Measure the Cookie Setting Behavior of Web Pages Showing Cookie Privacy Warnings

Smilte Petronyte

4th Year Project Report
Software Engineering
School of Informatics
University of Edinburgh

2021

Abstract

Nowadays, many websites display cookie privacy warnings to conform to the General Data Protection Regulation and the ePrivacy Directive introduced by the European Union, requiring companies to obtain the user's consent before setting any non-essential cookies. For the purposes of this research, an automated web crawler was developed to systematically identify and collect cookie notices and cookie data.

In this study, I perform an empirical analysis of the cookie setting behavior and cookie notices found on the top 1,000 most visited websites in 2020. The web crawler found that approximately 54% of websites displayed a cookie notice and any interaction with it usually results in an increased number of cookies. The data analysis primarily focuses on the options presented within these cookie notices and evaluates the impact of choosing one of these options in terms of the number and the type of cookies set afterwards.

Acknowledgements

I would like to thank my supervisor Kami Vaniea for support, suggestions and guidance over the course of this project. I would also like to thank my project group for insightful discussions and ideas.

Table of Contents

Chapter 1

Introduction

Cookies have become an essential part of the internet, they enhance user's browsing experience by allowing websites to function more efficiently. Cookies help web servers to identify their clients between subsequent visits, consequently, allowing implementation of functionalities such as remembering user's language preferences, wishlists, creating personalised user interfaces and advertisements. Admittedly, this requires cookies to store some data about the visitors hence creating an environment where visitor's preferences and browsing behaviour can be easily tracked, in fact, the tracking can span across multiple websites if sufficient actors are involved. This shortcoming is often exploited by advertising and web analytics companies that use cookie-based tracking to deliver their services. User tracking raises some obvious privacy concerns since the tracking ecosystem is seldom transparent to the visitor [32], can lead to personal data leakage [16][23] and is mostly controlled by very few major companies [15][28][20].

As a result, governing bodies introduced legislation to protect users and their personal data online. In 2018, the European Union introduced the General Data Protection Regulation (GDPR) which states that in some cases cookies are considered to be personal data, also, the GDPR requires that every entity collecting personal data obtains a person's consent before storing the data. In 2002 EU introduced ePrivacy Directive, which was later amended in 2009, stating that a company must obtain the user's consent before setting any cookie that is not essential for the basic website's functionality. The consent must be freely given and obtained by a positive affirmative action and the visitor should be able to access the website even after refusing to give consent. To comply with the GDPR and ePD websites introduced cookie notices.

Research suggests that not all websites that use cookies show cookie notices [14][10][27] and others demonstrate behaviour possibly violating the GDPR despite displaying cookie dialogs. Studies have found that many websites place tracking cookies before receiving the user's consent [25][29] and the majority of cookie notices do not give visitors an option to refuse consent [33][25][29]. Majority of the research studying cookie notices focus on options presented to the users or their perception of cookie dialogs but seldom focuses on the cookie setting behaviour resulting from the interactions with the cookie dialog. The studies that do inspect the cookie setting behaviour either perform the dialog interactions manually [29][25], or are limited to the collection of only a fraction of third-party cookies [27].

Therefore, my goal was to create a web crawler capable of detecting cookie dialogs, identifying options presented to the visitor, interacting with the dialog and collecting cookies in order to measure the cookie setting behaviour prior and after interactions with the cookie dialogs. Further, I wanted to design a tool that collects data with no or very limited human interaction hence possibly serving larger-scale research projects.

Using the implemented program I performed an automated crawl on the 1,000 most visited websites in 2020[1], with the aim to answer the questions about cookie notices and cookie setting behaviour, namely:

- How often websites display cookie notices?
- How often cookie notices block the contents of the web page hence forcing the user to make a cookie management choice?
- What consent options are presented within the cookie notices? Do the consent options differ for dialogs that do and do not block website's contents? How often cookie notices nudge the visitor to give consent?
- How many websites set cookies before the visitor gives consent? And how many of those cookies could potentially be used for tracking?
- How cookie setting behaviour changes when the visitor gives or refuses consent, or closes the dialog?
- What are the most prevalent cross-site trackers upon loading the page and after giving or refusing consent?

I have found that 54.3% of the crawled websites displayed a cookie dialog and the automated tool performed dialog identification with 98% accuracy. However, system's performance results are biased towards the set of websites used during the development process, therefore, I evaluated the tool's performance on a non-overlapping subset of 100 websites. The tool detected cookie notices with the accuracy of 93.0% with no human interaction and successfully identified 92.8% of the interactive elements presented in the cookie notices. Moreover, I have found that on the first page of the dialog, 88.6% display an opt-in button and only 10.8% contain an opt-out button. In addition, 47.0% of the cookie dialogs presenting a binary choice and 66.3% of the dialgos displaying a cookie management button are nudging the visitor towards giving consent. 95.8% of websites set at least one cookie upon loading the web page and 91.7% set at least one ID-like cookie. Moreover, based on the data collected, the top 20 most prevalent trackers identified upon loading the website place cookies on an approximately equal number of websites that do and do not display cookie dialogs.

In summary, the main contributions of the work performed include:

- Designing and implementing a semi-automated cookie notice detection and analysis tool that can also work well in fully-autonomous mode.

[1]According to the Tranco [24] list available at https://tranco-list.eu/list/52XN

- A dataset of cookie notices and cookies set by the top 1,000 most visited websites.
- Measurement of differences in cookie setting behaviour upon loading the website and after interacting with the cookie notice.

Chapter 2

Background

2.1 Web Cookies

World Wide Web is probably the best-known application of the Internet. WWW is based on an application-layer protocol – HTTP, which supports the communication between a client and a server. HTTP is a stateless protocol because no information about the client is stored on the server's side. However, to improve user experience on the Internet it is helpful for a website to identify its clients between subsequent visits. This is where HTTP cookies come into place.

Cookies are an addition to HTTP that help to maintain its state. More precisely, cookies are small data structures that allow the web server to track user's activity, and are stored on the user's local hard drive as name-value pairs, which are generated upon the initial visit to the specific website. In practice cookies enhance people's browsing experience: they help the server to remember the user's language preferences or the contents of the user's shopping cart, also they allow personalized user interface and ads. In addition, cookies allow the server to continue from where it let off the last time the user visited the website.

2.2 Types of Web Cookies

HTTP cookies can be classified into different categories based on some attributes of the cookie or the purpose it serves.

2.2.1 Cookie classification based on lifetime

Based on Max-Age attribute which indicates the maximum lifetime, cookies are divided into:

- Session cookies which do not have Max-Age or Expires attribute and are stored only for the duration of the session. In other words, these are deleted after the browser is closed.

- Persistent cookies which are stored on the user's local hard drive and are accessed every time the user visits the specific web page. These can have a lifetime ranging from a few minutes to a few years and are stored until they expire or are deleted manually.

2.2.2 Cookie classification based on origin

In addition, cookies are classified according to their origin, namely into:

- First-party cookies which are set by the domain of the website that the user is visiting.
- Third-party cookies which are set by a different domain than the user is accessing.

2.2.3 Cookie classification based on purpose

Cookies can be classified based on the purpose they serve. This classification is slightly more ambiguous, however, these cookie subgroups are often mentioned in cookie consent notices and cookie policies and in general can help the user to understand how their data is collected and used when browsing the internet. Cookie classification based on their intended purpose, is closely related to cookie laws, for instance GDPR and ePrivacy Directive (see Section 2.4), where there are different requirements for storing different types of cookies. The EU outlines four cookie categories [8]:

- Strictly necessary cookies - cookies that are essential for delivering the basic website functionality.
- Preference cookies - cookies that help the web server to recall some information about the user based on their earlier choices.
- Analytics cookies - cookies that help the website to improve its functionality. Although they are used to track user's browsing activity, collected information is said to be anonymous.
- Marketing cookies - cookies that help advertising companies to select personalised ads based on users browsing patterns and activity.

2.3 Privacy concerns with regards to cookies

Cookies used for marketing or website traffic analytics purposes do track user's browsing behaviour. Analytics cookies are usually pseudo-anonymous, whereas marketing cookies assign each user an ID that can then identify them throughout the web and thus help marketing companies build user profiles that can be used for behavioural advertisement targeting. Tracking user's behaviour has become a big privacy concern since most of the tracking cookies are serviced by third-party companies, which collect and store information about the user even if the user never visits that specific company's website. It is even more concerning if the same third-party company is being used by many different websites or when multiple first-party and third-party companies use cookie-syncing. In fact, a study performed by Narayanan et al. [15] showed that most third-party companies implement cookie-syncing, however, Urban et al. [32] suggest that the number of relation between third-parties has decreased since the introduction of the GDPR. Cookie-syncing allows cross-site tracking which helps to build more comprehensive user profiles that can later be used by or sold to other advertisement companies hence it can be seen as a violation of user's privacy since the tracking process is often not transparent to the user.

2.4 EU Cookie Regulations

In 2018 European Union introduced General Data Protection Regulation (GDPR), which states that despite not containing any personal information cookies that are used as user identifiers are considered to be personal data. By GDPR companies need to obtain the user's consent before storing any of their personal information or show a legitimate interest to store data. Moreover, consent must be freely given and obtained by positive affirmative action. The ePrivacy Directive (ePD) introduced by the EU in 2002 and amended in 2009, requires websites to obtain the user's consent before setting any cookies other than those that are necessary for the basic website functionality, in addition, websites must accommodate visitors even if they refuse consent unless there is a reasonable justification not to. However, there is little guidance defining what makes a cookie strictly necessary hence this is a grey area that is yet to be resolved. GDPR and ePD were the motivation behind cookie privacy warnings displayed on many websites. These privacy warnings must identify the clear purpose and precise information of any cookies used and any data collection that occurs as a consequence. More importantly, companies must allow the user to give or withdraw consent easily at any given time in the most user-friendly way possible. When third-party cookies are used it is required to specify the organizations that are allowed to access this data. Lastly, EU suggests that websites should not store cookies with an expiration date exceeding 1 year.

The UK similarly to other European countries has its versions of the GDPR and the eDP, called UK GDPR and PECR. The regulations introduce similar principles regarding the use of cookies that hold even after the UK has left the EU.

2.5 Discussion on previous work

Extensive research in the field of user security and web privacy focuses on cookies and other tracking technologies. As a result, multiple privacy management tools, for instance, Ghostery[1] and Disconnect[2], were introduced to help minimize the prevalence of the online trackers. Englehardt et. al [15] built a tracking measurement platform OpenWPM and analysed different tracking technologies, including cookies and cookie-syncing, over the top 1-million websites, while evaluating the effectiveness of previously mentioned privacy management tools. S. Mattu [26] created a web application Blacklight based on OpenWPM which demonstrates what tracking technologies

[1] *Ghostery*. URL: `https://www.ghostery.com`.

[2] *Disconnect*. URL: `https://disconnect.me/`.

are used in different websites. Similarly other studies on cookies focus on privacy issues arising from the use of third-party cookies that are used to track users online. Englehardt et al. [16] showed that if the number of unrelated third-party identifying cookies is high then they can be connected and used to track a user across the internet. Hu et al. [20] measured the interconnectedness of the third-party cookies and suggested that it is mostly influenced by a few major companies. Roesner et al. [28] investigated the capabilities of cookie-based tracking and found that some third-party cookie setting techniques are resistant to third-party cookie blocking.

Since the introduction of the GDPR and later amendment of ePrivacy Directive researchers inspected cookie dialogs and cookie setting behaviour, while evaluating the effects of the regulations and identifying possible violations. Eijk et al. [14] extended OpenWPM to automatically detect cookie notices using I don't care about cookies[3] CSS selector list. This extended tool was used to crawl 1800 websites, failed to connect to approximately 15% and found a cookie dialog on 40.2% websites with the accuracy of 91%. The lower accuracy is due to some limitations of the implementation of the tool, namely the CSS selector list not including selectors corresponding to cookie-walls. However, neither OpenWPM or the later version of the tool explores cookie dialogs by investigating how dialog interactions impact cookie-based tracking.

Dabrowski et al. [9] compared cookie setting behaviour for EU and USA based visitors before and after the GDPR. Study concludes that overall the number of cookies decreased for both EU and USA users. However, the EU visitors experience a greater effect of the GDPR since 49.3% of the Alexa Top 1,000 websites[4] that do set cookies upon loading the website for visitors outside EU, do not set any cookies for users based in EU. Degeling et al. [10] discussed the effect of the GDPR fot the top 500 websites in 28 EU countries, focusing on changes in privacy policies and cookie banners. The study concluded that 63.2% of websites display a cookie dialog and the majority of the cookie banners are confirmation only or do not display any consent options. Further, Degeling et al. concluded that it is challenging to generate fully compliant cookie notices even when using cookie consent libraries and highlighted the ambiguity of strictly necessary cookies since the definition can be used to website owners advantage.

Matte et al. [25] performed semi-automatic crawl on approximately 23,000 EU-based websites, identifying cookie dialogs and measuring if the cookie dialogs and cookie setting behaviour conforms to the GDPR and the ePrivacy Directive. The study focuses on cookie banners generated by known Consent Management Providers (CMPs) and the banner identification is automated, however, it did not include other cookie dialogs and the dialog interactions are performed manually. Cookie banners generated by CMPs were found on 6.2% of the crawled websites. Moreover, the inspection of the subset containing 560 banner displaying websites suggests that 6.8% do not give the visitor an option to opt-out. 9.9% of the websites register consent before visitor's agreement, for at least one and 5.3% register consent for all five of the following cookie purposes: information storage, personalisation, advertising, content selection or analytics. Further, 4.8% of the websites store consent after the user chooses to opt-out.

[3] Daniel Kladnik. *I don't care about cookies.* URL: `https://www.i-dont-care-about-cookies.eu/`.

[4] *Alexa Top Sites.* URL: `https://www.alexa.com/topsites`.

Last, the research highlights some of the dark patterns of the cookie banner design, for example, hiding the second page of the cookie notice, which contains the opt-out button, by using ambiguous names for the cookie settings buttons, for instance, *Learn more*. Continuation of the previous study performed by Utz et a. [33] focuses on user's perception on the choices and information displayed within the dialog, dialog's location and design. In addition, the study presents statistical information on the sample of 1,000 cookie dialogs found in the former study. These differ from the findings of this study and are briefly discussed in Section 5.1.

Sanchez-Rola et al. [29] manually inspected 2,000 websites, focusing on privacy policies and cookie notices, and tried to opt-out from the use of tracking cookies. The study found that 92% of websites perform some form of cookie-based tracking. Moreover, the influence of the GDPR can also be seen on US-based websites, whereas China-based websites did not contain any cookie notices or consent options. In addition, very few cookie banners display an opt-out button and clicking opt-out button does not result in a decreased number of cookies. The findings suggest that sometimes tracking cookies are set prior to receiving the user's consent and since opting-out has no effect on the tracking cookies that have been saved, the user does not have an option to opt-out even if such option is presented in the cookie notice. Last, B. Molnar [27] built a semi-automated web crawler that identifies cookie notices with 92.1% precision and allows cookie collection prior and after interactions with a cookie notice. However, the tool is limited to the collection of only a fraction of third-party cookies.

Chapter 3

Design and Implementation

The main goal of the project was to collect cookie dialogs, identify the options user has and measure how cookie setting behaviour changes depending on the interactions with the dialog. This requires web scraping large amount of web pages and performing this manually is time consuming, therefore, I have build semi-automated tool which collects all the necessary data. The manual steps help to achieve higher accuracy when identifying cookie dialogs and evaluate the correctness and reliability of the data collected. These are recommended but not required, thus the tool can be used as fully automated if a greater error margin is allowed.

3.1 Requirements

System design and implementation decisions are based on the functional and non-functional requirements. Iterative requirement verification and validation during the development process ensured the final system is of the expected functionality and quality.

3.1.1 Functional Requirements

The system should autonomously crawl a list of websites, collect cookies and detect cookie dialogs. If cookie dialog is displayed, the system should identify the interactive elements (buttons and hyperlinks) on the first page of the dialog and measure changes in cookie setting behaviour when different interactive elements are clicked. In addition, the system should have a user interface to assist the user with manual cookie dialog validation and identification.

3.1.2 Non-functional Requirements

The main purpose of the system is to crawl a variety of websites hence it should have certain properties, namely robustness, fault tolerance and recoverability, to allow the system to successfully operate in this dynamic environment. Network connection and website server errors can have an impact on the response time and the reliability of the data collected, thus failures should be expected at any time of the execution. The system crawls an arbitrary number of websites every time it is executed hence it should be stateful to ensure recovery when failures occur and it should keep the copy of the data collected to prevent data corruption or loss.

