Kingdom, this cookie is even counted 169 times. Even though this is a high number, it is only a small percentage of the 3,309 unknown UK cookies. It is also the most common unknown cookie for the Netherlands. Further investigation via search engines reveals that the purpose of this cookie is related to Facebook. The Facebook cookie policy page[9] states that the "fr" cookie is used to display, measure, and improve the relevance of advertisements and has a life span of 90 days. Not all descriptions of the purposes seem to match between different policies. E.g., the cookie policy of the company MSCI states that Akamai uses the cookie "ak_bmsc" to optimize site performance and security[10], whereas EU4Digital states that it is a functionality cookie placed by Mailchimp to manage and control lists[11]. For our observed countries, the top 10 does not cover a high percentage of the total unknown cookies. We would therefore need to manually examine many cookies to uncover more purposes. Due to time constraints, we leave this examination for future research.

[9] https://www.facebook.com/policies/cookies
[10] https://www.msci.com/cookie-policy
[11] https://eufordigital.eu/cookies-policy

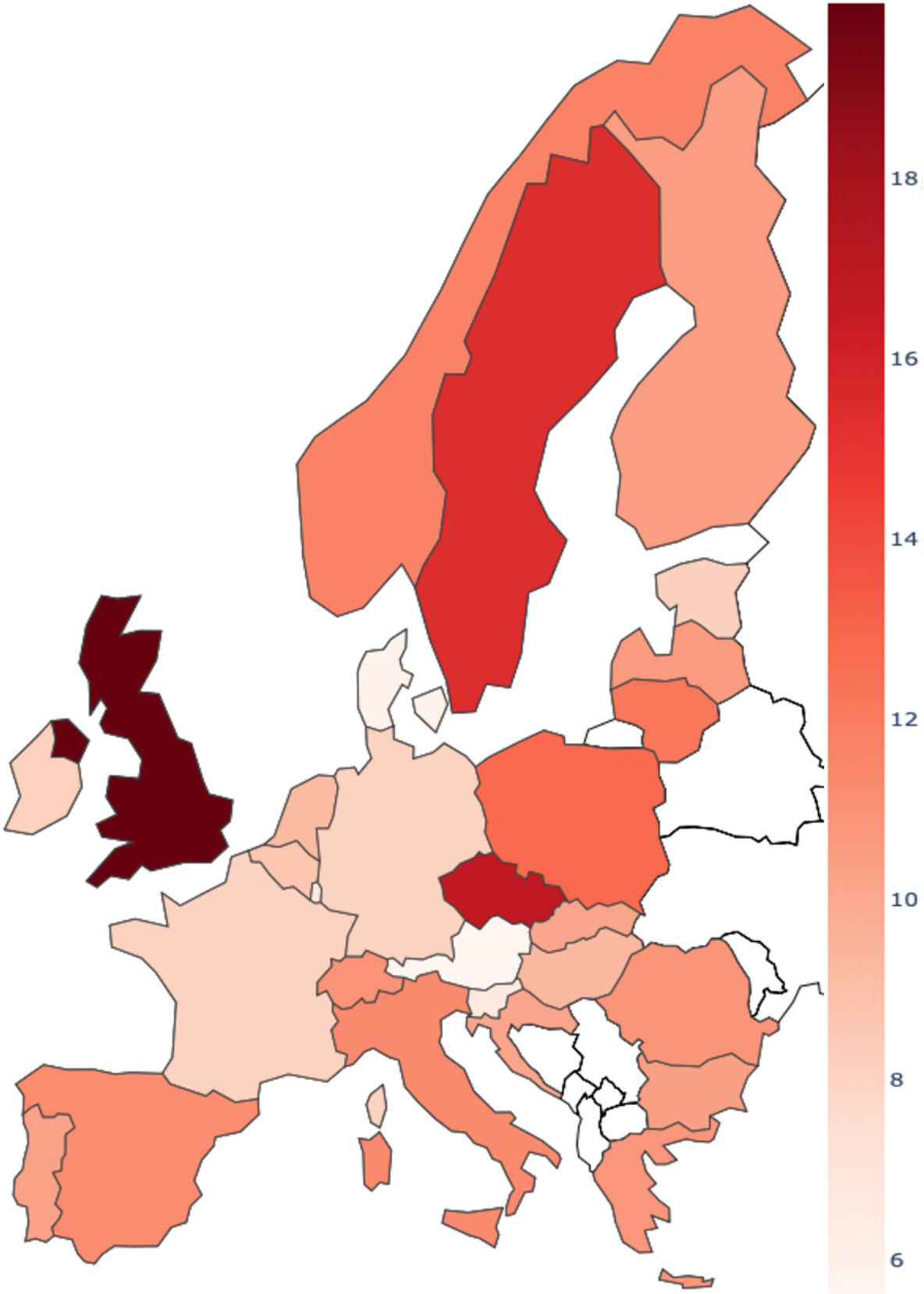
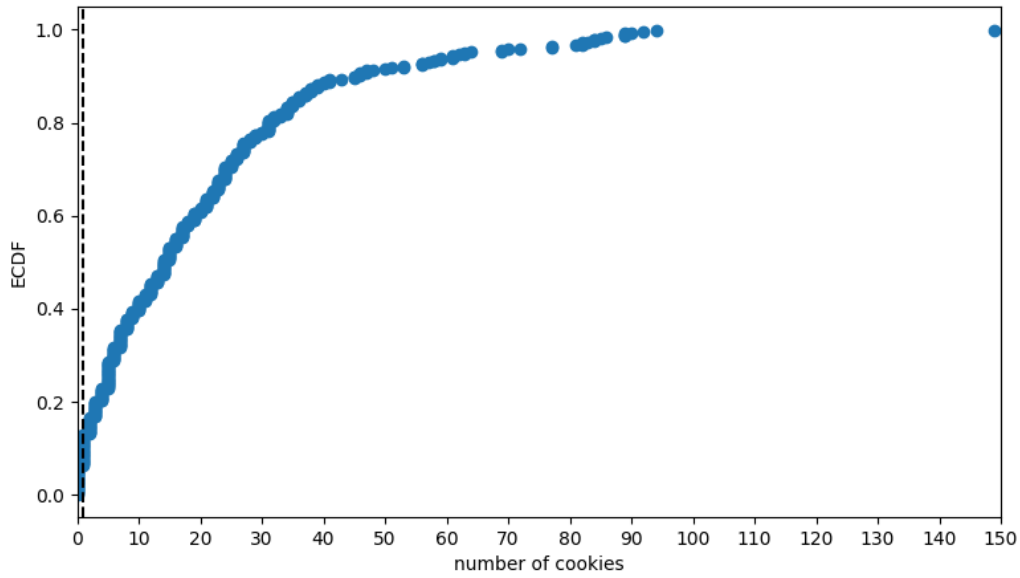
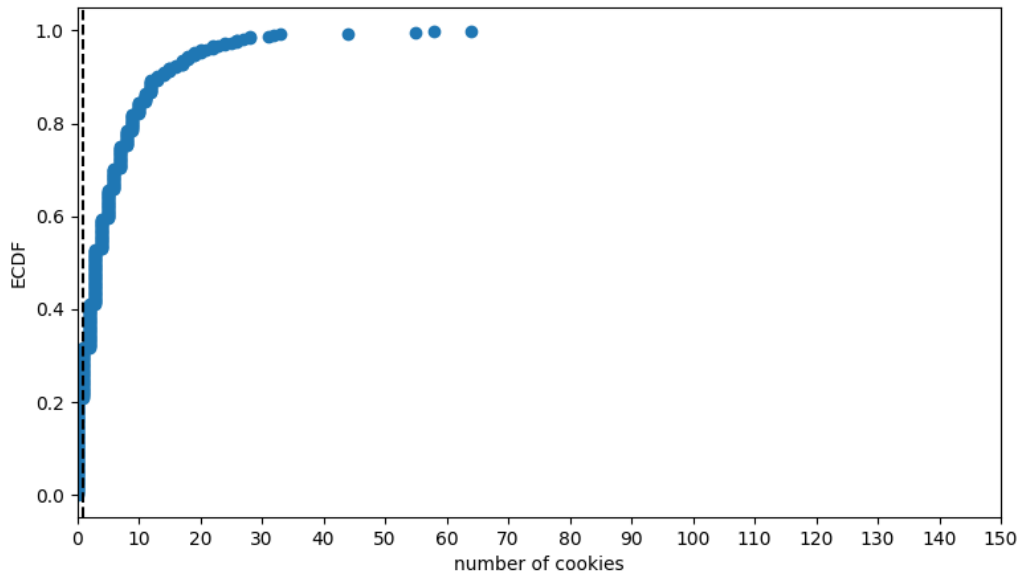


