Detection and measurement of web tracking

Imane Fouad

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Détection et mesure des techniques avancées du suivi dans le web

Imane FOUAD

Laboratoire: Université Côte d'Azur / Inria

Présentée en vue de l'obtention du grade de docteur en Informatique **de l'**Université Côte d'Azur
Dirigée par:
Nataliia Bielova, Arnaud Legout
Soutenue le: 29 Juin 2021

Devant le jury, composé de:
Thorsten Holz, Professeur, Ruhr-Universität Bochum
Martin Johns, Professeur, Technical University Braunschweig
Christo Wilson, Professeur Assistant, Northeastern University, Boston
Oana Goga, Chercheuse (CR), CNRS
Pierre Laperdrix, Chercheur (CR), CNRS
Yves Roudier, Professeur, EURECOM

Université Côte d'Azur/Inria

THÈSE DE DOCTORAT

présentée par

Imane Fouad

soutenance prévue le 29 Juin 2021
pour obtenir le grade de docteur EN INFORMATIQUE D'UNIVERSITÉ CÔTE D'AZUR

Détection et mesure des techniques avancées du suivi dans le web

Thèse dirigée par:

Dr. Nataliia Bielova	Chercheuse (Chargé de Recherche)	Inria Sophia Antipolis - Méditerranée
Dr. Arnaud Legout	Chercheur (Chargé de Recherche)	Inria Sophia Antipolis - Méditerranée

Rapporteurs:

Prof. Thorsten Holz	Professeur	Ruhr-Universität Bochum
Prof. Martin Johns	Professeur	Technical University Braunschweig

---

Membres du jury:

Dr. Christo Wilson	Professeur Assistant	Northeastern University, Boston
Dr. Oana Goga	Chercheuse (Chargé de Recherche)	CNRS (Laboratoire d'Informatique de Grenoble)
Dr. Pierre Laperdrix	Chercheur (Chargé de Recherche)	CNRS (CRIStAL laboratory)
Prof. Yves Roudier	Professeur	EURECOM

Résumé

Le web est devenu une partie essentielle de nos vies : des milliards utilisent quotidiennement des applications web et, ce faisant, plaçent des traces numériques sur des millions de sites web. De telles traces permettent aux entreprises de suivi de recueillir des données liées à l'utilisateur en utilisant une gamme de techniques de suivi, par conséquent, d'inférer les préférences et les habitudes de l'utilisateur. La collecte de ces données permet aux entreprises de fournir aux utilisateurs des publicités ciblées. Ceci est rentable pour les entreprises de publicité, mais intrusif pour la vie privée des utilisateurs.

Dans cette thèse, nous avons détecté et mesuré les technologies de suivi web. Nous avons également vérifié la conformité juridique des sites web dans le cadre juridique de la protection des données de l'UE en évaluant leur conformité avec le règlement général sur la protection des données (RGPD) et la directive ePrivacy.

Tout d'abord, nous avons proposé une classification comportementale du suivi basée sur l'analyse des pixels invisibles. Nous avons utilisé cette classification pour détecter de nouvelles catégories de suivi et découvrir de nouvelles collaborations entre les domaines sur un ensemble de 4,264,454 requêtes vers des domaines tiers. Nous avons démontré que les méthodes populaires pour détecter le suivi, basées sur EasyList&EasyPrivacy et sur Disconnect respectivement, échouent à détecter 25,22% et 30,34% des traqueurs que nous détectons.

Suite à ce premier travail, nous avons développé ERnie - une extension qui visualise les six techniques de suivi détectées à l'aide des pixels invisibles. Nous avons ensuite réalisé une étude qualitative et fait une analyse avec ERnie sur 176 sites web de médecins et d'hôpitaux. Nous avons constaté qu'au moins une forme de suivi se produit sur 64% des sites web avant d'interagir avec la bannière de consentement, et que 76% de ces sites web ne respectent pas les exigences du RGPD sur le consentement explicite valide.

Ensuite, nous avons étudié la combinaison des techniques de suivi web sans état et avec état. Au meilleur de notre connaissance, notre étude est la première à détecter

et à mesurer la recréation de cookies via les empreintes digitales de la machine et du navigateur. Nous avons mis au point une méthode de détection qui nous a permis de détecter la dépendance des cookies aux fonctionnalités du navigateur et de la machine. Nous avons découvert que cette technique peut être utilisée pour suivre les utilisateurs à travers les sites web même lorsque les cookies tiers seront obsolètes.

Enfin, nous avons évalué la conformité juridique des cookies qui sont au cœur des techniques de suivi précédemment étudiées. Nous enquêtons sur la conformité juridique des finalités pour 20,218 cookies tiers. Étonnamment, seulement 12,85% des cookies tiers ont une politique de cookies correspondante. Dans l'ensemble, nous avons constaté que les finalités déclarées dans les politiques de cookie ne sont pas conformes au principe de spécification de finalité dans 95% des cas dans notre audit automatisé. En outre, nous avons analysé les recommandations des services tiers mises en œuvre pour exercer le droit d'accès. Nous avons observé que certaines procédures d'authentification sont dangereuses ou douteuses.

mots-clés: suivi en ligne, pixels invisibles, synchronisation des cookies, RGPD, ePrivacy, consentement explicite, données sur la santé, empreintes digitales, recreation de cookie

Abstract

The web has become an essential part of our lives: billions are using web applications on a daily basis and while doing so, are placing *digital traces* on millions of websites. Such traces allow tracking companies to collect data related to the user using a range of tracking techniques, thus, to infer user's preferences and habits. The collection of this data allows companies to provide to the users targeted ads. Which is profit making for advertising companies, yet it is very intrusive for users privacy.

In this thesis, we detected and measured web tracking technologies. We further audited the legal compliance of websites within the EU data Protection legal framework by assessing their compliance with the General Data Protection Regulation (GDPR) and the ePrivacy Directive.

First, we proposed a fine-grained behavioral classification of tracking based on the analysis of invisible pixels. We used this classification to detect new categories of tracking and uncover new collaborations between domains on the dataset of 4,216,454 third-party requests. We demonstrated that popular methods to detect tracking, based on EasyList&EasyPrivacy and on Disconnect lists respectively miss 25.22% and 30.34% of the trackers that we detect.

As a follow up of this first work, we developed Ernie - a browser extension that visualises the six tracking techniques detected using the invisible pixels. We then made a qualitative study, and reported on the analysis with Ernie on 176 websites of medical doctors and hospitals. We found that at least one form of tracking occurs on 64% websites before interacting with the consent banner, and 76% of these websites fail to comply with the GDPR requirements on a valid explicit consent.

Next, we studied the combination of both stateful and stateless web tracking techniques. To the best of our knowledge, our study is the first to detect and measure cookie respawning via browser and machine fingerprint. We developed a detection methodology that allowed us to detect cookies dependency on browser and machine features. Our results showed that over 3.8% of the top 30,000 Alexa websites deployed this tracking mechanism. We found out that this technique can be used to track users

across websites even when third-party cookies are deprecated.

Finally, we assessed the legal compliance of the cookies that are the core of the previously studied tracking techniques. We investigate the legal compliance of purposes for 20,218 third-party cookies. Surprisingly, only 12.85% of third-party cookies have a corresponding cookie policy where a cookie is even mentioned. We found that purposes declared in cookie policies do not comply with the purpose specification principle in 95% of cases in our automatized audit. Furthermore, we analyzed the authentication practices implemented in third-party tracking services to exercise the access right. We observed that some data controllers use unsafe or doubtful procedures to authenticate data subjects.

Keywords: online tracking, ad-blocker, invisible pixels, cookie syncing, browser extension, GDPR, ePrivacy, explicit consent, health data, fingerprinting, cookie respawning

Acknowledgements

In a first place, I would like to thank my supervisors Nataliia Bielova and Arnaud Legout. Thank you for introducing me to the research world during my master's internship, and showing me the beauty of science! Thank you for believing in me and offering me the PhD position! Thank you for your commitment, kindness, and support! To be supervised by a great researcher is the dream, to be supervised by two great researchers is a life chance. Nataliia and Arnaud helped me to grow not only from a professional perspective, but also as a person. They offered me a productive and unique research environment that combines Arnaud's long experience and Nataliia's energy and passion. To you I'll be eternally grateful.

To my colleagues, and my previous office mates: Minh Ngo, Francis Dolière Somé, Jayanth Krishnamurthy, Yoon Seok Ko, Feras Al Kassar, Michael Tooth, Célestin Matte. Thank you for making my integration in the team as easy! thank you for the incredible time and the great discussions! I would also like to thank Cristiana Santos, our external collaborator, it was a real pleasure working with you.

To my husband, thank you for always putting forward the achievement of my dreams and my well-being. Thank you for always believing in me, even during the moments when I lost faith in myself! Thank you for sharing my moments of happiness and especially for raising me from my moments of sorrow!

To my sister, my closest friend, thank you for making a foreign country feel like home! Thank you for being always by my side, thanks to you I never felt alone. Even if I dedicate you a whole chapter, I would never be able to thank you properly.

À mes parents, qui étaient toujours présents malgré la distance. Je vous remercie et je vous dois toute réussite dans ma vie. Je vous serai éternellement reconnaissante.

Contents

Chapter 1

Introduction

1 Motivation

In 1989, while working at CERN, Tim Berners-Lee built the first prototype of what we know today as the World Wide Web. The web was first conceived to automate the information sharing between scientists around the world. One of the key ideas of the early web is the universality: all computers involved in the communication have to speak the same language independently of their hardware characteristics and independently of the user's location and background. The HTTP (Hypertext Transfer Protocol) protocol was built to answer this need [130].

The HTTP protocol is considered to be a stateless protocol, that is every request/response are handled by the server separately. As a result, the server is not able to link requests sent from the same browser. In 1994, HTTP cookies [76] were developed for the first time by Lou Montulli to make e-commerce shopping carts possible. The HTTP cookies were accepted by default in all browsers enabling services such as e-commerce services, but also allowing to track users by assigning them with unique identifiers. In early 1996, the Financial Times published an article [219] to highlight the privacy risks inherent to the user's tracking. Overtime, tracking techniques evolved and became more elaborated, alike, the tracking ecosystem grew, and the collaboration between tracking domains lead to more intrusive privacy concerns [208].

In the last decade, numerous studies measured the prevalence of third-party trackers on the web [193, 19, 172, 96, 146, 50, 51, 150, 149, 243]. To detect trackers, the research community applied a variety of methodologies. The most known web tracking technique is based on *cookies*, but only some cookies contain unique identifiers and hence are capable of tracking the users. Some studies detect trackers by analysing cookie storage, and third-party requests and responses that set or send cookies [193, 146], while other works measured the mere presence of third-party cookies [150, 149]. However the detection of identifier cookies and analysing behaviors of third-party domains remained a complex task. Therefore, most of the state-of-the-art works that aim at measuring trackers at large scale rely on filter lists. *But, how efficient are filter lists at detecting trackers? How can we detect tracking behaviors and uncover collaborations between third parties?* In this context, and to answer these research questions, we designed our first study on detection of cookie based tracking technique (Chapter 3).

Web tracking is happening on different categories of data including the most sensitive ones. Health data is known to be one of the most sensitive types of data, and massive health data leaks is recognized to be of particularly high severity to the users' privacy, according to the French Data Protection Authority (CNIL) [71]. Searching for doctors online has become an increasingly common practice among web users since telemedicine peaked in 2020 during the global Covid-19 pandemic [128]. However, the mere visit to a doctor's website can reveal a lot about its visitor: one can infer which diseases a visitor has or is interested in. The sensitivity of data processed in such websites, thus the privacy implication of tracking in these websites driven our second work on detection of cookie based tracking on health related websites (Chapter 4).

Followed by the privacy concerns introduced by web tracking, and in order to answer user's privacy concerns, browsers are moving toward the deprecation of third party cookies [199]. But, *can this deprecation prevent cross-site tracking?* Our third work on cookie respawning with browser fingerprinting answers this question. To overcome the third party cookies prevention, we showed that trackers deploy a tracking technique that relies on the combination of both stateless and statefull tracking mechanisms. We found that such practice is deployed, and can help track users across websites without relying on third party cookies (Chapter 5).

In 2018, the General Data Protection Regulation (GDPR) went effective with the aim at protecting users personal data. With the GDPR in place, the rights of the European users have been strengthened. The GDPR defined a number of rights for data subjects, and obligations for data controllers. On the one hand, data controllers, as part of their accountability and transparency obligations, need to declare the purposes of cookies deployed in their websites. This leads to relevant questions such as: *How should purposes be described according to the purpose specification principle of GDPR (Art 5(1)(b))? And how to ensure a scalable auditing, enabled by automated means, for legal compliance of cookie purposes?* On the other hand, data subjects would like to benefit from the rights specified in the GDPR, but still wonder: *How do I exercise my access right? How do I prove my identity to the controller?* Our last contribution answers these research questions (Chapter 6).

2 Thesis outline and contributions

In this thesis, we tackle the privacy threats related to tracking users on the web from three different angles:

- Design of a methodology to detect the privacy invasive practice,
- Automatic websites auditing,
- Assessment of legal compliance of a detected practice.

To ease the understanding of the thesis, we start with a background and related work chapter that cover the concepts and technologies used through the thesis (Chapter 2).

The main goal of our work is to detect and measure new advanced forms of tracking techniques on the web. In Chapter 3, we studied cookie based tracking techniques, we deployed a new detection methodology based on the analysis of the tracking behavior of domains serving invisible pixels. Such methodology allowed us to uncover new trackers, and evaluate the efficiency of the existing tracking protection and detection mechanisms. We summarize the contributions of this chapter in Section 2.1.

As a follow up of this first work, in Chapter 4 we built an extension that detects and visualize the different detected tracking behaviors. We then used this extension to evaluate health related website, that are considered as one of the most sensitive category of websites. We summarize the contributions of this chapter in Section 2.2.

With the emergence of the deprecation of cookies in multiple browsers which will result on the removal of third party cookies, we made a study on the detection and measurement of cookie respawning via browser fingerprinting in Chapter 5, and we showed that such technique can overcame the deprecation of third party cookies. We summarize the contributions of this chapter in Section 2.3.

In this thesis, we covered not only the technical aspect of tracking techniques on the web, but we also made an interdisciplinary work with a legal scholar. In Chapter 6, we studied the legal compliance with the GDPR and ePrivacy directive in two aspects. First, we measured the compliance regarding the usage of cookies that are the core of the studied tracking techniques. We assessed such practice by auditing websites and analysing the compliance with the purpose specification principle. Second, we evaluated the compliance with the Subject Access Request principle that ensures to the users the right to access the data collected about them by the third party domains tracking them in the web. We summarize the contributions of this chapter in Section 2.4.

An overview of this thesis organization is described in Figure 1.1.

2.1 *Designing a new fine-grained behavior-based tracking detection*

Invisible pixels are the *perfect suspect for tracking*. They are routinely used by trackers to send information or third-party cookies back to their servers. Using the invisible pixels dataset, that is the requests and responses that lead to invisible pixels, we proposed a classification of tracking behaviors. Our results showed that pixels are still widely deployed: they are present on more than 94% of domains and constitute 35.66% of all third-party images. We found out that pixels are responsible for 23.34% of tracking requests, and that the most popular tracking content are scripts: a mere loading of scripts is responsible for 34.36% of tracking requests. By applying this classification on more than 4M third-party requests collected in our crawl, we have detected six categories of tracking and collaborations between domains. We showed that domains *synchronize first party cookies with third party cookies.* This tracking appears on 67.96% of websites. We then evaluated the effectiveness of filter lists and privacy browser extensions at blocking the detected tracking request. Our evaluation of the effectiveness of EasyList&EasyPrivacy [90, 91] and Disconnect [86] lists showed that they respec-

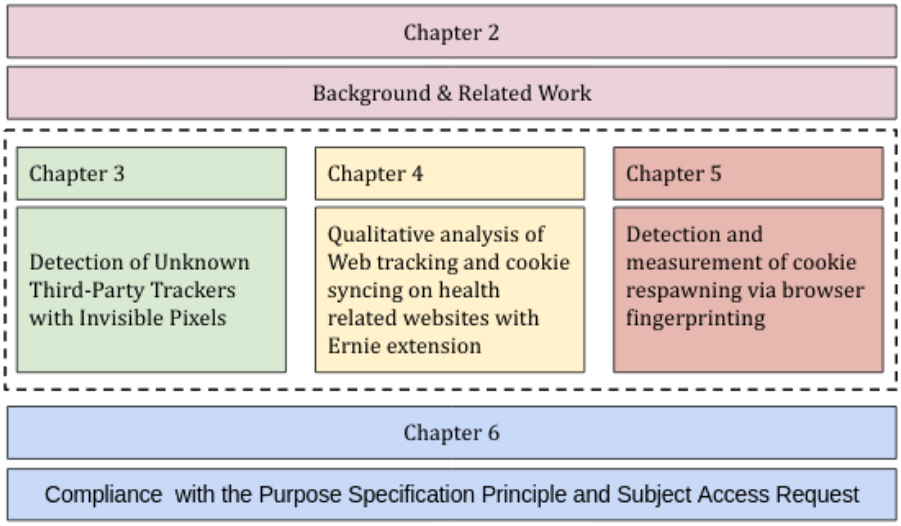


Figure 1.1: **Overview of the thesis structure.**

tively miss 25.22% and 30.34% of the trackers that we detect. Moreover, we find that if we combine all three lists, 379,245 requests originating from 8,744 domains still track users on 68.70% of websites. We further evaluated the popular privacy protection extensions: Adblock [21], Ghostery [120], Disconnect [86], and Privacy Badger [181], we showed that Ghostery is the most efficient among them and that all extensions fail to block at least 24% of tracking requests (Chapter 3).

This work was published at the Privacy Enhancing Technologies Symposium (PETs'20):

- **Imane Fouad**, Nataliia Bielova, Arnaud Legout, Natasa Sarafijanovic-Djukic. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. *Privacy Enhancing Technologies Symposium (PoPETS 2020)* [113].

2.2 *Auditing health related websites*

Based on the behavior-based tracking techniques, we made a technical and legal qualitative study on websites processing special category of data. We analyzed the presence of tracking behaviors on 176 health related websites, and we identified practices that can potentially violate the GDPR and ePrivacy directive. We found that 64% of the websites track users before any interaction with the banner. Moreover, 76% of these websites fail to comply with the legal requirements for a valid explicit consent: out of 176 studied websites, 46% do not display a cookie banner, and 75% thereof still contain tracking, thus violating the *explicit consent* legal requirement; 26% of the websites

provide a cookie banner without a reject button, and 86% of these websites include tracking, hence violating the requirement to give users *the possibility to reject tracking*. Moreover, we show that the *user choice is not respected* on health related websites: 33 (19%) websites still contain tracking after cookie rejection. We further analyzed in depth 5 case study websites, and for each of these websites we provide a comprehensive and detailed legal and technical analysis. We found that in every 45 webpages wherein doctors include a Google map to help locating their office, google.com receives its identifier cookie. While Google maps doesn't explicitly track users, tracking happens because of the NID cookie of google.com that is already present in the user's browser, and the HTTP standard [130] that requires cookies to be automatically attached to every outgoing HTTP(S) request (Chapter 4).

This work is under submission :

- **Imane Fouad***, Vera Wesselkamp*, Cristiana Santos, Nataliia Bielova, Arnaud Legout. Qualitative analysis of Web tracking and cookie syncing on health related websites with Ernie extension. Under submission.

2.3 *Detecting and measuring the prevalence and the privacy implications of cookie respawning via browser fingerprinting.*

Several related works studied stateful and stateless tracking techniques, however, to the best of our knowledge, we are the first to study how trackers can benefit from the combination of the both stateful and stateless tracking techniques. First, we designed a robust method to identify which features are used to respawn a cookie. This methodology allowed us to automatically identify the set of fingerprinting features used to generate a cookie. We strengthen our detection methodology by adding a permutation test ($N=10{,}000, p<0.05$). Next, we showed that the stateful and stateless tracking techniques that were studied separately are, in fact, actively used together by trackers on 1,150 (3.83%) out of the Alexa top 30,000 websites. Then, we identified who is responsible of respawning the cookies with the browser fingerprinting, and showed that this tracking technique is highly deployed in popular websites. Finally, we assessed the legal consequences of such practice together with a legal expert co-author (Chapter 5).

This work is under submission to the IEEE Symposium on Security and Privacy (S&P'22):

- **Imane Fouad**, Cristiana Santos, Arnaud Legout, Nataliia Bielova. My Cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning via browser fingerprinting. Under submission.

*: Co-first authors.

2.4 *Auditing the compliance with the Purpose Specification Principle and Subject Access Request.*

The enforcement of the General Data Protection Regulation (GDPR) and the ePrivacy Directive relies upon auditing legal compliance of websites. Data controllers, as part of their accountability and transparency obligations, need to declare the purposes of cookies that they use in their websites.

We investigated the legal compliance of purposes for 20,218 third-party cookies. Surprisingly, we found that only 12.85% of third-party cookies have a corresponding cookie policy where a cookie is even mentioned. Overall, we found out that purposes declared in cookie policies do not comply with the purpose specification principle in 95% of cases in our automatized audit.

Finally, we provide recommendations on standardized specification of purposes following the recent draft recommendation of the French Data Protection Authority (CNIL) on cookies.

The GDPR defines a number of rights for data subjects, including the subject access request (SAR). Such right allows individuals to obtain personal data collected about them. Every data subject would like to benefit from the rights specified in GDPR, but still wonders: *How do I exercise my access right? How do I prove my identity to the controller?* To answer these questions, we evaluated the threats introduced by the SAR, then we assessed the authentication practices implemented by third party services (Chapter 6).

This work is the basis of two submissions. Published at the International Workshop on Privacy Engineering (IWPE 2020), and at the Annual Privacy Forum (APF 2019):

- **Imane Fouad**, Cristiana Santos, Feras Al Kassar, Nataliia Bielova and Stefano Calzavara. On Compliance of Cookie Purposes with the Purpose Specification Principle. *International Workshop on Privacy Engineering (IWPE 2020)* [116].
- Coline Boniface, **Imane Fouad**, Nataliia Bielova, Cédric Lauradoux, and Cristiana Santos. Security Analysis of Subject Access Request Procedures. *Annual Privacy Forum (APF 2019)* [58].

Chapter 2

Background & Related Work

In this chapter, we present different concepts and technologies used throughout this thesis in order to ease the understanding of the following chapters. We complement this chapter with a presentation of the state of the art of these concepts and technologies. Figure 2.1 describes the organization of this chapter. Our work focuses on the study of the web tracking techniques (Section 2). First, we present an introduction to the web technologies that make tracking possible (Section 1). Next, we explain how web tracking techniques work (Section 2). Then, we introduce the existing tracking detection and protection mechanisms (Section 3). Furthermore, we assess the EU legal requirements background (Section 4). Finally, we position our work compared to the stated related works (Section 5).

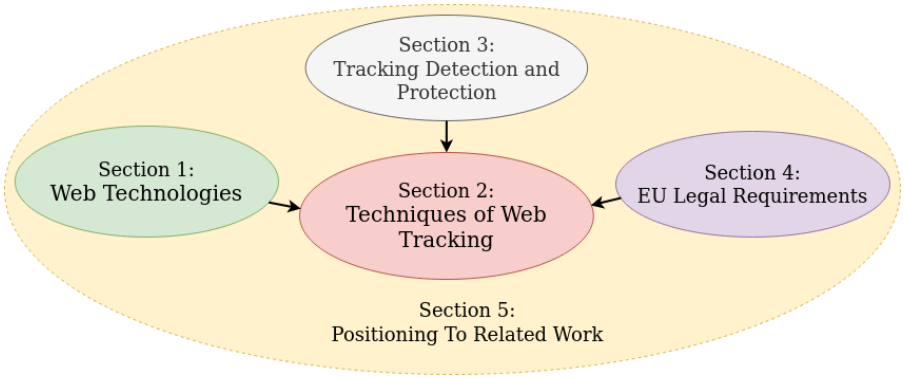


Figure 2.1: **Overview of the background & related work chapter structure.**

1 Web Technologies that make tracking possible

In this section, we describe the different web technologies that makes web tracking possible. Figure 2.2 summarizes the technologies and concepts presented in the section. First, we introduce the HTTP protocol. Next, we describe the scope of the cookie. Then, we explain the redirection and inclusion process. Finally, we distinguish between the first- and third-party content.

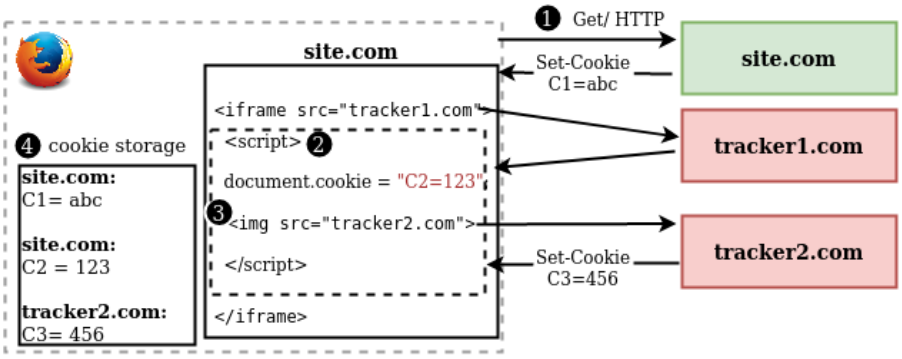


Figure 2.2: **Web technologies.** Upon a visit to website site.com, ❶ the browser sends a request to the web server using the HTTP protocol (Section 1.1). The browser then fetches the page content including the iframe from tracker1.com. ❷ The tracker1.com iframe includes a script that runs in the context of site.com and sets a cookie on the user's browser using *document.cookie* API. The resulting cookie C2 is stored in the user's browser with site.com as host (Section 1.2). ❸ In addition, the script from tracker1.com includes content from a different tracker tracker2.com that sets its own third-party cookie *C3* (Section 1.3). As a result, ❹ both first-party (*C1* and *C2*) and third-party cookies *(C3)* simultaneously used for within and cross-site tracking are stored in the users browser (Section 1.4).

1.1 *HTTP protocol*

The HTTP (HyperText Transfer Protocol) [130] protocol establishes a link between a web browser (client) and a web server. The browser sends a request to the server which gives it back a response. In other words, the HTTP communication protocol is what allows an Internet user to access content (a Web page, a CSS file, etc.). Both HTTP requests and responses carry a number of information including the URL (Uniform Re-source Locator) or web address of the resource to be accessed on the server, a list of key/value pairs HTTP headers providing additional information about the requests and responses, and eventually data (request body/response body).

An HTTP request and response are composed of a number of headers fields. Section 14 of RFC 2616 [191] defines the syntax and the semantics of all used standard HTTP header fields. Table 2.1 presents an example of HTTP request/response exchanged between the browser and the server in order to access www.google.com. In the following, we detail the HTTP fields that can be exploited for cookie based web tracking.

Cookie/Set-Cookie header: The HTTP protocol is considered to be a stateless protocol. That is, it handles each request/response independently from the previous one,

HTTP request

Host: www.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.24 Safari/537.36
Referer: https://developer.mozilla.org/
Cookie: NID=123456

HTTP response

Host: www.google.com
Status code: 200 OK
Set-Cookie: NID=123456; Max-Age=31449600; Path=/; secure

Table 2.1: **HTTP request/response exchanged between the client and the server upon visit to google.com.** Not all HTTP headers are presented. *User-Agent* header indicates browser and machine information such as the browser name, type, version and the OS on which it runs. *Referrer* header indicated the URL of the website from which the request is made. *Cookie* and *Set-Cookie* headers are used to set and receive cookies).

and therefore do not make the link between past and current requests sent from the same browser. However, most modern applications rely on the ability to identify the user either for functionality purposes such as authentication or for tracking purposes. In order to fulfill the deficiency introduced by the stateless mode, HTTP cookies were introduced. HTTP cookies are a small piece of information stored in the user browser. Using the *Set-Cookie* HTTP response header, a domain can set a cookie in the user's browser. This cookie is defined by the triplet (host, key, value), where host refers to the domain that sets the cookie. Next, when the browser sends a request to the same domain, it will automatically attach the cookies with a corresponding host in the *Cookie* header of the outgoing HTTP request. The usage of cookies helped website owners provide services to users and made web browsing easier. For instance, using cookies in e-commerce website helped to keep track of purchased items. Cookies are also used for authentication, and therefore the user is not required to resend her credential while browsing in a given website. Similarly to every emerging technology, several domains exploited the cookies and deployed them to track users across websites.

Referer header: The HTTP *Referer* header indicates the URL of the website from which the request is made. For instance, when a website site.com includes content from a different domain such as tracker.com, then the request made to load the content from tracker.com will indicate that the origin of the request is site.com using the *Referer* header.

To perform web tracking, a third party do not only need to uniquely identify the user across websites, but it should also be able to identify which website the user is visiting in order to recreate her browsing history. In order to detect what website the user is visiting, trackers may use an HTTP *Referrer* header, which indicates the URL of the content that embeds the tracker. By default, the browser sends the *Referer* field in every HTTP request. Third parties may also use other techniques to retrieve the visited page such as JavaScript calls to document.location.