Further, the time needed for each of the main tasks, namely identifying the cookie dialog and interacting with it, should not exceed thirty minutes. The time limit on individual tasks ensures that the program executes within a reasonable amount of time, also, it helps to deal with unexpected system failures, for example, unresponsive WebDriver that needs to be restarted. Last, it could be argued that it is more cost-efficient to manually inspect a website than wait over 30 minutes.

Moreover, it is desirable to achieve cookie dialog detection accuracy of at least 70%, however, this figure might not be achievable without manual interaction for any truly random arbitrary set of websites due to the fast-changing nature of web development and other problems discussed later (Sections 3.4.1.2 and 4.1).

3.2 Data sets

3.2.1 Top Websites

To identify the most popular websites, I have used the Tranco rankings[1]. It combines rankings of the three providers: Alexa[2], Cisco Umbrella[3] and Majestic[4]. Tranco rankings are designed to minimize the effect of website traffic manipulation, hence resulting in a more stable and consistent list over time. For the purpose of this study, I chose to use the list of most popular websites in 2020. It should be noted that the list includes multiple domains that only differ by TLDs (e.g. google.com and google.co.uk) and domains with their corresponding shortened versions (e.g. facebook.com and fb.me).

3.2.2 Cookie Dialog Identifiers

One of the system requirements was to have an automated program, therefore it was necessary to perform cookie dialog identification with no human interaction. Every web page uses custom HTML structure and element naming practices, hence making it more difficult to find a systematic approach. Other research in the field, suggests using community maintained CSS selector lists, which are designed for browser extensions that hide cookie dialogs [14] [27].

I have used combination of two CSS selector lists, namely EasyList[5] and I don't care

[1]The Tranco list generated on 4th of January 2021 available at `https://tranco-list.eu/list/52XN`

[2]*Alexa Top Sites*. URL: `https://www.alexa.com/topsites`.

[3]*Cisco Umbrella 1 Million*. URL: `https://umbrella.cisco.com/blog/cisco-umbrella-1-million`.

[4]*The Majestic Million*. URL: `https://majestic.com/reports/majestic-million`.

[5]*EasyList*. URL: `https://easylist.to/`.

about cookies[6]. Combination of these lists results in two sets of CSS selectors: domain specific, which are stored as domain, CSS selector pairs, and global CSS selectors. Although designed for a different purpose, these together with certain assumptions and validation procedures have proven to work relatively well for cookie notice detection (more details in Section 3.4.1.2).

3.3 Browser Automation

Although web crawling process is automated, it is important that all web interactions appear similarly to those of a human agent. To simulate this behaviour the program uses Selenium Chrome Webdriver (version 89.0.4389.23), which allows automated interaction with web pages and the Chrome browser itself, such as connecting to web pages, clicking DOM elements, taking screenshots and collecting cookies, all of which are essential for the purpose of this study. The reason for using Google Chrome browser for this study is two-fold. First, it has the highest browser market share among desktop users [4]. Second, at the time of the study Chrome has looser privacy settings by default compared to other popular browsers. For example, its main competitors Firefox and Safari have features to reduce the tracking activity, including blocking third-party cookies by default [31][35]. In addition, Safari restricts the expiration date of cookies by setting it to a maximum of 7 days [35]. Since I want to capture cookie setting behaviour as it is and collect all cookies set by first and third parties without altered retention periods, Chrome browser's behaviour is desirable.

When building any web automation tool it is a good practice to consider each website's robots exclusion protocol specified in robots.txt file. However, this tool is essentially a simple web crawling tool: it only visits the home page of each website and the interaction speed does not exceed the capabilities of a human user, therefore I have decided to disregard robots.txt file.

3.3.1 Visiting websites

The Tranco list contains only the domain names of the websites (e.g. domain.com), however, Selenium ChromeDriver requires a complete URL to open a website. Since not all domains use HTTPS yet, the system constructs 4 different URLs and then tries to open a website in the order specified:

1. https://www.domain.com
2. http://www.domain.com
3. https://domain.com
4. http://domain.com

Only when opening all URLs results in a connection failure the connection is considered unsuccessful and the domain is excluded from further analysis.

[6]Daniel Kladnik. *I don't care about cookies.* URL: `https://www.i-dont-care-about-cookies.eu/`.

3.3.2 Unsuccessful Connections

The Selenium, which is used for web automation, is detectable, thus some websites block web requests sent by ChromeDriver, making the domain unreachable to the web crawler. These and other connection failures are detected by inspecting the header of the web page, which in Chrome is set to *This site can't be reached* whenever a network error occurs, or checking for HTTP error codes. In addition, I have noticed that there are web pages that redirect to Google search engine despite having a domain name which does not indicate any relation to Google (e.g breedsthey.com), thus these are also considered to be unsuccessful connections.

Although, automatic procedures identify most of the websites that should be excluded from this study, there are few exceptions, where human judgement is required, namely: content delivery websites or other websites that are not meant for users browsing the internet, as well as web pages that exist even though they mention that the service is no longer available or unavailable in the visitor's area. In addition, some websites use Capthca tests, such as reCAPTCHA, to distinguish real users from malicious bots. It is difficult to predict when Capthca tests will occur and automated Captcha bypassing scripts do not conform to ethical programming practices hence are not implemented. Therefore, it is left as a responsibility of the user running the program to identify such cases and exclude domains from the study.

3.3.3 Cookie collection

Selenium WebDriver API provides a method to collect cookies, however, WebDriver's interactions are limited to the scope of current domain, thus the majority of third-party cookies remain uncaptured. To get more accurate measurements, system takes advantage of Chrome DevTool Protocol command Network.getAllCookies[7] which can be executed using the ChromeDriver.

When collecting cookies it is important to allow the website a reasonable time to set all the cookies, thus the program waits 30 seconds after loading the website before collection. Afterwards, the cookies are collected in 30 second intervals until the number of cookies present does not change between two latest intervals or until the cycle is repeated five times.

To distinguish between cookies set by different websites, or even by the same website when interacting with different dialog options, the cookies are cleared before every collection, as the method mentioned above returns all cookies present in the browser at the time.

[7] *Chrome DevTools Protocol.* URL: `https://chromedevtools.github.io/devtools-protocol/`.

3.4 System structure

The system is divided into two smaller subprograms, each corresponding to a different functional requirement. Since the consecutive execution of the two subprograms form the whole system, this can be viewed as the program running in two phases, namely:

1. Phase 1: Visit the websites, collect cookies and identify cookie dialog if present.
2. Phase 2: Visit the websites displaying a cookie dialog, interact with a cookie dialog and collect cookies after each action.

The specification of the system does not require two-phase design, however, the separation of these processes allows user to perform some manual procedures, namely:

1. Validate correctly identified cookie dialogs, detect false positives and websites where the dialog was not found.
2. Manually add cookie dialog CSS tags.

Since the goal of the project was to design an automated system, these manual procedures are optional, however, for the purpose of this study it was important to find majority of the cookie dialogs and make sure that the data collected is indeed correct, therefore both of these steps were performed.

Lastly, from the developer's perspective, two-phase system allows to test both subprograms in isolation. As each subprogram has different functionality, testing them separately helps to determine which functional requirements are satisfied.

3.4.1 Phase 1: Visit top websites, collect cookies and identify cookie dialog if present.

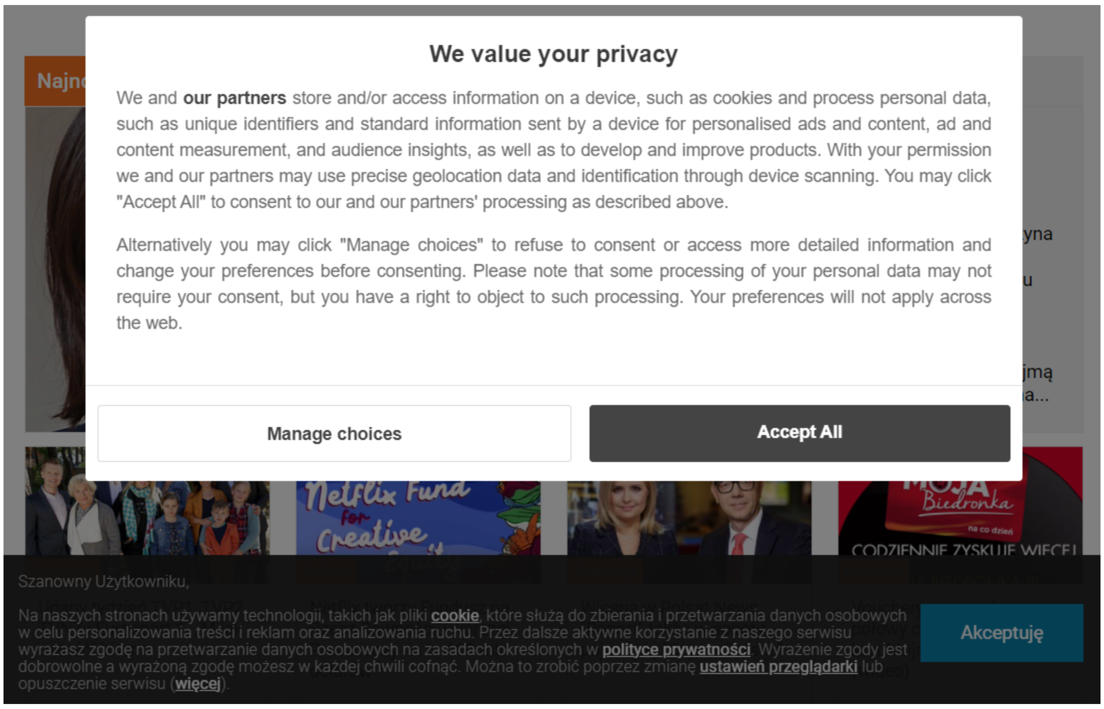
The main objective of the Phase 1 is to accurately identify the cookie dialogs, this includes detecting the dialogs as well as minimizing the number of falsely identified non-dialog elements [8].

For the purpose of this study it was important to capture the exact position and size of the dialog in addition to its contents, including text and buttons, therefore, this study required extensive manual dialog analysis and tagging which normally would not be necessary.

3.4.1.1 Assumptions about Cookie Dialogs

First, I assume a website displays at maximum one cookie dialog. Although this sometimes may not be the case (see Figure 3.1), it is difficult to differentiate between the two elements representing the same dialog and two different dialogs without manual verification. In addition, interacting with multiple dialogs increases the number of requests sent to the website possibly raising the likelihood of Captcha tests and adds complexity in the Phase 2 and data analysis, therefore, I have decided that it is beyond the scope of this study.

[8] Later I refer to these as True Positives and False Positives respectively.



We value your privacy

We and **our partners** store and/or access information on a device, such as cookies and process personal data, such as unique identifiers and standard information sent by a device for personalised ads and content, ad and content measurement, and audience insights, as well as to develop and improve products. With your permission we and our partners may use precise geolocation data and identification through device scanning. You may click "Accept All" to consent to our and our partners' processing as described above.

Alternatively you may click "Manage choices" to refuse to consent or access more detailed information and change your preferences before consenting. Please note that some processing of your personal data may not require your consent, but you have a right to object to such processing. Your preferences will not apply across the web.

Szanowny Użytkowniku,

Na naszych stronach używamy technologii, takich jak pliki cookie, które służą do zbierania i przetwarzania danych osobowych w celu personalizowania treści i reklam oraz analizowania ruchu. Przez dalsze aktywne korzystanie z naszego serwisu wyrażasz zgodę na przetwarzanie danych osobowych na zasadach określonych w polityce prywatności. Wyrażenie zgody jest dobrowolne a wyrażoną zgodę możesz w każdej chwili cofnąć. Można to zrobić poprzez zmianę ustawień przeglądarki lub opuszczenie serwisu (więcej).

Figure 3.1: Website displaying two cookie dialogs: one in the center and one at the bottom of the web page. Source: wirtualnemedia.pl

Second, if the cookie dialog is not visible upon loading a website, I assume that the website does not display a cookie dialog. This includes web pages where there is another pop-up window requiring interaction before loading the dialog or a user is required to choose a country specific website.

Third, similarly to the observation made by Sanchez-Rola et al. [29] that cookie notices on China-based websites are uncommon, I have noticed that East Asian websites rarely display cookie dialogs. Many countries outside the EU do not have strict regulations with regards to cookies, in addition, most of the East Asian websites appear to be targeted to the domestic audience. This is likely to be the reason why very few websites show a cookie dialog. There are many Chinese, Korean and Japanese websites that score high places in top rankings and extensive search for a cookie dialog that does not exists consumes a large amount of resources, therefore, I have assumed that websites containing Chinese, Korean or Japanese characters in the title do not contain cookie dialogs.

Last, websites contain different types privacy warnings and cookie setting tools, therefore it is important to define a cookie dialog in the scope of this study. Cookie dialog is a pop-up window or a banner which clearly states that cookies or tracking tools are used, or includes information on storing or using user's personal data. In addition, the contents of a cookie dialog are focused on cookies hence the decision of an interaction is based on privacy preferences and nothing else, however, this requires human judgement hence is not detected automatically. Examples are included in Appendix A.

3.4.1.2 Identifying cookie dialogs

The system uses Beautiful Soup[9] library to parse and search website's HTML code, as it is considered to be more time efficient than similar methods from the Selenium WebDriver API. However, some cookie dialogs are wrapped within an `iframe` and the HTML code inside is not visible to ChromeDriver, and consequently to Beautiful Soup. Therefore, the system needs to iterate through `iframe` elements and search for the dialog inside. There might be an arbitrary number of `iframe` tags, some of which could be nested, and exploring each adds time complexity, thus only level 1 of HTML tree structure is explored.

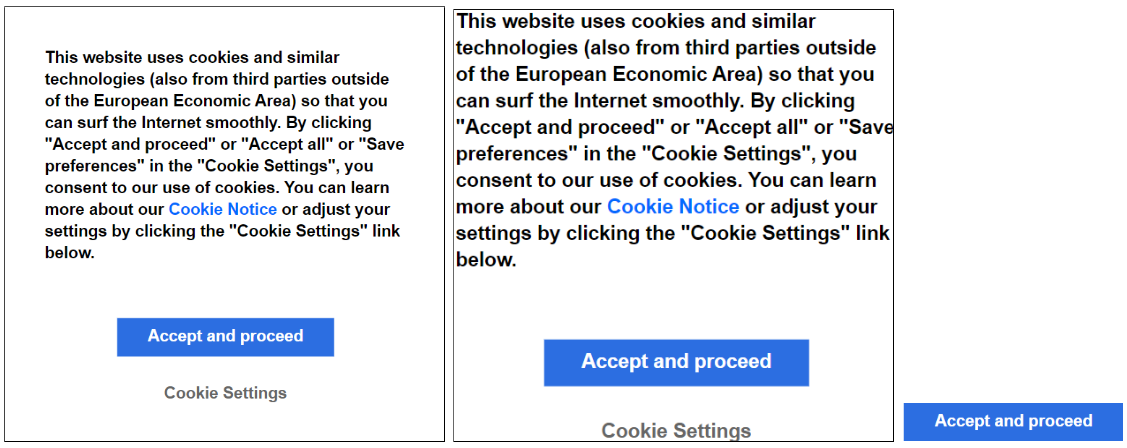
For cookie dialog identification system uses two sets of CSS selectors, namely, domain specific and global (Section 3.2.2). The global CSS selector list consists of almost 18,000 entries, hence at the worst-case the program needs to search the web page HTML approximately 18,000 times. Further, the number of iterations increase linearly, depending on the number of `iframe` elements explored. In fact, the average-case scenario is likely to be close to the worst-case since the websites are updated frequently, the domain specific selectors can become out of date, in addition, the system iterates over global selector list for all websites that do not display a cookie dialog. To illustrate, only 3.7% of crawled websites contained a cookie dialog which was successfully identified by a domain specific CSS selector.

First, if visited domain is in the domain specific list, the system searches for an element with the specified CSS tag, this includes inspection of the `iframe` elements. If no dialog elements are found using the domain specific CSS selector or the domain is not in the list, the system iterates through global selectors and find all possible matches. Last, if no dialog elements are found, the system inspects the `iframe` elements using global CSS selectors. Since this step computationally expensive, it is executed only when absolutely needed.