Figure 8.3: Average number of cookies set prior to consent per country.



(a) United Kingdom - number of cookies - ECDF.



(b) Austria - number of cookies - ECDF.

Figure 8.4: United Kingdom and Austria - Number of cookies (empirical cumulative distribution function).

8.3. VALIDITY

To validate our results, we handpicked several random websites from our datasets to manually check the number of cookies set. Examing the cookies in the Firefox browser reveals mostly the same number of cookies measured as our crawl output. However, sometimes a minor difference is observed. E.g., our manual check for the website https://nos.nl observed a cookie named "_chartbeat2", but our crawl output detected a second similar

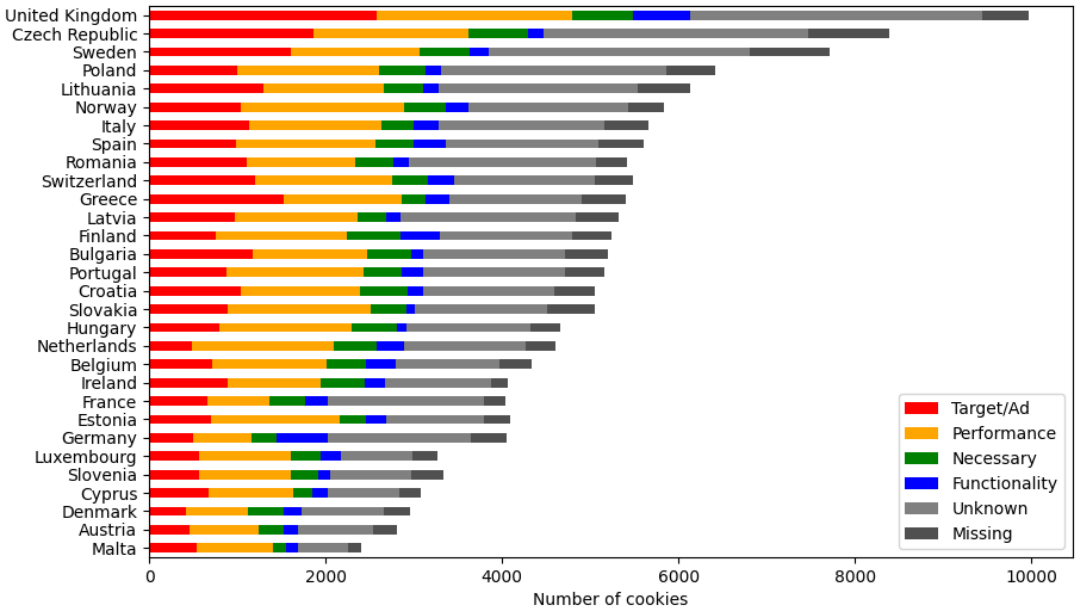
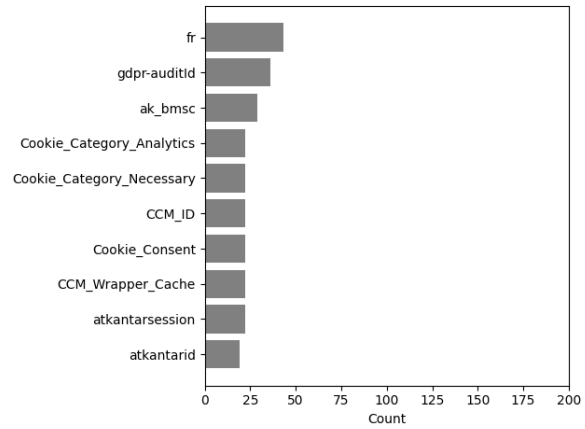


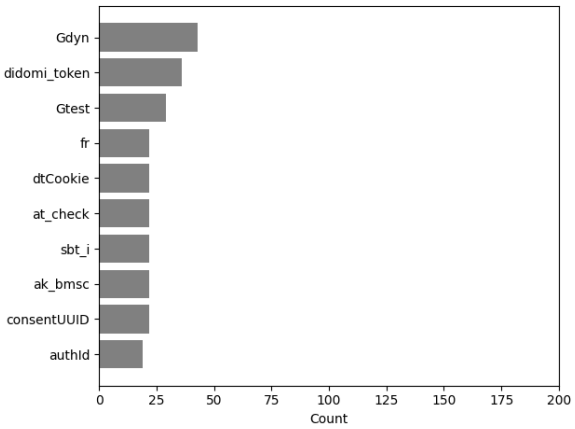
Figure 8.5: Cookie purposes according to Cookiepedia for cookies set prior to consent.

cookie with the name "_chartbeat4". Also, distinct sets of cookies are detected between different browsers. Again, when we analyze the cookies set for the NOS website in Google Chrome, only six cookies are shown in the developer console compared to twelve in Firefox. A reason for this diversity is that each browser handles cookies differently and has different default configurations regarding privacy. As our OpenWPM crawler uses a version of Firefox Nightly, it can have other privacy configuration options enabled. This config variation in browsers does not make it a straightforward task to validate the output manually. Also, cookies are split up according to their domain in the developer console. The number of cookies in each domain cannot always be counted together, as some cookie names are duplicated in multiple domains. Figure 8.7 depicts the list of domains observed in the Firefox developer console when visiting the website www.thelocal.fr.