1.2 *Scope of the cookie*

When a cookie is stored in the browser, it is identified by a tuple (*host*, key, value). If the cookie is set via an HTTP(S) response header, then the *host* of the cookie represents a domain that sets the cookie. However, when the cookie is set programmatically via a script included in the website, the script gets executed in the context, or "origin", where it is included. Due to the Same Origin Policy (SOP) [197]), the *host* of a cookie set by the script is the origin of the execution context of the script, and not the domain that contains the script. Given a cookie stored in the browser with its (*host*, key, value), when a browser sends a request to a domain, it attaches a cookie to the request if the cookie *host* matches the domain or the subdomain of the request [161].

In this thesis, we differentiate between *cookie host* and *cookie owner*. More precisely, this terminology is used in Chapter 5. A cookie *owner* is either a domain that sets a cookie via HTTP(S) response header (and in this case, matches with cookie *host*), or the domain that hosts a script that set the cookie programmatically (generally speaking, here the owner is different from the host). This practice is described in Figure 2.2: site.com is a website that includes a third-party script from tracker1.com. After loading, the script sets a cookie *C2=123* in the context of the visited website site.com. In this case, the cookie owner is tracker1.com, but the cookie host is site.com.

1.3 *Redirection/Inclusion*

Redirection process is used to push the client to resubmit a request to a new location. To do so, the server sends a special response with a *status code* HTTP header that starts with 3, such as 301 (Moved Permanently), or 302 (Temporary redirection) [188]. Along with the redirection response, the server sends a *Location* header that indicates the URL to redirect to. Using the redirection process, trackers can collaborate and include each other on websites where they were not initially included.
When the user visits a website that includes content from a third party, the third party can redirect the request to a second third party tracker directly via HTTP redirection as described in Figure 2.3. We detect redirection chains as follows.

1. If HTTP redirect, that is the status code is set to 30x, preserve the current URL and follow location URL.

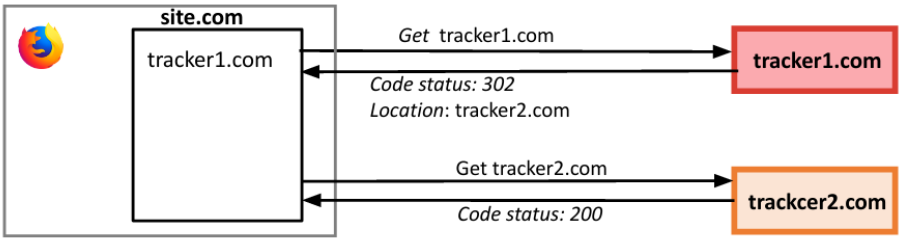


Figure 2.3: **Redirection process.** First, site.com includes tracker1.com. Then, tracker1.com redirects the request to tracker2.com using HTTP redirection process

2. We continue to the next location URL as long as response status code is equal to 30X.

To include another domain, in addition to the redirection process, trackers can also rely on inclusion as described in Listing 2.1 and 2.2. To detect the sender of the request in case of inclusion, we use the referer field.

In the following, we consider both HTTP redirection and inclusion process, for simplicity sake, we will use the term *redirection* to refer to both.

Listing 2.1: **Site.com HTML document**

```
1 <html>
2     <head>
3         <title> site.com </title>
4     </head>
5     <body>
6         <iframe src="tracker1.com"
              ></iframe>
7     </body>
8 </html>
```

Listing 2.2: **Tracker1.com HTML document**

```
1 <html>
2     <head>
3         <title>tracker1.com</title
              >
4     </head>
5     <body>
6         <img src="tracker2.com">
7     </body>
8 </html>
```

Bashir et al. [50] designed an instrumented version of Chromium that produces redirection trees directly from Chromium's resource loading code. Their solution is only running under Chromium, thus can not be implemented in our study under Firefox. In fact, to crawl websites, we used OpenWPM, an open source crawler that is running under Firefox.

1.4 *First- and third-party content*

The websites are composed of a first-party content and numerous third-party contents, such as advertisements, web analytic scripts, social widgets, or images. Following the standard naming [146], for a given website we distinguish two kinds of domains: the *first-party* domain that is the domain of the website, and *third-party* domains that are domains of the *third-party* content served on the website. Using HTTP request (or response), any content of the webpage can set (or receive) cookies. Additionally, cookies can be set programmatically via an included JavaScript library. Every cookie is stored in the browser with an associated domain and path, so that every new HTTP request sent to the same domain and path gets a cookie associated to it attached to the request.

First-party cookies set by first-party domains or programmatically via scripts running in the context of the website are capable to track users *within the same website*. Third-party cookies set by third-party domains allow third parties to track users *cross-websites* [193]. First- and third-party cookies are defined in the context of the visited website. In fact, a given cookie can be considered both as first- and third-party cookie depending on the visited website (for example, google.com's cookie is a first-party cookie when visiting google.com, but it is considered as a third-party when it is sent with the request to google.com upon a visit to a different website).

2 Techniques of Web Tracking

In this section, we present different tracking techniques. We first introduce the analytics and cross-site tracking. Next, we describe the cookie syncing, followed by the cookie respawning, and finally we present browser fingerprinting.

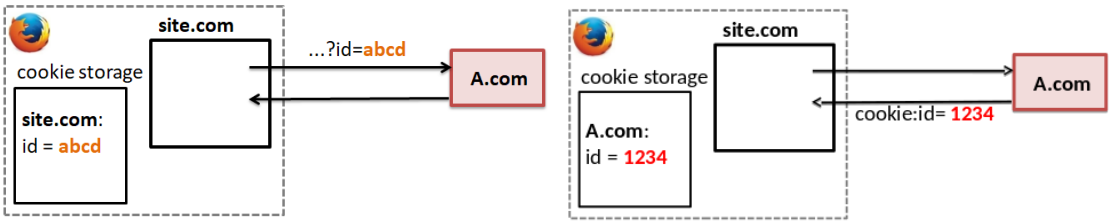
2.1 *Within-site and cross-site tracking*

There are two main categories of statefull tracking techniques: within site tracking (or analytics), and cross-site tracking.

Analytics services deploy first-party cookies to track repeat visits within a website. Figure 2.4a describes the analytics behavior. The website site.com directly visited by the user includes content from the analytics service A.com. A.com first sets a first-party cookie in the user's browser in the context of site.com. The cookie is then sent as part of the request parameters to A.com. Using this tracking technique, one can link user's browsing activity only within the same website.

Using third-party cookies, domains can track user's activity cross websites. Figure 2.4b presents the cross-site tracking behavior. When the user directly visits site.com that includes content form a third-party tracker A.com, the browser automatically sends a requests to A.com to fetch the content. Using the HTTP request/response, A.com sets a third-party cookie on the user's browser. A.com receives this cookie

when the user visits different websites that include A.com content. Therefore, A.com is able to link user's activity across multiple websites.



(a) Analytics tracking behavior (b) Cross-site tracking behavior

Figure 2.4: **Analytics and cross-site tracking behaviors**

Analytics and cross-site tracking are the main cookie based tracking categories. Therefore they have been widely studied in the past decade [210, 193, 46, 174, 96, 92, 19, 172, 146, 187]. Previous works showed that Google is the top organisation performing these tracking behaviors. In fact, Englehardt et al. [96] showed that Google is tracking users on over 70% out of the 1 million visited websites

2.2 *Cookie syncing*

Every cookie stored in the user's browser is only accessible to the domain that set it due to the Same Origin Policy (SOP) [197]. However, third parties are interested in merging information they collected about the users and recreate a more complete history of the user's browsing. To do so, third parties exploit *cookie syncing* or *cookie matching* [18, 96, 174, 50, 178] to share identifiers of the same user among the third parties. A typical demonstration of cookie syncing that has been analysed in the related works in shown in Figure 3.7. Cookie syncing is often used in the Real-Time-Bidding auction for targeted advertisement [131].

Previous studies [18, 174, 96, 50, 178] measured cookie syncing. Olejnik et al. [174] consider cookies with sufficiently long values to be identifiers. If such identifier is shared between domains, then it is classified as cookie syncing. Additionally, Olejnik et al. [174] studied the case of doubleclick to detect cookie syncing based on the URL patterns. Bashir et al. [50], used retargeted ads to detect cookie syncing. Papadopoulos et al. [178] used a year long dataset from real mobile users to study cookie syncing.

2.3 *Cookie respawning*

Cookie respawning is the process of automatically recreating a cookie deleted by the user (usually by cleaning the cookie storage). The term *cookie respawning* was first introduced in 2009 by Soltani et al. [210]. Several techniques can be used to respawn a cookie.

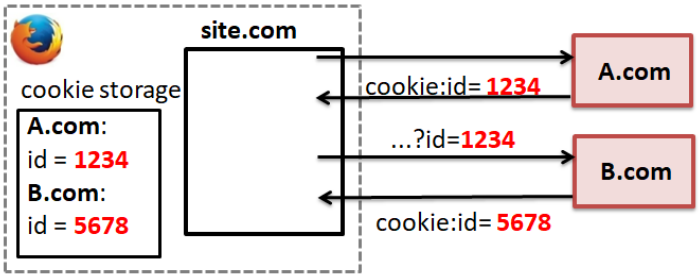


Figure 2.5: **Third to third party cookie syncing behavior.** First, site.com includes A.com, a request is then sent to A.com along with it's cookie. Next, A.com redirects the request to B.com and includes it's identifier *1234* in the request URL. Then, B.com receives the request along with it's cookie. This behavior allows A.com and B.com to combine the data collected about the user.

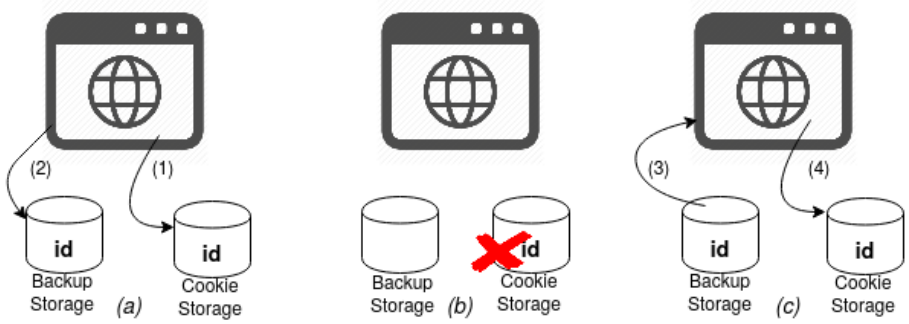


Figure 2.6: **Cookie respawning.** (a) The tracker (1) sets an identifier cookie in the user's cookie storage, and (2) stores a backup of the identifier in a different storage. (b) The user cleans her cookie storage. (c) (3) the tracker retrieves the identifier from the backup storage and (4) resets the removed cookie.

Figure 2.6 describes the cookie respawning tracking mechanism. When the user visits a website site.com, that uses cookies respawning tracking technique, the website generates a user identifier and stores it in multiple storages including cookies. Consequently, when the user deletes the cookie, the website can recover it using the backup storage.

Soltani et al. [210] showed that trackers are abusing the usage of the Flash cookies in order to respawn or recreate the removed HTTP cookies. This work attracted general audience attention [162, 163] and triggered lawsuits [144, 145]. Following Soltani et al. work, other studies started analyzing the usage of other storages for respawning such as ETags and localStorage [46]. Sorensen studied the usage of the browser cache

in cookie respawning [213]. Acar et al. automated the detection of cookie respawning and found that IndexedDB can be used to respawn cookies as well [19]. Roesner et al. showed that cookies can be respawned from the local and Flash storage [193].

2.4 *Browser fingerprinting*

Browser fingerprinting is a stateless tracking technique that provides the ability to identify and track users without using their browser storage [92, 19, 64, 31], unlike cookie-based tracking. When a user visits a web page that performs fingerprinter, this fingerprinter will return to the server a list of features composed of user's browser and machine characteristics such as a user agent or a time zone. The trackers use the collected features to build a unique identifier.

Following a definition of Laperdrix et al. [139], a browser fingerprint is a set of information related to a user's device from the hardware to the operating system to the browser and its configuration. Trackers often perform *browser fingerprinting*, that is collecting information through a web browser to build a unique identifier from a fingerprint of a device. To build this unique identifier, a tracker relies on several browser and machine characteristics, such as the user agent, WebGL or the IP address. In the rest of this thesis, we use the term *feature* to refer to these characteristics.

When a user visits a web page with some content located on a tracker's server, the user's browser sends a request to the server to fetch this content. This HTTP(S) request contains several HTTP headers, such as user agent, and an IP address that tracker's server receives *passively*. We refer to such information as *passive features*. To collect additional information, the tracker can include in the visited web page a JavaScript script that is then executed on the user's browser. The script retrieves multiple browser and machine information, such as the time zone, and sends them to a server of the remote tracker. We refer to such information as *active features*.

In 2010, the Panopticlick study [92] showed that fingerprints can be potentially used for web tracking. Following this study, several fingerprinting tracking techniques were discovered. Acar et al. [19] studied canvas based fingerprinting. Englehardt et al. [96] presented a new fingerprinting technique based on the AudioContext API. Cao et al. [64] presented a fingerprinting study mainly based on hardware features including WebGL. Al-Fannah et al. [31] studied fingerprinting in Majestic top 10,000 websites. Most recently, Solomos et al. [209] combined browser fingerprinting and favicons caches to identify users.

3 Tracking Detection and Protection

In this section, we present the existing tracking detection and protection mechanisms. First, we explain how filter lists are used to detect and block trackers (Section 3.1). Then we introduce the privacy protection browser extensions (Section 3.2).

3.1 *Detection and blocking of tracking with filter lists*

Filter lists contain a list of regular expressions or domain names that define content to be blocked. The most widely used filter lists are EasyList [90] and EasyPrivacy [91]. EasyList&EasyPrivacy are based on a set of rules originally designed for Adblock [21]. They are maintained by 4 people and are complemented by a community effort. EasyList is the primary filter list, it is a Ad-blocking list used to remove ads in the website. It is complemented by EasyPrivacy which is a tracker-blocking list used to remove tracking behaviors. In the following, we use EL&EP to refer to the two filter lists EasyList&EasyPrivacy.

To detect domains related to tracking or advertisement, most of the previous studies [96, 50, 141, 187, 132, 95, 51, 49, 135] rely on filter lists, such as EasyList and EasyPrivasy (EL&EP) that became the *de facto* approach to detect trackers. We differentiate between the usage of filter lists to detect and to block tracking requests:

- **Detection:** We consider a request as detected by the filter list if it is directly matching the list.
- **Blocking:** We consider that a request is blocked if it matches one of the following cases:
  - the request directly matches the list (detected).
  - the request is a consequence of a redirection chain where an earlier request was blocked by the list.
  - the request is loaded in a third-party content (an iframe) that was blocked by the list.

Only from 2016 to 2020 we identified *16 papers that rely on EL&EP* to detect third-party tracking and advertising. Table 2.2 presents the list of these papers. Englehardt and Narayanan [96] seminal work on measuring trackers on 1 million websites relies on EL&EP as a ground truth to detect requests to trackers and ad-related domains. Three papers by Bashir et al. [50, 49, 51] customize EL&EP to detect 2^{nd}-level domains of tracking and ad companies: to eliminate false positives, a domain is considered if it appears in the dataset more than 10% of the time in the dataset. Lauinger et al. [141] use EL&EP to identify advertising and tracking content in order to detect what content has included outdated and vulnerable JavaScript libraries in Web applications. Razaghpanah et al. [187] use EasyList as an input to their classifier to identify advertising and tracking domains in Web and mobile ecosystems. Ikram et al. [132] analysed how many tracking JavaScript libraries are detected by EL&EP based on 95 websites. Englehardt et al. [95] applyed EL&EP on third-party leaks caused by invisible images in emails. Iordanou et al. [135] relyed on EL&EP as a ground truth for detecting ad- and tracking-related third party requests. Only one work by Papadopoulos et al. [179] use Disconnect list [87] to detect tracking domains. Roesner et al. [193] and Lerner et al. [146] were the first to analyze trackers based on their behavior. They have proposed a classification of tracking behaviors that makes a distinction between analytics and cross-domain tracking. Yu et al.[243], identify trackers by detecting unsafe data

Paper	Venue	EasyList	EasyPrivacy	Usage	Dependency
Englehardt and Narayanan [96]	ACM CCS 2016	✓	✓	Detected	Rely
Moghaddam et al.[166]	ACM CCS 2019	✓	✓	Custom	Rely
Bashir et al. [50]	USENIX Security 2016	✓		Custom	Rely
Lauinger et al. [141]	NDSS 2017	✓	✓	Blocked	Rely
Razaghpanah et al. [187]	NDSS 2018	✓		Custom	Rely
Degeling et al.[84]	NDSS 2019	✓		Detected	Rely
Ikram et al. [132]	PETs 2017	✓		Blocked	Verif
Englehardt et al.[95]	PETs 2018	✓	✓	Blocked	Verif
Bashir and Wilson [51]	PETs 2018	✓	✓	Custom	Rely+Verif.
Cook et al.[75]	PETs 2020	✓	✓	Custom	Rely
Yang et al.[242]	PETs 2020	✓	✓	Custom	Rely
Bashir et al.[49]	IMC 2018	✓	✓	Custom	Rely
Iordanou et al.[135]	IMC 2018	✓	✓	Blocked	Rely
Alrizah et al.[34]	IMC 2019	✓	✓	Custom	Verif
Vallina et al.[228]	IMC 2019	✓	✓	Detected	Verif
Bashir et al.[48]	IMC 2019	✓		Detected	Rely

Table 2.2: **Usage of EL&EP lists in security, privacy, and web measurement communities (venues from 2016-2020).** "Usage" describes how EL&EP was used to detect trackers: whether the filter lists were applied on all requests, ("Detected"), on requests and follow-up requests that would be blocked, ("Blocked") or whether filter lists were further customised before being applied to the dataset ("Custom"). In the dependency column, "Rely" means that the authors use the EL&EP to build their results, "verify" means that the authors only use EL&EP lists to verify their results.

without taking into account the behavior of the third-party domain and the communications between trackers.

3.2 *Privacy protecting browser extensions*

Browser extensions are third-party programs that allows users to extend the functionalities of their browser. In particular, privacy protection browser extensions are used

to improve user's privacy online by detecting and blocking tracking behaviors. We distinguish two types of privacy protecting browser extensions.

- **Ad Blockers:** They are used to block ads on a given website. The most popular Ad blocker extension is AdBlock Plus [23]. It relies on EasyList filter list and is used to block ads from being loaded on the website.
- **Tracker Blockers:** The browser extensions in this category focus on blocking trackers. Ghostery is one of the most popular extension in this category.

Several browser extensions provide a visualization of the tracking detected on the visited website. Disconnect [85] shows third-party inclusion chains, while uBlock Origin [129] shows which part of a URL is responsible for tracking. The Lightbeam extension [169] visualizes which third parties are included on which websites. All these extensions only provide a very limited overview of the tracking on a website. Website scanners [108, 77, 180, 233] allow a user to see what cookies are set on a website in order to determine if the website is compliant with the GDPR. The EDPS Inspection Software [216] gives detailed information about web traffic caused by a website, as well as trackers based on the EasyPrivacy filter list. The CNIL's Cookieviz 2 [151], visualizes which third-party domains occur on which websites on a sequence of visits. It also shows if the domains dropped a third-party cookie and if that cookie is listed in an ads.txt file, indicating that it is used for advertisement.

4 EU Legal Requirements for web Tracking and consent

The **General Data Protection Regulation** (GDPR) [218] (that came into force in May 2018) is a European legislation that regulates the processing of personal data of data subjects. It aims to *"strengthen individuals' fundamental rights in the digital age and facilitate business by clarifying rules for companies and public bodies in the digital single market"*.

The scope of the GDPR is applicable to organisations located inside and outside of the EU, as long as they collect and process personal data from European users regarding activities related to offering goods or services or monitoring user's behavior (Article 3 of GDPR) [38]. It requires organizations need to choose a legal basis to lawfully process personal data (Article 6(1)(a)). The GDPR is one of the most severe regulations in the world as it levy harsh fines against those breaching it – that can max out to €20 million or 4% of global annual turnover, which ever is higher [39].

The **ePrivacy Directive** (ePD) [99] provides *supplementary* rules to the GDPR, in particular in the electronic communication sector, such as websites, and requires that website publishers have to rely on the *user consent* when they collect and process personal data using non-mandatory (non strictly necessary for the service requested by the user) cookies or other tracking technologies (Article 5 (3) ePD). Exceptions to consent refer to functional cookies which are used for communications and strictly necessary purposes.

Purpose. The European Data Protection Board (EDPB) [13] asserts that "it is the purpose (and its practical implementation) that must be used to determine whether or not a cookie can be exempted from consent". While analysing cookies present on a website, any auditor needs to capture the *purpose* of each cookie. The defined purpose can then help to determine whether processing is legally compliant, what safeguards the GDPR imposes, and which legal basis can be used.

Consent is defined in Article 4(11) and complemented by Articles 6 and 7 of the GDPR which states that for consent to be valid, it must satisfy the following requirements: it must be prior, freely given, specific, informed, unambiguous, revocable and finally, readable and accessible [201]. As such, consent cannot be implicit [201], it *"should be given by a clear affirmative act establishing a freely given, specific, informed and unambiguous indication of the data subject's agreement to the processing of personal data relating to him or her [...]. Silence, preticked boxes or inactivity should not therefore constitute consent."*

To collect user's consent, websites implement cookie banners. The presence of such banners are thus mandatory on websites performing tracking behaviors (as the ePD so demands).

In this section, we first introduce the concept of special categories of data defined by the GDPR. Next, we define cookies and privacy policies followed by a description of cookies purposes.

4.1 *Processing special categories of data*

Special categories of data. The GDPR [218] stipulates that personal data which are particularly sensitive by their nature merit specific protection, as their processing could create *significant risks to the fundamental rights and freedoms* (Recital 51). Such data include personal data revealing sensitive information such as racial or ethnic origin, political opinions, religious or philosophical beliefs, trade union membership, genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health and data concerning a natural person's sex life or sexual orientation [218, Article 9].

Explicit consent. Processing of such categories of data are *forbidden*, unless allowed by several exceptions (Article 9 (2) (b-j)), for example, the *explicit consent* of the user [218, Article 9(2)(a)]. Explicit consent is required in certain situations where *serious data protection risk* emerge, hence, where a high level of individual control over personal data is deemed appropriate, which is the case when considering processing special categories of data [106]. The GDPR prescribes that a "clear affirmative action" is a prerequisite for "regular" consent which was raised to a high standard. However, it needs to be clarified what *extra efforts* a controller should undertake in order to obtain the explicit consent of a data subject in line with the GDPR.

The EDPB [106] in its guidelines state that the term "explicit" refers to the way con-

sent is expressed by the data subject. Explicit consent then means that the data subject must give an express statement of consent. More tangibly, it proposes an extra effort to obtain explicit consent:

- Two stage verification of consent, *i.e.*, a double opt-in,
- A specific confirmation from the data subject.

In this line, in a digital era, explicit consent would be obtainable by filling an online form, scanning a written and signed statement, sending an e-mail, or even using an electronic signature. The EDPB alerts that such effort is justified *"to remove all possible doubt and potential lack of evidence in the future"*.

Previous works explored the tracking behaviors in sensitive websites. Vallina et al. [228] analyzed a set of 6,843 pornographic websites. They found that 72% of the websites include basic tracking and 58% of the top 100 porn websites contain cookie syncing. Matic et al. [215] built a classifier that identifies sensitive URLs. They found that 40% of the cookies used on 20K detected health related websites are persistent third-party cookies and 5% were set by trackers known from the Disconnect [86] and Ghostery [120] filter lists. Sanchez et al. [198] performed a manual analysis of 2000 websites. They found that only 4% of websites offer an easy way to reject in the cookie notice. They also looked at websites by category and found that more than 50% of health websites do not have a banner while still performing tracking, and 40% even create more cookies upon rejection.

4.2 *Policies*

Cookie or privacy policies consist in the typical way for website owners to be transparent about the data they process their users personal data. Under Article 13 and 14 of the GDPR, a website must have a cookie policy that is easily accessible for end users, and that contains information regarding *"what information you collect, what you do with their information, how you protect their information, if you disclose any information to third parties, how you store their information, how users may access, migrate, request rectification, restriction or deletion of information"* [44] [45].
There is substantial prior work in the area of analysis of privacy policies. Reidenberg et al. [190] considered several categories of privacy policies. Following Brodie et al. [60], The Usable Privacy Policy project [237] combines technologies, such as crowd sourcing to develop browser plug-in technologies to automatically interpret policies for users. Ammar et al. [36] performed a pilot study, followed by a website privacy policy corpus [238]. This corpus has been later used by the Polisis tool that automatically extracts information flows described in privacy policies [127]. Morel and Pardo [167] made an extensive overview of privacy policies and tools used to analyse these at scale. Degling et al. [84] analyzed the availability of privacy policies on the top 500 websites before and after GDPR came in force. Libert [148] analyzed over 200,000 websites' privacy policies.

4.3 *Cookie purposes*

Requirements to define purposes. Article 5(1)(b) of the GDPR and the 29WP [40] elaborates on the *"Purpose Limitation"* principle. This principle mandates personal data to be collected for specified, explicit and legitimate purposes only *purpose specification principle*. This *purpose specification* principle focuses on the initial purpose of data collection. It identifies three criteria for describing a purpose: *explicitness, specificity, and legitimacy*.

Data Protection Authorities (DPAs) advocate that websites should – as a best practice – declare the purpose of *all* cookies in the website. In effect, the UK, Greek, Finnish and Belgian DPAs [224, 109, 125, 53] endorse as a good practice disclosure of clear information about the purposes of cookies, including strictly necessary ones.
Basin et al. [52] analysed the purpose specification principle and propose a methodology for auditing GDPR compliance. Koops [138] analysed the purpose specification principle and, in an effort of techno-regulation, applied it within technical frameworks. The author suggested that purposes need to be specified, using a list of predefined domain specific purpose types. Grafenstein [230, 231] discussed this principle and propounds for a standardization of data purposes.

5 Positioning compared to related works

Tracking detection: In Chapter 3, we present our work on detection of cookie based tracking behaviors. In this work, instead of relying on filter lists, we made a tracking detection based on analysis behavior and we used invisible pixels to detect different tracking techniques. Invisible pixels have been extensively studied since 2001 [171, 25, 155, 88, 195]. Invisible pixels, called "web bugs" in previous works, were primarily used to set and send third-party cookies attached to the request or response when the browser fetches such image. The significant number of studies on invisible pixels showed that it is a well known problem. Though, the goal of our study is different: *we aim to use invisible pixels that are still widely present on the Web to detect different tracking behaviors and collaborations*. Moreover, differently from the previous works that detected trackers by analysing behavior [146], we proposed a more *fine-grained classification of tracking behaviors* that includes not only previously known behaviors, but also new categories of cookie sharing and syncing. Furthermore, our extension Ernie (Chapter 4) is the first tool that visualises several types of cookie synchronization techniques, and additionally shows which cookies and identifiers trigger tracking. It also shows the origin of cookie syncing requests and thus allows a detailed live overview of the tracking on a given website.

Cookie syncing: The detection of cookie syncing tracking technique relies on (1) the detection of identifier cookies, and (2) the detection of the sharing of these identifiers. In Chapter 3, we describe the methodology we used to both detect identifier cookies

Paper	Analysis of sensitive websites	Analysis of consent banners	Detection of tracking techniques
Vallina et al. [228]	Lists and manual labelling of porn websites	✓	BT, Cookie syncing (FTCS & TTCS)
Matic et al. [215]	Content classifier	×	BT
Sanchez et al. [198]	Symantec RuleSpace DB	✓	BT
Our paper	User simulation for health websites	✓	TA, TTCS, FTCS, BT, BTIT, TF

Table 2.3: **Overview of related works on analysis of special categories of data.** The abbreviations of tracking techniques are described in Section 2.1.2.

and how they are shared. Previous studies [18, 174, 96, 50, 178] measured cookie syncing on websites and users. In our study, we show that domains are using more complex techniques to store and share identifier cookies. We based our technique for detecting identifier cookies on the work of Acar et al. [18], and Englehardt and Narayanan [96], and differently from their work where they only checked for the identifiers that are stored and shared in a clear text, in our work, we detect more cases of cookie synchronization because we detect encoded cookies and even encrypted ones in the case of doubleclick.net. Papadopoulos et al. [178] detected not only syncing done through clear text, but encrypted cookie syncing as well. Hence, they cover DS, PC and ES sharing techniques detected in our work (see Figure 3.2), but they miss the remaining techniques that represent 39.03% of the cookie sharing that we detect. Bashir et al. [50], used retargeted ads to detect cookie syncing. To detect these ads, the authors filtered out all images with dimensions lower than 50×50 pixels, then they studied the information flow leading to these images, which limit their study to chains resulting to a retargeted ad. In our work, we analyse all kind of requests.

Cookie respawning: Cookie respawning refer to the recreation of removed cookies using a backup storage. Previous works [210, 46, 213, 19, 193] studied cookie respawning with different browser storages such as IndexedDB, ETags and localStorage. Unlike previous works that studied the usage of browser storages to respawn cookies, our study analyzes the usage of browser fingerprinting to respawn cookies. To the best of our knowledge, we are the first to make a study of cookie respawning with browser fingerprinting (Chapter 5).

Analysis of special categories of data: Table 2.3 summarizes the positioning of work on the analysis of special categories of data compared to related works. Our work

analyzes health related websites collected by simulating real users search behaviour. While previous works [215, 198] only investigated the presence of identifying third-party cookies on health related websites, we detected complex cookie syncing techniques from [113].