When system identifies a possible cookie dialog, it saves the CSS tag and takes a screenshot of a dialog for manual verification. Since the system iterates the global CSS selector list in full, it might find multiple elements and it is user's responsibility to decide if any of these elements represent a cookie dialog, by inspecting the screenshots. Alternatively, when system is executed in fully-automated mode, it assumes the first element found is the dialog.

Moreover, by assumption stated in Section 3.4.1.1 a website can only display one dialog and in fact it is rare for the system to find more than one distinct dialog per website. Usually when multiple cookie dialog elements are found all of these are different HTML nodes of the same dialog. To illustrate some of these nodes might be ascendants or descendants of the actual dialog, thus their visual representation and captured contents (text and buttons) might differ (see Figure 3.2). Admittedly, in some cases it does not matter which HTML node is chosen to represent the cookie dialog as long as it captures its contents and is of a similar size as the cookie dialog displayed (see Figure 3.3).

[9] *Beautiful Soup*. URL: `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`.



This website uses cookies and similar technologies (also from third parties outside of the European Economic Area) so that you can surf the Internet smoothly. By clicking "Accept and proceed" or "Accept all" or "Save preferences" in the "Cookie Settings", you consent to our use of cookies. You can learn more about our Cookie Notice or adjust your settings by clicking the "Cookie Settings" link below.

This website uses cookies and similar technologies (also from third parties outside of the European Economic Area) so that you can surf the Internet smoothly. By clicking "Accept and proceed" or "Accept all" or "Save preferences" in the "Cookie Settings", you consent to our use of cookies. You can learn more about our Cookie Notice or adjust your settings by clicking the "Cookie Settings" link below.

Figure 3.2: Three dialog elements found: the leftmost element represents the cookie dialog best, whereas the element in the middle is a valid option only if the size of the element does not have to resemble the exact size of the cookie dialog. Since the rightmost element does not include the text, it is considered to be poor a dialog representation. Source: docusign.com

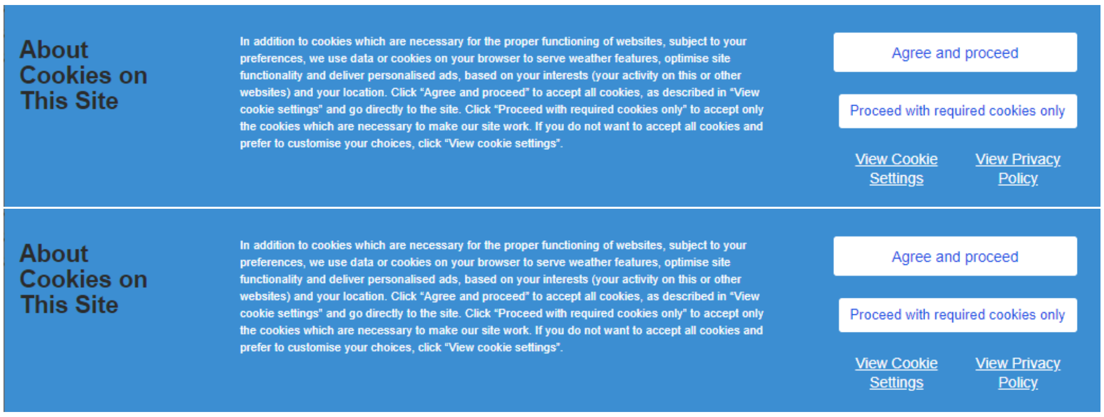
3.4.1.3 Cookie Dialog Identification Improvements

The method described above does not find all the dialogs and there is a significant number of falsely identified non-dialog elements. However, careful result analysis and manual inspection of the HTML code of many websites lead to further improvements.

unigrams	bigrams	trigrams
cookie	collects data	your personal data
cookies	personalised ads	
track	personalised content	
tracking	personal information	
	legitimate interest	
	improve products	
	your consent	
	your preferences	
	precise geolocation	
	geolocation data	
	your experience	

Table 3.1: Most common n-grams in 500 cookie dialogs.

First, to reduce the number of false positives, I added textual verification step using a modified list of the most common n-grams generated using around 500 cookie dialogs. Only the n-grams that are likely to help distinguish a cookie dialogs from other elements where chosen (see Table 3.1). An element found using non-specific CSS selector is considered to be a cookie dialog if it contains at least one of these n-grams. In addition, the text is required to be of a certain length, this helps to discard smaller elements such as buttons which would pass text verification otherwise (see rightmost


About Cookies on This Site

In addition to cookies which are necessary for the proper functioning of websites, subject to your preferences, we use data or cookies on your browser to serve weather features, optimise site functionality and deliver personalised ads, based on your interests (your activity on this or other websites) and your location. Click "Agree and proceed" to accept all cookies, as described in "View cookie settings" and go directly to the site. Click "Proceed with required cookies only" to accept only the cookies which are necessary to make our site work. If you do not want to accept all cookies and prefer to customise your choices, click "View cookie settings".

About Cookies on This Site

In addition to cookies which are necessary for the proper functioning of websites, subject to your preferences, we use data or cookies on your browser to serve weather features, optimise site functionality and deliver personalised ads, based on your interests (your activity on this or other websites) and your location. Click "Agree and proceed" to accept all cookies, as described in "View cookie settings" and go directly to the site. Click "Proceed with required cookies only" to accept only the cookies which are necessary to make our site work. If you do not want to accept all cookies and prefer to customise your choices, click "View cookie settings".

Figure 3.3: Two dialog elements which look identical despite being different HTML nodes. In this case both elements are good representations of the cookie dialog. Source: wunderground.com

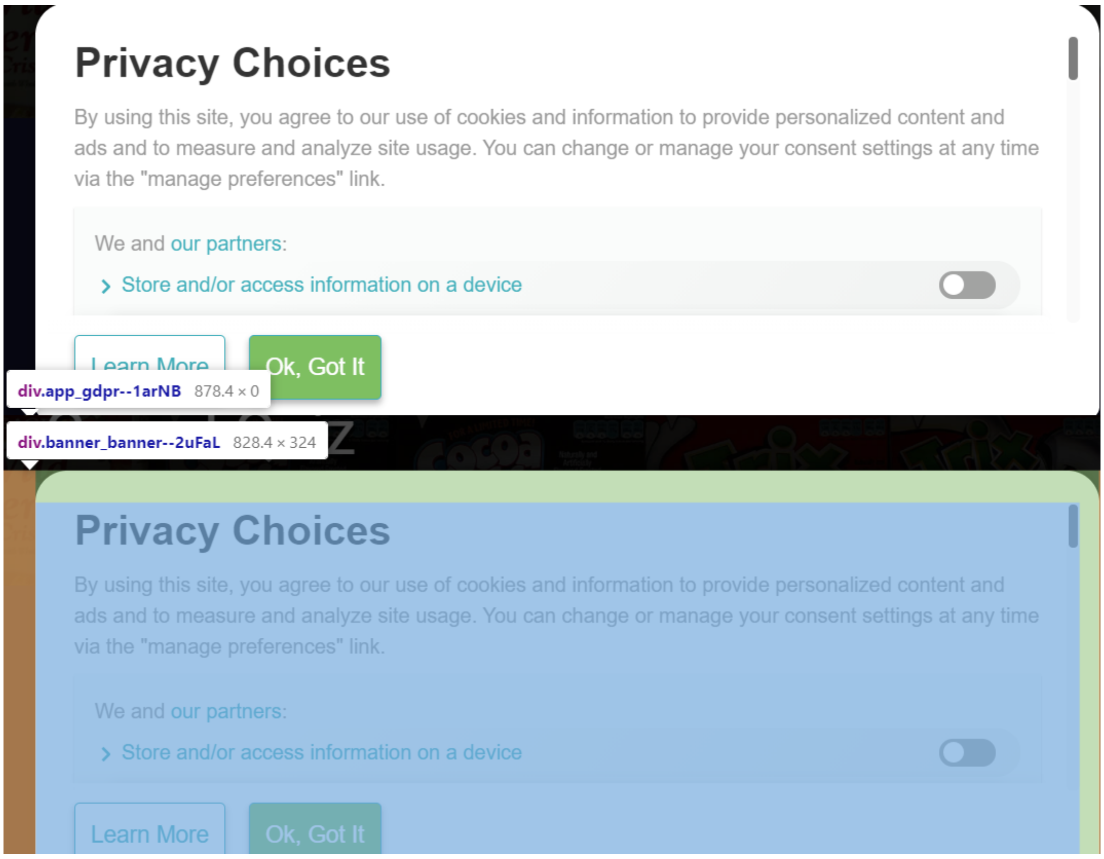
element in Figure 3.2). Since the list consists of English n-grams only, non-English elements are translated by calling Google Translate API[10] prior to performing textual verification. Using translator creates some challenges in textual verification method, for instance, word ordering in a sentences differ for different languages and there is no guarantee the translator will account for this hence bigram or trigram search might not be as effective. One of the possible solutions would be to extend the n-gram list beyond English language, however, this requires a comprehensive data set covering multiple languages in addition to ability to understand the language itself. Admittedly, any of the n-grams could appear at any place on the web page, however, the system only investigates candidate cookie dialogs, therefore, it is very unlikely that an element found using a typical dialog CSS selector and containing a typical dialog text is not a cookie dialog.

Second, although element screenshots are vital for manual verification, the process of taking a screenshot may introduce faulty behaviour. Selenium throws ScreenshotException when it is called to take a screenshot of an element having length or height equal to zero. This usually happens in one of the following scenarios:

- There is a cookie dialog that is purposely not being displayed, for instance, a website might contain a separate dialog for a mobile device, or a dialog used in an older version of the web page. In this case, system benefits from ScreenshotException since the element is indeed not an actual cookie dialog.
- There is a zero sized wrapper containing a cookie dialog that itself is visible (see Figure 3.4). This is an issue since the raised ScreenshotException causes system to fail when finding the dialog.

Therefore, I implemented additional functionality allowing system to inspect the children of a possible cookie dialog when the ScreenshotException is raised (see Figure 3.4). Similarly to the process of finding a cookie dialog, children elements are required to be visible and pass textual verification.

[10] *Google Translate API for Python*. URL: `https://pypi.org/project/googletrans/`.


Privacy Choices

By using this site, you agree to our use of cookies and information to provide personalized content and ads and to measure and analyze site usage. You can change or manage your consent settings at any time via the "manage preferences" link.

We and our partners:

Store and/or access information on a device

Learn More
Ok, Got It
div.app_gdpr--1arNB 878.4 × 0
div.banner_banner--2uFaL 828.4 × 324

Privacy Choices

By using this site, you agree to our use of cookies and information to provide personalized content and ads and to measure and analyze site usage. You can change or manage your consent settings at any time via the "manage preferences" link.

We and our partners:

Store and/or access information on a device

Figure 3.4: System finds a dialog wrapper of size zero (top) using a CSS selector `div[class^="app_gdpr-"]`. Then by inspecting children elements it finds the actual cookie dialog (bottom), its size and location. Source: howstuffworks.com

Third, manually investigating websites where the system failed to find a cookie dialog helped me to recognize patterns within the dialog's HTML structure and HTML element naming conventions. A lot of websites use Consent Management Providers (CMPs) to manage cookies and display cookie notices. Each CMP use a very similar dialog structure and HTML element names across all websites using their services hence making it easier to form a generalised approach in finding the cookie notices. Therefore, I extended the global CSS selector list by adding CSS selectors for CMP cookie dialogs I have observer. In addition, I added some selector specific behaviour, for instance, some cookie dialogs created by one of the CMPs can be found using a selector `.truste_box_overlay`, however, the dialog text and buttons are hidden within an `iframe`. Therefore, ChromeDriver needs to switch to the `iframe` before accessing the contenst of the cookie dialog. This is essential for performing textual verification and interacting with buttons in Phase 2. To illustrate, all improvements mentioned increased the number of cookies dialogs found by a global CSS selector from 191 to 371.

Last, I formed a list of keywords (see Figure 3.5) that could help to identify the cookie dialogs that were missed otherwise. If there are no dialog elements found using the specific and the global selectors, the system finds the HTML nodes that are most likely to contain a cookie dialog, more specifically, the `div` nodes, and iterates through them searching for an element with `class` or `id` attribute containing any of the keywords. If such element is found and it passes the textual verification it is likely to be a cookie dialog. Consequently, a CSS selector identifying the cookie dialog is formed to make the search process more efficient during the repeated executions or in the Phase 2.

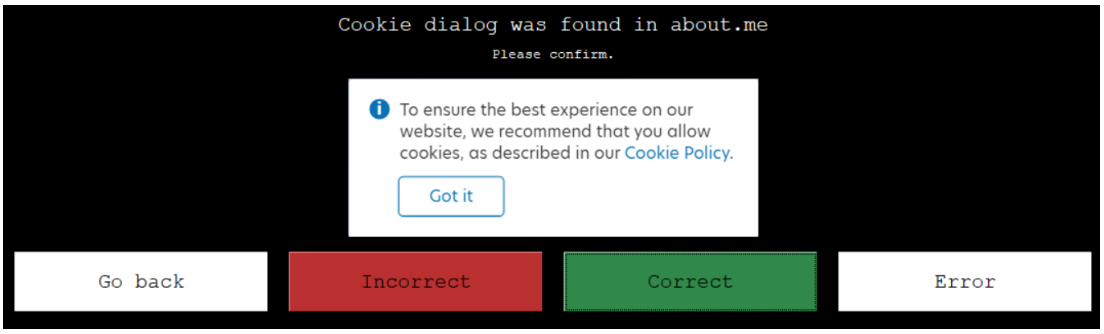
Keywords: `gdpr`, `cookie`, `privacy`, `policy`, `consent`, `notice`.

Figure 3.5: List of keywords that help identify cookie dialogs.

3.4.2 Manual Verification

Manual verification step provides user with an interface to help evaluate cookie dialog identification results, find errors or Captcha tests and choose the most appropriate CSS selectors for cookie notice detection. Further, verification of the cookie dialogs gives the user an opportunity to decide whether each element should indeed be classified as a cookie dialog.

For each website successfully crawled website, the user is presented with a screenshot and four choices, namely go back, incorrect, correct and error (see Figure 3.6). First, the program iterates through websites where a cookie dialog was found and requires the user to make a decision based on the cookie dialog screenshots. If there are multiple elements found using different selectors, the program iterates through multiple screenshots until the user identifies the correct one. Since the CSS selector is embedded within the metadata of the corresponding .png file, the list of the domain specific CSS selectors is modified simultaneously. Alternatively if the element found is not a cookie dialog, the screenshot of the full web page is displayed to confirm that the website indeed does not contain a cookie dialog. Subsequently, the program iterates through the websites where no dialog elements were found and asks the user to confirm that the web page indeed does not display a dialog.


Cookie dialog was found in about.me

Please confirm.

To ensure the best experience on our website, we recommend that you allow cookies, as described in our Cookie Policy.

Figure 3.6: UI for manual verification step. Here user is presented with a cookie dialog screenshot and four buttons for classification. Source: about.me

As a result, accuracy is computed, the list of domains containing cookie dialog is altered and another list of domains where the dialog was not found and hence require manual tagging is created.

There are certain limitations of this method. A screenshot is the visual representation of a web element hence inspecting many screenshots is fast and easy, however, a screenshot might create a false idea of what information is inside the captured HTML element. As a result, system might fail to find some or all the contents of the dialog in the Phase 2. For example, in Figure 3.7 the system fails to find the close button since it is not a descendant of the found HTML node. Although this rarely happens, it is important to acknowledge the fact, that not all system faults can be identified by simply inspecting cookie dialog screenshots. Thus, the manual verification tool outputs accuracy with an arbitrary error, which varies depending on the dataset but is expected to be small. Cookie notice identification accuracy is an approximate indicator of how well the system is performing rather than the precise measure. Alternatively, a user could be presented with HTML code in addition to a screenshot, however, manual inspection would require more time and resources which contradicts the main objective of building an automated system.

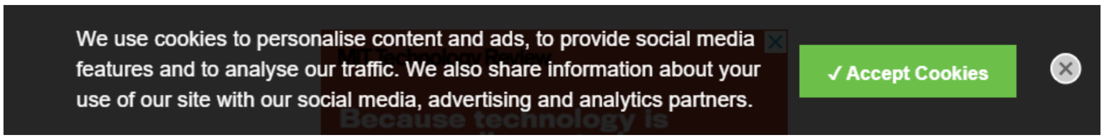

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners.