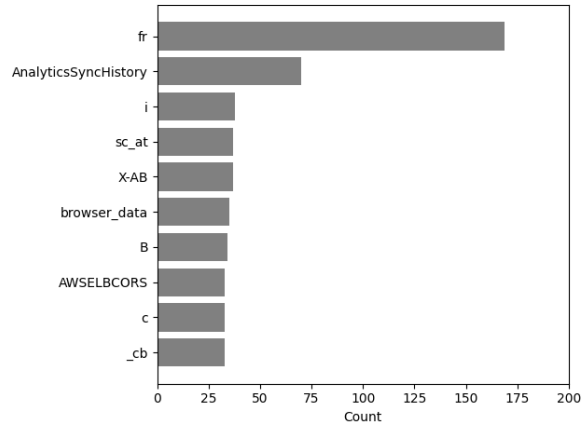
Counting the exact number of cookies of all domains manually without duplication is a tedious task and error-prone. Fortunately, the extraction of cookies during a crawl is a standard option in OpenWPM. It is regularly used in previous privacy research studies and is, therefore, more tested than our extended commands. Another peculiarity observed is the difference in the behavior of a website between Firefox in Ubuntu and other operating systems. Our crawler detected a high number of cookies for the website matchdirect.fr. Manual verification revealed that no cookie dialog was shown in Firefox on Ubuntu and immediately set cookies on the machine. On the other hand, a cookie dialog did show up in Firefox on other operating systems. It was tested in private mode to disable possible cache from previous sessions. Although we did not encounter similar cases in our manual analysis, some websites in our automatic crawl could behave differently in Ubuntu.



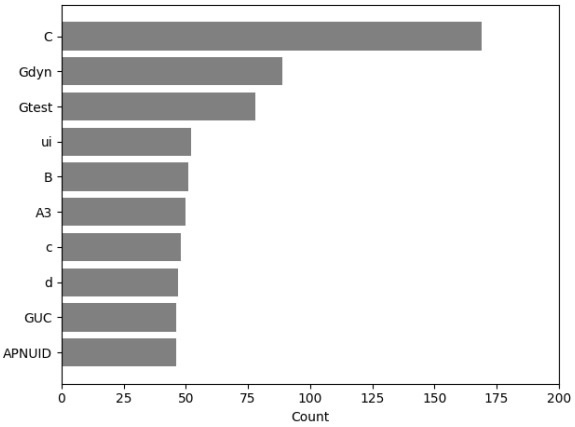
(a) The Netherlands top 10 unknown cookies.



(b) Belgium top 10 unknown cookies.



(c) United Kingdom top 10 unknown cookies.



(d) Czech Republic top 10 unknown cookies.

Figure 8.6: Top 10 unknown cookies.

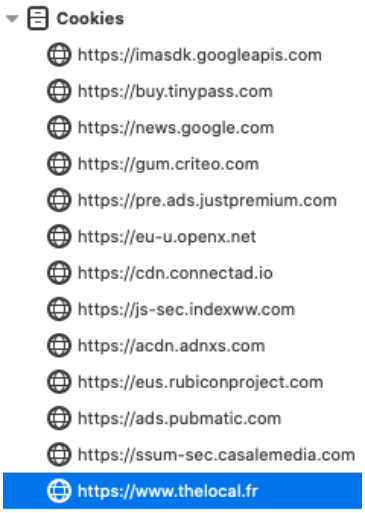


Figure 8.7: Cookie domains from `www.thelocal.fr` displayed in the Firefox developer console.

8.4. DISCUSSION

Although there currently exist many technologies for tracking online user behavior, cookies are still widely used. Legislators that will have to implement coming updates of current regulations, such as the ePrivacy Regulation, await the difficult task of increasing transparency and diminishing privacy concerns. Our outcomes indicate that transparency regarding the purpose of cookies is still below par. A large portion of the cookies we collected cannot be related to any purpose. This unknown gap is problematic and diminishes values of trust promoted by recognized institutions such as the IAB. Hereby related is that the largest database of cookie purposes is maintained by OneTrust, an organization that provides a CMP and therefore not completely neutral. Even manual research for a specific cookie purpose results in various descriptions. Further, for all observed countries, a significant amount of cookies are set for tracking and advertisement purposes before consent is given. Although the value of the cookie could not yet hold any personal information, the presence of such a cookie is certainly suspicious. Such websites could be in direct violation of the law.

9

RQ4: To what extent are cookie dialogs using interface elements to nudge users in giving their consent?

For this last subquestion, we examine the existence of dark patterns that website publishers implement to nudge users in giving their consent, i.e., the practice where design elements are used to persuade a user to click on a particular element. R2 of our defined ruling system from Section 5.1, based on R13 of the research of Santos et al., states that a cookie dialog needs to provide balanced consent and refuse choices. As discussed, the exact interpretation of balanced is a grey area as there is no standardization in the design elements of cookies dialogs. Therefore, to cover a wide range of dark patterns, manual analysis is still needed. Santos et al. state that it is not possible to verify the requirement automatically because of the lack of standards. As our goal is to identify violations automatically, we limit the scope of our detection mechanism to observe whether unambiguous options are available to consent and reject.

9.1. Experiment

To determine whether a cookie dialog presents a balanced choice for consent and reject, we collect the color, width, and height for further analysis. In earlier research, Turland et al. [TCJ+15] rearranged the presentation of wireless networks to users by placing the most secure options at the top. Therefore, they used color codes to mark unsecured networks as red and secured as green. The combination of positioning and color increased the rate of secured network selection by 60 percent. Nudging participants only by changing the position had a limited effect. Other research studies show that nudging people's behavior by using colors can be used in various environments. E.g., Thorndike et al. [TSR+12] showed that using color-coded labels improved sales of healthy items in a hospital cafeteria. Currently, there seems to be no research performed specifically towards the interaction of consent and reject elements in cookie dialogs and how color, positioning, and other attributes

influence users' behavior. Nevertheless, previously mentioned studies show that different design choices influence behavior.

We extended our OpenWPM implementation with a new command "detect_dark_patterns". We used a broad naming for the command as new detection patterns can be added in the future. To detect whether a consent and reject element exists on a website, we search for certain text strings. An essential part of this detection mechanism is the strings we use to identify an element on the webpage that results in consent or refusal when the user clicks on it. We based our list of strings by manually analyzing the words or sentences used on consent and reject elements in a partial list of our top websites of the Netherlands and Belgium. As not all websites use the native language of the country but instead use English terms, we also added the translations of the observed strings. We observed that many websites display the same terms, which results in a limited set of strings. The text of a subset of HTML elements that contain an item of our string list, depicted in Table 9.1, is identified as a consent element.

Dutch terms	English terms
akkoord	agree
accepteer	allow
toestaan	accept
accepteren	accepted
aanvaard	fine
aanvaarden	okay
prima	grant
stem toe	consent
oké	
toestemming	

Table 9.1: Dutch and English terms used to detect consent elements.

To search and select an element, we use an XPath query which Selenium executes. Listing 9.1 shows our composed XPath query.

Listing 9.1: XPath query to select consent elements.

```
elements = webdriver.find\_elements\_by\_xpath(
        "//button[(contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',
             'abcdefghijklmnopqrstuvwxyz'), {value}) or "
        "contains(translate(@aria-label, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',
             'abcdefghijklmnopqrstuvwxyz'), {value})) and "
        "not("
        "contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
            abcdefghijklmnopqrstuvwxyz'), 'niet') or "
        "contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
            abcdefghijklmnopqrstuvwxyz'), 'not') or "
        "contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
            abcdefghijklmnopqrstuvwxyz'), '...')"
        ")]|"
```

```
"//button[normalize-space(translate(text(),'
    ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'))
    ='ok']|"
"//a[contains(translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','
    abcdefghijklmnopqrstuvwxyz'), {value}) and "
"not("
"contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
    abcdefghijklmnopqrstuvwxyz'), 'niet') or "
"contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
    abcdefghijklmnopqrstuvwxyz'), 'not') or "
"contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
    abcdefghijklmnopqrstuvwxyz'), '...')"
")]|"
"//a[normalize-space(translate(text(),'
    ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'))
    ='ok']|"
"//span[contains(@class,'a-button-inner') and "
"contains(translate(.,'ABCDEFGHIJKLMNOPQRSTUVWXYZ', '
    abcdefghijklmnopqrstuvwxyz'), {value})]|"
"//input[contains(translate(@value,'ABCDEFGHIJKLMNOPQRSTUVWXYZ
    ', 'abcdefghijklmnopqrstuvwxyz'), {value})]"
    .format(
    value='\"' + b + '\"'))
```