Analysis of privacy policies: Previous works analyzed privacy policies [190, 60, 237, 36, 238, 127, 167, 84, 148]. However, to the best of our knowledge our work is the first to analyse purposes of cookies within cookie policies. Websites, as part of their accountability and transparency obligations defined by the GDPR, need to declare the purposes of cookies that they use in their websites. We complement previous works by i) analysing the legal and theoretical framework on purpose specification for cookie purposes and denouncing their current ill-defined formulation; ii) providing recommendations and observations to policy makers to mitigate the current state of the description of cookie purposes (Chapter 6).

Chapter 3

Detection of Unknown Third-Party Trackers with Invisible Pixels

Preamble

In this chapter, we propose a fine-grained behavioral classification of tracking based on the analysis of invisible pixels. We use this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of $4,216,454$ third-party requests. We then assess the efficiency of the most popular filter lists and browser extensions in the detection of these tracking techniques. This chapter is a replication of the paper titled "Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels" which was published in the Privacy Enhancing Technologies Symposium (PoPETS 2020) [113].

1 Introduction

The Web has become an essential part of our lives: billions are using Web applications on a daily basis and while doing so, are placing *digital traces* on millions of websites. Such traces allow advertising companies, as well as data brokers to continuously profit from collecting a vast amount of data associated to the users. Recent works have shown that advertising networks and data brokers use a wide range of techniques to track users across the Web [210, 193, 46, 174, 96, 92, 19, 172, 146, 187], from standard stateful cookie-based tracking [193, 97], to stateless fingerprinting [172, 64, 96, 19].

In the last decade, numerous studies measured prevalence of third-party trackers on the Web [193, 19, 172, 96, 146, 50, 51, 150, 149, 243]. Web Tracking is often considered in the context of targeted behavioral advertising, but it's not limited to ads. Third-party tracking has become deeply integrated into the Web contents that owners include in their websites.

But what makes a tracker? How to recognize that a third-party request is performing tracking? To detect trackers, the research community applied a variety of methodologies. The most known Web tracking technique is based on *cookies*, but only some cookies contain unique identifiers and hence are capable of tracking the users. Some

studies detect trackers by analysing cookie storage, and third-party requests and responses that set or send cookies [193, 146], while other works measured the mere presence of third-party cookies [150, 149]. To measure *cookie syncing*, researchers applied various heuristics to filter cookies with unique identifiers [97, 18, 96]. However, this approach has never been applied to detect tracking at large scale. Overall, previous works provide different methods to identify third-party requests that are responsible for tracking [193, 243]. Detection of identifier cookies and analysing behaviors of third-party domains is a complex task. Therefore, most of the state-of-the-art works that aim at measuring trackers at large scale rely on *filter lists*. In particular, EasyList [90] and EasyPrivacy [91] (EL&EP) and Disconnect [87] lists became the *de facto* approach to detect third-party tracking requests in privacy and measurement communities [96, 50, 141, 187, 132, 95, 51, 49, 135]*. EasyList and EasyPrivacy are the most popular publicly maintained blacklist of known advertising and tracking requests, used by the popular blocking extensions AdBlock Plus [23] and uBlockOrigin [223]. Disconnect is another very popular list for detecting domains known for tracking, used in Disconnect browser extension [86] and in tracking protection of Firefox browser [110].

Nevertheless, filter lists detect only known tracking and ad-related requests. Therefore, a tracker can avoid this detection by using a different subdomain for tracking, or wholly register a new domain if the filter list block the entire domain. Even though, the second option is quite challenging because in such case, all the associated publishers would need to update their pages. Third parties can also incorporate tracking behavior into functional website content, which is never blocked by filter lists because blocking functional content would harm user experience. Therefore, it is interesting to evaluate how effective are filter lists at detecting trackers, how many trackers are missed by the research community in their studies, and whether filter lists should still be used as the *default tools* to detect trackers at scale.

Our contributions: To evaluate the effectiveness of filter lists, we propose a new, fine-grained behavior-based tracking detection. Our results are based on a stateful dataset of 8K domains with a total of 800K pages generating 4M third-party requests. We make the following contributions:

1- We analyse all the requests and responses that lead to invisible pixels (by "invisible pixels" we mean 1×1 *pixel images or images without content).* Pixels are routinely used by trackers to send information or third-party cookies back to their servers: the simplest way to do it is to create a URL containing useful information, and to dynamically add an image tag into a webpage. This makes invisible pixels *the perfect suspects for tracking* and propose a new classification of tracking behaviors. Our results show that pixels are still widely deployed: they are present on more than 94% of domains and constitute 35.66% of all third-party images. We found out that pixels are responsible only for 23.34% of tracking requests, and the most popular tracking content are scripts:

*We summarize the usage of filter lists in security, privacy and web measurement community in Table 2.2 in Section 2.

a mere loading of scripts is responsible for 34.36% of tracking requests.

2- We uncover hidden collaborations between third parties. We applied our classification on more than 4M third-party requests collected in our crawl. We have detected new categories of tracking and collaborations between domains. We show that domains sync first party cookies through a *first to third party cookie syncing*. This tracking appears on 67.96% of websites.

3- We show that filter lists miss a significant number of cookie-based tracking. Our evaluation of the effectiveness of EasyList&EasyPrivacy and Disconnect lists shows that they respectively miss 25.22% and 30.34% of the trackers that we detect. Moreover, we find that if we combine all three lists, 379,245 requests originating from 8,744 domains still track users on 68.70% of websites.

4- We show that privacy browser extensions miss a significant number of cookie-based tracking. By evaluating the popular privacy protection extensions: Adblock, Ghostery, Disconnect, and Privacy Badger, we show that Ghostery is the most efficient among them and that all extensions fail to block at least 24% of tracking requests.

2 Methodology

To track users, domains deploy different mechanisms that have different impacts on the user's privacy. While some domains are only interested in tracking the user within the same website, others are recreating her browsing history by tracking her across sites. In our study, by "Web tracking" we refer to both within-site and cross-site tracking.
To detect Web tracking, we first collect data from Alexa top 10,000 domains, then by analyzing the invisible pixels we define a new classification of Web tracking behaviors that we apply to the full dataset. In this section, we explain the data collection process and the criteria we used to detect identifier cookies and cookie sharing.

2.1 *Data collection*

Two stateful crawls: We performed passive Web measurements using the OpenWPM platform [96]. It uses the Firefox browser, and provides browser automation by converting high-level commands into automated browser actions. We launched *two stateful crawls on two different machines with different IP addresses.* For each crawl, we used one browser instance and saved the state of the browser between websites. In fact, measurement of Web tracking techniques such as cookie syncing is based on re-using cookies stored in the browser, and hence it is captured more precisely in a stateful crawl.
Full dataset: We performed a stateful crawl of Alexa top $10,000$ domains in February 2019 in France [32] from two different machines. Due to the dynamic behavior of the websites, the content of a same page might differs every time this page is visited. To reduce the impact of this dynamic behavior and reduce the difference between the two crawls, we launched the two crawls at the same time. For each domain, we visited the

home page and the first 10 links from the same domain. The timeout for loading a homepage is set up to 90s, and the timeout for loading a link on the homepage is set up to 60s. Out of $10,000$ Alexa top domains, we successfully crawled $8,744$ domains with a total of $84,658$ pages.

For every page we crawl, we store the HTTP request (URL, method, header, date, and time), the HTTP response (URL, method, status code, header, date, and time), and the cookies (both set/sent and a copy of the browser cookie storage) to be able to capture the communication between the client and the server. We also store the body of the HTTP response if it's an image with a *content-length* less than 100 KB. We made this choice to save storage space. Moreover, in addition to HTTP requests, responses and cookies, we were only interested in the storage of invisible pixels. In our first dataset, named *full dataset*, we capture all HTTP requests, responses, and cookies.

Prevalence of invisible pixels: As a result of our crawl of $84,658$ pages, we have collected $2,297,716$ images detected using the field *content-type* in the HTTP header. We only stored images with a *content-length* less than 100 KB. These images represent 89.83% of the total number of delivered images. Even though we didn't store all the images, we were able to get the total number of delivered images using the content-type HTTP header extracted from the stored HTTP responses.

Figure 3.1 shows the distribution of the number of pixels in all collected images. We notice that invisible pixels (1×1 pixels and images with no content) represent 35.66% of the total number of collected images.

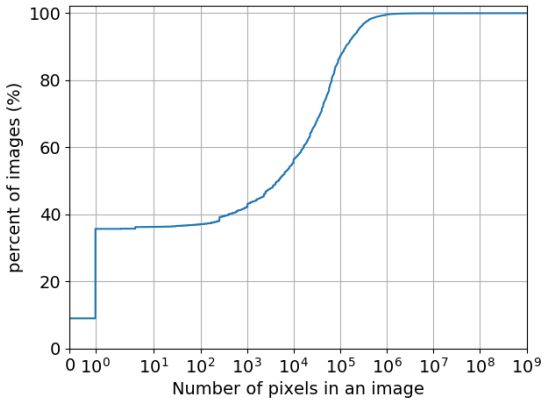


Figure 3.1: **Cumulative function of the number of pixels in images with a *content-length* less than** 100 **KB**. 35.66% *of the images are invisible pixels,* 9.00% *have no content (they are shown as zero-pixel images), and* 26.66% *are of size* 1×1 *pixel.*

We found that out of $8,744$ successfully crawled domains, $8,264$ (94.51%) domains contain at least one page with one invisible pixel. By analyzing webpages independently, we found that 92.85% out of $84,658$ visited pages include at least one invisible

pixel.
Invisible pixels subdataset: The invisible pixels do not add any content to the Web pages. However, they are widely used in the Web. They generally allow the third party to send some information using the requests sent to retrieve the images. Moreover, the user is unaware of their existence. Hence, every invisible pixel represents a threat to the user privacy. We consider the set of requests and responses used to serve the invisible pixels as a ground-truth dataset that we call *invisible pixels dataset*. The study of this *invisible pixels dataset* allow us to excavate the tracking behaviors of third party domains in the web.

2.2 *Detecting identifier cookies*

Cookies are a classical way to track users in the Web. A key task to detect this kind of tracking is to be able to detect cookies used to store identifiers. We will refer to these cookies as *identifier cookie*s. In order to detect identifier cookies, we analyzed data extracted from the two simultaneous crawls performed from two different machines. We refer to the owner of the cookie as host, and we define a cookie instance as (host, key, value).
We compare cookies instances between the two crawls: A tracker associates different identifiers to different users in order to distinguish them. Hence, an identifier cookie should be unique per user (user specific). We analyzed the $8,744$ crawled websites where we have a total of $607,048$ cookies instances belonging to $179,580$ (host, key) pairs. If an identical cookie instance appears in the two crawls, that is, the host, key and value of both cookies are identical, we consider that the cookie is not used for tracking. We refer to such cookies as *safe cookies*. We extracted $108,252$ safe cookies from our dataset. They represent 17.83% of the total number of cookies instances.

Due to the dynamic behavior of websites, not all cookies appear in both crawls. We mark the cookies (host, name) that appear only in one crawl as unknown cookies. In total, we found $15,386$ unknown cookies (8.56%). We exclude these cookies from our study.
We don't consider the cookie lifetime: The lifetime of the cookie is used to detect identifier cookies in related works [18, 97, 96]. Only cookies that expire at least a month after being placed are considered as identifier cookies. In our study, we don't put any boundary on the cookie lifetime because domains can continuously update cookies with a short lifetime and do the mapping of these cookies on the server side which will allow a long term tracking.
Detection of cookies with identifier cookie as key: We found that some domains store the identifier cookie as part of the cookie key. To detect this behavior, we analyzed the cookies with the same host and value and different keys across the two crawls. We found $5,295$ (0.87%) cookies instances with identifier cookie as key. This behavior was performed by 966 different domains. Table 8.1 in Appendix presents the top 10 domains involved. The cookies with identifier cookie as key represent only 0.87% of the total

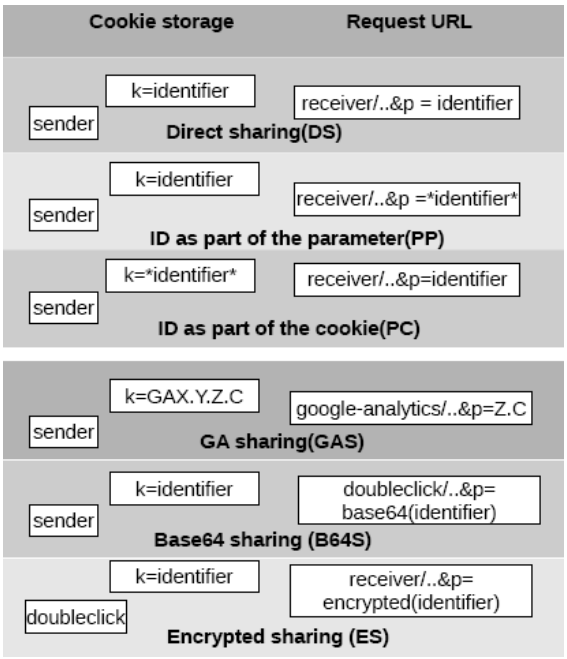


Figure 3.2: **Detecting identifier sharing.** "sender" is the domain that owns the cookie and triggers the request, "receiver" is the domain that receives this request, and "identifier" is the *identifier cookie* value. "*" represents any string.

number of cookies. Therefore, we will exclude them from our study.

2.3 *Detecting identifier sharing*

Third party trackers not only collect data about the users, but also exchange this data to build richer users profiles. Cookie syncing is a common technique used to exchange user identifiers stored in cookies. To detect such behaviors, we need to detect the *identifier cookies* shared between domains. A cookie set by one domain cannot be accessed by another domain because of the cookie access control and Same Origin Policy [197]. Therefore, trackers need to pass identifiers through the URL parameters.

Identifier sharing can be done in different ways: it can be sent in clear as a URL parameter value, or in a specific format, encoded or even encrypted. To detect identifiers, we take inspiration from [96, 18]. We split cookies and URL parameter values using as delimiters any character not in [a-zA-Z0-9,'-'',''_'.']. Figure 3.2 shows six different techniques we deployed to detect identifier sharing. The first three methods are generic: either the identifier is sent as the parameter value, as part of the parameter value or it's stored as part of the cookie value and sent as parameter value.

We noticed that the requests for invisible images, where we still didn't detect any cookie sharing, originate mostly from google-analytics.com and doubleclick.net. In-

deed, these domains are prevalent in serving invisible pixels across websites (see Figure 8.1 in Appendix). We therefore base the next techniques on these two use cases. First, we notice that first party cookies set by google-analytics.com have the format GAX.Y.Z.C, but the identifier sent to it are of the form Z.C. We therefore detect this particular type of cookies, that were not detected in previous works that rely on delimiters (**GA sharing**). Second, by base64 decoding the value of the parameter sent to doubleclick.net, we detect the encoded sharing (**Base64 sharing**). Finally, by relying on Doubleclick documentation [5] we infer that encrypted cookie was shared (**Encrypted sharing**). For more details see the Section 8 in the Appendix.

2.4 *Limitations*

We detected six different techniques used to share the identifier cookie. However, trackers may encrypt the cookie before sharing it. In this work, we only detected encrypted cookies when it's shared following a specific semantic set by doubleclick [5].

We do not inspect the payload of POST requests that could be used to share the identifier cookie. For example, it's known that google-analytics.com sends the identifier cookie as part of the URL parameters with GET requests or in the payload of the POST requests [37] – we do not detect such a case in this work.

To detect the sender of the request in case of inclusion, we use the referer field. Therefore, we may miss to interpret who is the effective initiator of the request, it can be either the first party or an included script.

3 Overview of tracking behaviors

In Section 2.1, we detected that invisible pixels are widely present on the Web and are perfect suspects for tracking. In this Section, we detect the different tracking behaviors by analyzing the *invisible pixels dataset*.

In total, we have $747,816$ third party requests leading to invisible images. By analyzing these requests, we detected 6 categories of tracking behaviors in $636,053$ (85.05%) requests that lead to invisible images.

We further group these categories into three main classes: explicit cross-site tracking (Section 4.1), cookie syncing (Section 4.2), and analytics (Section 4.3). In the following, we call BehaviorTrack our detection method of these behaviors.

After defining our classification using the *invisible pixels dataset*, we apply it on the *full dataset* where we have a total of 4,216,454 third-party requests collected from $84,658$ pages on the $8,744$ domains successfully crawled. By analyzing these requests, we detected 6 tracking behaviors in $2,724,020$ (64.60%) requests.

Figure 3.3 presents an overview of all classes (black boxes) and categories of tracking behaviors and their prevalence in the full dataset. Out of $8,744$ crawled domains, we identified at least one form of tracking in 91.92% domains. We further analyzed preva-

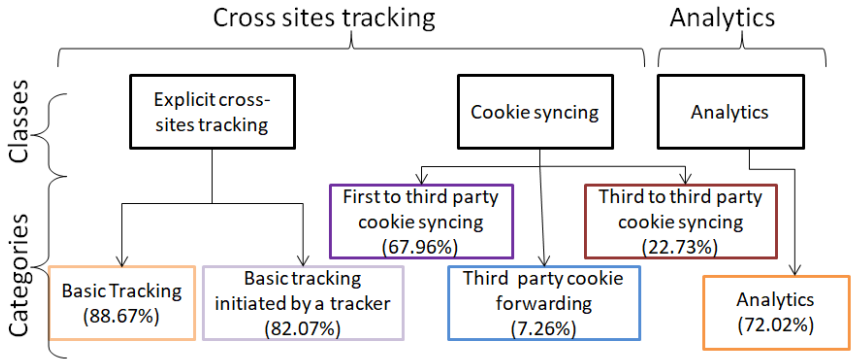


Figure 3.3: **Classification overview.** *(%) represents the percentage of domains out of* $8,744$ *where we detected the tracking behavior. A tracking behavior is performed in a domain if it's detected in at least one of its pages.*

lence of each tracking category that we report in Section 4. We found out that *first to third party cookie syncing (see Sec. 4.2.3) appears on 67.96% of the domains!*

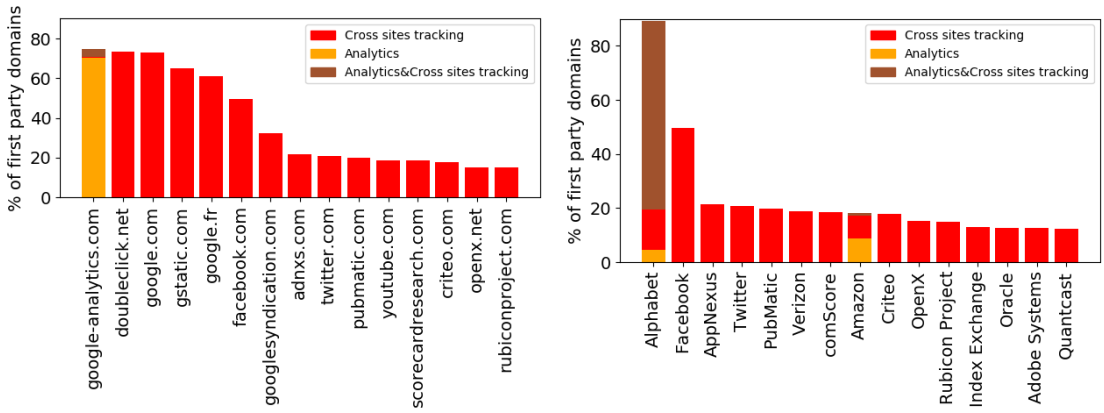


Figure 3.4: **Top 15 domains and companies involved in analytics, cross-site tracking, or both on the same first-party domain.**

In addition, we analyzed the most prevalent domains involved in either cross-site tracking, analytics, or both behaviors. Figure 3.4 demonstrates that a third party domain may have several behaviors. For example, we detect that google-analytics.com exhibits both cross-site tracking and analytics behavior. This variance of behaviors is due to the web site developer, as it's the case for cookie syncing and analytics behaviors. It can also be due to the domain's partners as it's the case for cookie forwarding. Google-analytics in that case is included by another third party, the developer is not

necessarily aware of this practice.

Content type	% requests
Script	34.36%
Invisible images	23.34 %
Text/html	20.01%
Big images	8.54 %
Application/json	4.32%

Table 3.1: **Top 5 types of content used in the $2,724,020$ third party tracking requests.**

We found that not all the tracking detected in the *full dataset* is based on invisible pixels. We extracted the type of the content served by the tracking requests using the HTTP header *Content-Type*. Table 3.1 presents the top 5 types of content used for tracking. Out of the $2,724,020$ requests involved in at least one tracking behavior in the full dataset, the top content delivered by tracking requests is scripts (34.36%), while the second most common content is invisible pixels (23.34%). We also detected other content used for tracking purposes such as visible images.

4 Classification of tracking

In this Section, we explain all the categories of tracking behaviors presented in Figure 3.3 that we have uncovered by studying the *invisible pixels* dataset. For each category, we start by explaining the tracking behavior, we then give its privacy impact on the user's privacy, and finally we present the results from the *full dataset*.

4.1 *Explicit cross-site tracking*

Explicit cross-site tracking class includes two categories: *basic tracking* and *basic tracking initiated by a tracker*. In both categories, we do not detect cookie syncing that we analyze separately in Section 4.2.

4.1.1 Basic tracking

Basic tracking is the most common tracking category as we see from Figure 3.3.
Tracking behavior: Basic tracking happens when a third party domain, say A.com, sets an identifier cookie in the user's browser. Upon a visit to a webpage with content from A.com, a request is sent to A.com with its cookie. Using this cookie, A.com identifies the user across all websites that include content from A.com.
Privacy impact: *Basic tracking* is the best known tracking technique that allows third parties to track the user across websites, hence to recreate her browsing history. How-

ever, third parties are able to track the user only on the websites where their content is present.

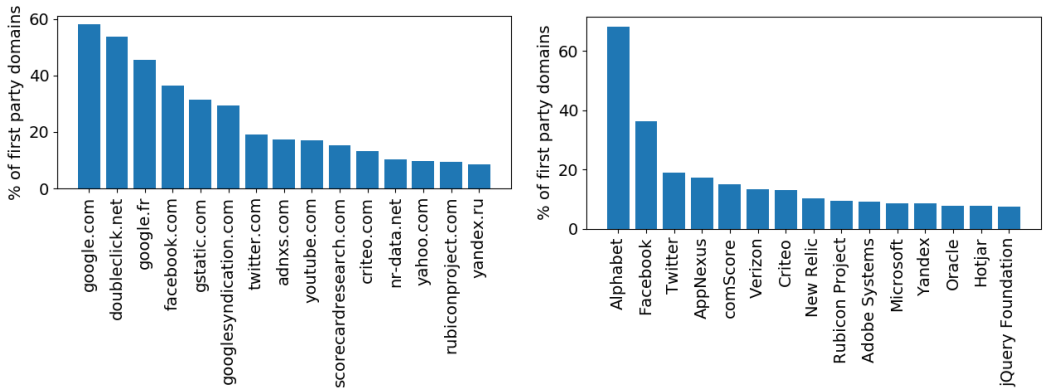


Figure 3.5: **Basic tracking:** Top 15 cross-site trackers and companies in charge of the trackers included in $8,744$ domains.

Results: We detected basic tracking in 88.67% of visited domains. In total, we found $5,421$ distinct third parties performing basic tracking. Figure 3.5 shows the top domains involved in basic tracking. We found that google.com alone is tracking the user on over $5,079$ (58.08%) domains. This percentage becomes more important if we consider the company instead of the domain (Figure 3.5). By considering companies instead of domains, we found that, by only using the *basic tracking* Alphabet (the owner of Google) is tracking users in 68.30% of Alexa top 8K websites.

4.1.2 Basic tracking initiated by a tracker

When the user visits a website that includes content from a third party, the third party can redirect the request to a second third party tracker or include it. The second tracker will associate his own identifier cookie to the user. In this case the second tracker is not directly embedded by the first party and yet it can track her.
Tracking behavior: *Basic tracking initiated by a tracker* happens when a basic tracker is included in a website by another basic tracker.
Privacy impact: By redirecting to each other, trackers trace the user activity on a larger number of websites. They gather the browsing history of the user on websites where at least one of them is included. The impact of these behaviors on the user's privacy could be similar to the impact of cookie syncing. In fact, by mutually including each other on websites, each tracker can recreate the combination of what both partners have collected using basic tracking. Consequently, through *basic tracking initiated by a tracker*, trackers get to know the website visited by the users, without being included in it as long as this website includes one of the tracker's partners. Hence,

through this tracking technique, the user's browsing history is shared instantly without syncing cookies.

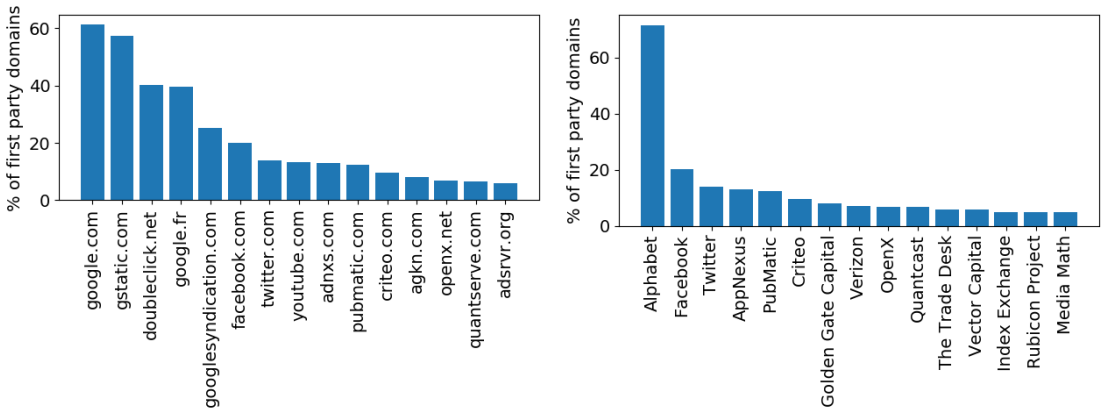


Figure 3.6: **Basic tracking initiated by a tracker:** Top 15 trackers and companies included in 8,744 domains.

Partners	**# requests**
pubmatic.com ↔ doubleclick.net	4,392
criteo.com ↔ doubleclick.net	2,258
googlesyndication.com ↔ adnxs.com	1,508
googlesyndication.com ↔ openx.net	1,344
adnxs.com ↔ doubleclick.net	1,199
rubiconproject.com ↔ googlesyndication.com	1,199
doubleclick.net ↔ yastatic.net	979
doubleclick.net ↔ demdex.net	790
adnxs.com ↔ amazon-adsystem.com	760
rfihub.com ↔ doubleclick.net	685

Table 3.2: **Basic tracking initiated by a tracker:** Top 10 pairs of partners from different companies that include each other. (↔) both ways inclusion.

Results: We detected Basic tracking initiated by a tracker in 82.07% of the domains. From Figure 3.6, we can notice that google.com is the top tracker included by other third parties. By only relying on its partners, without being directly included by the developer, google.com is included in over 5,374 (61.45%) of the Alexa top 8k domains and its owner company Alphabet is included in over 71.56% of the visited domains. Google.com is included by 295 different third party trackers in our dataset. In our results, we found that doubleclick.net and googlesyndication.com, both owned by

Google, are the top domains including each other (176,295 requests in our dataset). Table 3.2 presents the top 10 pairs of partners from different companies that are mutually including each other on websites. Note that in Table 3.2 we don't report mutual inclusion of domains that belong to the same company.

4.2 *Cookie syncing*

To create a more complete profile of the user, third party domains need to merge profiles they collected on different websites. One of the most known techniques to do so is cookie syncing. We separate the previously known technique of cookie syncing [18, 96] into two distinct categories, *third to third party cookie syncing* and *third party cookie forwarding*, because of their different privacy impact. We additionally detect a new type of cookie syncing that we call *first to third party cookies syncing*.

4.2.1 Third to third party cookie syncing

When two third parties have an identifier cookie in a user's browser and need to merge user profile, they use third to third party cookie syncing.

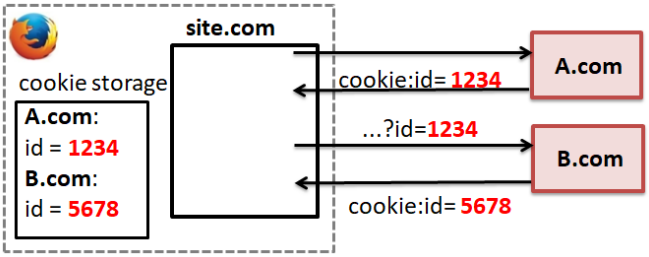


Figure 3.7: **Third to third party cookie syncing behavior.**

Tracking explanation: Figure 3.7 demonstrates cookie syncing[†]. The first party domain includes a content having as source the first third party A.com. A request is then sent to A.com to fetch the content. Instead of sending the content, A.com decides to redirect to B.com and in the redirection request sent to B.com, A.com includes the identifier it associated to the user. In our example, B.com will receive the request B.com?id=1234, where *1234* is the identifier associated by A.com to the user. Along with the request, B.com will receive its cookie *id = 5678*, which will allow B.com to link the two identifiers to the same user.
Privacy impact: *Third to third party cookie syncing* is one of the most harmful tracking techniques that impact the user's privacy. In fact, third party cookie syncing can

[†]Notice that in figures that explain the tracking behaviors, we show cookies only in the response, and never in a request. This actually represents both cases when cookies are sent in the request and also set in the response.

be seen as a set of trackers performing *basic tracking* and then exchanging the data they collected about the user. It's true that a cross sites tracker recreates part of the user's browsing history but this is only possible on the websites on which it was embedded. Using cookie syncing, a tracker does not only log the user's visit to the websites where it's included, but it can also log her visits to the websites where its partners are included. What makes this practice even more harmful is when a third party has several partners with whom it syncs cookies. One example of such behavior is rubiconproject.com, that syncs its identifier cookie with 7 partners: tapad.com, openx.net, imrworldwide.com, spotxchange.com, casalemedia.com, pubmatic.com and bidswitch.net.