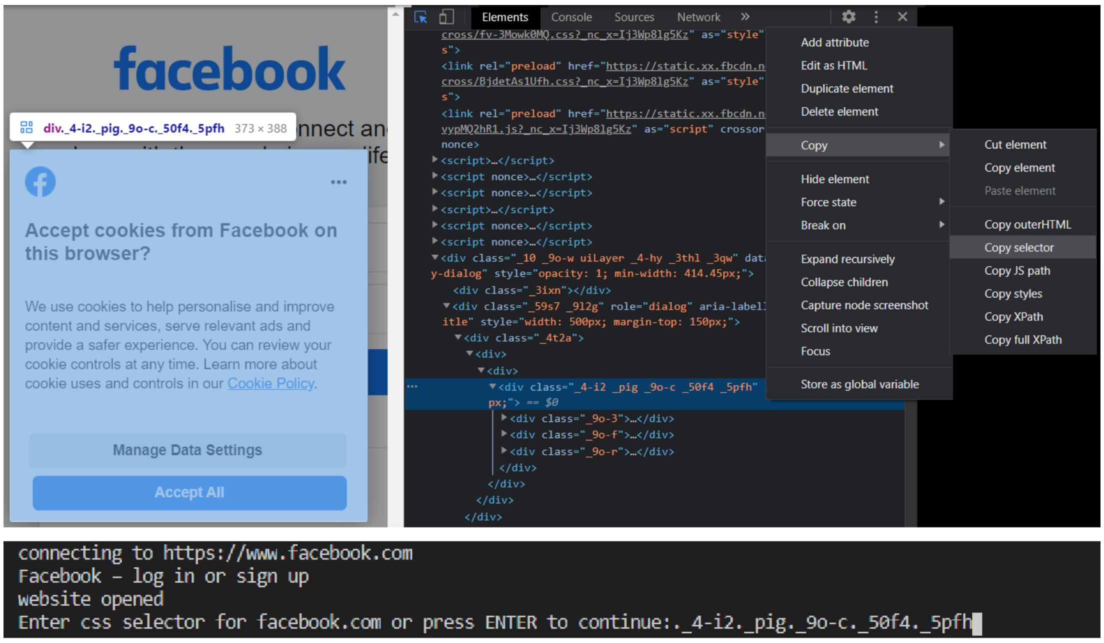
Figure 3.7: A screenshot of a web element falsely suggests that the element represents the complete cookie dialog. The close button is actually the sibling of this element hence the cookie notice is not captured in full. Source: technologyreview.com

3.4.3 Manual Tagging

Manual tagging helps to find cookie dialogs on websites where these were missed or incorrectly identified. The list of domains requiring manual tagging is generated in manual screenshot verification step. It is the user's responsibility to find a suitable CSS selector, therefore manual tagging requires some previous knowledge of HTML. To make process easier ChromeDriver opens each website and system asks to input a corresponding CSS selector. To find a selector the user can use Elements section of Chrome DevTools to first locate the dialog element and then either copy the selector or construct an suitable alternative by inspecting HTML code (see Figure 3.8). As can be seen, the CSS selector for the cookie dialog on facebook.com is a string of random characters, in fact, approximately 13% of detected cookie dialogs in the top 1,000 websites have non-human readable CSS selectors. Apart from a few cases where certain assumptions can be made (e.g. all cookie dialogs on google have the same selector independently of the TLD), these dialogs are not detected by any of the automated methods described in this chapter. Manually adding such CSS selector is a temporary solution since the selector can find the dialog only until the class names or id does not change.

3.4.4 Phase 2: Visit websites displaying a cookie dialog, interact with a cookie dialog and collect cookies after each action.

The main objective of the Phase 2 is to interact with the cookie dialogs and collect cookies that are set after clicking different types of buttons. The system is designed to simulate the behaviour of a typical internet user who is usually more concerned with



Accept cookies from Facebook on this browser?

We use cookies to help personalise and improve content and services, serve relevant ads and provide a safer experience. You can review your cookie controls at any time. Learn more about cookie uses and controls in our Cookie Policy.

```
cross/fv-3Mowk0MQ.css?_nc_x=Ij3Wp8lg5Kz" as="style"
s">
<link rel="preload" href="https://static.xx.fbcdn.n
cross/BjdetAs1Ufh.css?_nc_x=Ij3Wp8lg5Kz" as="style"
s">
<link rel="preload" href="https://static.xx.fbcdn.n
vypMQ2hR1.js?_nc_x=Ij3Wp8lg5Kz" as="script" crossor
nonce>
▶<script>…</script>
▶<script nonce>…</script>
▶<script nonce>…</script>
▶<script>…</script>
▶<script nonce>…</script>
▶<script nonce>…</script>
▼<div class="_10 _9o-w uiLayer _4-hy _3thl _3qw" dat
y-dialog" style="opacity: 1; min-width: 414.45px;">
   <div class="_3ixn"></div>
  ▼<div class="_59s7 _9l2g" role="dialog" aria-label
  itle" style="width: 500px; margin-top: 150px;">
    ▼<div class="_4t2a">
      ▼<div>
        ▼<div>
          ▼<div class="_4-i2 _pig _9o-c _50f4 _5pfh"
          px;"> == $0
            ▶<div class="_9o-3">…</div>
            ▶<div class="_9o-f">…</div>
            ▶<div class="_9o-r">…</div>
            </div>
          </div>
        </div>
      </div>
```

```
connecting to https://www.facebook.com
Facebook - log in or sign up
website opened
Enter css selector for facebook.com or press ENTER to continue:._4-i2._pig._9o-c._50f4._5pfh
```

Figure 3.8: Manual cookie dialog tagging process: user finds a cookie dialog and corresponding block of code within HTML using Chrome DevTools. Then can user either copy the selector as demonstrated or in this case construct their own and enter it in the terminal. Source: facebook.com

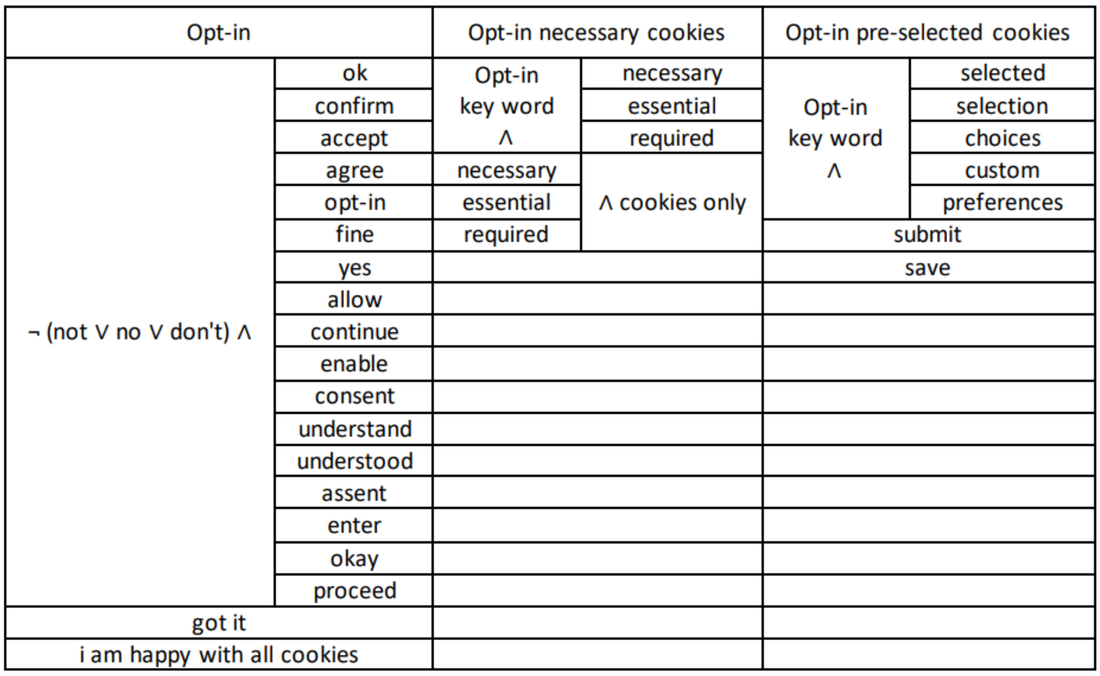
accessing the content of the web page than setting cookie preferences hence the system only interacts with buttons available on the first page of the dialog.

3.4.4.1 Identifying options

Since different cookie dialogs contain different buttons, it is helpful to define button categories. Categorising helps to differentiate between different types of buttons hence making the button identification, cookie data storage and access easier. I defined seven button categories, namely:

1. Opt-in
2. Opt-in necessary cookies
3. Opt-in pre-selected cookies
4. Cookie settings
5. Opt-out
6. Cookie or Privacy policy
7. Close

The system searches a button according to the order outlined above. To illustrate, when a possible button element is found, the system first checks if it is an opt-in button. If not, the system will continue going down the list until the appropriate category is found, otherwise, it is determined that the element is not a button element. There are three different categories representing the choice to opt-in since semantically these buttons are different and some cookie dialogs display more than one opt-in button.



Opt-in		Opt-in necessary cookies		Opt-in pre-selected cookies	
¬ (not ∨ no ∨ don't) ∧	ok	Opt-in key word ∧	necessary	Opt-in key word ∧	selected
	confirm		essential		selection
	accept		required		choices
	agree	necessary	∧ cookies only		custom
	opt-in	essential			preferences
	fine	required		submit	
	yes			save	
	allow				
	continue				
	enable				
	consent				
	understand				
	understood				
	assent				
	enter				
	okay				
	proceed				
got it					
i am happy with all cookies					

Table 3.2: Keywords and keyword combinations for the following button categories: Opt-in, Opt-in necessary cookies and Opt-in pre-selected cookies.

To find buttons available within the cookie dialog, first, the system needs to find the dialog itself using the list of domain specific CSS selectors generated in the Phase 1. This step is crucial to the whole process, especially, in cases where the contents of the dialog are within an `iframe` since the buttons are not visible to ChromeDriver otherwise. To minimise the set of elements that need inspection, the system considers the elements within the dialog and only explores the text nodes that do not have any descendant text nodes. If the node or its parent has a typical button HTML tag, namely `button`, `div`, `a` or `span`, or has an attribute `role=option`, system attempts to place button within appropriate category by checking if the element's text contains any of the keywords specified (see Tables 3.2 and 3.3). The set of keywords was partially adopted from the study performed by B. Molnar [27] in combination with careful analysis of the dialogs collected over the course of this study. Similarly to textual verification process described in Section 3.4.1.3, button classification uses Google Translate API for foreign websites.

Since each button exclusively belongs to one category, the order of button categorisation matters. In addition, it helps to reduce the number of keyword combinations that need to be checked. For instance, `Save & Exit` is categorised as an opt-in pre-selected cookies button since it contains keyword `save`, it also contains a keyword `exit` corresponding to another category. Therefore, if the categorisation order was reversed, system would need to introduce further text requirements, in this case, require Close buttonss to contain a key word combination ¬`save` ∧ `exit`.

The method describe retrieves most of the buttons with the exception of buttons not

Cookie settings	Opt-out		Cookie or Privacy policy	Close
let me choose	(not ∨ don't) ∧	track	privacy	dismiss
select cookies		consent	policy	close
manage		accept	notice	exit
preferences	disagree		here	x (letter)
settings	reject		use of cookies	× (multiplication sign)
choice(s)	decline		data protection	hide
preferences	refuse		terms of service	
options	turn cookies off		read more	
customize			learn more	
personalize			tell me more	
configure			more information	
configuration			see more	
advanced			more details	
vendor(s)				
partner(s)				
purposes				
tool				
manager				

Table 3.3: Keywords and keyword combinations for the following button categories: Cookie settings, Opt-out, Cookie or Privacy policy and Close.

containing text. I have observed two types of buttons without text node, namely:

1. Buttons represented by HTML `input` tag with a `type` attribute equal to button or submit. Similarly to the method described, these can be categorised by examining the value of either `title` or `value` attribute since it contains the text displayed on the button.

2. Buttons that contain an icon or an image instead of textual value.

Finding and categorising icon-based buttons require a different approach since in this case textual analysis is not applicable. I have observed that in the set of top 1,000 websites, icons are used exclusively for Close buttons and the icon itself is usually a variation of symbol X. Therefore, one possible solution is to perform image recognition on either the image extracted from the source code or screenshot of the element. However, websites rarely store icons as simple .png or .svg files, icons are often stored as base64 encoded strings in .css file or stored in large sprite sheet together with other smaller icons. On the other hand, screenshots have low resolution because the icons and hence the web elements representing icons are very small and ChromeDriver does not produce high-resolution images. As a result, image recognition would be complex and may not yield accurate results. Therefore, I implemented a different approach based on cookie notices' HTML structures I have observed.

To find icon-based close buttons, the system searches HTML nodes that do not have any text descendants and are likely to contain the icon, namely HTML tags: `button`, `a`, `svg`, `img`, `i`, `span`. System iterates the list of HTML tags and finds elements matching the query, when there is more than one element the left-most is assumed to be the close button. In addition, graphic HTML tag elements (`svg` and `img`) are required to be square. Since this method is heavily based on assumptions, ChromeDriver performs a sanity check by clicking the element and checking if dialog is no longer displayed after the click.

3.4.4.2 Dialog Interactions

After all buttons are identified, the system starts interacting with the dialog. As mentioned previously, system simulates behaviour of a typical internet user who wants to close the dialog with one click and access the contents of the web page. This means, the ChromeDriver does not set any cookie preferences even if there is an option on the first page of the dialog. Therefore, the system does not click Cookie settings or Cookie or privacy policy buttons. For each of the buttons in other categories, ChromeDriver clicks the button, collects and saves the cookie data, clears browser cookies and cache between the subsequent clicks.

Chapter 4

Evaluation

The system described in the previous chapter was developed and tested using the list of the top 1,000 websites, in fact, all assumptions and generalisations about the cookie dialogs and web design are based on the patterns seen within this subset of web pages and my previous knowledge. In addition, the implementation process included some manual tagging, verification and dialog HTML structure analysis, as a result, the system was improved to achieve greater efficiency, accuracy and a higher level of automation. However, this suggests that the methods for cookie dialog and button detection might be biased towards the top 1,000 websites, therefore, to evaluate the system it was tested on a subset of 100 websites (list included in Appendix B) randomly selected from the set of top 1 million domains.

The system failed to open 16 websites and 2 of these failures were detected manually since the websites were blocked by the network firewall which was not one of the expected failure conditions.

4.1 Cookie Dialog Detection

System found 25 cookie dialog elements out of 25 present, not including one false positive element. The number of possible cookie dialog elements is 25% lower compared to the dialog elements found with no dialog text verification, this illustrates how simple textual analysis can help eliminate a fraction of false positives. In addition, 6 cookie dialogs were foreign and passed the text verification. Further, all websites where no dialog was found, do indeed not display a dialog.

On average, system found 2 possible cookie dialog elements per website. Depending on whether the system operates in fully automated mode or not, the decision on which of these elements is considered to be the cookie dialog differs. Therefore, it is important to evaluate the performance for each mode separately.

4.1.1 Fully-automated mode

As mentioned in the previous chapter, in fully automated mode the first element found is assumed to be the dialog. This is indeed true for 91.7% of the crawled websites and the two exceptions were elements that include the contents of of the dialog but are larger than the actual dialog. However, such errors could be disregarded if the tool is used for a purpose where the exact size of the dialog is insignificant. Further, the tool detected one non-dialog element, in addition, the system detected a dialog on surf.nl that disappears in less than 30 seconds after loading the website hence making it impossible to perform the dialog interactions in the Phase 2. However, in fully-automated mode the system fails to identify false positives or cookie notices with atypical behaviour. Nevertheless, in fully-automated mode the system detected cookie dialogs with 93.0% accuracy[1].

4.1.2 Semi-automated mode

In semi-automated mode the user can confirm which element represents dialog best, add a better tag manually or declare that there is no cookie dialog at all. This helped to achieve 100.0% accuracy but required time and human interaction. Assuming the user has prior experience with the screenshot verification tool and knowledge about finding suitable CSS selectors it takes approximately 100 seconds to inspect all screenshots and around 90 seconds to find a CSS selector for one cookie dialog, this includes ChromeDriver's website loading time which is artificially extended to make sure the website is fully loaded. Since in this case there is only one website which needs manual tagging, the semi-automated mode adds a cost of approximately 3 minutes.