The translate function is needed to convert HTML elements text to lowercase, as our search needs to be case insensitive. A subgroup of DOM elements and related attributes is examined to minimize the chances of selecting the wrong elements. I.e., buttons and the related aria-label attribute, anchor elements, a specific span element with class "a-button-inner", and input elements. Further, we only examine div elements when there is no match for the previously mentioned elements as div elements often contain text related to the actual website content. Therefore, we also limit the string length to 20 characters. If no element is found, we further search for the presence of iframes related to cookie dialogs. If present, the same analysis via xPath is performed on the content of the iframe. Finally, the width, height, and background color attributes are selected from the element. We save the color by its RGB and converted HEX values.

In the first implementation, we used the Python library google_translator to translate our list of strings to different languages. Such a technique increases extendibility as a language parameter has to be provided to our command to translate the list of strings to the native language of a certain country. However, in subsequent crawler runs, the translate function of the library yields an error indicating that too many requests are performed[1]. Therefore, our current implementation uses different lists for each language, which is a manual translation of our Dutch list via Google Translate[2]. We acknowledge that our translations could contain words or phrases uncommonly used in cookie dialogs for the asso-

[1] https://github.com/lushan88a/google_trans_new/issues/28
[2] https://translate.google.com

ciated country. To improve our literal translations, a native speaker or professional interpreter should verify and correct our lists.

9.2. Results and analysis

Differentiating colors into categories that nudge users on different levels is a grey area and requires specific research, which we do not perform here. We base our analysis on colors used in previous research and use our own judgement for deviances. We divide colors into two self-created categories. Friendly colors such as green and blue relate to nudging users. Hard colors such as red and black relate to visualizing danger to drive away a user from clicking on the button. Further, we measure the width and height to observe discrepancies between consent and reject elements.

First, we are interested in how many accept and reject buttons we can identify in a cookie dialog. If we cannot find a reject button but can find an accept button, this would suggest a dark pattern and result in non-compliance. For this analysis, we use the same datasets, i.e., the top 500 websites of the European countries filtered from the global Tranco list. Figure 9.1 shows an overview of the number of consent and reject elements discovered by our crawler. The top 8 of the list are countries situated in the western part of Europe. France is at the top, for which we discovered 306 websites that have a direct option to consent and 142 websites with a reject option. Other countries reveal higher differences, e.g., about 270 websites of the UK offer a consent option and 70 a reject option, slightly less than The Netherlands. Notice that we only detect consent and reject elements in the first layer of a cookie dialog. E.g., Figure 9.2a shows a consent and reject option for the website rijksmuseum.nl. Figure 9.2b shows the cookie dialog of the website nieuws.nl with no direct reject option. We do not detect a possible reject option in a second or third layer behind the cookie configuration link. Malta, with its 305 websites, has a relatively high detection rate. We detected 110 elements, of which 97 are for consent and 13 for rejection. Thus, approximately one-third of the websites of Malta offers a consent option, a rejection option, or both. In comparison, we detect only 53 websites with a consent option and two websites with a direct option to reject for Norway. On average, our crawler detected 171 elements per country.

Figure 9.3 zooms in on the number of reject elements found per country. With 142 reject elements found, France is at the top of the list. The top four of the list are the same as in Figure 9.1. Following are Switzerland and Luxembourg, with respectively 41 and 39 reject elements found. After Cyprus, which ends the top ten, the number of reject elements found further declines sharply. At the bottom is Poland, for which we did not detect any reject element.

Figure 9.4 depicts the background colors of the consent and reject elements for The Netherlands, France, Ireland, and Austria. These colors are detected automatically by our crawler. To observe the context where the elements are used, we manually obtained the background color of the cookie dialog. Further, we only show the elements of websites with both a consent and reject option to compare the colors and detect possible dark patterns. Therefore, we filtered out the records that have elements without a color or with

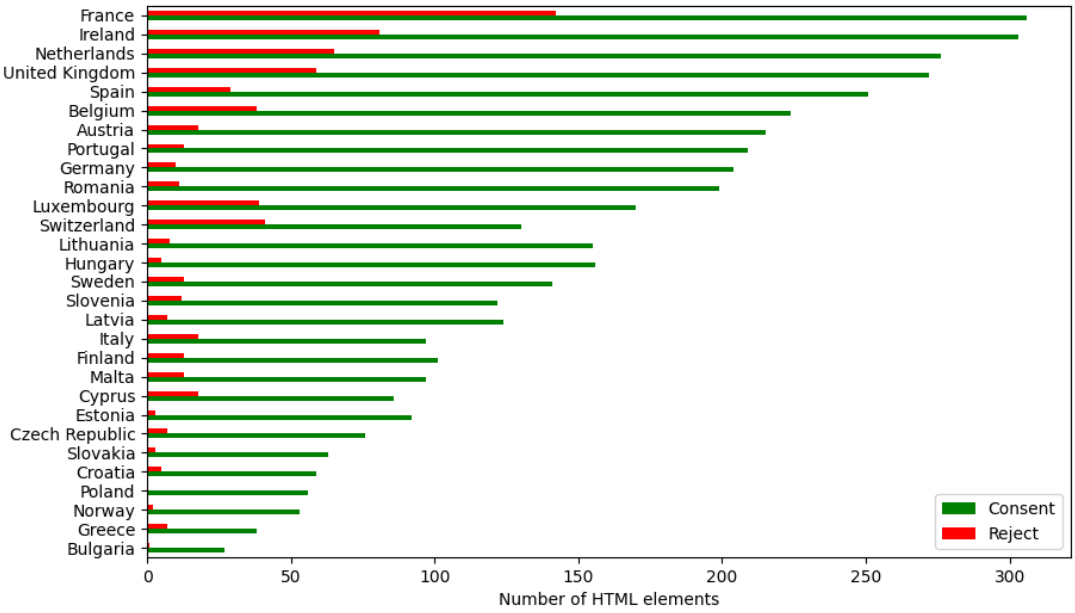
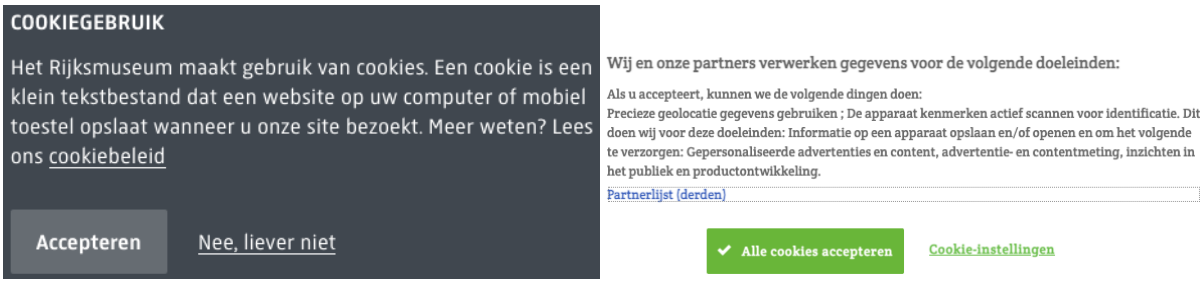


Figure 9.1: Number of consent and reject elements per country.