Partners	**# requests**	**Sharing technique**
adnxs.com → criteo.com	1,962	→DS
doubleclick.net → facebook.com	789	→DS
casalemedia.com → adsrvr.org	778	→DS
mathtag.com↔adnxs.com	453	→DS
pubmatic.com → lijit.com	321	→DS
adobedtm.com → facebook.com	269	→DS
doubleclick.net ↔ criteo.com	250	→ DC, PCS; ← DS
mmstat.com → cnzz.com	233	→ DC
sharethis.com → agkn.com	233	→ DC
mathtag.com → lijit.com	109	→ DC

Table 3.3: **Third to third party cookie syncing:** Top 10 partners. The arrows represent the flow of the cookie synchronization, (→) one way matching or (↔) both ways matching. DS (Direct sharing), PCS (ID as part of the cookie), PPS (ID sent as part of the parameter) are different sharing techniques described in Figure 3.2.

Results: We detected third to third party cookie syncing in 22.73% websites. We present in Table 3.3 the top 10 partners that we detect as performing cookie syncing. In total, we have detected 1,263 unique partners performing cookie syncing. The syncing could be done in both ways, as it's the case for doubleclick.net and criteo.com, or in one way, as it's the case for adnxs.com and criteo.com. In case of two ways matching, we noticed that the two partners can perform different identifier sharing techniques. We see the complexity of the third to third party cookie syncing that involves a large variety of sharing techniques.

4.2.2 Third-party cookie forwarding

The purpose of the collaboration between third party domains in *third party cookie forwarding* is to instantly share the browsing history. Cookie forwarding has always

been called "syncing" while instead it simply enables a third party to reuse an identifier of a tracker, without actually syncing its own identifier.

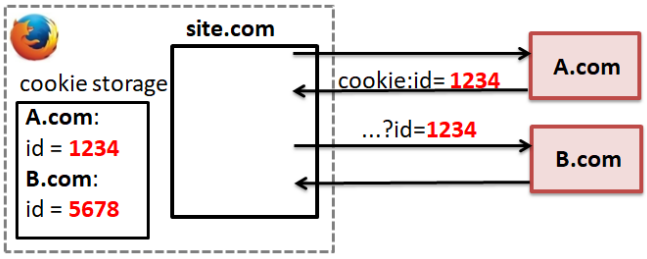


Figure 3.8: **Third party cookie forwarding behavior.**

Tracking explanation: The first party domain site.com includes A.com's content. To get the image, a request is sent to A.com along with its cookie. A.com then redirects the request to its partner (B.com) and sends the identifier cookie it associated to the user (*1234*) as part of the URL parameters (Figure 3.8).

Third party cookie forwarding differs from *Third to third party cookie syncing* depending on whether there is a cookie set by the receiver in the browser or not. This category is similar to third-party advertising networks in Roesner et al. and Lerner et al.'s works [193] [146], in the sense that we have a collaboration of third-party advertisers. However, in our study we check that the second tracker do not use its own cookie to identify the user. This means that this tracker (B.com) is relying on the first one (A.com) to track the user. In fact, B.com uses A.com's identifier to recreate her browsing history.
Privacy impact: *Third party cookie forwarding* allows trackers to instantly share the browsing history of the user. A.com in Figure 3.8 does not only associate an identifier cookie to the user, but it also redirect and shares this identifier cookie with it's partner. This practice allows both A.com and B.com to track the user across websites. From a user privacy point of view, *third party cookie forwarding* is not as harmful as cookie syncing, because the second tracker in this case does not contribute in the user's profile creation but passively receives the user's browsing history from the first tracker.
Results: We detected third party cookie forwarding in 7.26% of visited websites. To our surprise, the top domain receiving identifier cookie from third parties is google-analytics.com (Figure 8.2 in Appendix). Google-analytics.com is normally included by domains owners to get analytics of their websites, it's known as a *within domain tracker*. But in this case, google-analytics.com is used by the third party domains. The third party is forwarding its third party cookie to google-analytics.com on different websites, consequently google-analytics.com in this case is tracking the user across websites. This behavior was discovered by Roesner *et al.* [193]. They reported this behavior in only a few instances, but in our dataset we found 386 unique partners that forward cookies, among which 271 are forwarding cookies to google-analytics.com. In Table 3.4, we present the top 10 third parties forwarding cookies to google-analytics

service.

Third parties	# requests
adtrue.com	298
google.com	123
architonic.com	120
bidgear.com	80
akc.tv	76
insticator.com	73
coinad.com	64
performgroup.com	52
chaturbate.com	47
2mdnsys.com	40

Table 3.4: **Third party cookie forwarding.** Top 10 third parties forwarding cookies to google-analytics.

4.2.3 First to third party cookie syncing

In this category, we detect that first party cookie get synced with third party domain.

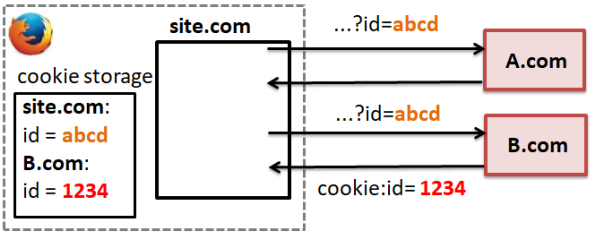


Figure 3.9: **First to third party cookie syncing behavior.**

Tracking explanation: Figure 3.9 demonstrates the cookie syncing of the first party cookie. The first party domain site.com includes a content from A.com?id=abcd, where A.com is a third party and *abcd* is the first party identifier cookie of the user set for site.com. A.com receives the first party cookie *abcd* in the URL parameters, and then redirects the request to B.com. As part of the request redirected to B.com, A.com includes the first party identifier cookie. B.com sets its own identifier cookie *1234* in the user's browser. Using these two identifiers (the first party's identifier *abcd* received in the URL parameters and its own identifier *1234* sent in the cookie), B.com can create a matching table that allows B.com to link both identifiers to the same user.

The first party cookie can also be shared directly by the first party service (imagine Figure 3.9 where A.com is absent). In that case, site.com includes content from B.com

and as part of the request sent to B.com, site.com sends the first party identifier cookie *1234*. B.com sets its own identifier cookie *1234* in the user's browser. B.com can now link the two identifiers to the same user.

Privacy impact: In our study, we differentiate the case when cookie shared is a first party cookie and when it is a third party cookie. We made this distinction because, the kind and the sensitivity of the data shared differs in the two cases. Using this tracking technique, first party websites get to sync cookies with third parties. Moreover, pure analytic services allow to sync in-site history with cross-site history.

Partners	# requests
First party cookie synced through an intermediate service	
google-analytics.com → doubleclick.net	8,297
Direct First to third party cookie syncing	
hibapress.com → criteo.com	460
alleng.org → yandex.ru	332
arstechnica.com → condenastdigital.com	243
thewindowsclub.com → doubleclick.net	228
digit.in → doubleclick.net	224
misionesonline.net → doubleclick.net	221
wired.com → condenastdigital.com	219
newyorker.com → condenastdigital.com	218
uol.com.br → tailtarget.com	198

Table 3.5: **First to third party cookie syncing**: Top 10 partners.

Results: We detected first to third party cookie syncing in 67.96% of visited domains. In Table 3.5, we present the top 10 partners syncing first party cookies. We differentiate the two cases: (1) first party cookie synced through an intermediate service (as shown in Figure 3.9) and (2) first party cookie synced directly from the first party domain. In total we found 17,415 different partners involved. The top partners are google-analytics.com and doubleclick.net. We found that google-analytics.com first receives the cookie as part of the URL parameters. Then, through a redirection process, google-analytics.com transfers the first party cookie to doubleclick.net that inserts or receives an identifier cookie in the user's browser. We found out that google-analytics.com is triggering such first party cookie syncing on 38.91% of visited websites.

4.3 *Analytics category*

Instead of measuring website audience themselves, websites today use third party analytics services. Such services provide reports of the website traffic by tracking the number of visits, the number of visited pages in the website, etc. The first party website includes content from the third party service on the pages it wishes to analyze the traffic.

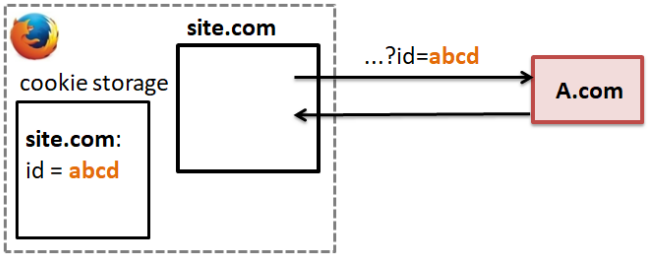


Figure 3.10: **Analytics behavior.**

Tracking explanation: Figure 3.10 shows the analytics category where the domain directly visited by the user (site.com) owns a cookie containing a unique identifier in the user's browser. Such cookie is called a first party identifier cookie. This cookie is used by the third party (A.com) to uniquely identify the visitors within site.com. The first party website makes a request to the third party to get the content and uses this request to share the first party identifier cookie.
Privacy impact: In analytic behavior, the third party domain is not able to track the user across websites because it does not set its own cookie in the user's browser. Consequently, for this third party, the same user will have different identifiers in different websites. However, using the first party identifier cookie shared by the first party, the third party can identify the user within the same website. From a user point of view, analytics behavior is not as harmful as the other tracking methods. The analytics service can not recreate the user's browsing history but it can only track her activity within the same domain, which could be really useful for the website developer.
Results: We detected analytics in 72.02% of the visited domains. We detect that google-analytics.com is the most common analytics service. It's used on 69.25% of the websites. The next most popular analytics is alexametrics.com, it's prevalent on 9.10% of the websites (see Figure 8.3 in the Appendix).

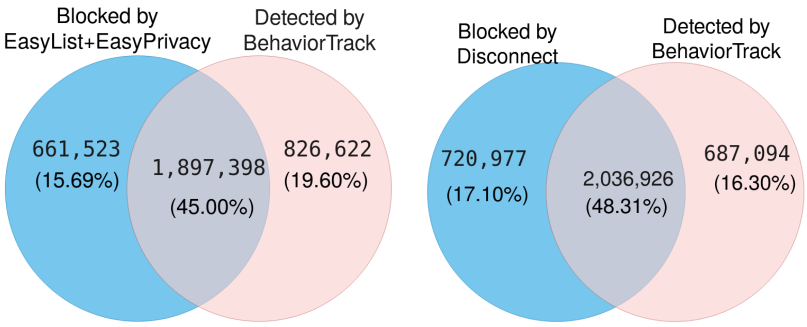
5 Are filter lists effective at detecting trackers?

Most of the state-of-the-art works that aim at measuring trackers at large scale rely on *filter lists*. In particular, EasyList [90], EasyPrivacy [91] and Disconnect [87] lists became the *de facto* approach to detect third-party tracking requests in the privacy and measurement communities [96, 50, 141, 187, 132, 95, 51, 49, 135]. Nevertheless, filter lists detect only known tracking and ad-related requests, therefore a tracker can easily avoid this detection by registering a new domain. Third parties can also incorporate tracking behavior into functional website content, which could not be blocked by filter lists because blocking functional content would harm user experience. Therefore, it is interesting to evaluate how effective are filter lists at detecting trackers, how many trackers are missed by the research community in their studies, and whether filter lists

should still be used as the *default tools* to detect trackers at scale.

In this Section, we analyze how effective are filter lists at detecting third-party trackers. Contrary to Merzdovnik et al.'s work [164], which measured blocking of third party requests without identifying whether such requests are tracking or not, we compare all the cross-site tracking and analytics behavior reported in Section 4 (that we unite under one detection method, that we call BehaviorTrack) with the third-party trackers detected by filter lists. EasyList and EasyPrivacy (EL&EP) and Disconnect filter lists in our comparison were extracted in April 2019. We use the Python library *adblockparser* [22], to determine if a request would have been blocked by EL&EP. For Disconnect we compare to the domain name of the requests (the Disconnect list contains full domain names, while EL&EP are lists of regular expressions that require parsing).

For the comparison, we used the *full dataset* of $4,216,454$ third party requests collected from $84,658$ pages of $8,744$ successfully crawled domains.



(a) EasyList and EasyPrivacy (b) Disconnect

Figure 3.11: Effectiveness of filter lists at detecting trackers on $4,216,454$ third party requests from $84,658$ pages.

Measuring tracking requests We apply filter lists on requests to detect which requests are blocked by the lists, as it has been done in previous works [96]. We then use the filter lists to classify follow-up third-party requests that would have been blocked by the lists. This technique has been extensively used in the previous works [95, 135, 141, 132]. We classify a request as blocked if it matches one of the conditions:

- the request directly matches the list.
- the request is a consequence of a redirection chain where an earlier request was blocked by the list.
- the request is loaded in a third-party content (an iframe) that was blocked by the list (we detect this case by analyzing the referrer header).

Figure 3.11 provides an overview of third party requests blocked by filter lists or detected as tracking requests according to BehaviorTrack. Out of all $4,216,454$ third party requests in the *full dataset*, $2,558,921$ (60.7%) requests were blocked by EL&EP, $2,757,903$ (65.4%) were blocked by Disconnect, and $2,724,020$ (64.6%) were detected

as performing tracking by BehaviorTrack.

Filter list(s)	# missed requests	% of 4.2M third-party requests	% of 2.7M tracking requests	# domains responsible for missed requests	# trackers follow up	# effective missed tracking requests
EL&EP	826,622	19.60%	25.22%	5,136	118,314	708,308
Disconnect	687,094	16.30%	30.34%	6,189	46,285	640,809

Table 3.6: **Overview of third-party requests missed by the filter lists and detected as tracking by** BehaviorTrack**.**

Requests blocked only by filter lists: Figure 3.11 shows that EL&EP block 661,523 (15.69% out of 4,216,454) requests that were not detected as performing tracking by BehaviorTrack. These requests originate from 2,121 unique third party domains. Disconnect blocks 720,977 (17.10%) requests not detected by BehaviorTrack. These requests originate from 1,754 distinct third party domains.

These requests are missed by BehaviorTrack because they do not contain any identifier cookie. Such requests may contain other non user-specific cookies (identical across two machines, see Sec. 2.2), however we assume that such cookies are not used for tracking. EL&EP and Disconnect block these requests most likely because they are known for providing analytics or advertising services, or because they perform other types of tracking through scripts such as fingerprinting, which is out of the scope of our study.

5.1 *Tracking missed by the filter lists*

Table 3.6 gives an overview of third-party requests missed by EL&EP and Disconnect filter lists and detected by BehaviorTrack as performing tracking. The number of third party domains involved in tracking detected only by BehaviorTrack (e.g., 6,189 for Disconnect) is significantly higher than those only detected by filter lists (e.g., 1,754 for Disconnect as reported earlier in this section). We define the term *trackers follow up* as the requests using identifying cookies set by previous requests blocked by the filter lists (note that our crawler is stateful). As a result, by simulating the blocking behavior of the filter lists, these cookies should be blocked and not included in the analysis of the following requests. Consequently, the follow up requests should not be categorized as tracking requests.
By further analyzing the requests only detected as tracking by BehaviorTrack and missed by EL&EP, we found that 118,314 requests (14.31% of the requests detected only by BehaviorTrack) are trackers follow up. Similarly, we found that 46,285 requests (6.73% of the requests detected only by BehaviorTrack) missed by Disconnect are trackers follow up. We exclude these requests from the following analysis and we

further analyze the remaining $708,308$ requests missed by EL&EP and the $640,809$ missed by Disconnect.

BehaviorTrack detects all kind of trackers including the less popular ones that are under the bar of detection of filter list. Because less popular trackers are less prevalent, they generate fewer requests and therefore remain unnoticed by filter lists. This is the reason why we detect a large number of domains responsible for tracking.

5.1.1 Tracking enabled by useful content

Content type	Missed by EL&EP	Missed by Disconnect
script	33.38%	35.27 %
big images	20.62%	21.73 %
text/html	13.77%	14.73 %
font	8.79%	0.09 %
invisible images	6.68%	12.21 %
stylesheet	6.17%	3.05 %
application/json	4.00%	4.83 %
others	6.59%	8.12%

Table 3.7: **Top content type detected by** BehaviorTrack **and not by filter lists.** The study is done on the $708,308$ requests missed by EL&EP and the $640,809$ missed by Disconnect.

We analyzed the type of content provided by the remaining tracking requests. Table 3.7 presents the top content types used for tracking and not blocked by the filter lists. We refer to images with dimensions larger than 50×50 pixels as Big images. These kinds of images, texts, fonts and even stylesheets are used for tracking. The use of these types of contents is essential for the proper functioning of the website. That makes the blocking of responsible requests by the filter lists impossible. In fact, the lists are explicitly allowing content from some of these trackers to avoid the breakage of the website, as it's the case for cse.google.com.

We categorized the top 30 third party services not blocked by the filter lists but detected by BehaviorTrack as performing tracking using Symantec's WebPulse Site Review [67]. Unlike in previous sections, where we analyzed the 2^{nd}-level TLD, such as google.com, here we report on full domain names, such as cse.google.com. That gives more information about the service provided. New domains such as consensu.org are not categorized properly so we manually added a new category called "Consent frameworks" to our categorization for such services. Table 3.8 represents the results of this categorization. Web Ads/Analytics represents 13.33% of the services missed by EL&EP and 23.33% of those missed by Disconnect. However, the remaining services are mainly categorized as content servers, search engines and other functional categories.

Service category	EL&EP	Disconnect
Content Servers	23.33 %	23.33 %
Social Networking	16.67 %	0.00%
Web Ads/Analytics	13.33 %	23.33 %
Search Engines/Portals	13.33 %	23.33 %
Technology/Internet	13.33 %	10.00 %
Consent frameworks	3.33 %	3.33 %
Travel	3.33 %	3.33 %
Non Viewable/Infrastructure	3.33 %	0.00%
Shopping	3.33 %	3.33 %
Business/Economy	3.33 %	6.67 %
Audio/Video Clips	3.33 %	0.00%
Suspicious	0.00%	3.33 %

Table 3.8: **Categories of the top 30 tracking services detected by** BehaviorTrack **and missed by the filter lists.**

They are tracking the user, but not blocked by the lists. This is most likely not to break the websites.

5.1.2 Why useful content is tracking the user

Tracking enabled by a first party cookie: A cookie set in the first party context can be considered as a third party cookie in a different context. For example, a site.com cookie is a first party cookie when the user is visiting site.com, but it becomes a third party when the user is visiting a different website that includes content from site.com. Whenever a request is sent to a domain, say site.com, the browser automatically attaches all the cookies that are labeled with site.com to this request.

For example, when a user visits google.com, a first party identifier cookie is set. Later on, when a user visits w3school.com, a request is sent to the service cse.google.com (Custom Search Engine by Google). Along with the request, Google's identifier cookie is sent to cse.google.com. The filter list cannot block such a request, and is incapable of removing the first party tracking cookies from it. In our example, filter lists do not block the requests sent to cse.google.com on 329 different websites. In fact, blocking cse.google.com breaks the functionality of the website. Consequently, an identifier cookie is sent to the cse.google.com, allowing it to track the user across websites.

By analyzing the requests missed by the lists, we found that this behavior explains a significant amount of missed requests: 44.61% requests ($316,008$ out of $708,308$) missed by EL&EP and 32.00% requests ($205,088$ out of $640,809$) missed by Disconnect contain cookies initially set in a first party context.

Tracking enabled by large scope cookies. A cookie set with a 2^{nd}-level TLD domain can be accessed by all its subdmains. For example, a third party sub.tracker.com

Tracking behavior	Prevalence
Basic tracking	83.90%
Basic tracking initiated by a tracker	13.50%
First to third party cookie syncing	1.42%
Analytics	1.00%
Third to third party cookie syncing	0.09%
Third party cookie forwarding	0.08%

Table 3.9: **Distribution of tracking behaviors in the $379,245$ requests missed by EL&EP and Disconnect.**

sets a cookie in the user browser with tracker.com as its domain. The browser sends this cookie to another subdomain of tracker.com whenever a request to that subdomain is made. As a result of this practice, the identifier cookie set by a tracking subdomain with 2^{nd}-level TLD domain is sent to all other subdomains, even the ones serving useful content.

Large scope cookies are extremely prevalent among requests missed by the filter lists. By analyzing the requests missed by the lists, we found that 77.08% out of 22,606 third-party cookies used in the requests missed by EL&EP and 75.41% out of 24,934 cookies used in requests missed by Disconnect were set with a 2^{nd}-level TLD domain (such as tracker.com).

5.2 *Panorama of missed trackers*

To study the effectiveness of EL&EP and Disconnect combined, we compare requests blocked by these filter lists with requests detected by BehaviorTrack as tracking according to the classification from Figure 3.3. These results are based on the dataset of $4,216,454$ third-party requests collected from $84,658$ pages of $8,744$ domains.

Overall, $379,245$ requests originating from 9,342 services (full third-party domains) detected by BehaviorTrack are not blocked by EL&EP and Disconnect. Yet, these requests are performing at least one type of tracking, they represent 9.00% of all $4,2$M third-party requests and appear in 68.70% of websites.

We have detected that the $379,245$ requests detected by BehaviorTrack perform at least one of the tracking behaviors presented in Figure 3.3. Table 3.9 shows the distribution of tracking behaviors detected by BehaviorTrack. We notice that the most privacy-violating behavior that includes setting, sending or syncing third-party cookies is represented by the basic tracking that is present in (83.90%) of missed requests.

Table 8.2 in Appendix presents the top 15 domains detected as trackers and missed by the filter lists. For each domain, we extract its category, owners and country of registration using the whois library [236] and manual checks. We also manually analyzed all the cookies associated to tracking: out of the 15 presented domains, 7 are tracking

the user using persistent first party cookies. The cookies of the search engine Baidu expires within 68 years, whereas the cookies associated to Qualtrics, an experience management company, expires in 100 years.

We found that content from code.jquery.com, s3.amazonaws.com, and cse.google.com are explicitly allowed by the filter lists on a list of predefined first-party websites to avoid the breakage of these websites. We identified static.quantcast.mgr.consensu.org by IAB Europe that rightfully should not be blocked because they provide useful functionality for GDPR compliance. We detect that the cookie values seemed to be unique identifiers, but are set without expiration date, which means such cookies will get deleted when the user closes her browser. Nevertheless, it is known that users rarely close browsers, and more importantly, it is unclear why a consent framework system sets identifier cookies even before the user clicks on the consent button (remember that we did not emit any user behavior, like clicking on buttons or links during our crawls).

We identified tag managers – these tools are designed to help Web developers to manage marketing and tracking tags on their websites and can't be blocked not to break the functionality of the website. We detected that two such managers, tags.tiqcdn.com by Tealium and assets.adobedtm.com by Adobe track users cross-sites, but have an explicit exception in EasyList.

6 Are browser extensions effective at blocking trackers?

In this Section, we analyze how effective are the popular privacy protection extensions in blocking the privacy leaks detected by BehaviorTrack. We study the following extensions: Adblock [21], Ghostery [120], Disconnect [86], and Privacy Badger [181]. The latest version of uBlock Origin 1.22.2 is not working correctly with OpenWPM under Firefox 52, which is the latest version of Firefox running on OpenWPM that supports both web extensions and stateful crawling. Hence we didn't include uBlock Origin in our study.

We performed simultaneous stateful Web measurements of the Alexa top 10K websites using OpenWPM in November 2019 from servers located in France. For each website, we visit the homepage and 2 randomly chosen links on the homepage from the same domain. Selection of links was made in advance.
We consider the following measurement scenarios:

1. Firefox with no extension.
2. Firefox with Adblock 3.33.0 (default settings).
3. Firefox with Ghostery 8.3.4 (activated blocking).
4. Firefox with Disconnect 5.19.3 (default settings).
5. Firefox with Privacy Badger 2019.7.11 (trained on the homepage and 2 random links from this homepage for the top 1,000 Alexa websites).
6. Firefox with all previous extensions combined.

Out of $30,000$ crawled pages, $25,485$ were successfully loaded by all the crawls. The

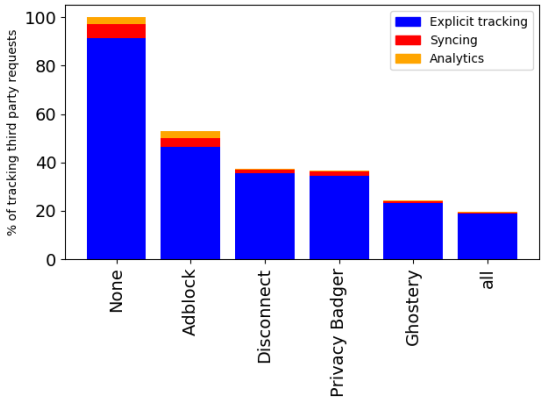


Figure 3.12: **Percentage of third party requests allowed by privacy protecting browser extensions out of 2,924,480 tracking requests.**

analysis in the following is done on this set of pages.

Figure 3.12 represents the effectiveness of the extensions in blocking the tracking requests detected by BehaviorTrack. Our results show that Ghostery is the most efficient among them. However, it still fails to block 24.38% of the tracking requests. All extensions miss trackers in the three classes, However, Disconnect and Privacy Badger have an efficient Analytics blocking mechanism: they are missing Analytics behavior on only 0.36% and 0.27% of the pages respectively. Most tracking requests missed by the extensions are performing Explicit tracking.

Conclusion: Similarly to Merzdovnik et al. [164], we show that tracker blockers (Disconnect, Ghostery and Badger) are more efficient than adblockers (Adblock) in blocking tracking behaviors. However, all studied extensions miss at least 24.38% of the tracking detected by BehaviorTrack. This shows that even though the extensions reduce the amount of tracking performed, they do not solve the problem of protecting users from tracking.

7 Discussion

Our results show that there are numerous problems in the cookie-based third party tracking. In this Section, we discuss these problems with respect to different actors.

Browser vendors. We observed that first party cookies can be exploited in a third party context to perform cross site tracking. In its Intelligent Tracking Prevention 2.0 introduced in 2018, Safari allowed cookies to be used in a third party context only in the first 24 hours of the cookie lifetime. Such time frame could be limited even further, however this approach requires rigorous testing with end users. Other browser vendors should follow Safari and prohibit the usage of cookies in a third party context.

Web standardization organizations. While third-party content provides useful features to the website, it is also capable of tracking users. We have shown that third party domains serving functional content such as Content Servers or Search Engines may track the user with identifier cookies. We have noticed that we detect such tracking because the domain behind such functional content does not set but only receives identifier cookies that are already present in the browser and were initially set with the 2^{nd}-level domain as host, which makes the cookie accessible by all subdomains. Even if the tracking is not intentional, and the domain is not using the identifier cookie it receives to create user's profile currently, this cookie leakage is still a privacy concern that could be exploited by the service anytime. We therefore believe that Web standardization bodies, such as the W3C, could propose to limit the scope of the cookies and not send it to all the subdomains.

Supervisory bodies. When a regulatory body, such as a Data Protection Authority in the EU, has to investigate and find the liable party for potential unlawful tracking happening on a website, it is a very complex task to identify who is the responsible for setting or sending the identifier cookie. In our work, we have identified *tracking initiators* – third party domains that only redirect or include other domains that perform tracking. Such tracking initiators, that we detected on 11.24% of websites, are partially liable for tracking. Another example are CDNs, we have observed that requests or responses for fetching a jQuery library from code.jquery.com contains identifier cookies. We found that it is the Cloudflare CDN that inserts a cookie named __cfduid into its traffic in order "to identify malicious visitors to their Customers' websites".

Conclusion. Our work raises numerous concerns in the area of tracking detection and privacy protection of Web users. We believe that our work can be used to improve existing tracking detection approaches, but nevertheless various actors need to revise their practices when it comes to the scope and usage of cookies, and third parties should exclude third party tracking from the delivery of functional website content.

8 Conclusion

Web tracking remains an important problem for the privacy of Web users. Even after the General Data Protection Regulation (GDPR) came in force in May 2018, third party companies continue tracking users with various sophisticated techniques based on cookies without their consent. According to our study, 91.92% of websites incorporate at least one type of cookie-based tracking.

In this chapter, we defined a new classification of Web tracking behaviors, thanks to a large scale study of invisible pixels collected from $84,658$ webpages. We then applied our classification to the full dataset which allowed us to uncover different relationships between domains. The redirection process and the different behaviors that a domain can adopt are an evidence of the complexity of these relationships. We show that even the most popular consumer protection lists and browser extensions fail to detect

these complex behaviors. Therefore, behavior-based tracking detection should be more widely adopted.