4.2 Identifying interactive elements

The system successfully identified 92.8% of the interactive elements and more detailed results are presented in Table 4.1. Performance of interactive element search was very similar for both automated and semi-automated modes. The only difference is the execution time since in fully-automated system explores a falsely identified dialog and a dialog that is known to disappear shortly which approximately takes an extra 12 minutes.

	Opt-in	Opt-in necessary	Opt-in pre-selected choice	Opt-out	Close	Settings	Privacy policy
found	24	1	0	2	4	17	16
total	24	1	0	2	4	20	18

Table 4.1: Number of successfully identified and the total number of buttons in each category.

[1] $accuracy = \frac{TrueNegatives + TruePositives}{SuccessfullyScrapedWebsites}$, since assumption of maximum one dialog per website holds, $\#TruePositives \leq \#WebsitesWhereDialogWasFound$

Chapter 5

Results

This chapter discusses findings of this study and the discussion is split into two parts, namely analysis of cookie dialogs and analysis of cookie setting behaviour.

All experiments were performed on the 15-16th of March 2021 using the tool described in Chapter 3 inspecting the list of top 1,000 websites of 2020. The program successfully crawled 907 websites and the following results and analysis are presented with respect to this subset. Since cookie setting behaviour, cookie dialogs and their contents vary depending on the location, to get consistent results data was collected using UK based VPN proxy.

5.1 Cookie dialog analysis

I have found that 54.3% of the successfully crawled websites displayed a cookie dialog. The automated tool performed identification with 98% accuracy, which increases to 99.9% after identifying erroneous website connections using the manual verification step. I consider cookie dialog to be content blocking if it is located in the center of the page, these include pop-up windows and cookie-walls. Some dialogs located at the top or the bottom of the page can also be content blocking since other website's clickable elements are disabled until the user interacts with the cookie dialog, however, in this study, the classification is based on dialogs location hence such cases are not considered. I have found that 194 out of 490, or 39.6% of the detected cookie dialogs are content blocking hence the user is required to acknowledge the use of cookies and set cookie preferences before accessing the contents of the web page. This figure is significantly higher compared to the cookie dialog analysis presented in the study done by Utz et al. [33] where only 3% of 1,000 notices collected in 2018 were content blocking, however, it is likely that the cookie notices have changed since more time after GDPR has passed.

The Table 5.1 presents the number of dialogs which contain buttons of different categories identified in Section 3.4.4.1. 88.6% of the dialogs display some sort of opt-in button and only 10.8% contain an opt-out button. In addition, 47.0% of the cookie dialogs displaying both these options highlight the opt-in button hence nudging the user to give consent (see Figure A.6). 10.2% of dialogs do not have any of the cookie management or consent options, however the majority of these no-option dialogs do contain a close button, again comparing with the findings by Utz et al. [33] where 27.8% of consent notices do not give users any option on the first page of the dialog. In addition, in 12.5% of the websites the cookie dialog reappears after clicking a close button and refreshing the page.

Further, 64.3% of the cookie dialogs contain a settings button allowing the user to set their cookie preferences, however, 66.3% of such dialogs are nudging the user to opt-in rather than click settings or opt-out buttons. Comparing the options available in content blocking versus non-blocking cookie dialogs, it can be seen that 92.3% of blocking dialogs display a cookie settings button whereas only 50% of non-blocking have this option. In addition, 87.9% of cookie dialogs classified as confirmation only (see Figure A.4) are non-blocking dialogs.

Dialogs containing button	Opt-in	Opt-in necessary	Opt-in pre-selected choice	Opt-out	Close	Settings	Privacy policy
Blocking	179	2	8	26	18	167	143
Non-blocking	252	3	0	27	78	148	224
All	431	5	8	53	96	315	367

Table 5.1: Number of dialogs containing buttons of different categories, namely Opt-in, Opt-in necessary, Opt-in preselected choice, Opt-out, Close, Settings and Privacy policy. The statistics are presented for the total of 490 cookie dialogs, where 194 dialogs are content blocking.

5.2 Cookie setting behaviour

This section discusses cookie setting behaviour of crawled websites. Analysis is split into further sections, namely cookie setting behaviour upon first loading the website and after interacting with the cookie consent options displayed on the first page of the dialog.

5.2.1 ID-like cookies

To have a better understanding which cookies could potentially be used for tracking, I needed to determine which cookies could be used as unique identifiers. There is no comprehensive database describing the purpose of each cookie, however, other studies suggest different approaches to identify potentially ID-like cookies that have a higher likelihood to be used for tracking. Sanchez-Rola et al. [29] distinguishes ID-like cookies by using a password strength evaluation algorithm called zxcvbn to determine whether the user could be identified among 1 billion internet users and checking whether the cookie is set by a known analytics or marketing domain. Other studies

[16][15] suggest a different approach, where ID-like cookie is required to have a retention period of at least 90 days, in addition to having a high-entropy value. Similarly, Gunes et al. [1] filter out cookies that expire within a month when detecting ID-like cookies.

My implementation of distinguishing ID-like cookies is based on the methods described above. I have considered cookies that have a life time of at least 30 days. The reason I have chosen a shorter period is two-fold. First, looking at the distribution of cookie retention periods under three months I identified distinct peaks at 30 and 90 days both of which appear relevant for this purpose. Although a longer period allows to build more comprehensive user profile, 30 days might be sufficient to track and exploit user's browsing behaviour for marketing purposes since the ad targeting data is often sold close to real-time. Second, I have observed cookies set by advertising and analytics service providers, such as tapad.com, adform.com and neustar.com, that have retention periods of 30 days. After filtering cookies by the expiration date, I have looked at the substrings comprising the cookie value, similarly to the methods described in other studies [15][1]. I extracted the subvalues separated by the list of delimiters, namely `: , = , . , % , |` and considered a cookie to be ID-like if the $8 \leq lenght(subvalue)$ and the subvalue is hard to guess according to the zxcvbn[1] score. I considered the scores equal or greater than 3 which is equivalent to at least 100 million guesses required to guess a password.

Not all ID-like cookies are profiling cookies recording identifiable user's behaviour. To illustrate, some analytics cookies are ID-like since each user is given a unique ID, however, the data collected is anonymous. For instance, Google Analytics first-party cookie `_ga`[2] assigns a user ID that is non-personally identifiable. In addition, ID-like cookies may be used to store user's cookie management preferences, for example, `CookieConsent`[3] cookie set by Consent Management Provider Cookiebot stores user's consent preference. Although it assigns an identifiable user ID, this cookie records user's cookie preferences rather than tracks their behaviour online.

5.2.2 Cookies set upon first visit

Cookie collection upon loading the website suggests that 95.8% of the websites set at least one cookie prior to any interaction with the web page and 91.6% websites set at least one cookie having 1 year or longer retention period. Figure 5.1 shows that the vast majority of the websites set at least one cookie in each of the following categories: persistent, first-party and ID-like. More detailed statistical information is presented in Table 5.2. It can be seen that on average websites set 22 cookies and the maximum number of cookies set is almost 10 times greater. In addition, on average a website sets more third-party than first-party cookies, and stores 19.1 persistent and 10.8 ID-like cookies.

To determine whether the websites that do not display a cookie dialog set more cookies and hence are the underlying cause of high values of measures outlined in Table 5.2,

[1] *zxcvbn: a realistic password estimator*. URL: `https://pypi.org/project/zxcvbn/`.

[2] *Google Analytics*. URL: `https://developers.google.com/analytics/`.

[3] *Cookie Declaration*. URL: `https://www.cookiebot.com/en/cookie-declaration/`.

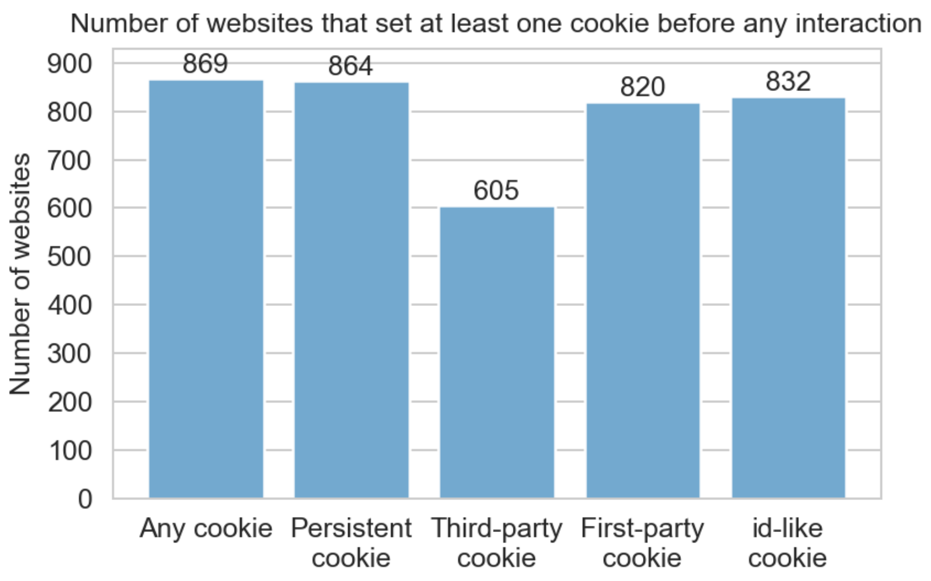


Figure 5.1: Number of websites that set at least one cookie prior to any interaction with the web page.

cookie	mean	median	mode	max	standard deviation
any	22.1	12	5	206	28.8
persistent	19.1	9	2	195	27.1
third-party	11.3	2	0	176	23.4
first-party	10.7	7	0	69	10.1
ID-like	10.8	5	1	117	16.1

Table 5.2: Statistical measurements on cookies set before any interaction with a cookie dialog. The data is collected from 907 successfully crawled websites.

I split the subset and compare distributions of the number of cookies set by websites that do and do not display a cookie dialog. This is based on a hypothesis, that a website displaying a cookie dialog is following the regulation guidelines, therefore, it is more likely to conform to the regulation with regards to cookie setting behaviour hence set less cookies upon loading the website. The distributions of the number of cookies set upon loading the website are presented in Figure 5.2. By performing a t-test between websites that do and do not display a cookie dialog I did not observe a statistically significant difference between the means of these two groups, however, the variance of no-dialog distribution is approximately 44% greater. In fact, if the subsets are split further by the cookie type, t-test results are similar for persistent, third-party and ID-like cookies. However, there is a statistically significant difference between the number of first-party cookies set by websites that do and do not display a cookie dialog ($p < 0.009$) with websites displaying a cookie dialog having a higher mean. Combining statistical measures with the cookie distribution demonstrated in Figure 5.2, I conclude that websites containing cookie dialogs set more first-party cookies upon loading the web page. However, there is no statistical evidence that this holds for other types of cookies.

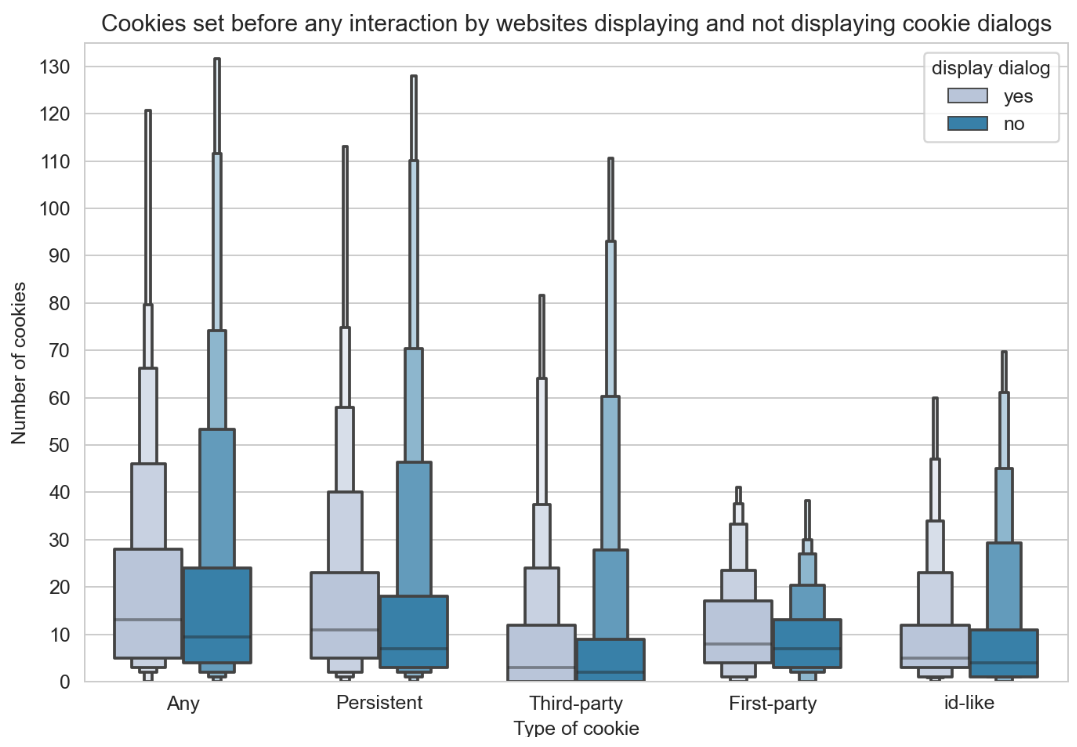


Figure 5.2: Distribution of cookies set prior to any interaction with the websites displaying and not displaying cookie dialogs. Outliers are not presented and the line inside the widest boxes denotes the median of each distribution.

The Figure 5.3 presents the top 20 trackers that set at least one ID-like cookie before any interaction with the cookie notice. I identify top cross-site tracker companies by inspecting the most common cookie name-domain pairs among ID-like third-party cookies and confirm that the cookie domains corresponds to a known tracker identified by WhoTracksMe[4] or DuckDuckGo Tracker Radar[5]. As can be seen, DoubleClick is the most prevalent tracker, setting cookies on 25.7% of crawled websites. In addition, most trackers store cookies on an approximately equal number of websites that do display and websites that do not display cookie dialogs.

[4] *WhoTracksMee*. URL: `https://whotracks.me/`.

[5] *DuckDuckGo Tracker Radar*. URL: `https://github.com/duckduckgo/tracker-radar/`.

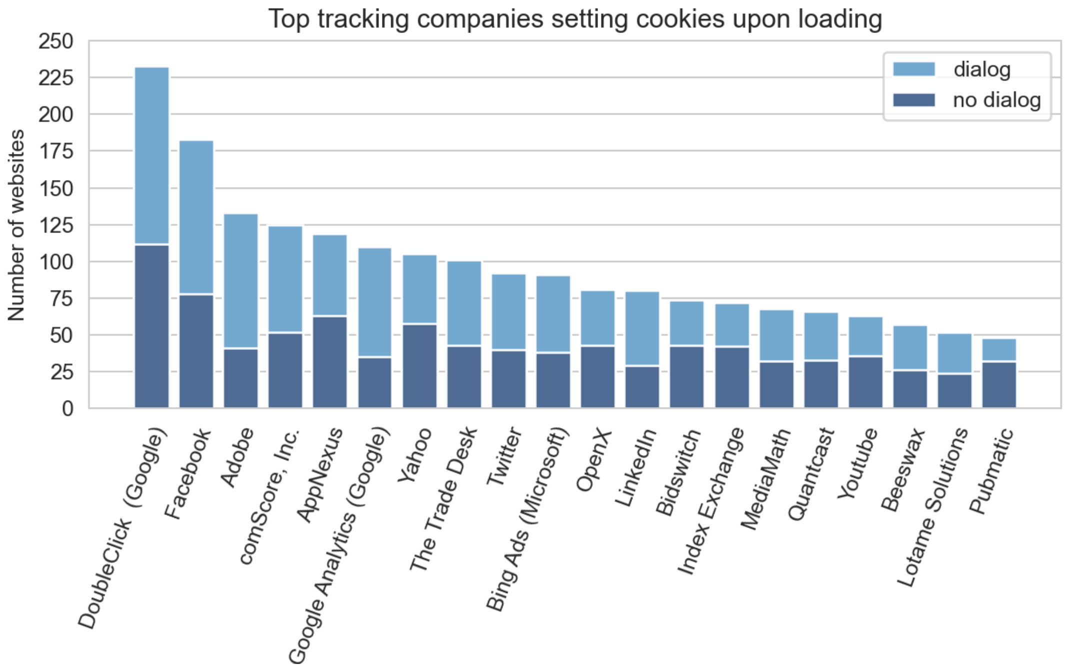


Figure 5.3: Top 20 cross-site trackers that set at least one ID-like cookie upon loading. Presented data corresponds to 907 successfully crawled websites.