(a) Cookie dialog of `www.rijksmuseum.nl`.

(b) Cookie dialog of `www.nieuws.nl`.

Figure 9.2: Cookie dialog examples with and without a direct reject option.

a transparent color. Overall, we notice that the color green is often used for the consent option. Only one French website uses a green color for its reject option. For the reject options, the colors red, white, grey, and black are the most prominent. The background color of the cookie dialog has mostly the color white or black, or is similar to the elements' color. Using the same color analysis from earlier research [TCJ+15] to identify dark patterns, we notice that most websites with both a consent and reject element nudge a user in giving consent. There are a few exceptions where the consent element has a red color. However, the number of these exceptions is negligible. Also, we notice that several records have the exact same RGB colors. E.g., several records from Ireland depicted in Figure 9.4c have the same green and black value for their elements. Manual analysis reveals that some of these websites use the CMP Cookiebot. Therefore, the cookie dialogs have a similar design.

Further, the difference in width and height is shown in the last two columns. In particular, we subtract respectively the width and height of the reject element from the consent element. This results several times in negative consent width values. Depending on the lan-

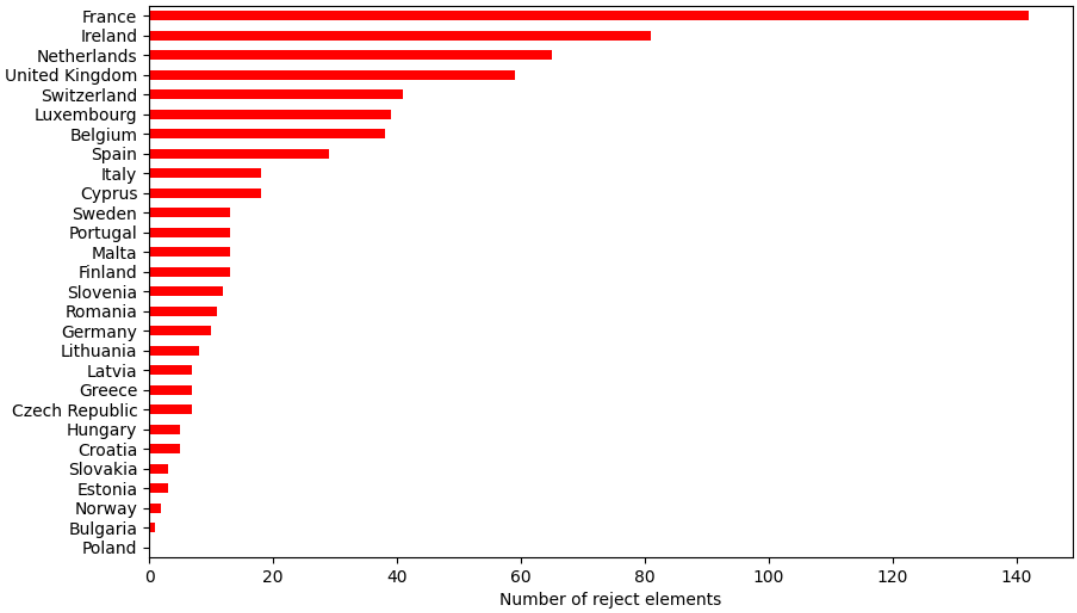


Figure 9.3: Number of reject elements per country.

guage, the text value of a reject element is often longer than the value of a consent element as it includes extra words such as "niet" or "not". Besides a few exceptions, most websites reveal no difference for the reject option height. Overall there seems to be no striking dark pattern related to the elements' width and height.

Background	Consent	Reject	Consent - reject (width)	Consent - reject (height)
			-30	0
			0	0
			0	0
			0	0
			0	0
			-34	0
			0	0
			0	0
			14	0
			0	0
			0	0
			200	2
			-28	0
			0	0
			0	0
			0	0

(a) Consent and reject HTML elements colors, width and height difference - The Netherlands.

Background	Consent	Reject	Consent - reject (width)	Consent - reject (height)
			14	0
			-20	0
			-33	0
			-20	0
			0	0
			-20	0
			14	0
			0	0
			0	0
			-20	0
			-20	0
			231	0
			-33	0
			-129	-10
			9	-6
			-234	16
			-20	0
			-117	1
			-1	0
			-20	0
			-20	0
			-20	0
			-20	0
			231	0
			-142	-2
			87	0
			-17	0
			-25	0

(b) Consent and reject HTML elements colors, width and height difference - France.

Background	Consent	Reject	Consent - reject (width)	Consent - reject (height)
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			15	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			0	0
			5	0

(c) Consent and reject HTML elements colors, width and height difference - Ireland.

Background	Consent	Reject	Consent - reject (width)	Consent - reject (height)
			-8	-26
			0	0
			80	0
			50	0
			172	22
			-12	0
			-29	0
			0	0

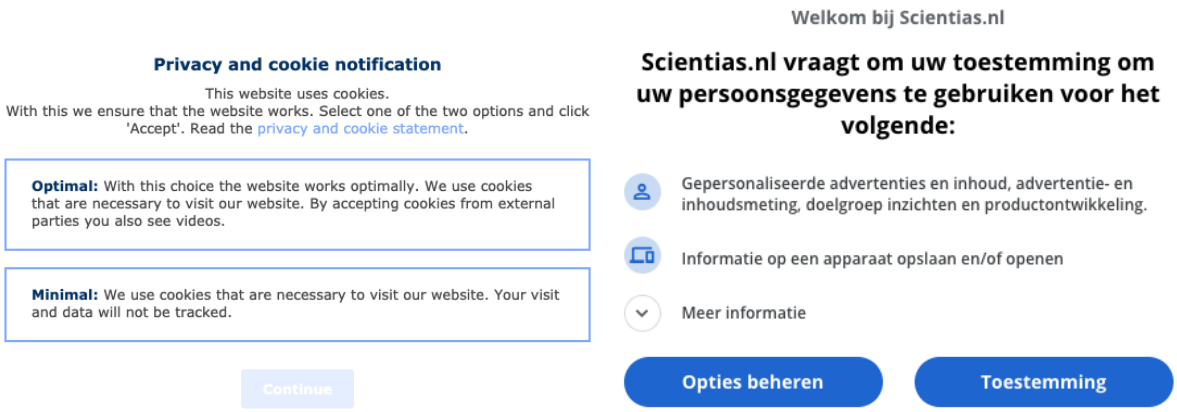
(d) Consent and reject HTML elements colors, width and height difference - Austria.

Figure 9.4: Consent and reject HTML elements colors, width and height difference.

9.3. VALIDITY

We validated our crawler results by manually visiting the first hundred websites of The Netherlands. The outcomes indicated that all explicit consent and reject options were correctly discovered. However, we encountered a few particularities.
The website www.radiologyassistant.nl shows a dialog with two options, as depicted in Fig-

ure 9.5a. Due to the text length and the fact that the options are no button or link element, our crawler did not detect these. However, the user needs to perform a second click on the button "Continue". Therefore, it is not wrong that our crawler did not detect these. Further, as we already mentioned, we do not detect options in a second or third layer. Figure 9.5b shows the cookie dialog from www.scientias.nl for which we only detect the consent button with the text "Toestemming". Also, we detected one false positive for a website that shows no cookie dialog. Our crawler detected the website due to the existence of the word "prima" in the string "Skip to primary content" which does not exceed our string limit. Our crawl does not catch such exceptions. Besides validating the presence of the elements, we also verified the color, width, and height attributes. Our first preliminary checks verify that our crawler records the width and height as recognized by the browser. The color attribute cannot always be correctly captured as the CSS attribute "background-color" is sometimes applied on a higher element in the HTML DOM structure. We observed this behavior on one website during validation.