Chapter 4

Qualitative analysis of Web tracking and cookie syncing on health related websites with Ernie extension

Preamble

In this chapter, we evaluate the tracking prevalence of the 6 tracking techniques classified in Chapter 3 on health related websites. We report on the analysis on 176 websites of medical doctors and hospitals that users would visit when searching for doctors in France and Germany. Moreover, we analyse in depth 5 case study websites, one per each type of tracking and legal violation, to provide a comprehensive explanation of why tracking is happening on health related websites. This chapter is a replication of a paper titled "Qualitative analysis of Web tracking and cookie syncing on health related websites with Ernie extension" which is under submission to the Privacy Enhancing Technologies Symposium (PoPETS 2021).

1 Introduction

Health data is known to be one of the most sensitive types of data, and massive health data leaks is recognized to be of particularly high severity to the users' privacy, according to the French Data Protection Authority (CNIL) [71]. Searching for doctors online has become an increasingly common practice among Web users since telemedicine peaked in 2020 during the global Covid-19 pandemic [128]. However, the mere visit to a doctor's website can reveal a lot about its visitor: one can infer which diseases a visitor has or is interested in. Whenever health websites integrate third-party trackers, they *expose their potential patients' medical secrets to third parties*[*]. When providing services or monitoring user's behaviour in the EU, health related websites integrating third-

[*]According to the French Code of Public Health [72, Article L1110-4], medical secret covers "all information the person coming to the knowledge of the professional, of any member of the staff of these establishments, services or organizations and of any other person in relation, by virtue of his activities, with these establishments or organizations. It applies to all professionals working in the health care system".

party trackers are in breach with the General Data Protection Regulation (GDPR) [218] because processing of sensitive health data (derived from a visit to a website) is *generally forbidden*, unless allowed by several exceptions therein considered (Article 9(2) GDPR).

In the last decade, research has been focused on quantifying the prevalence of tracking based on cookies or lists of known tracking domains [96, 193, 50, 141, 187, 132, 95, 51, 49, 135], while several recent studies detected sophisticated forms of cookie syncing and ID sharing [178, 113, 177]. These studies were performed with customized large-scale crawlers and hard to replicate for non-experts. Moreover, quantitative studies measure the prevalence of various tracking techniques, but rarely explain *the reason why tracking is included.* This question is particularly important for health related websites that, differently from commercial websites, do not have an incentive to include targeted advertisement.

As a result, owners of health related websites, such as doctors and hospitals, have the urgent need to be able to detect tracking and advanced cookie synchronisation techniques on their website in order to determine whether the included third parties may be leaking their patients' health data. While some browser extensions visualise known tracking third parties or third party cookies [151, 85, 119, 94, 169], *no browser extension exists as of today that is able to visualise sophisticated forms of cookie synchronisation and sharing of user's identifiers* [178, 113, 177] across third parties. Therefore, owners of health-related websites are in a difficult position where it is close to impossible to determine tracking and complex cookie syncing included in their websites.

Moreover, since processing health data is forbidden by the GDPR, health website owners can only rely on one exception and implement a specific type of consent mechanism, called *explicit consent*, to make such processing lawful for all third parties included in the website. However, even for a basic consent to be legally valid, it has to comply with at least 22 different fine-grained requirements [201]. While general websites implement cookie banners to comply with the legal requirement of consent, recent works made evident that in practice websites often do not contain any cookie banners, or contain banners that do not respect the user's choice [157, 173, 198, 177]. Therefore, doctors and hospitals need to ensure that if their health websites contain tracking or any form of sophisticated cookie syncing, a *valid and explicit consent must be collected* before any of such activities are included.

In this paper, we perform the first in-depth qualitative study of third-party tracking, including complex cookie syncing and ID sharing techniques on health related websites, that are mostly owned by doctors and hospitals in two EU countries: France and Germany. We designed a new Firefox browser extension called Ernie that performs a state-of-the-art detection and visualizes sophisticated forms of tracking and ID sharing on a visited website, based on 6 different categories of third-party tracking from Fouad et al. [113].

Instead of relying on categorisation services [198, 215], we carefully selected 176 websites that Web users would find whenever searching for particular doctors in 2

major French and German cities, and manually visited them with Ernie extension. With Ernie, we monitored and recorded all 6 categories of tracking techniques before and after interacting with the cookie banner. Finally, we performed a detailed legal analysis together with a legal expert, co-author of this paper, to understand when each technique is potentially violating the GDPR.

Unlike previous works that measured tracking *quantitatively* on a large scale, we opted for *a deep technical and legal qualitative analysis* of one case study website for each type of potential violation. This analysis helped us (1) to uncover the mechanisms used by trackers that circumvent Firefox's Enhanced Tracking Protection [111] used in our experiments; and (2) to identify the reasons why tracking is included in otherwise unsolicited health websites. This approach demonstrates the usefullness of Ernie browser extension that is a first prototype of an extension that can be further used by non-expert users.

In summary, we make the following contributions:

(1) We propose the first browser extension Ernie[†] that visualizes complex cookie syncing and ID sharing tracking techniques. Ernie detects 6 categories of such tracking behaviors – Basic tracking, basic tracking initiated by another tracker, first to third party cookie syncing, third to third party cookie syncing, third party cookie forwarding, and third party analytics– following to the state-of-the-art methodology from Fouad et al. [113].

(2) We perform a legal and technical analysis of consent collection on 176 health related websites and identify practices potentially violating the GDPR and the ePrivacy directive. We found that 64% of the websites track users before any interaction with the banner. Moreover, 76% of these websites fail to comply with the legal requirements for a valid explicit consent: out of 176 studied websites, 46% do not display a cookie banner, and 75% thereof still contain tracking, thus violating the *explicit consent* legal requirement; 26% of the websites provide a cookie banner without a reject button, and 86% of these websites include tracking, hence violating the requirement to give users *the possibility to reject tracking*. Moreover, we show that the *user choice is not respected* on health related websites: 33 (19%) websites still contain tracking after cookie rejection.

(3) We analyse in depth 5 case study websites, one per each type of tracking and legal violation, to provide a comprehensive explanation of why tracking is happening on health related websites. Such in depth analysis helped us to conclude which techniques companies use to deploy tracking even in privacy-friendly browsers, such as Firefox ETP [111]. We found that in every 45 webpages wherein doctors include a Google map to help locating their of-

[†]The main goal of this extension is to provide an easy-to-use tool for the non-experts, such as doctors, the end users and research community, NGOs and legal experts to visualise complex tracking and the regulatory authorities to evaluate compliance.

fice, tracking occurs. While Google maps doesn't explicitly track users, tracking happens because of the NID cookie of google.com that is already present in the user's browser, and the HTTP standard [130] requires cookies to be automatically attached to every outgoing HTTP(S) request. Moreover, we found that such practice not only enables tracking with Google map content, but it also enables explicit tracking on 84 (47.73%) websites.

2 Methodology

2.1 *Ernie Extension*

The browser extension Ernie has been designed to detect the sophisticated cookie based tracking mechanisms described by Fouad et al. [113]. Ernie detects six categories of tracking (see Section 2.1.2).

Ernie collects all first-party and third-party HTTP(S) requests and responses during a page visit in a specific browser tab. A page visit can be triggered by entering a new URL in the navigation bar, clicking a URL, clicking the forward/backward browser buttons, reloading a page, or a redirection event. All requests send and responses received in that tab after the page visit and before the next one are considered part of the current page visit. As a result, Ernie provides a visualization that attributes to one of the six considered categories the HTTP(S) requests and responses, and the corresponding cookies.

2.1.1 Detection of ID cookies and ID sharing

Detection of ID cookies. Ernie extension implements a standard approach to detect cookies that are likely to identify a user [113, 96, 18, 97] by comparing cookies between two different users. Ernie simulates a different users by opening a hidden tab in a separate container for each page visit, which is only used by the extension. To create the container, the extension uses the Firefox API contextualIdentities [4]. *Contextual identities* are containers within a browser profile which have a separate cookie storage, localStorage, indexedDB, HTTP data cache, and image cache. In the following, we refer to the hidden tab as *shadow tabs*. If the cookies with the same key and domain have different values for the two users, Ernie concludes that the cookie is "user-specific", we call in the following such cookies *ID cookies*. The extension displays and analyses all (first-party and third-party) ID cookies set in the browser (via HTTP(S) requests, HTTP(S) responses, or Javascript).

If the value of a cookie is the same between the main and shadow tabs, then the cookie is categorized as *Safe* and is simply saved in a local database of the extension.
Detection of ID Sharing. To recognize if an ID cookie is shared via a URL parameter, the extension implements an ID sharing algorithm inspired by prior works [113, 18, 96]. All cookie values and URL parameters are split using as delimiters any character not in

[a-zA-Z0-9-_.]. Differently from [113] and in order to reduce the chance of coincidental matches, after splitting, we don't consider values that are shorter than 4 characters or that are only the value *true* or *false*. Fouad et al. [113] considered three additional ways to share an identifier in the parameters: Google Analytics (GA) sharing, base64 sharing, and encrypted sharing. The extension implements these detection methods as well, and extends GA sharing to all the domains listed on the privacy policy of Google [7], because we observed this type of sharing not only on google-analytics.com, but also on doubleclick.net and google.com owned by Google.

All the requests, responses, and corresponding cookies where ID sharing is detected, are stored in an an external database located on the same device.

2.1.2 Tracking detection

While detecting ID cookies and ID sharing, the Ernie extension can identify six types of tracking behaviours presented by Fouad et al. [113]. In order to identify a tracking behavior, the extension first needs to discover the initiator of the request, that is, the resource which caused the request. Ernie finds the initiator as follows.

1. If the request is caused by a 30x HTTP redirect, the initiator is the source of the redirection. Ernie labels the previous request that caused the redirection as the initiator.
2. If there is no redirection, but the HTTP-*Referer*-header of the request is set, Ernie labels as the initiator the previous request with the same URL as the one in the referer header.
3. For requests whose initiator cannot be found by either of the two previous steps, Ernie considers that the initiator is the first party.

Once the initiator of a request is identified, Ernie detects whether the request is responsible for one of the six tracking behaviours presented below.

Basic tracking (BT) is the most common tracking technique. To detect Basic tracking, the extension checks whether a third-party ID cookie is sent in a third-party request or set in a third-party response.

Basic tracking initiated by another tracker (BTIT) occurs when (1) a basic tracker initiates a third party request to another third party domain and (2) this other third party domain sets or sends an ID cookie. To detect the Basic tracking initiated by another tracker, the extension performs algorithm 1.

First to third party cookie syncing (FTCS) occurs when (1) a first party ID cookie is shared with a third party domain via the request URL (either in the key or value of the parameter, or the path of the URL - see Section 2.1.1 for details), and (2) the third party domain sets or sends its own ID cookie (See Figure 4.1). To detect the first to third party cookie syncing, the extension performs algorithm 2.

Third to third party cookie syncing (TTCS) occurs when an ID cookie of a third party is shared in the request URL of another third party request, either in the key or value of the parameter, or in the path of the URL (see the ID sharing section above).

Algorithm 1: Detection of Basic tracking initiated by another tracker in website *site*

```
Let C be the set of ID cookies Detected in site;
for Every request r in site do
    if r is sent to a third party: Tracker1 then
        Extract all cookies sent/received by Tracker1 and put them in set C1;
        Extract initiator of Tracker1: Tracker2;
        Extract cookies sent/received by Tracker2 and put them in set C2;
        if C1 ⋂ C ≠ ∅ and C2 ⋂ C ≠ ∅ then
            Tracker1 and Tracker2 are performing Basic tracking initiated by
             another tracker
        end
    else
        Continue to the next request;
    end
end
```

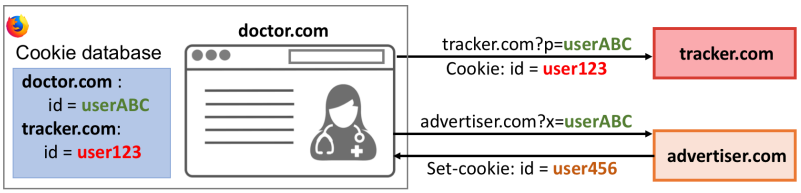


Figure 4.1: **Two examples of first to third-party cookie synchronisation:** either the third-party cookie is already present in the browser and hence automatically sent to a third party (case of tracker.com) or is actively set by a third-party domain (case of advertiser.com).

Algorithm 2: Detection of First to third party cookie syncing

```
Let C be the set of ID cookies Detected in site;
Let's note C_site the set of identifier cookies set by site.;
if C_site ≠ ∅ then
    for Every request r in site do
        if r is sent to a third party: Tracker1 then
            Extract the chain of initiators to Tracker1: T_i with i the length of the
              chain;
            while j <= i do
                if ∃ c in C_site shared with T_j and T_j received/set its own third
                  party ID cookie then
                    First party cookie is synchronized with T_j
                end
            end
        else
            Continue to the next request;
        end
    end
end
```

The third party request additionally sets its own ID cookie. We detect the sharing of the cookie through all the initiators chain.

Third party cookie forwarding (TF) occurs when an ID cookie of a third party is shared in the request URL of another third party request, either in the key or value of the parameter, or in the path of the URL. Unlike the case of third to third party cookie syncing, the third party request does not set its own ID cookie. We detect the sharing of the cookie through all the initiators chain.

Third party analytics (TA) occurs when an ID cookie of the first party is shared in the request URL of a third party request, either in the key or value of the parameter, or in the path of the URL. The third party request does not set its own ID cookie.

2.1.3 Limitations of the Ernie extension

The limitation of using a *shadow tab* to simulate a different user is that even if requests on the shadow tab are sent with different cookie values, they are still sent from the same IP address and the same device. If the website uses browser fingerprinting to recognise users, the requests from the shadow tab will likely be recognized as being from the same user as the original requests.

Using the Referer header has some limitations. If a third party makes a request to another third party, the Referer is often still set to the URL of the first party. Additionally, due to privacy concerns, the Referrer header is often not set at all by the websites

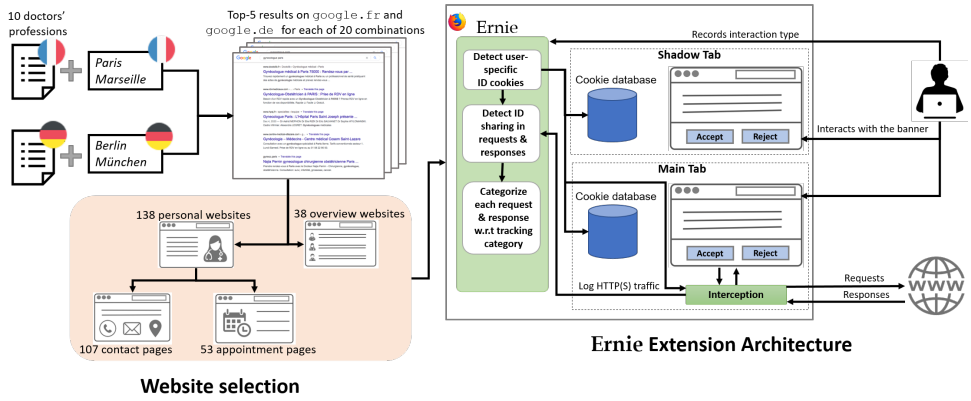


Figure 4.2: **High level overview of our experimental setup.** Website selection process as well as browser setup and website analysis is described in the remainder of this section. ERNIE extension architecture is presented in details in Section 2.1.

that serves the request. We therefore may miss some of the initiators and label them as first-party. As a result, our method may miss some of the tracking categories.

2.2 *Experimental setup*

Figure 6.2 presents an overview of our experimental setup. We first select health related websites (Section 2.2.1). Next, we setup the browser (Section 2.2) and collect data upon different interaction modes (Section 2.2.3).

2.2.1 Websites Selection

Simulating user search for a doctor in a city. Recent work have shown that classifiers need to be used to detect whether a given website belongs to a sensitive category, such as health, automatically [215]. We instead have opted for a method that closely simulates a user that is interested to find information about a given medical profession in a given city. We decided to simulate typical users in two EU countries: France and Germany.

Notably, the Germany and French Data Protection Authorities are allocated with the highest tech specialists in Europe to face GDPR infringements [59]. Authors are fluent in both French and German, so they are able to analyse the type of visited website, find contact information and analyse the content of cookie banners.

Table 4.1 shows the list of 10 doctor professions that we have built from a list of long term illnesses that are fully covered by the French health insurance due to their severity [1]. We then simulate users in two major cities in France (”Paris”, ”Marseille”) and Germany (”Berlin”, München”). For each of the studied doctor professions, we

English	French	German
gynaecologist	gynécologue	Frauenarzt
urologist	urologue	Urologe
infectologist	infectiologue	Infektiologe
oncologist	oncologue	Onkologe
cardiologist	cardiologue	Kardiologe
endocrinologist	endocrinologue	Endokrinologe
psychiatrist	psychiatre	Psychater
neurologist	neurologue	Neurologe
orthopaedist and traumatologist	chirurgien orthopédiste et traumatologue	Orthopäde und Traumatologe
pulmonologist	pneumologue	Pneumologe

Table 4.1: **Doctors professions in English, French and German.**

pretend to be a user that makes a Google search of one doctor in one city. Specifically, we make the following search ⟨ city ⟩ ⟨ doctor ⟩ on google.fr, using a French VPN for French cities and doctors' professions in French, and on google.de, using a German VPN for German cities and doctors' professions in German.

We then automatically extract the URL links of the top 5 results of each search with Pupeteer version 5.4.1 [10] running on Chromium 87.0.4272.0. As a result, we have a list of 200 URLs. This process is represented in the top-left corner of Figure 6.2.

Further analysis of collected websites. By manually analysing content of each of the 200 websites, we categorise each site as either as doctor's Personal website, or an Overview website, where a user can search for doctors in an area and potentially book appointments. An example for an overview website is doctolib.fr.
We found that many of Personal websites have

- *contact pages*, where potential patients can find phone number or other contact information. These pages are usually visible via "Contact"/"Where to find us" link of menu item.
- *appointment pages*, which include external content to book an appointment. These pages are found via searching for "Book an appointment" information on the website.

We have therefore added contact and appointment subpages to each visit to a Personal doctors websites. We imitate a user's behaviour and access these two subpages only by navigating within the visited Personal website.

During our manual analysis of 200 websites, we removed sites that are not related to our interest, such as news websites, PDF documents, job offerings and websites of doctors unions. After removal of these websites, we obtained 176 websites in our dataset.

Personal	138
with Contact	107
with Appointment	53
Overview	**38**
Total websites	**176**

Table 4.2: **Visited websites by type.** We successfully visited at least one subpage of 176 websites, among them 138 are Personal and 38 are Overview websites. Out of the 138 Personal websites, 107 include a contact subpage and 53 include an appointment subpage. The full list of 176 analysed websites can be found in support materials [9].

Table 4.2 presents the list of visited websites that are also shown in an orange box of Figure 6.2.

2.2.2 Browser setup

Browser settings. We use Firefox version 78.4.1 on Debian, which has *Enhanced Tracking Protection* activated by default, meaning that Firefox already blocks some cross-site and social media trackers based on the *Disconnect* list [6]. Additionally the *Web Page Language Settings* are set to the languages that authors are fluent with: English [en], German [de], French [fr] and English (United States) [en-us] to be able to analyse the visited websites and their policy.

Simulation of a base browsing profile. Instead of visiting websites with a clean browser, we simulate real users by install generic browsing profile to insures that their profile already has common cookies set when visiting health related websites.

To build the base browsing profile, we first collect a list of popular websites globally, in France and Germany, by combining the top-30 global, the top-30 websites in France, and top-30 websites in Germany from the Alexa top list [2]. To build the user profile, we visited the 90 collected websites on the 13th of November 2020. The full list of unique websites visited to build the profile can be found at [3].

We then visit each health related website collected in Section 2.2.1 with the browsing profile in place, but the follow up visiting is *stateless*, that is we don't keep the state between two websites. We visited health related websites with the browsing profile between the 13th and 17th of November, 2020.

Reachable websites. If a website times out 3 times with the standard browser settings, it is defined as unreachable for both the browser profile collection as well as the visits of health related websites. This occurred only once for microsoftonline.com.

2.2.3 Data Collection

With the browsing profile in place, we visit each of the collected websites with version 2.1 of our extension and log the tracking behaviour that the extension finds. For all websites, we reload the page once after the initial visit. We do reloading because after interacting with a cookie banner, some websites include additional content only on the next page load.

Interactions with the cookie banners. Previous works explored the interaction with the cookie banners [83, 198]. However, automated interaction with banners remains challenging: Matic *et al.* [215, Sec. 3.1] report that only 4.4% of websites contain a cookie banner we can automatically interact with via advanced tools like Consent-O-Matic [173, 74].

Given the relatively small number of websites included in our study, we decided to manually label the type of banners and interactions. The EU legislation requires consent before setting or sending tracking cookies. We therefore evaluate the types of banners and changes in the tracking behaviour based on the choice made by the user in the cookie banner. We interact with the banners in three ways, and also record each interaction type in our dataset.

- No Interaction: We don't interact with the cookie banner, but still visit the website and the contact or appointment subpages on Personal websites. This is not possible on every website, as cookie banners sometimes block the access to a website until the user has made a choice in the cookie banner.
- Accept All: We accept all cookie preferences the cookie banner suggests to us. Most of the time, that means clicking the "Accept All" button. This is only possible on websites that have a cookie banner.
- Reject All: We reject as many cookie categories and vendors as proposed in the banner interface. This is not possible on all websites that have cookie banners, because many banners only describe their use of cookies and other tracking technologies, but do not offer a possibility to reject them.

For each type of interaction, we visit as many page types as possible. This means that we have at least two page visits (initial visit and reload) and at most 12 page visits per website (three interaction types on a maximum of four page types).

Data collection from manual analysis. The Ernie extension saves all collected data to a local database on the same device with which we visit the health related websites. The database contains data related to page visits (described in Section 2.1) as well as data about manual analysis of the website content. We collect the following data upon each manual visit to a health related website:

- the URL and the country of the website (France or Germany depending on which search has lead to the website - see Section 2.2.1);
- the site type (Personal, Overview - see Table 4.2),
- whether the website contains a banner, and the type of consent banner the website employs (according to the classification of banners by Degeling et al. [83]),

- URLs of contact and appointment subpages for Personal websites.

2.2.4 Limitations of the experimental setup

The methods we used to select websites and interact with them have some limitations. First, our site selection may be biased because we rely on search results from google.de and google.fr. Secondly, to imitate French and German users, we used the VPN of a German and a French institution. These IP addresses might be recognized as not belonging to a private household, which might introduce bias in the content being served, as shown in [244].

In our experiments we used a Firefox browser with Enhanced Tracking Protection on, however users of other browsers without any tracking protection, such as Google Chrome, could experience much more tracking that Ernie extension is also able to detect.

3 Results

In this section we present the main findings regarding consent collection and potential illegal tracking occurring on health websites where we observed, at least, one type of tracking (see Section 2.1 for the full set of tracking categories Ernie detects). We say that *a website includes tracking* if we detect, at least, one type of tracking behavior on, at least, one page of a website. We refer to domains that participate in tracking as *tracking domains*.

Distinctly, we found that before any interaction with the website, tracking occurs on 65% of the 176 visited websites. Notice that we include Third party analytics category in these findings because it requires consent according to several Data Protection Authorities [133, 126, 61] and to the European Data Protection Board (EDPB) [105].

We present each finding firstly with a technical description, followed by a legal analysis and alleged violations triggered by tracking practices, alongside with a case study demonstrating such violations. The legal analysis is performed together with a legal expert co-author of this paper.

Legal requirements for online tracking. To comply with the GDPR and the ePrivacy Directive (ePD), websites must obtain *consent* from users located in the EU when monitoring users' behavior (Article 5(3) ePD). A common method to obtain consent is through the use of ubiquitous consent banners. For consent to be legally valid, it must be prior to any data collection, freely given, specific, informed, unambiguous, readable and accessible and finally, should be revocable (Articles 4(11) and 7 GDPR) [201].

Though consent is generally needed for tracking, some types of trackers are exempted of consent, and the only way to assess with certainty whether consent is required, is to analyse the *purpose* of each tracking technology on a given website [41]. To determine a purpose of each tracking cookie in our case study, we analyse privacy policies of third parties that set such cookie.

Data concerning health status. Health status of users are particularly sensitive by their nature, and under the GDPR [218, Article 9], merits specific protection, as their processing could create *significant risks to the fundamental rights and freedoms* of users (Recital 51 GDPR). *Data concerning health* means personal data related to the physical or mental health of a person, including the provision of health care services, which reveal information about her health status (Article 4(15), Recital 35 GDPR). When a user visits a health related website, this mere visit surely reveals information about the health condition of this visitor. It might be argued that this information is not 100% certain. However, when health websites integrate third-party trackers, they expose their potential patients' health condition to third parties. Considering the large number of websites and the large number of users a single third party can follow, the collected information will undoubtedly be very informative on the health condition of a very large number of users.

Legal requirements for online tracking on health websites. The processing of data concerning health is *forbidden* by the GDPR, unless allowed by several exceptions (Article 9 (2)(a-j)). For the purposes of online tracking in health related websites, only the *explicit consent* exception seems to be the applicable legal basis to process this special category of data [218, Article 9(2)(a)]. An *explicit consent* request should abide to the following requirements [30, 89, 133]: i) include double confirmation or verification from the user, ii) consist of a separated request from any other consents [43] (Recital 43 GDPR) iii) specify the nature of the special category of data through a specific legend. This additional effort is justified *to remove all possible doubt and potential lack of evidence in the future* [107].

Without explicit consent from users, tracking on health websites infringes the lawfulness principle (Article 9 (2)(a) GDPR), rendering any forthcoming processing *unlawful*, and consequently such websites will be subject to administrative fines up to 20,000,000 EUR, or in the case of an undertaking, up to 4 % of the total worldwide annual turnover of the preceding financial year, whichever is higher (Article 83 (5)(a) GDPR).

Methodology used for the legal analysis and case study. In the legal analysis of the following subsections we take a double approach. First, we analyse *straightforward violations* independently of whether tracking requires or is exempted of consent. Then, we additionally analyse further violations related to the presence of trackers that definitely require (not exempted of) consent. Pursuant to this, we analyse the *purpose* of each cookie to determine whether consent is needed. We name this later analysis as *violations depending on the purpose of the cookie*. We then report in a case study only cookies responsible for tracking or syncing that definitely require consent.

3.1 *No consent banner and tracking*

Technical description and prevalence. By manually analyzing the studied health related websites, we found that out of the 176 visited websites across all the website

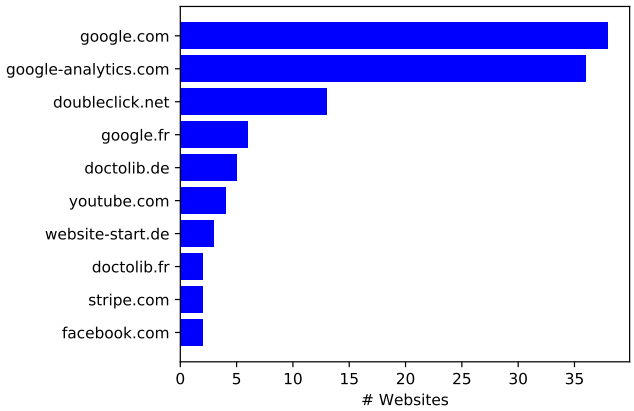


Figure 4.3: **Top 10 trackers on websites that do not include a consent banner.** *In total, we detected 81 websites that do not display a cookie banner and include tracking.*

categories (both Personal and Overview- see Section 2.2.1), 81 (46.02%) do not have any cookie banner, and 61 (75.31%) thereof include at least one of the studied tracking categories before interaction. Figure 4.3 presents the top 10 domains performing at least one of the studied tracking behaviors (see Section 2.1.2) on the 61 websites where no banner is displayed.

Legal analysis. *Straightforward violation*: The absence of *any* method set forth to collect the user's explicit consent renders any forthcoming tracking unlawful due to the lack of legal basis (Article 9(2)(a) GDPR), hence, allegedly violating the lawfulness principle. *Violations depending on the purpose of the cookie*: if the purpose of all the cookies used in a website does not require consent, the absence of a banner would not entail any legal violation. However, if the purpose of at least one cookie requires consent, then the violations would consist of: i) lawfulness principle, due to lack of any method to collect the user's consent; ii) prior consent, as tracking becomes unlawful if carried out before consent is requested (Article 6(1)(a) GDPR) [106].

Case study. We analyzed in depth the health related website logicrdv.fr [152]. Logicrdv.fr is an intermediate french website between doctors and patients that is specialized in the management of appointments. Through this website, the user can search for doctors of a given profession in a specific region, and set an appointment. The particular page we have visited provides a list of cardiology doctors near Marseille. When we first visited the website we found that no banner was included, and there were no means for the user to express her privacy preferences regarding tracking. The website moreover did not have any privacy policy.

With Ernie extension we detected tracking from 3 different third party domains: stripe.com, google.com and google-analytics.com. On a further analysis, we found

that google.com is responsible for 89 tracking requests, while stripe.com and google-analytics.com exhibit only 2 tracking requests each on this website. Moreover, all tracking by google.com is Basic tracking (see Section 2.1.2) caused by its own cookie named NID.