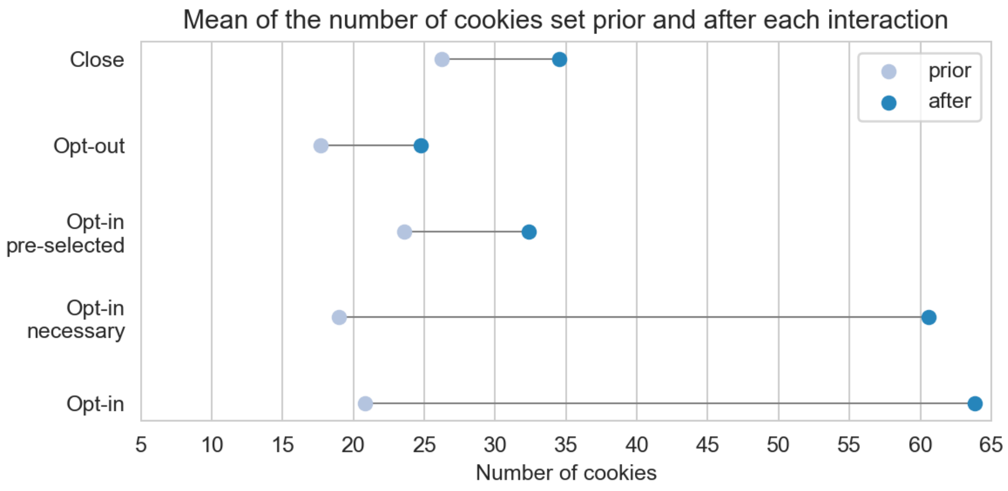
5.2.3 Cookies set after dialog interactions

This section discusses cookie setting behaviour in more detail, especially how this behaviour changes after user interacts with the options displayed within cookie dialog.

In Phase 2 system the failed to find a cookie dialog on one website due to Capthca test. In addition, three cookie dialogs required additional interactions before clicking any buttons, therefore there are no data on cookie setting behaviour after interactions for these three websites. Thus, further discussion considers the subset of 486 websites displaying a cookie dialog which the system managed to successfully engage with.

Figure 5.4 presents how the average number and the median of cookies changes after clicking different dialog buttons. As can be seen, all dialog interactions result in an increased number of cookies and on average clicking opt-in or opt-in necessary results in the highest amount of cookies. The subset of cookies set after clicking opt-in necessary button have the highest median, in addition, this subgroup demonstrates the greatest increase of median compared to other buttons. On the other hand, only 1% of the crawled websites display a cookie dialog with an opt-in necessary button hence there is not enough data to draw conclusions about cookie setting behaviour after clicking opt-in necessary and compare it with the behaviour of other button types. In this specific data set, it was found that opt-in necessary button is always accompanied with an opt-in button. When only this small cookie dialog subset is considered it can be seen that on average clicking opt-in necessary button results in significantly lower amount of cookies than clicking opt-in (see Figure 5.5).

As presented in Figure 5.4, clicking opt-out button does not remove all cookies that were set upon loading the website, this is expected since some cookies are necessary



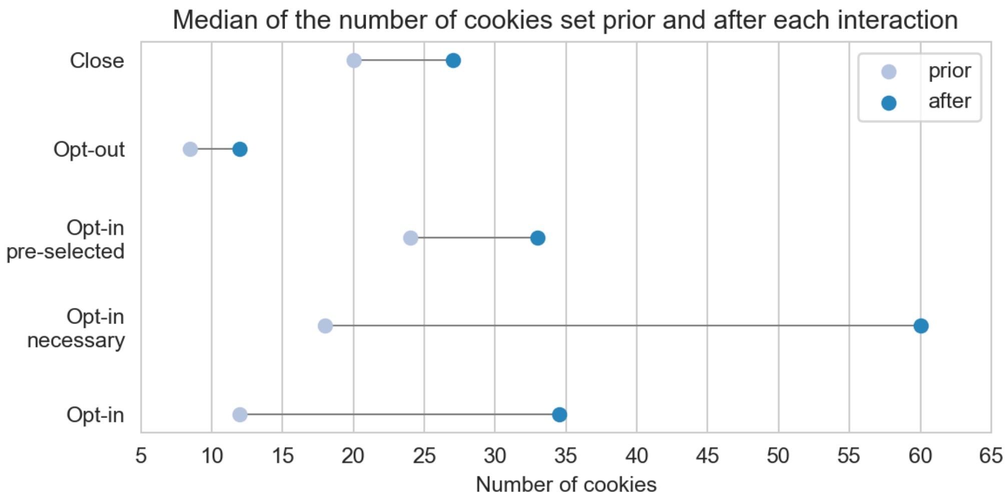


Figure 5.4: Mean and median of the number of cookies set before and after clicking button in each of the following button category: Opt-in, Opt-in necessary, Opt-in pre-selected, Opt-out and Close.

for a website to operate. However, the option to opt-out or reject cookies semantically implies removal of cookies or at least no added cookies, therefore I inspect the cookie setting behaviour in a little more detail, more specifically I was interested how clicking the opt-out button affected the number of cookies set in each subgroup and these findings are presented in Table 5.3. As can be seen, on average websites do set additional cookies of every type and the least difference is in the number of ID-like cookies. Comparing modes, medians and means suggests that the cookie distributions remain positively skewed and the mode of third-party, first-party and ID-like cookies does not change, however the overall maximum number of cookies doubles.

By clicking the close button the user is acknowledging the presence of the cookie consent notice, however, the meaning of close button is ambiguous since it does not imply giving or refusing consent. It can be seen in Figure 5.4 that the mean and the median of cookies set after the click increases, yet the difference is smaller compared to that of opt-in. The number of dialogs containing the close button is far smaller than the

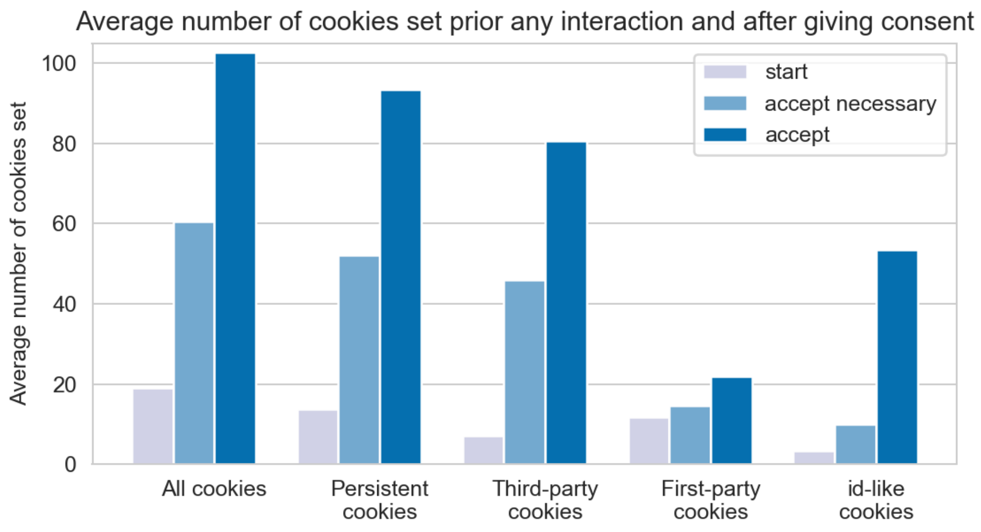


Figure 5.5: The average number of cookies before and after clicking Opt-in and Opt-in necessary buttons.

	mean		median		mode		max		standard deviation	
cookie type	prior	after	prior	after	prior	after	prior	after	prior	after
any	17.7	24.8	8.5	12.0	3	4	81	167	20.0	32.6
persistent	15.4	21.7	8.0	11.0	2	1, 3 and 6	80	137	17.9	21.7
third-party	8.4	13.6	2.0	2.0	0	0	78	128	15.1	13.6
first-party	9.3	11.2	6.0	8.5	0	0	36	39	9.5	11.2
ID-like	8.5	10.1	4.0	4.5	1	1	49	58	10.9	13.5

Table 5.3: Statistical measurements on cookies set before and after clicking an Opt-out button. Cookie data before the click is adjusted to only span websites that do display an Opt-out button in the cookie dialog.

number containing the opt-in button hence simply comparing changes in the mean and the median is not sufficient to state that these two buttons result in different cookie setting behaviour. One way to reveal what close button implies with regards to user's consent is to compare post-click cookie distribution with post-click distributions of opt-in and opt-out buttons. T-test between cookies set after clicking close and after clicking opt-in reveal statistically significant difference ($p < 10^9$) with cookie distribution after opting-in having the higher mean. Similar results were produced for different types of cookies with the exception of first-party cookies set after closing the dialog and after opting-in where no significant statistical difference was observed. Similarly, a t-test between cookies set after clicking close and after opting-out reveal statistically significant difference ($p = 0.04$) with post-click close distribution having the higher mean. However, by performing a t-test between third-party cookies set after closing the dialog and after opting-out I did not observe a statistically significant difference between the means of these distributions. Figure 5.6 demonstrates the discussed cookie distributions. It can be concluded, that clicking close button results in a cookie setting distribution different from opt-in and opt-out, however, in this dataset the number of cookies saved after clicking close falls somewhere in between opt-out and opt-in.

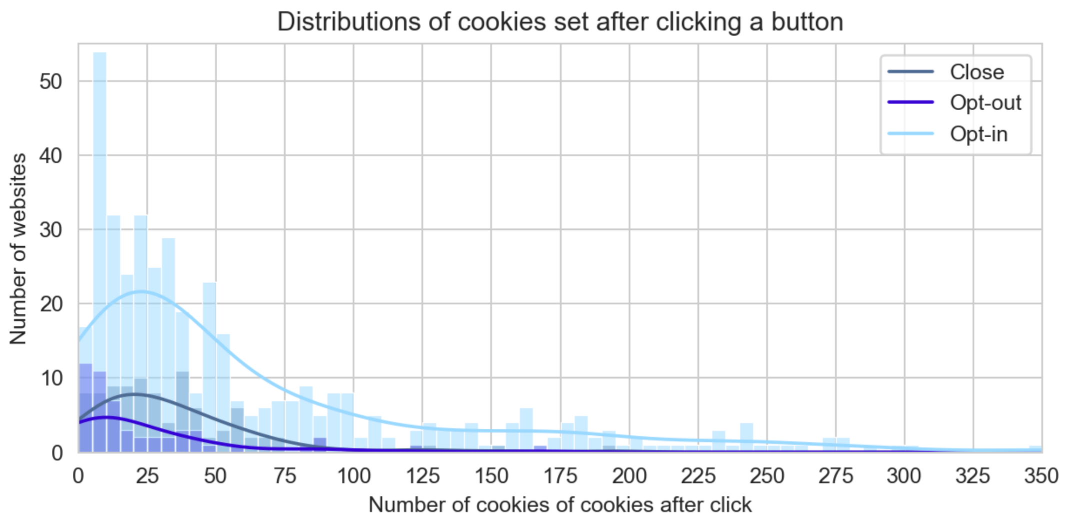


Figure 5.6: Distributions of cookies set after clicking Opt-in, Opt-out and Close button. Since the data on other button types is insufficient, Opt-in necessary and Opt-in preselected are excluded.

The Figure 5.7 presents the top 20 trackers identified after clicking opt-in button. As can be seen, for the top 20 trackers the number of websites where the tracker was identified after opting-in increased and on average is 3.7 times greater than pre-click. Comparing to the trackers found upon loading (see Figure 5.3), DoubleClick, Facebook and Adobe remain the most prevalent trackers for this dataset. Similarly, DoubleClick is the most prevalent tracker after clicking opt-out and close buttons, where it was found on 23.1% and 51.2% crawled websites respectively (see Figure 5.8). Google (the owner of DoubleClick) and Facebook were also identified as the most prevalent trackers in studies carried out by Roesner et al. [28], Karaj et al. [21] and Englehardt et. al [15], despite their measurements including a wider range of trackers, namely within-site trackers and trackers using tracking technologies beyond cookies.

The Figure 5.8 presents the top 20 trackers found after opting-out and after clicking close. As can be seen, the number of websites where the tracker was found increased for 15 out of 20 top trackers after refusing consent and for all top 20 trackers after closing the dialog. Moreover, approximately half of the most prevalent tracker observed after closing the dialog or refusing consent were not included in the list of the top 20 most common trackers upon loading or opting-in. However, the subset of the cookie notices displaying an opt-out or close button is far smaller compared to those containing an opt-in button, in addition, the overlaps between these subsets are limited. Therefore, it can not be concluded that the websites displaying the dialogs with opt-out or close buttons employ different trackers from those displaying the dialogs containing an opt-in option.