(a) Cookie dialog of `www.radiologyassistant.nl`.

(b) Cookie dialog of `www.scientias.nl`.

Figure 9.5: Cookie dialog validation examples.

As our research is performed in a dutch environment, we have to acknowledge that we are more familiar with languages from surrounding countries. Therefore, our translation lists for other languages can contain words not often used in cookie dialogs from the respective country. We asked a native speaker from Romania to review our terms, which showed that some translations are unlikely to be used in our context. E.g., our Romanian translation "admis" is rather uncommon in the cookie dialog context. The verb form, "admite", would be more suitable. Professional interpreters should further validate the strings from our list to filter out strange translations.

9.4. Discussion

Although our detection mechanism for identifying dark patterns needs further enhancements to increase the scope, our implementation shows that it is possible to detect possible usage of dark patterns by website publishers. Our investigation focused on observing the use of balanced consent and reject choices. Many websites show a green or blue color for the consent option and a black or red color for the reject option. As previous

research [TCJ+15] shows that the use of these particular colors influences the behavior of users and which option they select, we can conclude that many website publishers consciously try to nudge users into giving their consent. As there are currently no design criteria for cookie dialogs, website publishers can use such methods for their benefit. Currently, the analysis of our gathered data is a manual process. If clear design criteria were established, these could serve as input to automate the analysis. E.g., certain ranges of RGB colors could be allowed for elements that our crawler could check. Further, the number of websites that offer a direct option to reject is disturbingly low. In the best case, users can reject in a second or third layer. Worst case, there is not option to reject at all. Such practices infringe the privacy choices of users and will often not be compliant with the regulations.

10

Conclusion

10.1. To what extent can an automatic scanning tool help perform an informed audit on cookie banners respecting the legal rules?

In this research, we explored several mechanisms to discover compliance violations in cookie dialogs automatically. Although earlier research studies already investigated the usage of cookie dialogs and possible violations according to the European legislation, our research focuses on using automation techniques that could support compliance checks performed by privacy auditors. Therefore, we used OpenWPM, a web privacy measurement framework based on Selenium that is scalable and often used in previous studies. As it is an open-source project, we extended the standard implementation with our own commands. Figure 10.1 shows the overview of the proof of concept we implemented.

First, we revealed significant differences in the number of websites that present a cookie dialog between different European countries. For The Netherlands and Belgium, our detection command indicates that 53.7% and 43.6% of websites from the top 500 .nl and .be websites worldwide display a cookie dialog. Other countries deviate from these detection numbers. In total, we scanned 14805 websites, for which we measure an average of 47.8%. Also, many websites seem to rely on JavaScript, although users are not obliged to enable it in their browser. A user who disables JavaScript will not see a large percentage of the cookie dialogs and is therefore not able to consent or decline. Further, detecting the specific usage of a Consent Management Provider (CMP) by website publishers shows that besides a few popular CMPs such as Quantcast and OneTrust, there is a high diversity between different European countries.

Next, to uncover violations, we used a ruling system based on two rules that originate from a previous study [SBM19] and the GDPR. For rule R1, we collected the cookies set before users' consent and linked them to a specific purpose from Cookiepedia. Disturbingly we found that a high number of cookies set are used for targeting and advertising purposes before consent is given. Thus, despite community efforts such as the IAB, users' data is

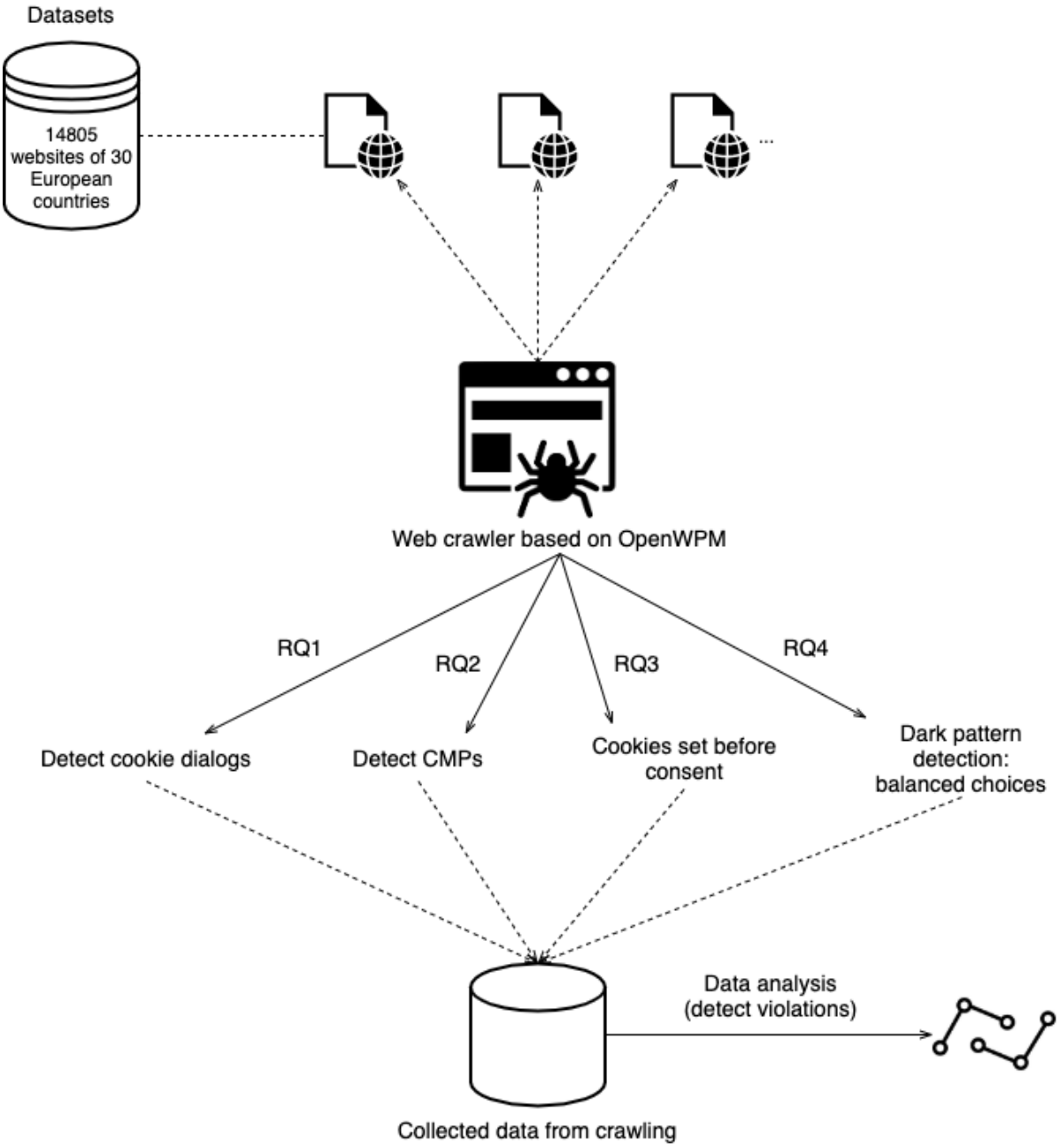


Figure 10.1: Overview of the proof of concept we implemented.