We found that the NID cookie is never set by google.com on the visited website, but it was always sent as part of the request. In fact, the NID tracking cookie was first set on the user's browser when we built the user profile and visited google.com website (see Section 2.2). Once stored in the user's browser, the cookie was automatically sent with every request to google.com's sub-domains as part of the management mechanism of the HTTP cookie standard [130]. As a result, when we visited logicrdv.fr – which includes Google maps to indicate the doctors location–, the browser automatically sent a request to google.com to fetch the content and automatically attached the NID cookie with every request. All tracking request sent to google.com from logicrdv.fr were used to fetch the google map. We never consented on the use of cookies neither on our visit to logicrdv.fr, nor on google.com. Google privacy policy states that "The NID cookie contains a unique ID we use to remember your preferences and other information, such as your preferred language, how many search results you prefer to have shown on a results page [..]", and at the very same time claims that "'NID' is used for these [advertising] purposes to show Google ads in Google services for signed-out users" [8]. Therefore, according to the purpose of this cookie, it requires consent since it is used, among other purposes, for advertising. As stipulated by regulatory guidance, such purpose is subject to the legal basis of consent [133, 89, 61, 41].

Findings. In our dataset, we detected 45 contact pages that include Google maps, and in all these websites tracking occurs because of the management mechanism of the HTTP cookie standard [130]. When the user first visits google.com, the NID is automatically set by google.com. Upon visits to websites containing Google maps, NID cookie is automatically attached with every request to a sub-domain of google.com to fetch the Google map. The impact of this practice is particularly severe for users' privacy because google.com is the default page visited upon installation of all major browsers: Google Chrome browser (used by 2.65 billion users in 2020 [69]), Safari browser (446 million users [196]), and Firefox browser (250 million users [112]).

3.2 *No possibility to refuse in a consent banner and tracking*

We found that 95 (53.98%), out of the 176 studied health related websites, include a consent banner. However, some of these banners are not designed to provide an unambiguous and freely given choice to the user, rendering unlawful such consent collection [124, 106]. Using the categorization of consent banner design proposed by Degeling et al. [83], we grouped the cookie banners detected in the visited health websites into 6 categories, which we depict in Table 4.3.

We further analyzed the 95 websites that include a cookie banner, and we found that on 49 (52%) websites this banner implements a reject button ("Binary", "Checkbox",

Banner	Accept	Reject	# of websites
No Option			6
Confirmation	✓		40
Binary	✓	✓	26
Slider	✓	(✓)	0
Checkbox	✓	(✓)	14
Vendor	✓	(✓)	7
Other			2
Total			95

Table 4.3: **Overview of banner types, and if they allow rejecting and accepting.** (✓) means that it is allowed for some categories in that banner, but not for others, e.g., one can reject cookies for some vendors in a "Vendor" banner, but not for all vendors.

"Vendor" and "Other" banner types). We found that rejection is actually possible on 44, out of 49 websites, and after rejection of cookies, 33 websites still include trackers. On 20 (60%) out of these 33 websites, the number of tracking domains before and after rejection of cookies remains the same. Therefore, even the presence of reject options is often ineffective.

Technical description and prevalence. Out of the 95 websites that include a cookie banner, 46 (48%) thereof display a cookie banner that either (i) is only informative ("No Option") and the user doesn't have any option; or (ii) only includes a confirmation button ("Confirmation") without giving the user any possibility to reject. Using Ernie, we detected at least one tracking behavior category on 40 (86.96%) out of 46 websites that display cookie banner without a possibility to reject.

Legal analysis. *Straightforward violation*: Considering the legal requirements for explicit consent, one should contend that the lack of any possibility to confirm a user rejection – as to make evident the user's choice regarding the processing of her sensitive data – would render such consent request unlawful (Article 9 (2)(a) GDPR). *Violations depending on the purpose of the cookie*: if the purpose of cookies does not require consent, the absence of a rejection button in a cookie banner does not seem to entail any legal violation. However, if the purpose of a cookie requires consent, then such practice allegedly is conflicting with the following consent requirements and data protection principles: i) requirements of "configurable banner" and "balanced choice" (Articles 4 (11), 7(3) GDPR) [16], which are compulsory for an unambiguous consent of a user; and ii) the principle of "data protection by design and by default" which demands the most privacy-friendly default settings to be used (Article 25 GDPR).

Case study. Ramsaygds.fr is a website of a private hospital in Marseille, France. The particular page we have visited [185] explains to patients why they should choose this hospital when they have health problems related to urology. When we first vis-

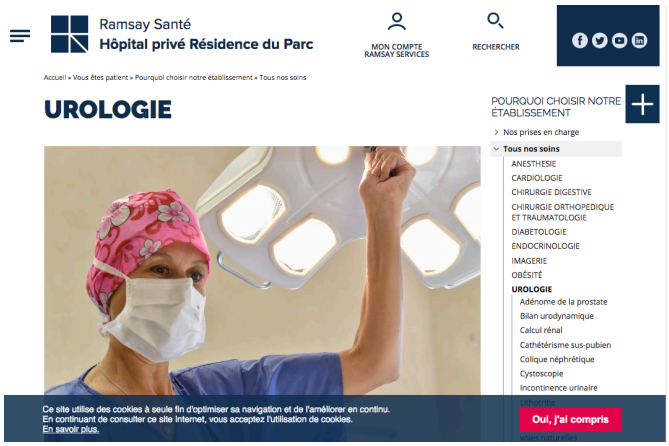


Figure 4.4: **Interface of the "Ramsay Santé Hôpital privé Résidence du Parc" private hospital website.** Captured on 18th February 2020 from https://hopital-prive-residence-du-parc-marseille.ramsaygds.fr/vous-etes-patient-pourquoi-choisir-notre-etablissement/urologie-22.

ited the website, we noticed that it presents a cookie banner to the user. However, the banner contains only one button "I understood", and does not include any reject button (see figure 4.4). Before interacting with the banner, Ernie detected tracking behaviors from 4 distinct domains. We found that ramsaygds.fr includes analytics performed by google-analytics.com and doubleclick.net and cross-site tracking behaviors by google.com and google.fr. We further analyzed the tracking behaviors on ramsaygds.fr after clicking on "I understood" button, and found that the tracking domains before interaction and after acceptance are identical. The website includes a privacy policy [186], however, they only state the use of Google analytics cookies for analytics purposes and do not mention the usage of other tracking forms detected on the website. In their policy they state that the user can manage and reject cookies in her browser, and block them using their browser storage according to the advice by the French Data Protection Authority (CNIL) on how to manage cookies [70]. The provided CNIL website is in fact a recommendation to users on how to protect their privacy in the web, and can not in any case replace the implementation of a reject button in the website cookie banner.

Findings: We found that cookie banners that do not provide a possibility to reject are only informative and do not affect the number of trackers. We compared the number of tracking domains before interaction and after accepting cookie on the 42 (23.86%) websites where there is no reject option and we successfully accepted cookies. We found that on 40 (95.24%) out of the 42 websites, the number of trackers remained the same before and after clicking the accept button. Moreover, 33 out of 44 websites that propose reject option still include trackers after rejection. Hence, cookie banners are

Senders	Receivers
Before interaction	
ramsaygds.fr	google.com
psychologies.com	facebook.com
rdvmedicaux.com	facebook.com
jameda.de	ioam.de
pagesjaunes.fr	facebook.com
After rejection	
jameda.de	ioam.de
pagesjaunes.fr	facebook.com
institutpaolicalmettes.fr	facebook.com
118000.fr	facebook.com
atos-kliniken.com	google.com

Table 4.4: **Cookie syncing.** Top 5 senders and receivers of cookie synchronization before interaction and after rejection. *All presented domains perform First to third party cookie syncing.*

not effective on these websites.

3.3 *Cookie Syncing before interaction or after rejection*

To create a more complete profile of the user, domains need to merge user's data they have collected on different websites. One of the most known techniques to do so is cookie syncing. In this section, we study all cookie syncing tracking categories (*First to third party cookie syncing, Third to third party cookie syncing, and Third party cookie forwarding*) performed on websites before any interaction with the banner or after rejection is selected on the banner.

Technical description and prevalence. Using Ernie, we detected cookie synchronization on 17 websites before interaction. This cookie synchronization is performed by 8 distinct third-party domains. Before interacting with the banner, we didn't detect any instance of Third to third party cookie syncing nor Third party cookie forwarding. The only synchronization activity we detected before interaction is First to third party cookie syncing, where google.com is the top domain that performs such syncing on 11 websites.

After rejection, to our surprise, we detected cookie synchronization on 8 websites performed by 3 distinct third party domains. We found that google.com is simultaneously performing First to third party cookie syncing and Third to third party cookie syncing on 4 and 1 websites respectively.

Legal analysis. *Straightforward violations*: Cookie syncing potentially breaches the following principles: *Lawfulness principle*: the absence of the user's explicit consent for cookie syncing, before interaction and after rejection, breaches this principle (pursuant

to Article 9 (2)(a) GDPR). *Fairness principle*: cookie syncing disregards the legitimate expectations of the data subject at the very time of data collection. Any (extensive) disclosure to third parties of sensitive data is out of any user reasonably expectations (Article 5(1)(a) GDPR). *Transparency principle*: in both scenarios users should be informed of the existence of cookie syncing operations and its purposes (Recital 60 GDPR), and should be made aware their personal data are shared with other third-parties. Moreover, users should be informed of the extent, risks and consequences of cookie syncing (Recital 39). In particular, considering the extent of data being sharing with third-parties, users should be informed of the existence of *profiling* and the rights and safeguards they are afforded with (Articles 13(2)(f), 22(1)(4) GDPR). The violation of these transparency obligations breaches the transparency principle and renders processing unlawful. *Minimization principle*: this practice contradicts expressly the minimization principle which requires personal data to be collected and processed limited to what is necessary, proportional and relevant to fulfil the data controller purpose (Article 5(1)(c) GDPR). *Violations depending on the purpose of the cookie*: if the purpose of cookies would not require consent, then no further breaches are accounted. However, if the purpose of a cookie requires consent, then such practice allegedly violates the following consent requirements: *Prior consent*: cookie syncing becomes unlawful if carried out before the request for consent due to the lack of a legal ground (Articles 4 (11), 6(1)(a) GDPR). *Informed consent*: users should be informed about third parties with whom the cookies are shared with – an obligation prescribed in the Court of Justice of the EU case law [16] and in the GDPR (Articles 4 (11), 13 (1)(e) GDPR). Users should also be informed about the purposes for which sensitive data will be collected for (Article 13 (1)(c) GDPR.

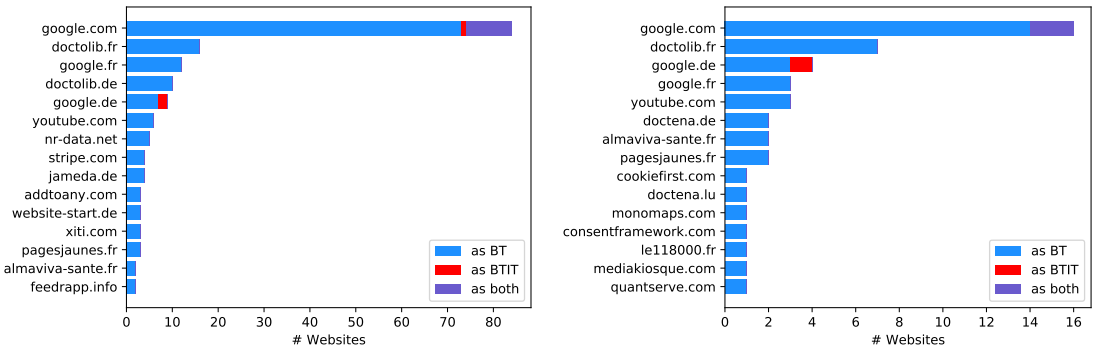
Case study. Lefigaro.fr is a phone book website that allows users to search for a doctor and make an online appointment by providing the user's phone number and address. The specific page that we visited [142] list Endocrinology doctors in Marseille. We noticed that no banner was displayed when we directly visited the subpage, however, the website does include a cookie banner in its home page. Due to this behavior users directly accessing the subpage through a Google search can not provide their consent. A cookie banner should be available through all website pages. Using Ernie, we detected first to third party cookie syncing between lefigaro.fr and two third parties before interaction: google.com and acpm.fr. We detected that lefigaro.fr shares the first party cookie that has as key measure with acpm.fr as part of the URL path. Acpm.fr then sets it's own cookie on the user's browser. Acpm.fr is a third party domain that provides to media websites a certification of the distribution, attendance, measuring of the audience by making it more visible to media agencies and advertisers [20]. Lefigaro.fr declares collaboration with acpm.fr in their policy [143] and they state that they are using acpm.fr cookies to measure audience in the website, but they do not provide information regarding cookies sharing.

Findings. First to third party cookie syncing is a common practice we detected before interaction on 17 (9.96%) websites with the Firefox ETP [111] protection activated. This

practice was shown before by Fouad et al. [113], but it didn't receive much attention. In this paper, we show that it still happens. We contacted the Firefox team and shared results for them to improve their tracking protection.

3.4 *Explicit Tracking before interaction or after rejection*

In this section, we analyse two categories of tracking together - Basic tracking and Basic tracking initiated by another tracker (see Section 2.1.2) – that we call *Explicit Tracking* in this section.



(a) No interaction

(b) After rejection

Figure 4.5: **Explicit tracking.** Receivers of explicit tracking before interaction and after rejection. *BT: Basic tracking, BTIT: Basic tracking initiated by another tracker*

Technical description and prevalence Using Ernie, we studied explicit tracking on the 176 websites, where at least one subpage is successfully visited. Before interacting with the banner, we found that explicit tracking occurs on 116 (66%) of the visited health related websites by 43 distinct domains. Figure 4.5 shows that google.com is the top domain performing explicit tracking, it is responsible of explicit tracking on 84 (47.73%) of the visited websites before any interaction. Moreover, after rejection, 29 (66%) out of 44 websites that provide possibility to reject, are explicitly tracking the user. Such tracking is performed by 24 distinct domains.

Legal analysis. *Straightforward violations*: We observe that explicit tracking before interaction and after rejection on health websites violates the following principles. *Lawfulness principle*: the absence of explicit consent for this tracking category in both scenarios, breaches the lawfulness principle (pursuant to Article 9 (2)(a) GDPR). *Fairness principle*: after rejecting tracking, users do not expect still to be tracked. Accordingly, such practice seems to infringe the fairness principle (Article 5(1)(a)). *Violation depending on the purpose of the cookie*: if the purpose of cookies does not require consent, then

no further breaches are accounted. However, if the purpose of a cookie requires consent, then such practice allegedly is in breach of the *prior* consent requirement (Articles 4 (11), 6(1)(a) GDPR).

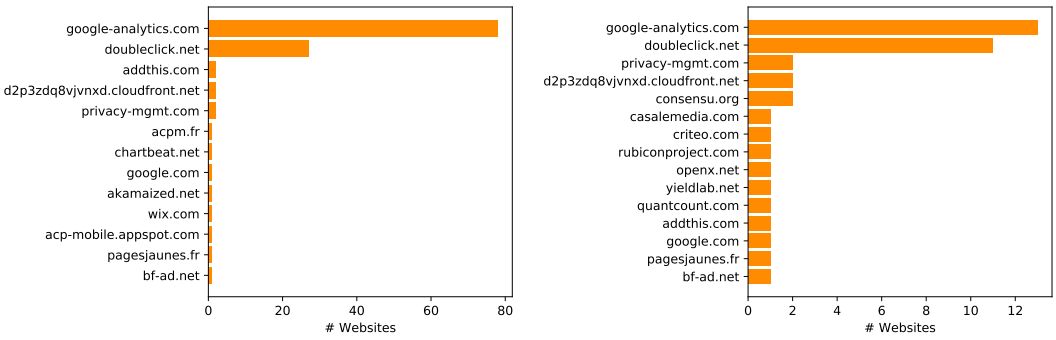
Case study. Ameli.fr is a major health website in France: it allows any French resident to access different health insurance services such as consulting reimbursements, downloading certificates, obtaining European card, etc. We analyzed a specific subpage of ameli.fr [35] that helps users search for doctors and medical institution using the doctor or institution name, profession or the required service. This ameli.fr website displays a banner, but no choice can be made by the user. The banner is only used to inform the user that if she continues browsing the website than she accepts the usage of cookies. Using Ernie, we detected Basic tracking before interacting with the website from the third-party domain xiti.com. Xiti.com define themselves as a web traffic measurement website [240]. We detected that xiti.com is performing Basic tracking on the Ameli.fr website using the following cookie: idrxvr, atidx, and atid. These cookies are classified as analytics cookies used to provide measurement on the website [147]. However, differently from standard first-party cookies used for analytics, these analysed cookies are third-party cookies, and therefore differently from analytics services, they can be used for cross-site tracking.

Finding. We found that all explicit tracking performed by google.com before interaction on the 84 (47.73%) visited websites were a result of the management mechanism of the HTTP cookie standard [130]. In fact, when we first visited google.com website upon profile creation (see Section 2.2), google.com set an ID cookie NID in the user browser. This cookie was then sent with every request to google.com in the 84 websites before interaction, thus following the same mechanism as described in Findings of Section 3.3.

3.5 *Third-party Analytics before interaction or after rejection*

As of today, website developers tend to use third party analytics services to measure audience in their websites. These analytics services provide report on the website traffic by measuring the number of repeated visits, the most popular pages, etc. Such practice allows tracking only within the same website. According to the ePrivacy Directive (Article 5(3)) websites owners are bound to request user consent before performing such tracking practices on their websites.

Technical description and prevalence. We analyzed the prevalence of the third-party analytics behavior in health related websites before interaction and after rejection of cookies. We found that analytics behavior is simultaneously performed on 81 websites before any interaction and 16 websites after rejection. google-analytics.com is the top domain responsible of third-party analytics on health related websites without user's consent (see figure 4.6). It is tracking users on 77 websites before interaction and on 13 after rejection. It is followed by doubleclick.net that performs analytic behavior on 26 websites before interaction and on 11 after rejection.



(a) No interaction

(b) After rejection

Figure 4.6: **Third-party analytics.** Receivers of analytics tracking before interaction and after rejection.

Legal analysis. *Straightforward violations*: We observe that third-party analytics before interaction or after rejection on health websites violates the lawfulness principle due to lack of explicit consent. *Violation depending on the purpose of the cookie*: Is it already determined that using third-party analytics requires consent from users. This stance is upheld by several Data Protection Authorities [133, 126, 61] that assert these technologies are not considered *strictly necessary* for a website to provide a functionality explicitly requested by the user, because the user can access all the functionalities provided by the website when such cookies are rejected. The French DPA [73] adds further that consent is required whenever tracers allow the overall monitoring of the navigation of the person using different applications or browsing different websites, or when data stemming from such tracers are combined with other processing operations or transmitted to third parties, these different operations not being necessary for the operation of the service.

Case study. kardiologie-praxiswestend-berlin.de is a joint medical office of several cardiologists. The website does not have a cookie banner. In their privacy policy they explain that their website uses google-analytics.com and googleadservices.com, and the data collected by google-analytics.com will not be linked to other data from Google.

Before interaction and after rejection, we detected analytics behavior on the studied website using the Ernie extension (see Figure 4.7). We found that google-analytics.com first receives the *__utma* first party cookie as part of the request, then google-analytics.com makes a redirection to doubleclick.net and shares the first party cookie *__utma* with it. According to google's policy [123], the *__utma* cookie is used to distinguish users. The two requests sent to google-analytics.com and doubleclick.net are categorized as analytics. Doubleclick.net then redirects to google.com, which again redirect to google.de. The first party cookie is shared with google.com and google.de, more-

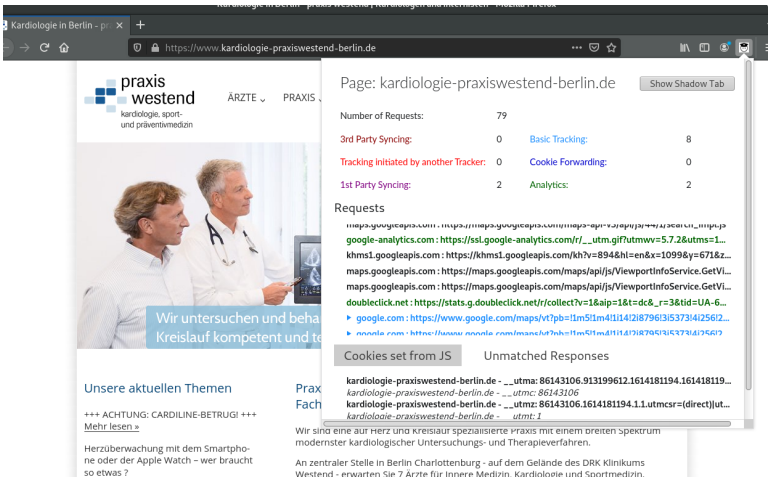


Figure 4.7: **Detection of third party analytics behavior on** kardiologie-praxiswestend-berlin.de **website using Ernie extension.**

over, the browser automatically attaches the NID cookie set in the browser in our base profile. These two requests are therefore first to third party cookie syncing-requests, effectively allowing the linking of the *__utma* first party cookie to the NID cookie.

Findings. Due to the redirection inclusion process, third party domains track users on websites where they were not initially included. Moreover, using this redirection, trackers share first party identifiers and link them with third party IDs. We found that on 25 websites out of the 26 websites where doubleclick.net is performing analytics, google-analytics.com is included as well, and both google-analytics.com and doubleclick.net receive the same first party identifier. We detected that all first party cookies _ga, _gid and ___utma shared with doubleclick.net on these 25 websites belong to google-analytics.com [122]. Therefore, we suspect that google-analytics.com is responsible of including and sharing the first party ID with doubleclick.net.

4 Conclusion

In this chapter we have gleaned robust evidence of tracking technologies deployed on health-related websites (before user consent interaction, and also after accepting and rejecting). Our open source browser extension Ernie can be used to collect further evidence and demonstrate cookie-based tracking technologies and sophisticated cookie syncing techniques employed on websites. We hope that Ernie extension can be beneficial to both policy-makers, to advance the enforcement of EU Privacy and Data Protection law, and to owners of health websites, such as doctors and hospitals that so far had no access to such visualisation tools. We have further contacted the website owners that we mention in our case studies and we are willing to help them changing

their practices towards improving the afforded protection of privacy and health data of Web users.

Chapter 5

Detection and measurement of cookie respawning with browser fingerprinting

Preamble

In this chapter, we make the first study on the detection and measurement of cookie respawning with browser and machine fingerprinting. We develop a detection methodology that allows us to detect cookies dependency on browser and machine features. We demonstrate how this technique can be used to track users across websites even when third-party cookies are deprecated, and together with a legal scholar, we show that such technique violates the GDPR and ePrivacy directives. This chapter is a replication of a paper titled "My Cookie is a phoenix: detection, measurement,and lawfulness of cookie respawning with browser fingerprinting" which is under submission to the IEEE Symposium on Security and Privacy (S&P 2022)

1 Introduction

In the last decades, the usage of the web on a daily basis has considerably increased, along with an increased sophistication of web browsers. In parallel, numerous companies built their business models on profiling and tracking web users. Therefore, browsers evolution does not only provide a better user experience, but also allows the emergence of new tracking techniques exploited by companies to collect users' data. There are two main categories of tracking techniques: stateful and stateless.

Stateful tracking is a standard technique that relies on browser storage such as cookies [193, 18, 46, 97]. Trackers store a unique identifier in the cookie and later use it to recognize the user and track her activity across, possibly, different websites. The simplest way to protect from such tracking is to delete the unique identifier by, e.g., cleaning the cookie storage.

However, trackers can recreate deleted cookie using a technique called *cookie respawning* to track users. For instance, a tracker can use multiple browser storages to store the identifiers, in addition to the cookie storage, such as the HTML5 localStorage [46]. Consequently, even if the user cleans the cookie storage, the tracker can still recreate the cookies using other storages [210, 46, 18, 193].

Stateless tracking allows to track a user without storing identifiers in her browser storage. Using *browser fingerprinting* [172, 64, 96, 19, 139, 136], trackers can identify a user through a combination of the user's browser and machine features such as the user agent or the machine IP address. Whereas it's very hard to protect against fingerprinting, this technique is not stable over time. Vastel et al. [229] showed that fingerprints change frequently. They show that out of 1,905 studied browser instances, 50% changed their fingerprints in less than 5 days, 80% in less than 10 days. This instability is caused either by automatic triggers such as software updates or by changes in the user's context such as travelling to a different timezone.

In summary, stateful tracking is a stable way to track a web user until she cleans cookies and other browser storages. Stateless tracking is not stable over time, but does not require any storage and can't be easily stopped by the user. So given that each technique is not perfect, *how can a tracker take advantage of the best of the two worlds?* The tracker can first use a browser fingerprint to create an identifier and store it in the user's browser. In this way, even if the user cleans all browser storages, the identifier can be re-created via a browser fingerprint. We refer to this tracking technique as *cookie respawning with browser fingerprinting*. This practice ensures the resumption of the tracking even after cleaning all browser storages.

Several studies measured the prevalence of stateful [18, 193, 97] or stateless [172, 64, 96, 19] tracking techniques separately. However, to the best of our knowledge, *we are the first to study how trackers profit from both stateful and stateless techniques* by combining them.

The focus of this chapter is to detect and measure the prevalence and the privacy implications of cookie respawning with browser fingerprinting. In this chapter, we make the following contributions.

1. **We designed a method to identify which features are used to respawn a cookie.** Our contribution lays in the design of a method to automatically identify the set of fingerprinting features used to generate a cookie, hence, to conclude what user's information is collected.
2. **We make the first study of cookie respawning with browser fingerprinting.** We show that the stateful and stateless tracking techniques that were studied separately are, in fact, actively used together by trackers. We found that $1,150$ (6.25%) of the Alexa top $30,000$ websites use cookie respawning with browser fingerprinting.
3. **We identify who is responsible of cookie respawning with browser fingerprinting.** We made a detailed study of the responsibility delegation of cookie respawning with browser fingerprinting. We show that multiple actors collaborate to access user features, set and own the cookies: we uncovered collaborations between 65 distinct domains that together respawn 115 different cookies.
4. **We show that cookie respawning with browser fingerprinting is highly deployed in less popular websites.** Cookie respawning with browser fingerprinting is also happening on websites from different categories including highly

sensitive ones such as adult websites.

5. **We show that cookie respawning with browser fingerprinting lacks legal interpretation and its use, in practice, violates the GDPR and the ePrivacy directive.** We are the first to assess the legal consequences of this practice together with a legal expert co-author. Despite the intrusiveness of this practice, it has been overlooked in the EU Data Protection Law and it is not researched in the legal scholarship, nor audited by supervisory authorities.

2 Methodology

When a user visits a web page with some content located on a tracker's server, the user's browser sends an HTTP(s) request to the server to fetch this content. This request contains several HTTP headers, such as user agent, and an IP address that tracker's server receives *passively*. We refer to such information as *passive features*. To collect additional information, the tracker can include in the visited web page a script that gets executed on the user's browser. The script retrieves multiple browser and machine information, such as the time zone, and sends them to a server of the remote tracker. We refer to such information as *active features*. In the following, we define a *browser fingerprint* as the set of active and passive features accessed by the tracker.

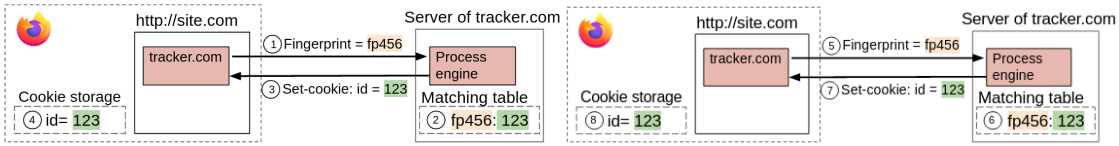
We say that a tracker *respawns a cookie* when it recreates the exact same cookie after the user revisit the website in a clean browser.

2.1 *How can trackers benefit from a combination of cookies and browser fingerprint?*

To benefit from both techniques, the tracker can first use a browser fingerprint to create an identifier and store it in the browser's cookie. In this way, even if a user cleans this cookie, the identifier can be recreated with a browser fingerprint. Moreover, even if the fingerprint changes over time, the identifier stored in the cookie can help to match the new fingerprint with the old fingerprint of the same user. We explain these scenarios and benefits in details below.

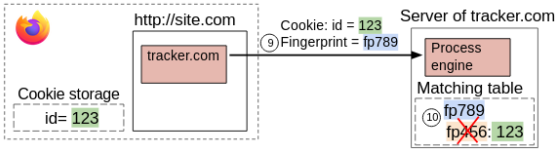
Figure 5.1(a) shows that the tracker first receives a set of user's active and\or passive features (step ①). In step ②, the tracker generates an identifier from the received features, that it might store on the server's matching table. The tracker then stores the created identifier in the user's browser cookie, either via the Set-cookie header (step ③) or programmatically via JavaScript (not shown in Figure 5.1(a)). As a result, an identifier is stored in the browser's cookie database (step ④).