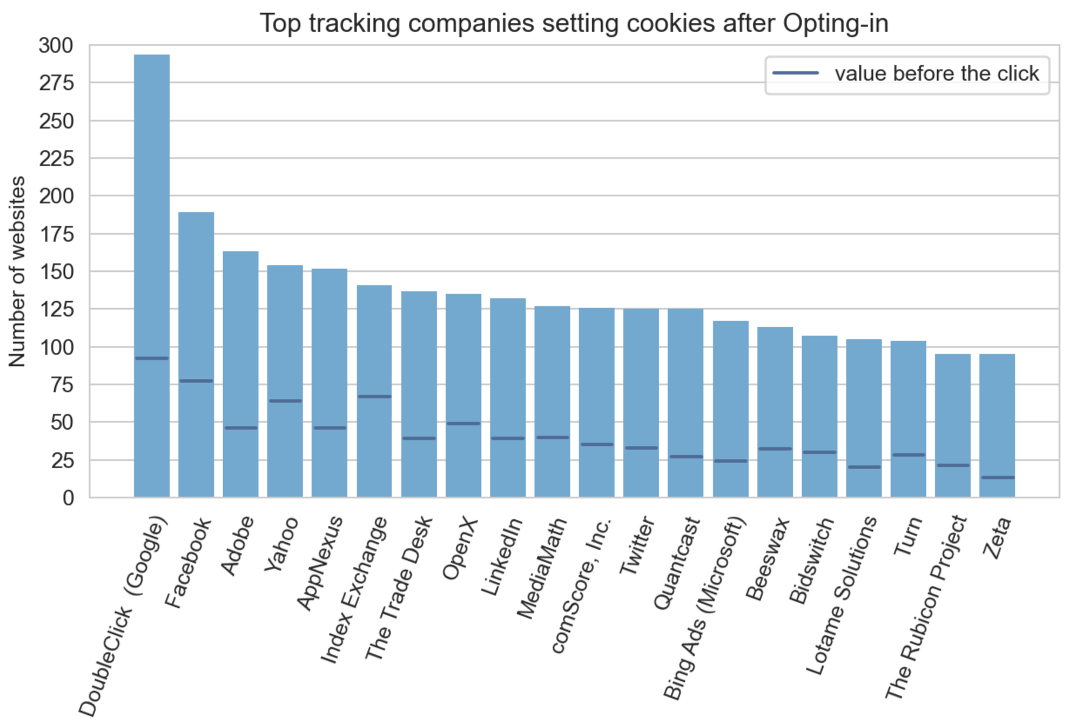
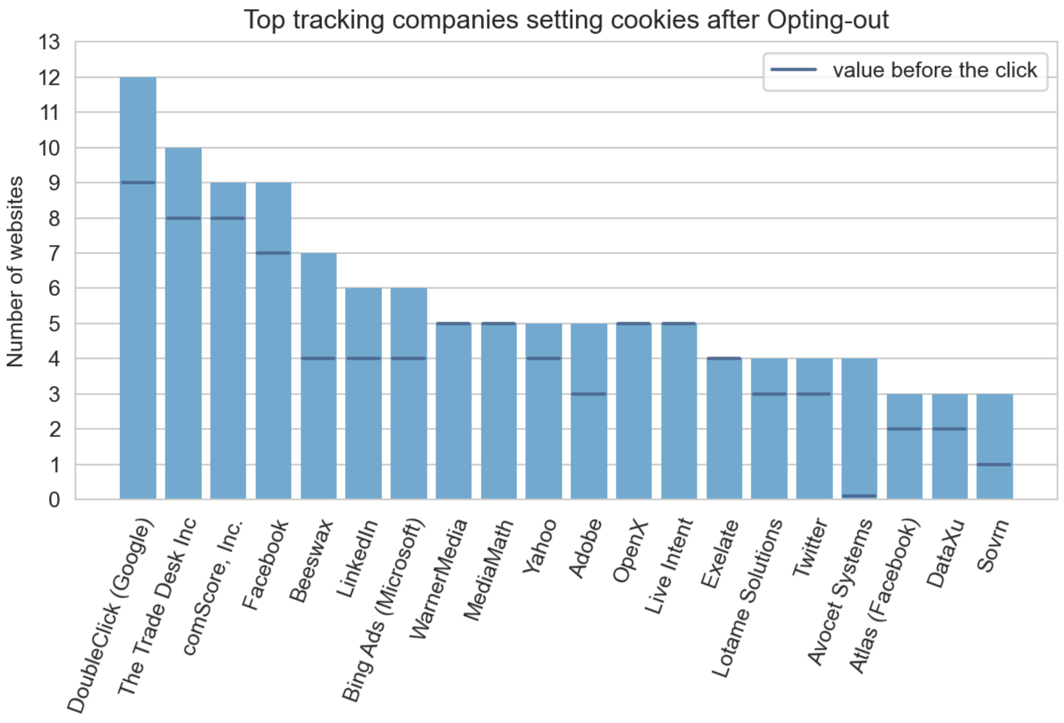
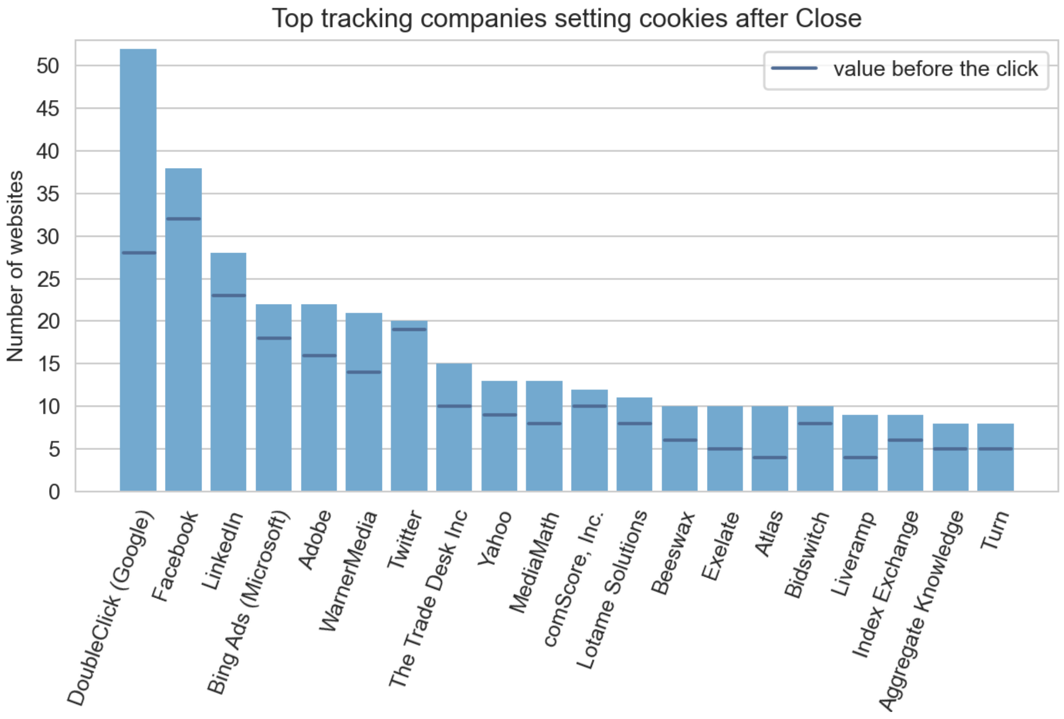


Figure 5.7: Top 20 cross-site trackers that set at least on ID-like cookie when clicking opt-in button. Presented data corresponds to 426 websites that display a cookie dialog with an opt-in button and the system managed to successfully interact with. The dark blue line indicates the number of websites within this subset containing the tracker before the click.



(a) Trackers after opting-out. Data presented for 52 websites.



(b) Trackers after closing the dialog. Data presented for 96 websites.

Figure 5.8: Top 20 cross-site trackers that set at least on ID-like cookie when clicking opt-out (a) and close (b) buttons. Data is presented with respect to the number websites that display a cookie dialog with the specific button and the system managed to successfully interact with. The dark blue line indicates the number of websites within this subset containing the tracker before the click.

Chapter 6

Conclusion

In this paper, I have analysed cookie consent options and the cookie setting behavior resulting from the interactions with the cookie notices. The main contribution of this study is the automated web crawler capable of finding cookie dialogs, identifying and interacting with the consent options displayed within the first page of the dialog, and collecting cookies. For a random subset of domains, the crawler is expected to identify the cookie notices with an accuracy of around 93.0% however, higher accuracy could be achieved if capturing the exact size of the cookie notice is not important.

Using the designed tool I performed an automated crawl on the top 1,000 most visited websites in 2020. I have found that 91.7% of successfully crawled websites set at least one ID-like cookie upon loading yet only 54.3% of the websites displayed a cookie notice. Further, any interaction with the cookie notice usually results in an increased number of cookies and often the design of the cookie notices nudges the user to give consent either by highlighting the opt-in button, or not giving an easy way to refuse consent by not including the opt-out button or possibly hiding this option within the settings page.

In general, some of the observed cookie setting behavior indicates possible violations of the GDPR and ePD. Many websites set ID-like or cross-site tracker cookies without informing the user about the use of tracking technologies or prior to the visitor giving or after refusing consent, however, many of these observations fall within the grey area of the legislation since it is the company's responsibility to decide whether certain cookies are strictly necessary for their website to operate.

Since the number of dialogs with an opt-in button is significantly greater compared to those containing other buttons that the system interacted with, the direct comparison between cookie setting behaviors is not completely straightforward. Further, the underlying behavior depends not only on the option the user chooses but also on the options available within the cookie notice. Therefore, it would be better to split the cookie notices into categories, namely, no-option, confirmation, binary, etc. and compare the cookie setting behavior based on the cookie dialog type in combination with the dialog interactions. However, as presented in Section 5.1, some of these cookie notice types are more common than others hence comprehensive analysis would require crawling a much larger set of domains.

The implemented tool lays the groundwork for future research of cookie setting behavior resulting from cookie notice interactions. However, cookie dialog detection in the Phase 1 has proven to be costly mainly due to a large amount of global CSS selectors. Iterating through a long list is computationally expensive and more importantly, it makes the system non-scalable to larger data sets. Both CSS selector lists used in this study are community managed and there seems to be a lack of maintenance, especially when it comes to deleting selectors that are no longer relevant hence the lists continue growing as websites change leaving a large number of redundant CSS selectors. In fact, less than 1% of the global selectors was used to identify cookie notices in this study. This suggests, that the cookie detection method could be optimised by pruning the global CSS selector list based on the selectors identifying cookie dialogs in the top 1,000 websites. Further developments could include automating the inspection of the subsequent pages of the dialog, namely the settings page which often includes the opt-out button or at least an option to object to some cookies. Moreover, the tool could be extended to identify the tracker companies setting first-party cookies, however, this would require inspecting HTTP requests to identify the domain of the cookie's origin.

Bibliography

[1] Gunes Acar et al. “The Web Never Forgets: Persistent Tracking Mechanisms in the Wild”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 674–689. ISBN: 9781450329576. DOI: 10.1145/2660267.2660347. URL: https://doi.org/10.1145/2660267.2660347.

[2] *Alexa Top Sites*. URL: https://www.alexa.com/topsites.

[3] *Beautiful Soup*. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[4] *Browser Market Share Worldwide*. URL: https://gs.statcounter.com/browser-market-share.

[5] *Chrome DevTools Protocol*. URL: https://chromedevtools.github.io/devtools-protocol/.

[6] *Cisco Umbrella 1 Million*. URL: https://umbrella.cisco.com/blog/cisco-umbrella-1-million.

[7] *Cookie Declaration*. URL: https://www.cookiebot.com/en/cookie-declaration/.

[8] *Cookies, the GDPR, and the ePrivacy Directive*. May 2019. URL: https://gdpr.eu/cookies/.

[9] Adrian Dabrowski et al. “Measuring cookies and web privacy in a post-gdpr world”. In: *International Conference on Passive and Active Network Measurement*. Springer. 2019, pp. 258–270.

[10] Martin Degeling et al. “We value your privacy... now take some cookies: Measuring the GDPR’s impact on web privacy”. In: *arXiv preprint arXiv:1808.05096* (2018).

[11] *Disconnect*. URL: https://disconnect.me/.

[12] *DuckDuckGo Tracker Radar*. URL: https://github.com/duckduckgo/tracker-radar/.

[13] *EasyList*. URL: https://easylist.to/.

[14] Rob van Eijk et al. “The impact of user location on cookie notices (inside and outside of the European union)”. In: *Workshop on Technology and Consumer Protection (ConPro’19)*. 2019.

[15] Steven Englehardt and Arvind Narayanan. “Online tracking: A 1-million-site measurement and analysis”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 1388–1401.

[16] Steven Englehardt et al. “Cookies that give you away: The surveillance implications of web tracking”. In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 289–299.

[17] *Ghostery*. URL: `https://www.ghostery.com`.

[18] *Google Analytics*. URL: `https://developers.google.com/analytics/`.

[19] *Google Translate API for Python*. URL: `https://pypi.org/project/googletrans/`.

[20] Xuehui Hu and Nishanth Sastry. “What a Tangled Web We Weave: Understanding the Interconnectedness of the Third Party Cookie Ecosystem”. In: *12th ACM Conference on Web Science*. 2020, pp. 76–85.

[21] Arjaldo Karaj et al. “WhoTracks. Me: Shedding light on the opaque world of online tracking”. In: *arXiv preprint arXiv:1804.08959* (2018).

[22] Daniel Kladnik. *I don’t care about cookies*. URL: `https://www.i-dont-care-about-cookies.eu/`.

[23] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. “Privacy leakage vs. protection measures: the growing disconnect”. In: *Proceedings of the Web*. Vol. 2. 2011. 2011, pp. 1–10.

[24] Victor Le Pochat et al. “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation”. In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. NDSS 2019. Feb. 2019. DOI: `10.14722/ndss.2019.23386`.

[25] 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)*. IEEE. 2020, pp. 791–809.

[26] Surya Mattu. *Blacklight: A Real-Time Website Privacy Inspector*. URL: `https://themarkup.org/blacklight`.

[27] Barnabas Molnar. “Measuring the Cookie-Setting Behaviour of Web Pages Showing Privacy Warnings”. 2020.

[28] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. “Detecting and defending against third-party tracking on the web”. In: *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*. 2012, pp. 155–168.

[29] Iskander Sanchez-Rola et al. “Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control”. In: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. 2019, pp. 340–351.

[30] *The Majestic Million*. URL: `https://majestic.com/reports/majestic-million`.

[31] *Third-party cookies and Firefox tracking protection*. URL: `https://support.mozilla.org/en-US/kb/third-party-cookies-firefox-tracking-protection?redirectslug=disable-third-party-cookies&redirectlocale=en-US`.

[32] Tobias Urban et al. “The unwanted sharing economy: An analysis of cookie syncing and user transparency under GDPR”. In: *arXiv preprint arXiv:1811.08660* (2018).

[33] Christine Utz et al. “(un) informed consent: Studying gdpr consent notices in the field”. In: *Proceedings of the 2019 acm sigsac conference on computer and communications security*. 2019, pp. 973–990.

[34] *WhoTracksMee*. URL: `https://whotracks.me/`.

[35] John Wilander. *Full Third-Party Cookie Blocking and More*. URL: `https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more/`.

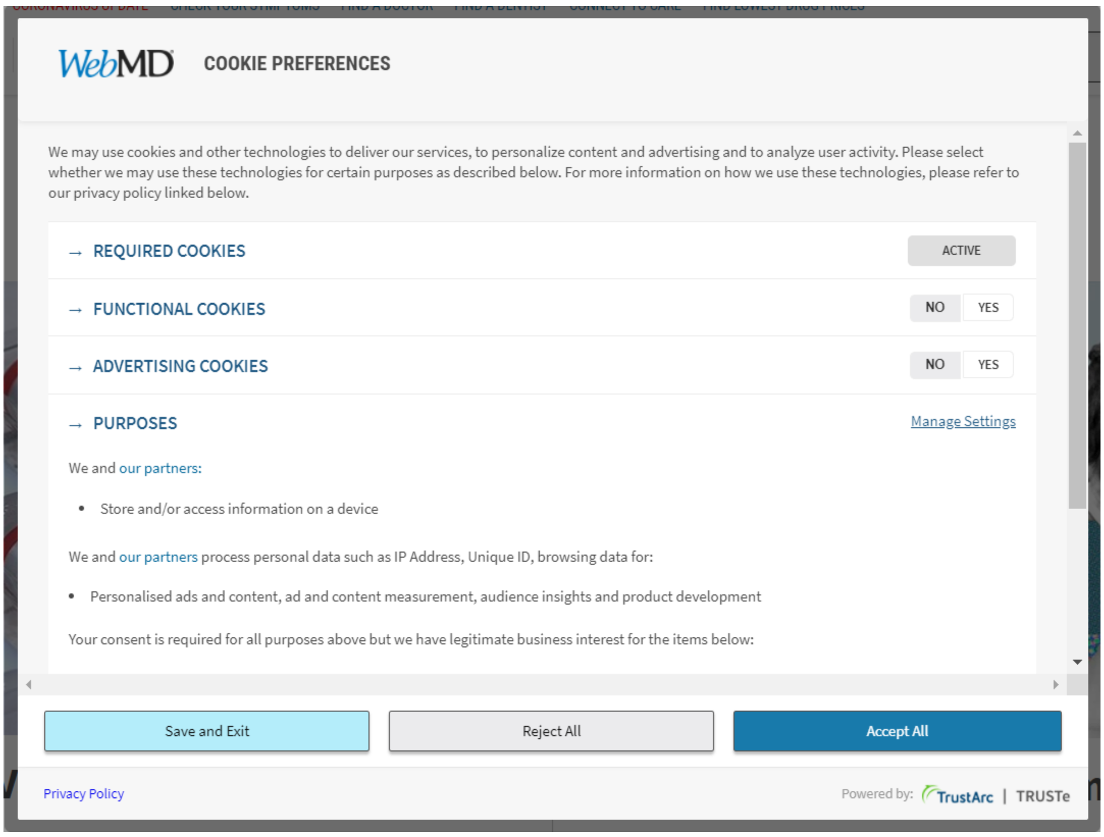
[36] *zxcvbn: a realistic password estimator*. URL: `https://pypi.org/project/zxcvbn/`.

Appendix A

Cookie Dialog Examples

A.1 Dialogs

This section includes some example cookie dialogs.



COOKIE PREFERENCES

We may use cookies and other technologies to deliver our services, to personalize content and advertising and to analyze user activity. Please select whether we may use these technologies for certain purposes as described below. For more information on how we use these technologies, please refer to our privacy policy linked below.