still used for illegal activities, often without their knowledge. Also, the purpose for a high number of cookies cannot be revealed, which could be problematic if the purpose is not declared on the corresponding website. Lastly, for rule R2, we automatically detected a specific dark pattern, namely the presence of balanced consent and reject options in a cookie dialog. Results showed that colors such as green and blue are regularly used for the consent element, whereby the color for the reject element is the same as the background color of the cookie dialog, or black or red. Also, many websites do not show a direct option to refuse. Such patterns are encountered in all European countries. Our focus on automation showed that it is possible to audit a large number of websites simultaneously to detect compliance violations in cookie dialogs. Although we only identified a subset of possible

violations, we believe that integrating more automation in current compliance processes could extend the detection scope. As a result, more website publishers could be triggered to follow the regulations, or at least they would be aware of existing violations.

10.2. Limitations

Cookie dialog detection. We use a predefined list of keywords from Adblock with manual additions as input to detect a cookie dialog. However, our list is not fully complete. As shown in our validation, not all cookie dialogs are found. As websites can use a wide range of keywords in their cookie dialog implementation, it is difficult to detect all cookie dialogs.

CMP detection. We base our CMP detection mechanism on using the TCF ping operation. Although it is a reliable API command, it is not available for all websites, even if it uses a CMP. Our implementation does not search within the network requests to analyze certain links or parameters to identify CMP usage without the ping operation.

Cookies. We only analyze cookies set before consent. Further, our analysis is based on the purposes provided by Cookiepedia. If a cookie is not recognized by Cookiepedia, we mark it as missing. However, the purpose of the cookie could be known by other databases or is available on the website itself.

Balanced consent and refuse options. The determination of the context color, i.e., the background color of the cookie dialog, is not an automatic process. We manually identified this color afterward for our analysis, as our crawler only selects possible consent and reject elements and not the cookie dialog itself. Further, analyzing the data for violations is still a grey area and a manual process as there are currently no fixed specifications.

System setup. For this research, we were able to run six crawlers at once. Although we can run multiple crawlers simultaneously, there is a limit on system resources. Therefore, to collect the data for all countries, multiple runs are required.

10.3. Future work

Tracking technologies. Our research is focused on the use of cookies and related dialogs. However, as new technologies constantly arise, website visitors can be tracked by other mechanisms besides cookies. Bellero et al. [BM18] investigated the privacy threats of three client storage mechanisms, namely, Web Storage, Web SQL Database, and Indexed Database API. Amongst these three, Web Storage is the most frequently used. Of the top 10K Alexa websites, 63 percent use Web Storage for tracking purposes. Consent is also needed for these tracking technologies. Further research could investigate to what extent such mechanisms are used for tracking purposes before user consent.

Data protection laws. Our research is based on European data protection legislation such as the ePrivacy Directive and the GDPR. However, other parts of the world are covered by different regulations. E.g., the California Consumer Privacy Act (CCPA) in the United States or the Brazilian General Data Protection Law (LGPD). Although some parts of these laws can have similar rules, each law will have its particularities which will affect how automation techniques can be used to detect violations. Therefore, we cannot determine to what extent our implementation is compatible with other laws. E.g., not all laws may require website publishers to use cookie dialogs. Therefore, future research could focus on different legislation, which then could be compared with our results to examine the difference in implementation techniques.

Cookies. For our examination in RQ3, we showed that the purpose for a high number of cookies could not be determined. This is problematic and is contrary to the transparency requirements from the IAB. Therefore, future work should address these unknown cookies to investigate whether they are used for tracking or advertising purposes. We performed such an analysis manually only for a few cookies. Also, we did not investigate cookies set after user consent. A more advanced implementation of our crawler is needed to click on the consent button for such an examination. In the same line, the number of cookies and their purposes could be gathered after clicking the reject option, if available, to analyze if only cookies for essential purposes are set. If no reject option is available, additional layers in the dialog could be inspected.

Dark patterns. Further, in RQ4, we detected one specific dark pattern, namely, whether balanced consent and reject options are available. Therefore, we only touched the surface of design choices website publishers can use to nudge users. E.g., Soe et al. [SNGS20] identify so-called forced action patterns where users are blocked from accessing a website. To continue, users have to click on it. Figure 10.2 shows an example of such behavior. Currently, our implementation does not detect such a pattern.

Borgesius et al. [GSZBB21] conducted online experiments to explore the effects of design nudges. They conclude that current cookie dialog implementations do not enable meaningful choices for users. Their research was recently conducted in 2021, which shows that dark patterns are still a broad issue. Further research should continue on such findings to explore how automation techniques could be integrated to support the detection of violations.

Mobile. Another possible research area is the use of cookies and dialogs in mobile environments. It would be interesting to examine if cookies are set in a different manner or quantity on a mobile phone. Also, the layout of the implemented dialogs can have different layouts due to the smaller screen size. Such differences could be intentional. A preliminary crawl for a small list of 17 websites with Puppeteer[1], a Node library used for crawling and testing, shows a marginal difference in the average cookie set. The average amount of GET

[1] https://github.com/koenae/crawler

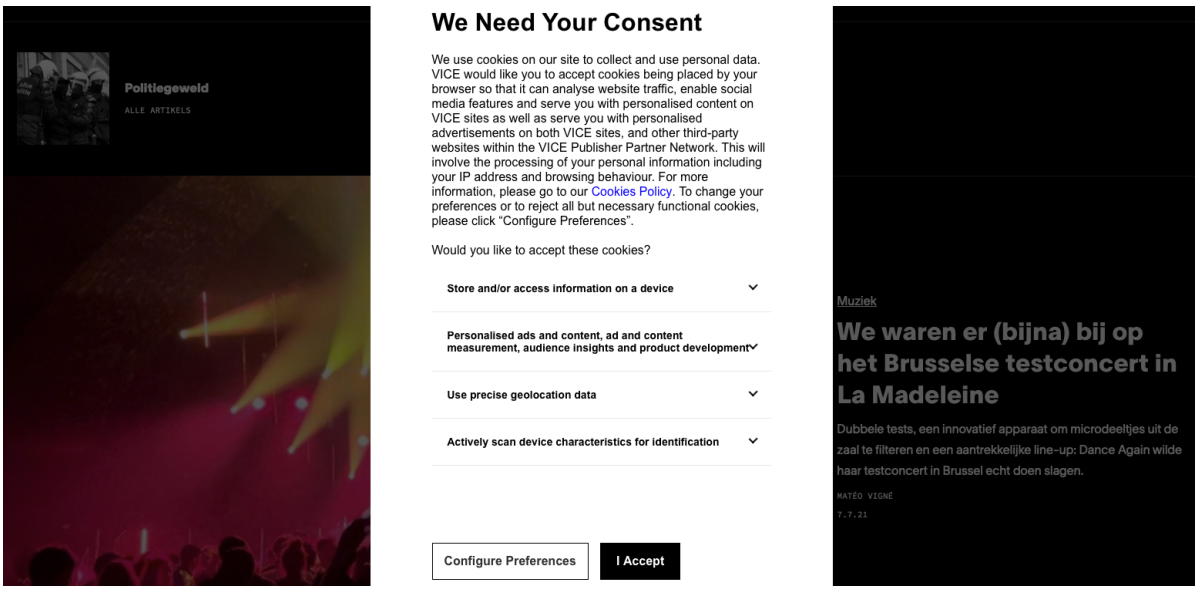


Figure 10.2: Cookie dialog from `www.vice.com` which forces a user to take an action.

requests does show a significant difference. However, such measurements are highly fluctuant and influenced by the timing of the requests. These findings could be a baseline for further in-depth research.