Figures 5.1(b) shows what happens when the user does not have a cookie *123* in her browser, however the fingerprint *fp456* remains the same. In this case, the fingerprint *fp456* is sent to the server of tracker.com (step ⑤), and it allows the tracker to match the known fingerprint and the cookie previously set for this user (step ⑥). As a result, the tracker is able to set again the same cookie *123*, previously deleted by the



(a) Cookie respawning with browser fingerprinting tacking mechanism

(b) Recreation of cookies using browser fingerprint



(c) Usage of cookies to ensure fingerprint stability

Figure 5.1: **Cookie respawning with browser fingerprinting tracking technique.** (a) (step 1) The tracker receives user's features, (step 2) then stores a fingerprint *fp456* associated with the features and generates a corresponding cookie *123*. (step 3) Next, the tracker sets the cookie in the user's browser. (step 4) As a result an identifier is stored in the browser cookie storage. (b) When the user cleans her browser and revisit the website, (step 5) the tracker receives the fingerprint *fp456*, (step 6) extracts the corresponding cookie from the matching table, (step 7) and re-sets it in the user's browser. (step 8) As a result, the cookie *123* is recreated in the user's browser. (c) The fingerprint is not stable over time, (step 9) thus the user fingerprint might change. (step 10) The tracker can use the cookie received with the fingerprint to update the latest on the server side.

user (step ⑦). This allows the tracker to respawn deleted user cookies with browser fingerprinting and continue tracking her via such cookies (step ⑧).

Figure 5.1 (c) presents the consequences of cookie respawning with browser fingerprinting. When the browser fingerprint of the user is updated from *fp456* to *fp789*, the server of tracker.com receives an old cookie *123* with a new fingerprint *fp789* (step ⑨). The cookie *123* helps the server to recognize the user's browser and update the corresponding record in the matching table and substitute a fingerprint *fp456* to *fp789* associated to cookie *123* (step ⑩). This allows the tracker to match different fingerprints of the same user, given that fingerprinting is not stable over time.

As a result, cookie respawning with browser fingerprinting allows trackers to respawn deleted cookies, and also to link different browser fingerprints of the same user. This makes the tracking robust to either cookie deletion or fingerprint change. Only in case the browser fingerprint changes and the cookie is deleted at the same time, the tracker will not be able to recognize the user and hence to continue tracking this user.

In this paper, we propose a robust methodology to detect the mechanisms presented in Figures 5.1 (a) and (b). In this section, we first introduce our methodology to crawl Alexa top $30,000$ websites (Section 2.2). Next, we present our method to detect cookie respawning with browser fingerprinting (Section 2.3). Then, we describe the fingerprinting features used in our study and spoofing techniques (Section 2.4). Finally, we list the limitations of our methodology (Section 3).

2.2 *Measurement setup*

We performed passive web measurement on March 2021 of the Alexa top $30,000$ websites extracted on March 2020*. All measurements are performed using the OpenWPM platform [175] on the Firefox browser. OpenWPM provides browser automation by converting high-level commands into automated browser actions. We used two machines to perform the crawls in our study. The versions of OpenWPM and Firefox, the time period of the crawl, and the characteristics of the two machines used in this study are presented in Table 8.3 of the Appendix.

We used different characteristics with two machines so that they appear as different users, as done by previous works [116, 18, 97, 96]. Ideally, we would have used two distinct machines with different locations to detect user specific cookies, however, both machine A and machine B are located in France. Hence, to change the Machine B geolocation, we spoofed the parameters latitude and longitude by modifying the value of $\mathrm{geo.wifi.uri}$ advanced preference in the browser and point it to Alaska.

All our crawls are based on the notion of *stateless crawling instances*. We define a stateless crawling instance of a website X as follows: (1) we visit the home page of the website X and keep the page open until all content is loaded to capture all cookies stored (we set the timeout for loading the page to 90s), (2) we clear the profile by removing the Firefox profile directory that includes all cookies and browser storages. The rational behind the stateless crawling instance is to ensure that we do not keep any state in the browser between two crawling instances. This guarantees that respawned cookies do not get restored from other browser storages.

We perform stateless crawling instances of the Alexa top $30,000$ websites and for *each stateless crawling instance*, we extract the following from the information automatically collected during the crawls by OpenWPM:

1. For each HTTP request: the requested URL, the HTTP header.
2. For each HTTP response: the response URL, the HTTP status code, the HTTP header.
3. All JavaScript method calls described in Table 5.1.
4. All cookies set both by JavaScript and via HTTP Responses. On these collected cookies, we perform the following filtering as shown in Figure 5.2: first, we select cookies recreated after cleaning the cookies database; second, we filter out cook-

*We made this list of websites publicly available [33].

JavaScript calls	API
HTML5 Canvas	HTMLCanvasElement, CanvasRenderingContext2D
HTML5 WebRTC	RTCPeerConntection
HTML5 Audio	AudioContext
Plugin access	Navigator.plugins
MIMEType access	Navigator.mimeTypes
Navigator properties	window.navigator
Window properties	Window.screen, Window.Storage, window.localStorage, window.sessionStorage, and window.name

Table 5.1: **Recorded JavaScript calls.**

ies that are not user-specific; finally, we filter out cookies that are not respawn with studied features (Section 2.3).

2.3 *Detecting cookie respawning with browser fingerprinting with sequential crawling*

Figure 5.2 presents our sequential crawling methodology that detects which fingerprinting features are used to respawn cookies. Our method consists of two main steps explained in this section:

- **Create the initial set of candidate respawned cookies:** we identify candidate respawned cookies by collecting all cookies that get respawned in a clean browsing instance, and we remove cookies that are not user-specific.
- **Identify dependency of each respawned cookie on each fingerprinting feature:** we spoof each feature independently to detect whether the value of a respawned cookie has changed when the feature is spoofed. We perform a permutation test ($N = 10,000$, $p < 0.05$) to add statistical evidence on the dependency between a feature and the respawned cookie.

2.3.1 Creation of the initial set of candidate respawned cookies

To build the initial set of candidate respawned cookies, we perform two stateless crawling instances from machine A as described in Figure 5.2 (see *initial crawl* and *reappearance crawl*). Via these two crawls, we ensure that all browser storages are cleaned and the only possible way for cookies to be respawn is with browser fingerprinting.

We define a cookie as the tuple *(host, key, value)* where *host* is the domain that can access the cookie. To create the set of candidate respawned cookies, we only collect cookies that appear in both the *initial crawl* and *reappearance crawl* when visiting the

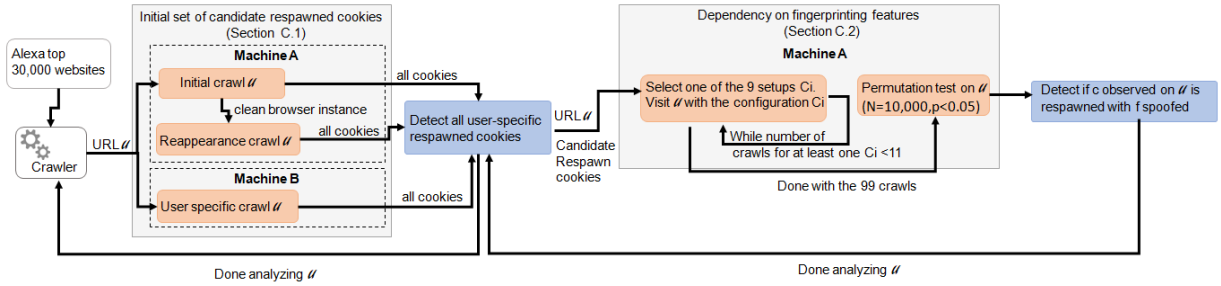


Figure 5.2: **Sequential crawling of $30,000$ top Alexa websites to identify cookie respawning with browser fingerprinting.** For each website, we perform an *initial crawl* from machine A and, a *user specific crawl* from machine B to detect machine unrelated cookies. After *initial crawl* finishes, we start a *reappearance crawl* from machine A to detect reappearance of cookies. Using *initial crawl*, *user specific crawl*, and *reappearance crawl* we detect *user-specific* cookies that reappear in *reappearance crawl*, but not in *user specific crawl*. For such cookies, we randomly chose one configuration C_i: either spoof one feature at a time or to set all features to initial value. We perform 99 stateless crawls (11 *spoofing crawl*s per feature and 11 *control crawl*s where the studied features are unspoofed). Finally, we perform a permutation test for each feature (N=10,000), and we consider that the cookie is *feature dependent* if the resulting p-value < 0.05. All these steps are discussed in Section 2.3.

same website in the two crawls. Note that due to our sequential crawling (that is, we visit websites in a sequence), we only consider candidate respawned cookies within the same website.

Previous research [116, 18, 97, 96] considered that cookies are non specific to the users and hence unlikely to be used for tracking when their values are identical for several users. Therefore, using distinct machines to remove non user-specific cookies became a common method in this research area. We follow this methodology and remove cookies that are not user-specific from our set of candidate respawned cookies. To do so, we performed an additional *user specific crawl*[†] from a different machine B that appears to trackers as a different user. It's important that machines A and B have different fingerprinting features (see Table 8.3 of the Appendix) to avoid wrong categorization of cookies that depend on these features as non user-specific.

We hence remove the following cookies from the candidate set of respawned cookies and keep only user-specific cookies:

- a cookie *(host, key, value)* if it appears on both the *initial crawl* on machine A

[†]Practically, we perform the *initial crawl* and *user specific crawl* in parallel, and the *reappearance crawl* right after the *initial crawl* completes (Figure 5.2).

and *user specific crawl* on machine B with the same *host*, *key*, and *value*.
- a cookie *(host, key, value)* if a cookie with the same *host* and *key* is not present in a *user specific crawl*. We adopt a conservative strategy to remove such cookies because we do not have a proof that such cookies are user-specific.

Our robust deletion method for cookies that are not user-specific or do not re-appear in a *user specific crawl* allows us to ensure that only user-specific cookies are further analysed.

2.3.2 Identifying dependency of each respawned cookie on each fingerprinting feature.

The set of candidate respawned cookies contains cookies that are both user-specific and respawn when crawled a second time after we used a new browser instance with a cleaned browser storage. Therefore, cookies in this set are very likely to be respawned with the use of browser fingerprinting. To detect which fingerprinting features are used to respawn the collected cookies, we performed the following steps. We first identified 8 fingerprinting features from previous research (see more details on the choice of features and methods to spoof them in Section 2.4). Then, for each website u where we have at least one candidate respawned cookie, we perform 99 crawls, 11 spoofing crawls per studied fingerprinting feature f, and 11 crawls with all features set to their initial values (as in *initial crawl*) that we refer to as *control crawls*. In each of the total 88 spoofing crawls, we first spoof the feature f and perform a stateless *spoofing crawl* of the website u. For each user-specific respawned cookie from the candidate set, we perform the following algorithm.

- For each of the 99 crawls, we label the cookie as respawned if the cookie's *host* and *key* are identical but *value* are different from the initial crawl. As a result we get 11 observations for each configuration (either one of the 8 features is spoofed or no feature spoofed.)
- For every feature, we perform a permutation test with the 11 observations from the *control crawls* using 10,000 permutations. The statistical test assess the difference of the probability to have the cookie respawned between the feature crawls and the control crawls.
- We consider that the cookie is *feature dependent* if the p-value for the test statistic is lower than 0.05.

2.4 *Selection of fingerprinting features and spoofing techniques*

To achieve a high uniqueness of an identifier built from a browser fingerprint, trackers use a combination of both passive and active browser and machine features. Though browser features are useful for fingerprinting, using them alone might be problematic for trackers because the usage of multiple browsers is recommended and common among users [227, 203, 64]. To improve the accuracy of the fingerprint, trackers also

Browser features	Accept language [140, 31]	Active/Passive
	Geolocation [31]	Active
	User agent [140, 57, 121, 31]	Active/Passive
	Do not track [140, 121]	Active/Passive
Machine features	WebGL [140, 64, 121, 31, 168]	Active
	Canvas [140, 64, 121, 18, 81, 96, 168]	Active
	IP address [57, 31, 81, 96]	Passive
	Time zone [140, 57, 121]	Active

Table 5.2: **Studied fingerprinting features.**

use machine related features such as the IP address, or the OS version [57, 31].

Table 5.2 presents a full list of studied browser and machine features that we selected based on the most common features in prior works for browser fingerprinting [140, 57, 121, 31, 64, 18, 81, 96, 168].

We have used two methods to spoof fingerprinting features: 1) via Firefox preferences and 2) add-ons. We have validated that each feature has been properly spoofed on our own testing website with a fingerprinting script and also by using *whoer* website [235] that verifies the information sent by the user's browser and machine to the web.

2.4.1 Spoofing using Firefox preferences

Firefox allows to change its settings in the browser preferences of *about:config* page. With this method, we spoofed the following features.

User agent. The *User-Agent* HTTP header allows the servers to identify the operating system and the browser used by the client. The *initial crawl* run in Firefox under Linux (see Table 8.3 in the Appendix for details). To spoof the user agent, we changed the general.useragent.override preference in the browser to Internet Explorer under Windows: (*Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; AS; rv:11.0) like Gecko*). We checked the spoofing efficiency on our testing website with an injected script. The script returns the user agent value using the navigator.userAgent API. We tested the user agent value returned with the HTTP header using the *whoer* website. We found that the user agent value was spoofed both in JavaScript calls and HTTP headers.

Geolocation. The geolocation is used to identify the user's physical location. The *initial crawl* has as location *Cote d'Azur, France*. We spoofed the geolocation parameters latitude and longitude by modifying the value of geo.wifi.uri preference in the browser and point it to the *Time Square, US* ("lat": 40.7590, "lng": -73.9845). We validated the spoofing efficiency using a script call to navigator.geolocation API.

WebGL. The WebGL API is used to give information on the device GPU. In our study, we focus on the WebGL renderer attribute that precises the name of the model of the GPU. We spoofed the WebGL renderer using the webgl.render-string-override preference in the browser. We changed the value of WebGL renderer to *GeForce GTX 650 Ti/PCIe/SSE2*. To retrieve information about the graphic driver and read the WebGL renderer value, we used the WEBGL_debug_renderer_info add-on. We validated the WebGL spoofing efficiency by using the add-on in our customized website.

Do Not Track. The Do Not Track (DNT) header indicates user's tracking preference. The user can express that she doesn't want to get tracked by setting the DNT to True. In the *initial crawl*, the DNT was set to *null*. We enabled the Do Not Track header, and we set it to True using the *privacy.donottrackheader.enabled* preference. We validated that the DNT returned value in the HTTP header is set to True using the *whoer* website.

2.4.2 Spoofing using browser add-ons

The browser preferences do not provide a spoofing mechanism for all fingerprinting features. We used browser add-ons to spoof such features.

Canvas. The HTML canvas element is used to draw graphics on a web page. The difference in font rendering, smoothing, as well as other features cause devices to draw images and texts differently. A fingerprint can exploit this difference to distinguish users. We spoofed the canvas by adding a noise that hides the real canvas fingerprint. To do so, we used the Firefox add-on *Canvas Defender* [63]. To test the add-on efficiency, we built a customized website where we inject a canvas fingerprinting script. The script first draws on the user's browser. Next, the script calls the Canvas API *toDataURL* method to get the canvas in dataURL format and returns its hashed value. This hashed value can then be used as a fingerprint. To evaluate the add-on efficiency against the canvas fingerprinting, we revisited the customized website and compared the rendered canvas fingerprint. We found that the returned canvas hashed values were different upon every visit.

IP address. We run the *initial crawl* with an IP address pointing to France. We spoofed the IP address using a VPN add-on called *Browsec VPN* [232]. We used a static IP address pointing to the Netherlands. Consequently, the spoofed IP address remain constant during the runs of spoofed crawls. We checked that the IP address changed using the *whoer* website.

Time zone. We launched the *initial crawl* with *Paris UTC/GMT +1* timezone. We spoofed the timezone to *America/Adak (UTC-10)* using the Chameleon add-on [68].

Accept-language. The *Accept-language* header precises which languages the user prefer. We used English as *Accept-language* in *initial crawl*. We spoofed the accept-language header using the Chameleon add-on [68] to Arabic. We checked that it was properly spoofed using the *whoer* website.

2.5 *Limitations*

Spoofing features and implementing the spoofing solution with the OpenWPM crawler requires substantial engineering effort. Therefore, we limit our study to 8 browser features that are commonly used by previous works and that can be spoofed either directly using browser settings, or using the add-on (Canvas Defender, Browsec VPN, and Chameleon) that we successfully run with OpenWPM. Consequently, cookies respawned using other features are excluded from this study. The number of excluded cookies is 2,976 (see Section 3.1). This is a limitation that does not impact the main goal of our study, as we do not intend to be exhaustive in the identification of respawned cookies, but we aim to understand and describe the mechanisms behind respawning, and propose a robust methodology to detect features that are used by trackers to respawn cookies.

Given that we spoof one feature at a time, we may introduce inconsistency between different features. For example, when we spoof the geolocation API, we don't modify the time zone or the IP address. This method doesn't invalidate our results because we detect dependency on each feature separately. Nevertheless, we may miss trackers that modify their behaviour when some features are spoofed.

Non user-specific cookies are not intrusive for the user's privacy because they are identical among different users. We are aware that the cookies we classify as non user-specific might have been respawned due to features we do not consider.

3 Results

In this section, we present findings on prevalence of cookie respawning with browser fingerprinting, identify responsible parties, and analyze on which type of websites respawning occurs more often. Our results are based on Alexa top $30,000$ websites where we extracted a total of $428,196$ cookies. We study the respawning of both first and third party cookies.

3.1 *How common is cookie respawning with browser fingerprinting?*

Table 5.3 presents an overview of the prevalence of cookie respawning with browser fingerprinting. We extracted $428,196$ cookies from the visited $30,000$ websites. Using the *reappearance crawl*, we extracted a set of cookies that did reappear in the crawl. As a result, we obtained a set of $88,470$ (20.66%) reappearing cookies that appear on $18,117$ (60.39%) websites.

Next, we filtered out cookies that are not user-specific – they appear with the same (host, key, value) on *initial crawl* and *user specific crawl* – and cookies that only appear on *initial crawl* but not in *user specific crawl* (Section 2.3.1). We found that out of $88,470$ reappearing cookies, $5,144$ (5.81%) are user specific. The set of user specific cookies is observed on $4,093$ (22.59%) websites.

Crawls	*Initial*	*Reappearance*	*User specific*	*Feature dependent*
Collected cookies	428,196	88,470	5,144	1,425
Occurrence on websites	30,000	18,117	4,093	1,150

Table 5.3: **Cookie respawning with browser fingerprinting is common on the web.** We detected 1,425 respawned cookies that appear on 1,150 websites. We define the Initial, Reappearance, User specific crawls and Feature dependent cookies in Section 2.3.

After filtering out non reappearing cookies and keeping only user specific cookies, we identified cookies whose value depend on at least one of the studied features following our methodology detailed in Sections 2.3.2. As a result, we extracted $1,425$ respawned cookies that appear on $1,150$ (3.83%) websites. Out of the remaining $3,719$ set of cookies, 743 were excluded from the statistical test because they did not appear on the 99 spoofing and control crawls. The remaining $2,976$ cookies that are user specific and not detected as feature dependent can be respawned via other features that are out of scope of our study.

Summary. We found $1,425$ cookies respawned using at least one of the studied features. These cookies were respawned in $1,150$ websites that represent 3.83% of the visited websites.

3.2 *Which features are used to respawn cookies?*

In this section, we present the results we obtained from the sequential crawling methodology (Section 2.3). For each of the $1,425$ respawned cookies, we detected features on which the cookie value depends (see all studied fingerprinting features in Table 5.2).

Given that a cookie can be respawned with several features, we consider that a cookie *C is respawned with a set of features F* if the value of *C* depends on every feature in *F* (such detection was done independently for each feature as described in Section 2.3.2).

Table 5.4 presents the number of times each feature is used to respawn a cookie. IP address is the most commonly used feature to respawn cookies and is used in respawning of 672 (47.15%) cookies. The second most popular feature to respawn cookies is User-Agent (UA) – it is observed with 486 (34.10%) cookies. Note that features that can be easily collected passively, like IP address and UA, are more frequently used than features that can only be accessed actively, such as Canvas or Geolocation.

We found that cookies are usually respawned with a set of different fingerprinting features. In our dataset, cookies are respawned with 184 distinct sets of features. Figure 5.3 shows the sets of features most often used for cookie respawning. We see

	Passive	Active/Passive			Active			
Features	**IP**	**UA**	**Lang**	**DNT**	**CV**	**GEO**	**GL**	**TZ**
Occurrence	672	486	278	277	231	249	292	310

Table 5.4: **IP address is the most commonly used feature to respawn cookies.** Occurrence: number of times a feature has been used to respawn a cookie (either independently or in combination with other features). CV: Canvas, IP: IP address, UA: User agent, GEO: Geolocation, GL: WebGL, TZ: Time zone, Lang: Accept language, DNT: Do Not Track.

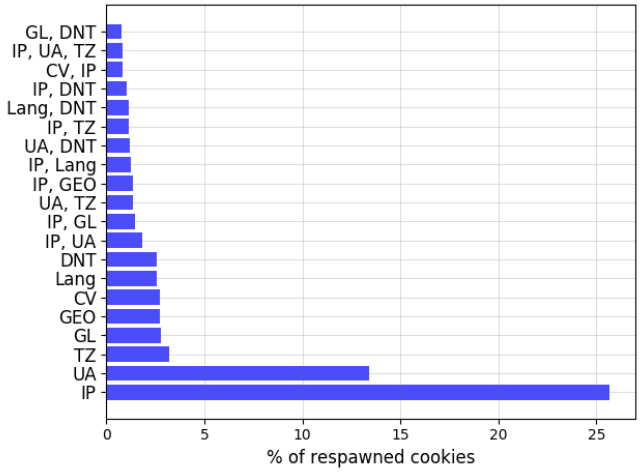


Figure 5.3: **Top 20 set of features used to respawn cookies.** IP addresses alone are used to respawn over 25% of the cookies. CV: Canvas, IP: IP address, UA: User agent, GEO: Geolocation, GL: WebGL, TZ: Time zone, Lang: Accept language, DNT: Do Not Track.

that the IP address alone is the most commonly used feature to respawn cookies, and moreover no other set of features is more popular than the IP address alone.

The IP address is used alone to respawn 366 (25.68%) cookies. Mishra et al. [165] studied the stability and uniqueness of the IP address over a duration of 111 days on a dataset of 2,230 users. They showed that 87% of participants retain at least one IP address for more than a month. Hence, IP addresses are both stable and unique, therefore, they can be used to uniquely identify and track user's activity. Interestingly, the top-2 sets of features, {IP}, and {UA}, contain only passive features that are easier to collect. Active features are rarely use, timezone, the most popular active feature for respawning, is used alone for 46 (3.23%) cookies.

Summary. We show that trackers use multiple combinations of features to respawn cookies and that the IP address, which is overlooked in a number of fingerprinting studies [140, 64, 121, 18, 168], is the most used feature to respawn cookies.

3.3 *Discovering owners of respawned cookies*

Cookie respawning opens new opportunities for different companies to collaborate together to track users. Usually, the *host* of a cookie defines the domain that can access the cookie. We introduce in this chapter a notion of cookie *owner* that has set the cookie via an HTTP header or programmatically via JavaScript (see Section 2). However, additional stakeholders can help to respawn a cookie by serving a fingerprinting script. We explore each of these new potential stakeholders in the rest of this section.

3.3.1 Identifying cookie owners

Due to the the Same Origin Policy (SOP) [197], the domain that is responsible for setting a cookie can be different from the domain that receives it (see Chapter 2). Therefore, we differentiate two stakeholders: *Owner* – the domain that is responsible for setting the cookie, and *Host* – the domain that has access to the cookie and to whom the cookie is sent by the browser. In the following, we define both owner and host as 2^{nd}-level TLD domains (such as tracker.com).

It's important to detect the cookie owner – for instance, in order to block its domain via filter lists [90, 91, 86] and prevent cookie-based tracking. Indeed, the notion of cookie owner is often overlooked when the reasoning is only based on the cookie host [62]. When one cookie owner sets a cookie in the context of several websites (the owner's script can be embedded directly on a visited website or in a third-party *iframe*), the host of this owner's cookie is the context where the cookie is set because of the SOP [197]. To identify cookie owners in the context of our work, we distinguish two cases, as described below.

Cookie set by a script. Document.cookie property is the standard way for a JavaScript script to set a cookie [137] programmatically. To check whether a cookie is set via JavaScript and to extract its *owner* (the domain who serves the script) when

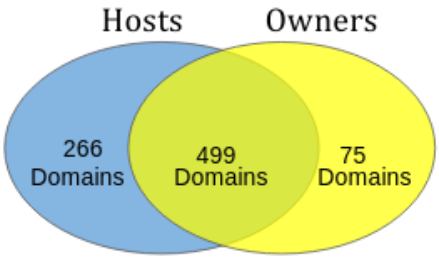


Figure 5.4: **Emergence of new domains when considering cookie owners.** The $1,425$ respawned cookies have 765 distinct hosts and 574 distinct owners. The notion of cookie owner allows to identify 75 cookie owner domains that never appear as a cookie host. We also found 266 cookie host domains that are never used to set the cookie.

crawling a website, we (1) extract the set of scripts S that set a cookie on the website using document.cookie, (2) for every script in S, we extract the set of cookies C set by this script, and (3) check whether there is an overlap between the set of respawned cookies identified in Section 3.1 and in the set C. If it is the case, we conclude that the cookies in the overlap are set via JavaScript, and their *owner* is the 2^{nd}-level TLD domain that served the script.

Cookie set by HTTP(S) header. If the cookie is set by the HTTP(S) Set-Cookie response header, then the *owner* of the cookie is the same as its host because it corresponds to the 2^{nd}-level TLD of the server that set the cookie.

For each of the $1,425$ respawned cookies, we identified its *owner* depending on how the cookie was set. Figure 5.4 shows domains appearing as host only (left blue part), as owner only (yellow part), or both (middle overlap). In total, $1,425$ respawned cookies are labeled with 765 distinct hosts, however they were set by 574 distinct owners. Figure 5.4 also depicts that 75 domains appear as owners and never as cookies hosts. These domains serve JavaScript scripts that set cookies, but never serve cookies directly via an HTTP(S) response header. Hence, when only considering cookies hosts, these domains are not detected. We evaluated the efficiency of disconnect [86] filter list in detecting these 75 domains. We found that disconnect miss 53 (70.66%) owners domains. We also found that 266 domains that appear as cookie hosts are never identified as cookie owners. Cookies associated with these domains were set in the context of the hosts domain because of the SOP, but these domains were never actually responsible of setting these cookies.

Figure 5.5 presents the top 10 domains responsible for cookie respawning that are either cookie hosts, cookie owners, or both. Two domains, rubiconproject.com and casalemedia.com, represent the largest fraction of websites. All cookies served by these two domains are served via HTTP(s). Three out of the top 10 domains are exclusively cookie owners: adobetm.com, bizible.com, and maricopa.gov. These domains

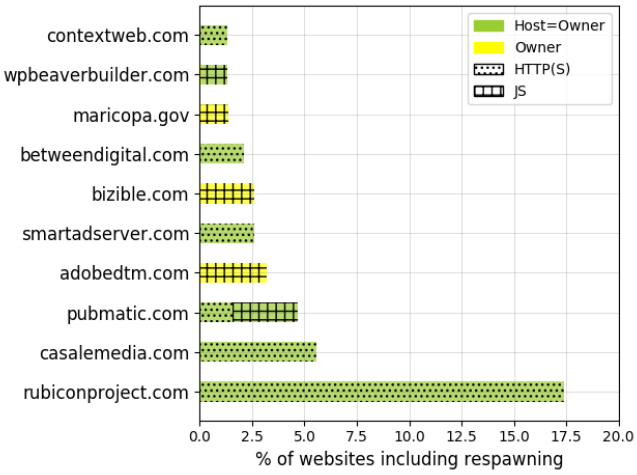


Figure 5.5: **The top 10 respawned cookies owners.** The bar is green when the domain is both host and owner, and yellow when the domain only appears as owner. For each domain, we show when cookies are set via an HTTP(S) header and when they are set via JavaScript. When considering cookie owners, new domains are identified such as adobedtm.com.

are only setting respawned cookies via JavaScript and never directly through HTTP(S). Out of the $1,425$ respawned cookies, 514 (36.07%) are set via JavaScript.

Summary. Previous studies that only looked at the cookie host can miss the trackers responsible for setting the cookies. In our study, 75 domains could be missed if we only considered cookie hosts. We found that disconnect miss 70.66% of these domains. Considering cookie owners improves the understanding of the tracking ecosystem.

3.3.2 Identifying scripts used for respawning

A cookie can be respawned using a set of different features. These features can be all accessed by a single script or by multiple collaborating scripts as we describe in this section. To identify the scripts that are responsible of accessing browser or machine features used for respawning a cookie, we use the recorded JavaScript calls described in Table 5.1.

Every feature can be accessed only actively, only passively or actively and passively (see Table 5.2). In this section, we focus *only on the active features* collected using the following JavaScript calls: window.navigator.geolocation (to access the Geolocation) and HTMLCanvasElement (to access the Canvas). As OpenWPM does not log calls to Time zone and WebGL, we do not consider these active features in this section. For every respawned cookie C, we identified the set of features F used for respawning C as described in Section 2.3.2. To extract the scripts that are responsible of respawning

Owner	# of cookies
adobedtm.com	10
ssl-images-amazon.com	3
hdslb.com	2
bitmedia.io	2
19 Others	20
Total	37

Table 5.5: **Top domains suspect to set and respawn the cookies.**

C via the set of features *F*, we analyze the features used to respawn each cookie. If the cookie is respawned with only passive or active/passive features, then no conclusions can be made for both HTTP(S) and JavaScript cookies. In fact, these features are sent passively, therefore no conclusion can be made on which scripts used them in respawning.