→ REQUIRED COOKIES — ACTIVE

→ FUNCTIONAL COOKIES — NO YES

→ ADVERTISING COOKIES — NO YES

→ PURPOSES — Manage Settings

We and our partners:

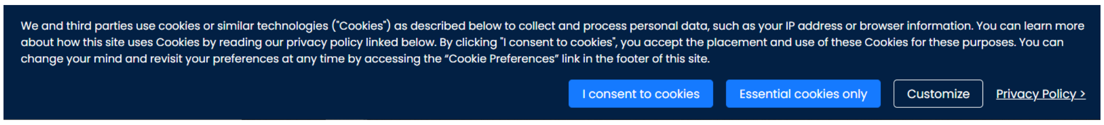
- Store and/or access information on a device

We and our partners process personal data such as IP Address, Unique ID, browsing data for:

- Personalised ads and content, ad and content measurement, audience insights and product development

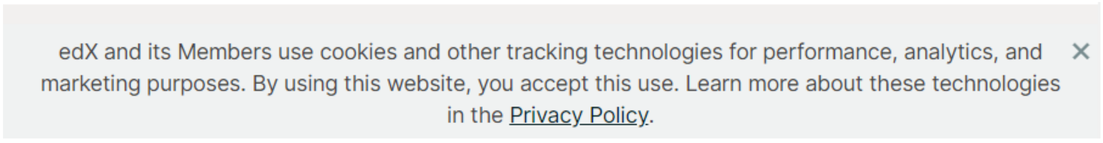
Your consent is required for all purposes above but we have legitimate business interest for the items below:

Figure A.1: Cookie dialog displaying cookie management options where the choices are pre-selected hence Save & Exit button is classified as Opt-in pre-selected. Source: webmd.com



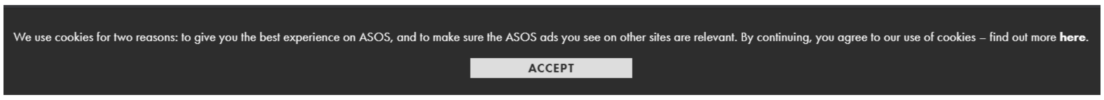
We and third parties use cookies or similar technologies ("Cookies") as described below to collect and process personal data, such as your IP address or browser information. You can learn more about how this site uses Cookies by reading our privacy policy linked below. By clicking "I consent to cookies", you accept the placement and use of these Cookies for these purposes. You can change your mind and revisit your preferences at any time by accessing the "Cookie Preferences" link in the footer of this site.

Figure A.2: Cookie dialog with Opt-in and Opt-in necessary buttons. Source: docker.com



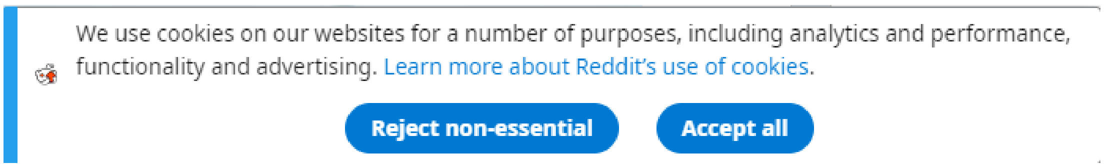
edX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the Privacy Policy.

Figure A.3: Cookie dialog with no consent or cookie management options available hence it is considered to be no-option cookie dialog. Source: edx.org



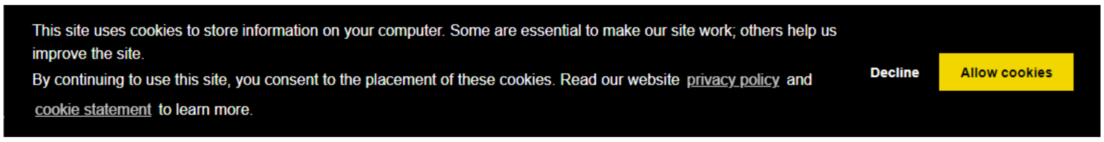
We use cookies for two reasons: to give you the best experience on ASOS, and to make sure the ASOS ads you see on other sites are relevant. By continuing, you agree to our use of cookies – find out more **here**.

Figure A.4: Cookie dialog where the only consent option available is to Opt-in hence this is considered to be confirmation cookie dialog. Source: asos.com



We use cookies on our websites for a number of purposes, including analytics and performance, functionality and advertising. Learn more about Reddit's use of cookies.

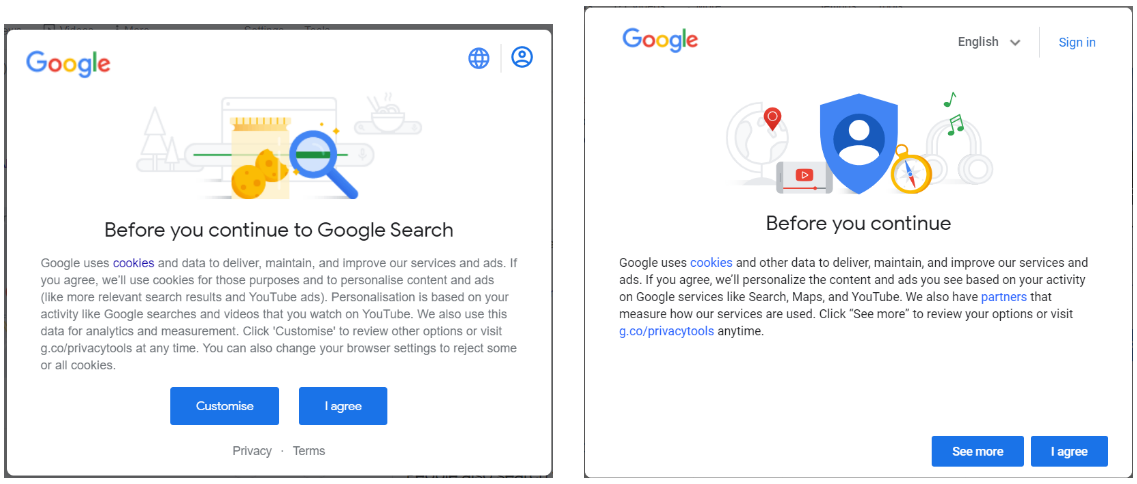
Figure A.5: Binary cookie dialog displaying both Opt-in and Opt-out options. Source: reddit.com



This site uses cookies to store information on your computer. Some are essential to make our site work; others help us improve the site.

By continuing to use this site, you consent to the placement of these cookies. Read our website privacy policy and cookie statement to learn more.

Figure A.6: Cookie dialog with both Opt-in and Opt-out options available. Allow cookie button has bright yellow color compared to Decline button which is presented in the similar style to other textual content of the dialog. The design of the cookie dialog highlights the Opt-in button hence nudging the user to consent. Source: rubiconproject.com



Before you continue to Google Search

Google uses cookies and data to deliver, maintain, and improve our services and ads. If you agree, we'll use cookies for those purposes and to personalise content and ads (like more relevant search results and YouTube ads). Personalisation is based on your activity like Google searches and videos that you watch on YouTube. We also use this data for analytics and measurement. Click 'Customise' to review other options or visit g.co/privacytools at any time. You can also change your browser settings to reject some or all cookies.

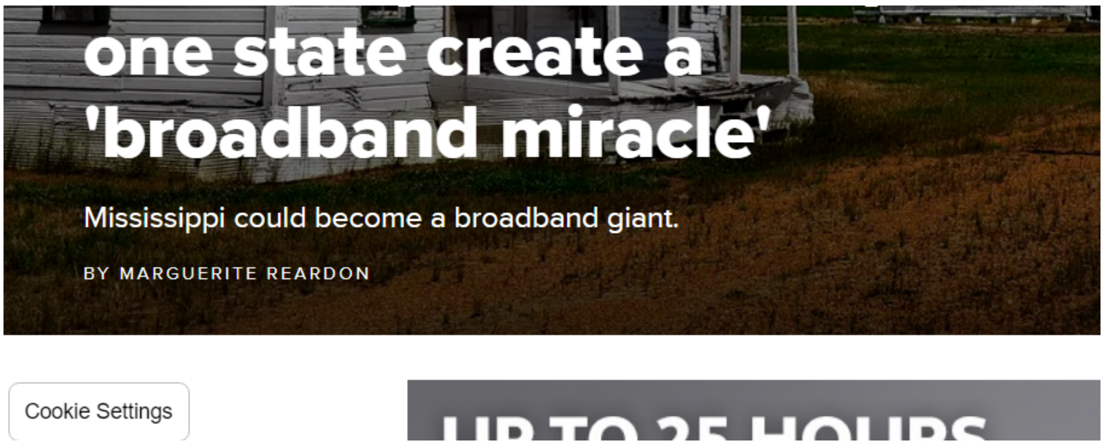
Before you continue

Google uses cookies and other data to deliver, maintain, and improve our services and ads. If you agree, we'll personalize the content and ads you see based on your activity on Google services like Search, Maps, and YouTube. We also have partners that measure how our services are used. Click "See more" to review your options or visit g.co/privacytools anytime.

Figure A.7: Two different cookie dialogs displayed on the same website using different connection endpoints. Cookie dialog on the left is displayed on google.com connecting from Lithuania, whereas the one on the right is displayed when connecting over the UK based VPN. Source: google.com

A.2 Non-dialog Elements

This section presents examples of pop-up windows and banners that are not considered to be cookie dialogs. For explanation refer to the title of the figure.

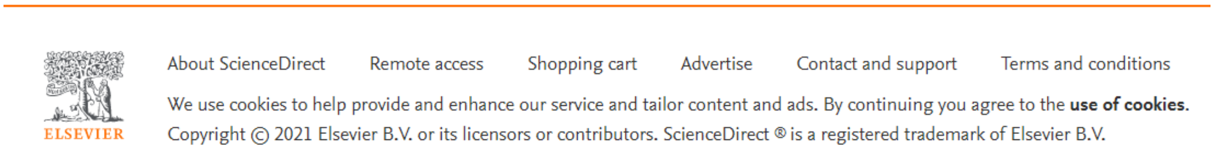


one state create a 'broadband miracle'

Mississippi could become a broadband giant.

BY MARGUERITE REARDON

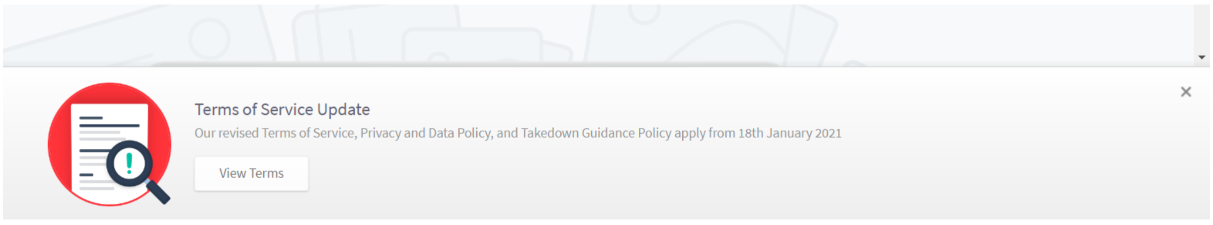
Figure A.8: Cookie Settings button (bottom-left) which opens a cookie dialog when clicked. Source: cnet.com


About ScienceDirect Remote access Shopping cart Advertise Contact and support Terms and conditions

We use cookies to help provide and enhance our service and tailor content and ads. By continuing you agree to the **use of cookies**.

Copyright © 2021 Elsevier B.V. or its licensors or contributors. ScienceDirect ® is a registered trademark of Elsevier B.V.

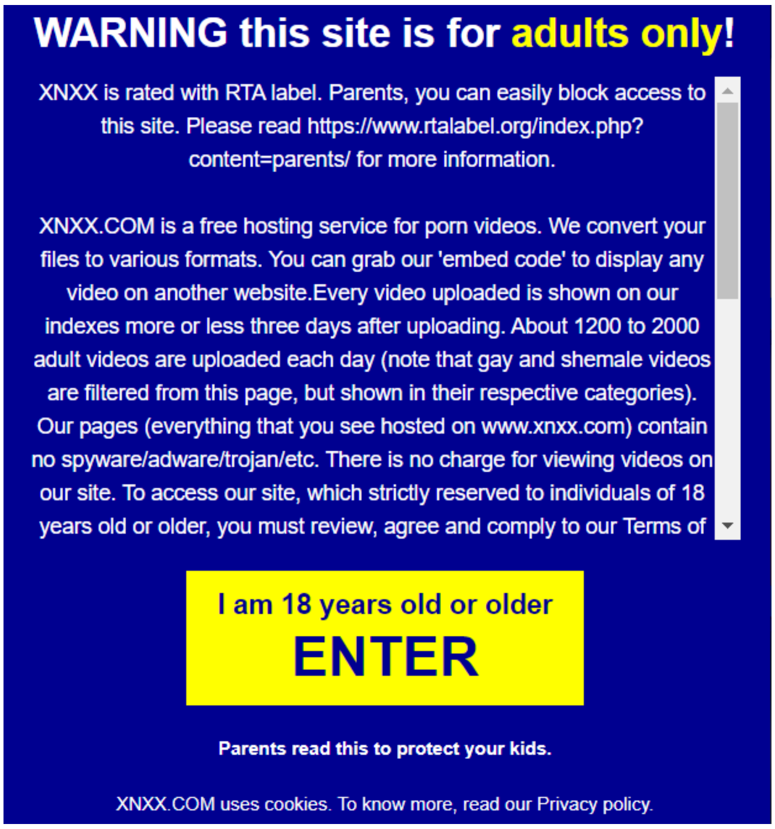
Figure A.9: Use of cookies mentioned in the footer of the web page. It is not interactive and in most cases not noticeable to a user. Source: sciencedirect.com


Terms of Service Update

Our revised Terms of Service, Privacy and Data Policy, and Takedown Guidance Policy apply from 18th January 2021

View Terms

Figure A.10: Banner informs in changes in Privacy and Data Policy. Although, policy itself is likely to include information about the use of cookies, it is not explicitly stated. Source: mega.nz


WARNING this site is for adults only!

XNXX is rated with RTA label. Parents, you can easily block access to this site. Please read https://www.rtalabel.org/index.php?content=parents/ for more information.

XNXX.COM is a free hosting service for porn videos. We convert your files to various formats. You can grab our 'embed code' to display any video on another website.Every video uploaded is shown on our indexes more or less three days after uploading. About 1200 to 2000 adult videos are uploaded each day (note that gay and shemale videos are filtered from this page, but shown in their respective categories). Our pages (everything that you see hosted on www.xnxx.com) contain no spyware/adware/trojan/etc. There is no charge for viewing videos on our site. To access our site, which strictly reserved to individuals of 18 years old or older, you must review, agree and comply to our Terms of

I am 18 years old or older
ENTER

Parents read this to protect your kids.

XNXX.COM uses cookies. To know more, read our Privacy policy.

Figure A.11: Pop-up window mentioning the use of cookies, however, the main contents, including the title and the text inside the enter button, are focused on the fact that the website contains age restricted content. Source: xnxx.com

Appendix B

List of evaluation domains

1. us2.net
2. mitech-ndt.com
3. bloodpressureuk.org
4. karnoenergy.com
5. motel-one.com
6. entrata.com
7. ss-ic.org.cn
8. hotesib.ru
9. unrxhwb.com
10. iheartcraftythings.com
11. tv-osaka.co.jp
12. qhnmdb.com
13. 01717.cn
14. hechingerreport.org
15. nyrr.org
16. zahav.ru
17. britishherald.com
18. cliffsnotes.com
19. caresoft.vn
20. btinternet.com
21. pymnts.com
22. kodakgallery.com
23. 51xuediannao.com
24. sverhestestvennoe.club
25. revealnews.org
26. bloter.net
27. fmhds.gov.ng
28. fortunegreece.com
29. fast2sms.com
30. newmatilda.com
31. knightfrank.co.uk
32. pengfu.com
33. pz.gov.pl
34. pixelation.org
35. quintly.com
36. eetop.cn
37. freeola.com
38. ccgp-shandong.gov.cn
39. samsungcloudsolution.com
40. wittenberg.edu
41. essay4you.net
42. reclaimthenet.org

43. porngem.com
44. 660citynews.com
45. leiphone.com
46. sportsgrid.com
47. diariodemallorca.es
48. sikporn.com
49. trackivation.com
50. elmastudio.de
51. technipfmc.com
52. hanjiecheng.com
53. fdcp.co.jp
54. vaporivape.com
55. bitsdujour.com
56. harlemglobetrotters.com
57. streamguys1.com
58. wirtualnemedia.pl
59. xhamster32.com
60. finra.org
61. codesandbox.io
62. finpecia365.com
63. mikrocontroller.net
64. wpsoul.net
65. anedot.com
66. aterm.jp
67. 13baba.com
68. managebac.com
69. yacht.de
70. chabad.org
71. seoreviewtools.com
72. teach.com
73. biopharmadive.com
74. coronavirus.gob.mx
75. minimalistbaker.com
76. taxjustice.net
77. cca.edu
78. 3y1.xyz
79. surferseo.com
80. reedsy.com
81. ipleer.com
82. surf.nl
83. 9news.com.au
84. totalplay.com.mx
85. medium.systems
86. exrwalq.net
87. noz.de
88. ecoticias.com
89. ubuntu-mate.org
90. kinoihoote2.site
91. 1tv.am
92. cheapauthenticmlbjerseys.com
93. openrepository.com
94. officialpayments.com
95. rubik.com.cn
96. pentagram.com
97. geoengineeringwatch.org
98. slots-usa-casino.com
99. mporg.ir
100. progressionstudios.com