Research threats. Lastly, we have to take into account that website publishers could manipulate our research output. As discussed in RQ2, a CMP ping operation can be mimicked to trick the scanning process. However, this is just one example we observed. Other techniques could exist to influence the outcomes of privacy research for the benefit of the publisher. We believe that such practices are currently not widely used as automation techniques are not standard built into regulation audit processes. Nonetheless, as automation techniques evolve and have an increasing impact on detecting violations, publishers could be tempted to explore backdoors in order to appear compliant. Such practices would threaten research results. Therefore, researchers should investigate the possible threats to safeguard correctness.

BIBLIOGRAPHY

[AEE+14] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juárez, Arvind Narayanan, and Claudia Díaz. The web never forgets: Persistent tracking mechanisms in the wild. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 674–689. ACM, 2014. 5

[BM18] Stefano Belloro and Alexios Mylonas. I know what you did last summer: New persistent tracking mechanisms in the wild. *IEEE Access*, 6:52779–52792, 2018. 64

[EN16] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1388–1401. ACM, 2016. 16, 17

[EPRF17] José Estrada-Jiménez, Javier Parra-Arnau, Ana Rodríguez-Hoyos, and Jordi Forné. Online advertising: Analysis of privacy threats and protection approaches. *Comput. Commun.*, 100:32–51, 2017. 15, 16

[ERE+15] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan R. Mayer, Arvind Narayanan, and Edward W. Felten. Cookies that give you away: The surveillance implications of web tracking. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 289–299. ACM, 2015. 5

[FFM00] Batya Friedman, Edward Felten, and Lynette I. Millett. Informed consent online: A conceptual model and design principles. Technical report, CSE Technical Report, 2000. 6

[GSB+20] Colin M. Gray, Cristiana Santos, Nataliia Bielova, Michael Toth, and Damian Clifford. Dark patterns and the legal requirements of consent banners: An interaction criticism perspective. *CoRR*, abs/2009.10194, 2020. 13, 15

[GSZBB21] Paul Graßl, Hanna Schraffenberger, Frederik Zuiderveen Borgesius, and Moniek Buijzen. Dark and bright patterns in cookie consent requests. *Journal of Digital Social Research*, 3(1):1–38, Feb. 2021. 65

[JKV19] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. Fingerprint surface-based detection of web bot detectors. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part II*, volume 11736 of *Lecture Notes in Computer Science*, pages 586–605. Springer, 2019. 14, 17

[KNS09] Chetan Kumar, John B. Norris, and Yi Sun. Location and time do matter: A long tail study of website requests. *Decis. Support Syst.*, 47(4):500–507, 2009. 20

[Lei19] Bart Leiser, Mark Custers. The law enforcement directive: Conceptual challenges of eu directive 2016/680. *European Data Protection Law Review (EDPL)*, 5:367, 2019. 8

[MAF+19] Arunesh Mathur, Gunes Acar, Michael Friedman, Elena Lucherini, Jonathan R. Mayer, Marshini Chetty, and Arvind Narayanan. Dark patterns at scale: Findings from a crawl of 11k shopping websites. *Proc. ACM Hum. Comput. Interact.*, 3(CSCW):81:1–81:32, 2019. 13

[MBS20] Célestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice?: Measuring legal compliance of banners from IAB europe's transparency and consent framework. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 791–809. IEEE, 2020. 9, 11, 14

[MFF01] Lynette I. Millett, Batya Friedman, and Edward W. Felten. Cookies and web browser design: toward realizing informed consent online. In Julie A. Jacko and Andrew Sears, editors, *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems, Seattle, WA, USA, March 31 - April 5, 2001*, pages 46–52. ACM, 2001. 5, 6

[MSB20] Célestin Matte, Cristiana Santos, and Nataliia Bielova. Purposes in IAB europe's TCF: which legal basis and how are they used by advertisers? In Luís Antunes, Maurizio Naldi, Giuseppe F. Italiano, Kai Rannenberg, and Prokopios Drogkaris, editors, *Privacy Technologies and Policy - 8th Annual Privacy Forum, APF 2020, Lisbon, Portugal, October 22-23, 2020, Proceedings*, volume 12121 of *Lecture Notes in Computer Science*, pages 163–185. Springer, 2020. 9, 11

[NLV+20] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark patterns after the GDPR: scraping consent pop-ups and demonstrating their influence. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–13. ACM, 2020. 12, 13, 14

[PC00] Weihong Peng and Jennifer Cisna. HTTP cookies - a promising technology. *Online Inf. Rev.*, 24(2):150–153, 2000. 5

[PvGT+19] Victor Le Pochat, Tom van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. 18

[SBM19] Cristiana Santos, Nataliia Bielova, and Célestin Matte. Are cookie banners indeed compliant with the law? deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners. *CoRR*, abs/1912.07144, 2019. 11, 15, 16, 40, 62

[SNGS20] Than Htut Soe, Oda Elise Nordberg, Frode Guribye, and Marija Slavkovik. Circumvention by design - dark patterns in cookie consent for online news outlets. In David Lamas, Hegle Sarapuu, Marta Lárusdóttir, Jan Stage, and Carmelo Ardito, editors, *NordiCHI '20: Shaping Experiences, Shaping Society, Proceedings of the 11th Nordic Conference on Human-Computer Interaction, Tallinn, Estonia, 25-29 October, 2020*, pages 19:1–19:12. ACM, 2020. 2, 12, 65

[TCJ+15] James Turland, Lynne M. Coventry, Debora Jeske, Pam Briggs, and Aad P. A. van Moorsel. Nudging towards security: developing an application for wireless network selection for android phones. In Shaun W. Lawson and Patrick Dickinson, editors, *Proceedings of the 2015 British HCI Conference, Lincoln, United Kingdom, July 13-17, 2015*, pages 193–201. ACM, 2015. 53, 57, 61

[TS06] Mike Thelwall and David Stuart. Web crawling ethics revisited: Cost, privacy, and denial of service. *J. Assoc. Inf. Sci. Technol.*, 57(13):1771–1779, 2006. 21, 22, 23

[TSR+12] Anne N. Thorndike, Lillian Sonnenberg, Jason Riis, Susan Barraclough, and Douglas E. Levy. A 2-phase labeling and choice architecture intervention to improve healthy food and beverage choices. *American Journal of Public Health*, 102:527–533, 2012. 53

[TTBM19] Martino Trevisan, Stefano Traverso, Eleonora Bassi, and Marco Mellia. 4 years of EU cookie law: Results and lessons learned. *Proc. Priv. Enhancing Technol.*, 2019(2):126–145, 2019. 2, 12, 14

[Uzu20] Erdinç Uzun. A novel web scraping approach using the additional information obtained from web pages. *IEEE Access*, 8:61726–61740, 2020. 14