In total, we found that 931 (65.33%) cookies are respawned with only passive or mixed features. For the remaining 494 cookies depending on active features, 95 (19.23%) are only using WebGL or Time Zone that are out of the scope of this study. In the rest of this section, we consider the 399 respawned cookies that are respawned with only active features for which we can access method calls.

We refer to the set of active features used to respawn a cookie as f_a. We extract the set of features accessed by every script on the website where the cookie is respawned, and distinguish three cases.

1 - The owner of the cookie is suspect to be the responsible of respawning. We identify such cases when (1) the cookie is set via JavaScript, and all active features f_a used to respawn it are accessed by the owner of the cookie, or (2) the cookie is set via HTTP(S) and a script hosted by the same 2^{nd}-level TLD accesses all active features. If one of the two cases is validated, then we suspect that the owner of the cookie is the responsible of respawning it. In total, we found 37 (9.27%) cookies that are respawned by their owners. Out of these 37 cookies, 17 are set via HTTP(S) and respawned by scripts that belong to the same domain. These 37 respawned cookies are owned by 23 distinct owners. Table 5.5 presents the top 4 owners that are suspect to respawn the cookies as well.

We found that adobedtm.com [27] (the tag manager owned by adobe) is the top domain that both owns and respawns cookies. Though respawning is not explicitly indicated in their policy, the policy states that they collect browser and machine features. We didn't find any information regarding cookies respawned either by ssl-images-amazon.com, hdslb.com or bitmedia.io [55].

2 - The respawning is a result of a potential collaboration between the cookie owner and other scripts. If the set of active features used to respawn the cookie are not accessed by the owner, but are accessed by other scripts on the same website,

Owner	Collaborator	# of cookies
rubiconproject.com	googlesyndication.com	8
rubiconproject.com	pushpushgo.com	3
adobedtm.com	morganstanley.com	2
adobedtm.com	provincial.com	2

Table 5.6: **Top domains suspect to collaborate to respawn cookies.** The reported domains are both first- and third- party.

then we suspect that the cookie is potentially a result of collaboration between the owner of the cookie and other scripts on the same website. In this study, we don't assess whether the domains are actively collaborating, or if one domain is leveraging scripts from other domains to glean fingerprint information. In total, we found that 67 (16.79%) cookies are suspect to be a result of collaboration between multiple domains. The 67 cookies are a result of collaboration of 35 distinct domains.

Table 5.6 presents the top domains that are suspect to collaborate in order to respawn cookies. We define the collaborator as the only domain accessing the features used for respawning the cookie and not accessed by the owner of the cookie. The top collaboration involves googlesyndication.com owned by Alphabet (parent company of Google). Googlesyndication.com is accessing and potentially sharing user's Canvas information.

3 - The responsible of respawning the cookie are not all known. If not all the active features used to respawn the cookie are accessed on the website where the cookie is respawned via JavaScript calls, then we assume that the owner is accessing the features via other means. This happens with 295 (73.93%) cookies. In 186 (63.05%) cookies out of the 295, the owner is not accessing the geolocation API and do access other active features it used for respawning the cookie. This can potentially be a result of the dependency between geolocation and IP addresses. When we spoof the geolocation to Time Square in the US, we keep an IP address that points to France because we only spoof one feature at time. Hence, companies may detect this incoherence, and not use the IP address to respawn the cookie, which, in our experiment will be identified as dependency on the geolocation feature.

Summary. Identifying the responsible of respawning can prove to be a complex task. While 23 owners respawn cookies themselves, 35 domains collaborate to respawn cookies.

3.4 *Where does respawning occur?*

In section 3.3, we studied the domains that are responsible of setting and respawning cookies. In this section, we analyse on which types of websites respawning occurs. In the following, we refer to these websites as *websites including respawning*. We analyse

Alexa rank interval	Websites including respawning	# of owners
0 — 1k	49 (4.9%)	49
1k — 10k	360 (4%)	213
10k+	741 (3.70%)	382

Table 5.7: **Popular websites are more likely to include cookie respawning.** Number of owners: presents the total number of distinct respawned cookies owners in the Alexa ranking interval.

Alexa ranking distribution and impact of websites category on the usage of cookie respawning, present websites including respawning that process special categories of data, and present the geolocation of owners of respawned cookies and websites.

Popularity of websites including respawning. We detected $1,150$ websites where at least one cookie is respawned. Table 5.7 presents the number of websites including respawning for each Alexa rank interval. We observe that cookie respawning with browser fingerprinting is heavily used on popular websites: out of the top 1k visited websites, 4.9% are websites including respawning. This percentage decrease to 3.70% in less popular websites.

Categorization of websites including respawning. We used the *McAfee service* [158] to categorize the visited websites. The McAfee uses various technologies and artificial intelligence techniques, such as link crawlers, and customer logs to categorise websites. It is used by related works [225]. A description of the reported McAfee categories can be found in the McAfee reference guide [159].

We successfully categorized Alexa 29,900 visited websites. For every category, we present the percent of respawn websites. We found that the visited websites belong to 669 categories and the $1,150$ websites including respawning belong to 143 different categories.

Figure 5.6 gives an overview of the 10 most prominent categories within the Alexa visited websites. We found that all top 10 categories contain websites that include respawning. Business is the top websites category, 8.62% of the visited websites are categorized as business.

Most of websites including respawning are categorized as *General News*. Out of the $29,900$ visited websites, 6.73% are categorized as *General News*, and 5.95% of these *General News* websites contain at least one respawned cookie. *General News* is known for using more third parties than other categories [211], which can be the reason behind the high deployment of respawning in this category of websites.

Websites processing special categories of data. The GDPR [218, Recital 51] stipulates that personal data which are particularly sensitive by their nature, merit specific protection, as their processing could create significant risks to the fundamental rights of users. Such data include personal data revealing sensitive information such as data

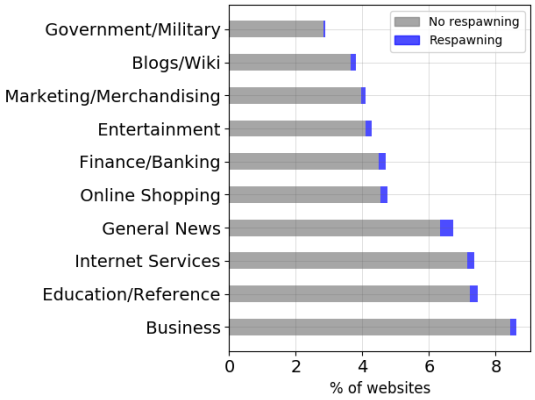


Figure 5.6: **General news is the top category including cookie respawning with browser fingerprinting.** We consider that a website U is including respawning if it contains at least one respawned cookie. The bar is gray when we don't detect respawning in the website, and is blue when we do.

concerning a natural person's sex life or sexual orientation [218, Article 9]. Processing such categories of data is *forbidden*, unless allowed by the user's explicit consent [218, Article 9(2)].

We studied tracking via the third-party respawned cookies on websites processing sensitive data. As a result, we detected 21 cookies respawned in *Adult* websites that are set by 19 different owners. The top domain respawning cookies on sensitive websites is adtng.com (no corresponding official website was found for adtng.com). It respawned cookies on 3 different adult websites, and therefore, can track and link user's activity within adult websites in a persistent way, *without explicit consent to legitimize such operation*, rendering such respawning practise unlawful.

Geolocation of websites including respawning and respawned cookies owners. Independently of the country of registration of a website, if a website *monitors* the behavior of users while they are in the EU, the GDPR applies to such monitoring [218, Article 3(2)(b)]. Notice that any form of web tracking will be deemed as "monitoring", including cookie respawning with browser fingerprinting. Since our experiments simulate users located in France (EU), both EU and non-EU organizations must comply with the GDPR.

We extracted the country of registration of the owners of respawned cookies and the websites including them using the *whois library* [236]. We successfully identified the country of registration of 362 (63.07%) out of 574 total distinct owners, and 670 (58.26%) out of 1,150 websites including respawning. We found that the owners and websites are distributed across the globe, ranging respectively over 29 and 47 different countries, including EU. Out of these 670 websites, 52 (7.76%) are in the EU.

Figure 5.7 presents the registration countries of respawned cookies owners and web-

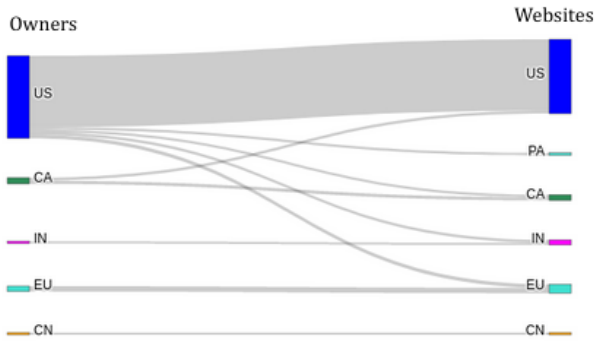


Figure 5.7: **Cookie respawning with browser fingerprinting is geolocaly distributed.** Corresponding countries of owners (left) and websites including respawning (right) of respawned cookies. We present the top 10 (owner,website) geolocation. "EU" label represents the 27 member states of the EU.

sites where they are set. We observe that top countries of both respawned cookies and websites including respawning are not in the EU: 356 (24.98%) of the respawned cookies are both originated and included by domains from the US. We also observed that respawned cookies on Chinese websites are only set by Chinese owners, and interestingly, websites registered in Panama are active in respawning as well (22 (3.28%) of the studied 670 websites including respawning are from Panama).

Summary. Cookie respawning with browser fingerprinting is commonly used: 5.95% of *General News* websites contain at least one respawned cookie. We found that cookies are respawned in sensitive *adult* websites as well, which leads to serious privacy implications: Cookie respawning with browser fingerprinting is distributed across the globe, however, only 7.76% of the websites that include respawning are in the EU. Nevertheless, both EU and non-EU websites must comply with the GDPR as it is applicable independently of the country of registration of the website where EU users are monitored.

3.5 *Tracking consequences of respawning*

3.5.1 Persistent cross-site tracking with respawned cookies

Basic tracking via third-party cookies [193, 114] is the most known tracking technique that allows third parties to track users across websites, hence to recreate her browsing history. When a third party cookie that enables cross-site tracking is respawned, such tracking becomes *persistent*. That is, in contrast to regular third-party tracking, the user can not prevent it by deleting cookies. Hence, respawned cookies enable persistent tracking that allows trackers to create larger users' profiles by linking users activity

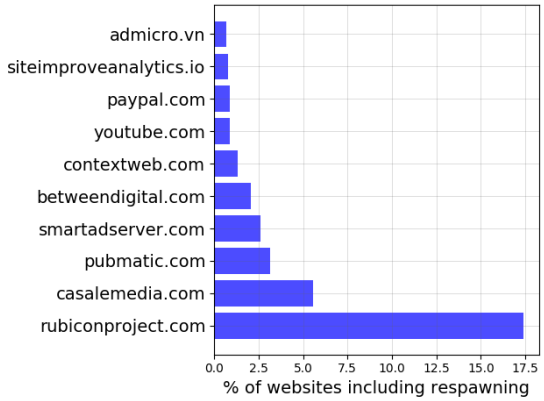


Figure 5.8: **Persistent third-party tracking based on respawned cookies.** Top 10 cross-site trackers using respawned cookies.

before and after they clean their browser. Since the host is the domain to whom browser automatically sends the cookies, we focus on the cookie *host* and not on cookie owner.

Third party cookies allow trackers to track users *cross-websites* [193]. In this section, we only analyse third-party respawned cookies that can be used to track users across websites. Note that all extracted respawned cookies are user specific (Section 2.3.1) and therefore can be considered as *unique identifiers*. Out of $1,425$ respawned cookies, 528 (37.05%) are third-party cookies. In total, we identified 144 unique hosts that have access to these cookies. Figure 5.8 presents the top 10 cross-site trackers that have access to respawned cookies. We found that rubiconproject.com is the top domain: it has access to at least one respawned cookie on 200 (17.39%) of the visited websites out of $1,150$. Rubiconproject.com defines itself as a publicly traded company, as it is automating the buying and selling of advertising [194].

3.5.2 Cookie respawning with browser fingerprinting beyond deprecation of third-party cookies

Web browsers are moving towards deprecation of third party cookies which are the core of cross-site tracking [199]. *Can this deprecation prevent cross-site tracking?* In the following, we show how cookie respawning with browser fingerprinting can overcome browsers preventions.

Via *persistent tracking with respawned cookies*, domains can track users across websites without third-party cookies. Consider the following scenario: example.com and news.com include a fingerprinting script from tracker.com. When the user visits these websites, the script from tracker.com accesses the user's browser and machine features, and sets a corresponding first-party cookie. As a result, two first-party cookies are set in the user's browser and labeled with two different hosts: example.com and

news.com, but the values of these two cookies are identical, because they are created from the user's browser and machine features. By respawning these two cookies on both websites, the owner tracker.com shows to be able to track the user in a persistent way across sites with a first-party cookie only.

We analyzed the usage of the same (owner, key, value) first-party respawned cookie across different websites. The $1,425$ cookies correspond to 1,244 respawned (owner, key, value) instances, out of which 40 (3.21%) are respawned on multiple websites in a first party context with the same value (see Table 8.4 in Appendix). wpbeaver-builder.com [239] is the top owner setting identical first party respawned cookies across websites. It respawned the same cookie on 15 distinct websites. It defines itself as a WordPress page builder. Its policy declare to collect user's information, but it does not precise the type of this information.

Summary. Cookie respawning with browser fingerprinting enable tracking across websites even when third party cookies are deprecated. We found 40 first party cookies that can serve for cross-site tracking.

4 Is respawning legal?

In this section, with a legal expert which is a co-author, we evaluate the legal compliance of $1,425$ respawned cookies and reflect upon the applicability of current regulations in practice. Our legal analysis is based on the General Data Protection Regulation (GDPR) [218] and the ePrivacy Directive (ePD) [99], as well as in its recitals (which help legal interpretation of provisions in a specific context, but they are not mandatory for compliance). The GDPR applies to the processing of personal data [104] and requires that companies need to choose a legal basis to lawfully process personal data (Article 6(1)(a)). The ePD provides *supplementary* rules to the GDPR in particular in the electronic communication sector, such as websites, and requires those, whether inside or outside the EU, to obtain *consent* from users located in the EU for processing of their personal data. We have additionally consulted the guidelines of both the European Data Protection Board (an EU advisory board on data protection, representing the regulators of each EU member state) [17] and the European Data Protection Supervisor (EDPS, the EU's independent data protection institution) [93]. While these guidelines are not enforceable, they are part of the EU framework for data protection which we apply in this work to discern whether respawning is compliant.

To assess the legal consequences of respawning, the legal expert analysed legal sources to interpret cookie deletion. To our surprise, we found that there is *no explicit legal interpretation of cookie deletion.* Only the EDPS [93, Section 4.3.4] noted that *"if cookies requiring consent have disappeared, this is most probably because the user deleted them and wanted to withdraw consent".* As a result, *cookie respawning also does not have a clear legal interpretation* and merits attention for its plausible legal consequences. These consequences can arguably be derived, not only from the consent perspective,

but also from the core principles of data protection, as discussed in the following sections (fairness, transparency and lawfulness principles). Thus, owners of respawned cookies and website owners that embed those may be jointly responsible for their usage (Article 26 [218]) and may then be subject to fines of up to 20 million EUR (or 4% of the total worldwide annual turnover of the preceding financial year, Article 83(5)[218]).

4.1 *Fairness Principle.*

This principle requires personal data to be processed fairly (Article 5(1)(a)). It requires that i) *legitimate expectations of users are respected* at the time and context of data collection, and ii) there are no "surprising effects" or potential negative consequences occurring in the processing of user's data.
Findings: We consider that *all* $1,425$ *respawned cookies plausibly violate the fairness principle*, as respawning seems to be inconsistent with the user's expectations regarding respawned cookies after its deletion from her browser, and also considering the cookie's duration.
Suggestions for policymakers: It's hard to operationalize the high-level fairness principle into concrete requirements for website owners and map it into legitimate expectations of users. Policy makers need to provide more concrete guidelines on the operationalization of this principle in the Web.

4.2 *Transparency principle.*

Personal data processing must be handled in a transparent manner in relation to the user (Article 5(1)(a)). This principle presents certain obligations for websites: i) inform about the *scope and consequences* [222] and the *risks* in relation to the processing of personal data (Recital 39); ii) inform about the purposes, legal basis, etc. before processing starts (as listed in Art. 13); iii) provide the above information in a concise, transparent, intelligible and easily accessible form (Art. 12).
Findings: We analyzed the privacy policies of the 10 top popular respawned cookie owners: *rubiconproject.com [194], casalemedia.com [66], pubmatic.com [183], adobedtm.com [27] smartadserver.com [207], bizible.com [56], betweendigital.com [54], maricopa.gov [154], wpbeaverbuilder.com [239], and contextweb.com* (Figure 5.5). Some policies [194, 66, 27, 207, 56] refer to the use of browser's features without referencing the consequences or risks thereof. Also, none of the policies refer to cookie respawning. As such, these seem to be in breach of the transparency principle.
Suggestions for policymakers: In practice, the description of data (purposes, legal basis, types of personal data collected, features used and its consequences) is often mixed within the text, which makes harder to extract concrete information therefrom [116]. Policy-makers need to converge on harmonized requirements and standard format for privacy policies.

4.3 *Lawfulness Principle.*

The ePD requires websites to obtain user consent to lawfully process personal data using cookies. When a cookie recreates itself without consent, every data processed henceforth could be considered unlawful due to lack of legal basis for personal data processing [103]. This practice incurs in violation with the lawfulness principle (Articles 5(1)(a) and 6(1) of the GDPR, and 5(3) of the ePD). The EDPS [93] already advised against the use of cookie respawning if the processing relies on users' consent. It mentions that *"cookie respawning would circumvent the user's will. In this case (...) institutions must collect again user's consent"*.

To evaluate compliance with the lawfulness principle, we need first to evaluate whether cookies are exempted or subject to consent. The 29WP [41] asserts that *"it is the purpose that must be used to determine whether or not a cookie can be exempted from consent*".

Given that only a small percentage of cookies include a description of their purposes [116], we adopted the Cookiepedia open database [78] which has over 11 million cookies used across 300,000 websites and has been used in prior work [225]. It uses the classification system developed by "The UK International Chamber of Commerce" (ICC) and relies on four common purposes of cookies: i) Strictly Necessary (which includes authentication and user-security); ii) Performance (also known as analytics, statistics or measurement cookies); iii) Functionality (includes customization, multimedia content); and iv) Targeting (known as advertising). Even though this classification is not binding, we point to the fact that it is the largest database of pre-categorized cookies.

The 29WP [41] adds two other characteristics contributing to determine whether cookies are exempted or subject to consent: duration (*session and persistent cookies*) and context (*first and third-party cookies*). Building on the analysis made by Santos et al. [201, Table 5] on the list of *purposes* that are subject to consent and those that are exempted therefrom, we firstly studied the Cookiepedia purposes and then we derived which are the purposes subject to consent according to their duration and context. Table 5.8 summarizes the Cookiepedia purposes requiring consent depending on the duration and the context on which it is running.

Findings: In our study we crawled websites and even if a website provided a consent banner, we did not give consent thereto. We evaluated whether respawned cookies are subject to or exempted from consent (as described in Table 5.8). As a result of our evaluation, we found that out of 336 respawned cookies categorized by Cookiepedia, 130 (38.69%) are subject to consent. Hence, these 130 cookies are in breach of the lawfulness principle.

Suggestions for policymakers: Companies can embed respawning and still claim respawned cookies are exempted of consent. We analysed that both the duration and context of cookies contribute to determine whether cookies are exempted or subject to consent. However, from a technical point of view, these criteria can be bypassed by do-

	Session	Persistent
First-party	Targeting/ Advertising	Targeting/ Advertising
Third-party	Targeting/ Advertising Performance Strictly necessary	Targeting/ Advertising Performance Strictly necessary Functionality

Table 5.8: **Purposes of Cookiepedia [78] that require consent according to their context and duration.**

mains that embed respawning. As per *duration*, session cookies can get recreated even after their elimination by the user. Functionality cookies are exempted of consent when used as session cookies and are subject to consent when used in a persistent way [41]. When respawned, such cookies can be used for a longer duration than previously envisaged. We found that out of $1,425$ respawned cookies, 446 (31.30%) are session cookies. Regarding *context*, performance cookies are exempted of consent when used in a first party context and are subject to consent when used as third party cookies. However, in practice, a cookie set in the first party context can be considered as a third party cookie in a context of a different website. We found that 4 respawned cookies (host,key,value) appear as first- and third-party in different websites. These cookies are respectively set by pornhub.com, mheducation.com, hujiang.com and fandom.com. Given that a cookie context and duration can be altered, these should not be used as a criteria to evaluate the need of consent.

5 Conclusion

This work presents a large scale study of cookie respawning with browser fingerprinting, a tracking technique that is devoid of a clear legal interpretation in the EU legal framework. We employed a novel methodology to reveal the prevalence of cookie respawning with browser fingerprinting in the wild. The detection of such behavior and the identification of responsible domains can prove to be hard to achieve, which impacts both the ability to block such behavior, and its legal assessment. We believe this work can serve as a foundation for improvement of future regulation and protection mechanisms.

Chapter 6

Compliance with the Purpose Specification Principle and Subject Access Request

Preamble

This chapter is a replication of the paper titled "On Compliance of Cookie Purposes with the Purpose Specification Principle" published in the International Workshop on Privacy Engineering (IWPE 2020) [117], and the paper titled "Security Analysis of Subject Access Request Procedures" published in the Annual Privacy Forum (APF 2019) [58] .

1 Introduction

Auditing legal compliance of websites within the EU Data Protection legal framework is of paramount importance. *Data Protection Authorities* (DPAs) are interested in making auditing as precise and scalable as possible to enable regulatory enforcement, and to react towards the expansion of complaints received since the General Data Protection Regulation (GDPR) [218] came into force in May 2018. *Data Protection Officers* (DPOs), who oversee and evaluate the overall compliance of the companies' websites, are also concerned in making the auditing scalable to ensure compliance.

While analysing the cookies present on a website, an auditor needs to capture the *purpose* of each cookie. This defined purpose can then help to determine whether processing is legally compliant, what safeguards the GDPR imposes, and which legal basis can be used. Ultimately, it is the *purpose* and the processing that must be used to determine whether or not a cookie can be exempted from consent [118]. Finally, only when it's declared which cookies require consent, one can verify whether a website is setting such cookies before any action of the user, and whether a cookie banner is compliant with the GDPR and with the ePrivacy Directive (ePD) [99, 41].

DPAs advocate that *all* cookies should – as a best practice – declare their purpose. The UK, Greek, Finnish and Belgian DPAs [224, 109, 125, 53] endorse as a good practice disclosure of clear information about the purposes of cookies, including strictly neces-

sary ones. The guidance of the 29 Working Party (29WP) [42] notes that although some cookies may be exempted from consent, they are part of a data processing operation, therefore publishers still have to comply with the obligation to inform users about the usage of cookies prior to their setting.

In practice, we observe that some websites describe the purposes of cookies in the corresponding *privacy policies* (or in *cookie policies*). *But how are such purposes supposed to be defined?* Article 5(1)(b) of the GDPR and the 29WP [40] elaborate on the *"Purpose Limitation"* principle. This principle mandates personal data to be collected (1) for specified, explicit and legitimate purposes only and (2) not further processed in a way incompatible with those purposes. In this work, we focus on the first component of this principle named *purpose specification*.

We first analyse the legal requirements of the purpose specification principle, and derive how cookie purposes should be described. With the aim of *automatic auditing of websites at scale* to ensure compliance, we then perform a large scale crawling: we collect 20,218 third-party cookies from 84,658 pages of the top 10,000 domains. Thereupon we search for cookie policies describing these cookies and extract their purposes to evaluate how many cookie purposes satisfy the legal requirements of purpose specification.

Our first result is concerning: only 12.85% of 20,218 third-party cookies have a corresponding cookie policy where a cookie is mentioned. Our second result exposes the illusion of the legal value of cookie policies: only 5% of cookies include a description of their purposes in well-structured tables. By processing such tables with automated means, we have extracted purposes for 997 third-party cookies out of 20,218 cookies collected in our experiment.

We conclude with guidelines to DPAs, DPOs and policy-makers to enable automatic auditing of websites. We substantiate that policy-makers should propose means to specify purposes in machine readable forms, and establish an ontology of purposes that comply with the legal requirements and reasoning under GDPR, ePD and other legal sources. For transparency and scientific purposes, we make available the dataset of 997 cookies and their purposes to the research community for further experiments [82].

2 Legal Requirements for Purposes

The following analysis on the legal requirements for purposes is based on the most authoritative legal documents in the domain of privacy and data protection law. In particular, we extract the arguments laid down in binding legal sources, such as the rulings of the Court of Justice of the EU (CJEU), and the legal rules laid down in legal provisions of the GDPR and the ePrivacy Directive (ePD). For a complementary analysis we resort to the non-binding guidelines by Data Protection Authorities (DPAs), 29WP and OECD.

Availability The 29WP [44] [45] recommends that organisations should publish a privacy or cookie policy on their websites, wherefrom users are able to access necessary information on the purposes of cookies being used, including the ones of third parties. From this recommendation, we derive a first requirement of *availability* stating that the purposes of cookies should be available to users. The OECD Privacy guidelines [14] and the GDPR re-enforce the *predetermination* of purposes – they specify that before, and in any case not later than at the time of data collection, it should be possible to identify the purposes for which these data are to be used. The requirement of 'availability of purposes' stems also from the *transparency principle* (Article 5(1)(a) of the GDPR, and Recital 39 thereto) [45] which mandates an obligation of data controllers to *inform* the purposes of processing to the data subject (Article 13 (1)(c) and Recitals 58 and 60 of the GDPR). The CJEU ruling on Planet 49 [101] asserts transparency obligations about cookie purposes, which also hold for third parties with whom cookies are shared.

In the following, we describe which are the legal requirements to define purposes lawfully (demanded by Article 5 (1)(b) of the GDPR and the 29WP [40]). The *purpose specification* principle focuses on the initial purpose of data collection. It identifies three criteria for describing a purpose: *explicitness, specificity, and legitimacy*. We analyse and contextualize each requirement in the context of purposes for cookies.

Explicitness The three following conditions must be met for a purpose to be explicit: i) Unambiguous: a purpose must be sufficiently unambiguous as to their meaning or intent; ii) Exposed: purposes need to be clearly expressed, revealed or explained. The 29WP [42] contends that it is not enough for information to be "available" somewhere in the website that the user visits; iii) Shared common understanding: the definition of the purposes must be understood in the same way by everybody. Criteria iii) could be measurable by user studies which are out of scope of this paper.

Specificity Purposes should be precisely identified and clearly defined. Their formulation must be detailed enough to determine what kind of processing is and is not included within the specified purpose [40]. *Violations* occur when a purpose is too vague, general or overly legalistic. The 29WP [40, 45] give such examples: "improving users' experience"; "marketing purposes", "IT-security purposes"; "future research"; "we may use your personal data to develop new services and products"; "we may use your personal data to offer personalized services".

Legitimacy Purposes should conform to a legal basis for processing and regarding cookies and tracking technologies, the eligible legal basis is consent (Article 5(3) ePD). In the context of cookies and cookie policies, this requirement of legitimacy is not directly applicable and therefore we do not study it in this paper, but scope it in our previous work [200].

Discussion on explicitness Controllers can take *"appropriate measures"* [45] for providing information in view of fair and transparent processing in a *"easily accessible"* way. As such, we claim that the positioning of cookies in a table signifies best how cookie purposes can be "clearly expressed and revealed", based on three reasons: i) Cookie purposes are hard to find inside of a text. Previous works showed privacy policies are typically long, complex documents laden with legal jargon [226, 202]. Reading privacy policies for all the websites a user visits annually would take about 244 hours/year [160]. As a result, these policies are ineffective at informing relevant information like as purposes [204, 189, 79]. ii) Auditing purposes: we interpret legal requirements in terms of usefulness for auditing and compliance automated procedures. iii) Commonly sustained and recommended practice: presentation of structured information in a table format is recurrent, even if non mandatory, either by commonly visited websites (such as Google, Wikipedia, LinkedIn), and it is also recommended by the UK DPA [224]. The Belgian, French and UK DPA websites present cookie purposes inside of tables which include, for example: name, expiry date, content and purpose of cookies. Legal scholars, as Koops, [138] underline that both controllers and end-users will benefit if purposes are consistently specified in a table, or even in a machine-readable form to avoid data controllers to hide behind vague or very abstract-level purposes or to function creep into new, unspecified ones.

3 Extraction of Cookie Purposes

Third-party cookies When a data subject visits a website, two types of cookies can be set in her browser: first and third-party cookies. *First-party* cookies are set in the user's browser by the site explicitly visited by the user or programmatically by the third party script included in the website (that however executes in the same "origin" of the visited website). When used in isolation, first-party cookies are capable to track users *only within one visited website*. *Third-party* cookies are set either (1) in the HTTP response by any third-party content (images, html files or even at the delivery of scripts [115]); or (2) included via scripts operated from a third-party "origin" (a third-party origin most often is ensured by including a third-party iframe element, that includes a third-party webpage in the content of the visited website). Third-party cookies are capable to track users *across visited websites*.

In Figure 6.1, step ❶ demonstrates a hypothetical example of a visited website, clothes.com, a third-party cookie id set by a third party tracker.com.

We study only third-party cookies and their purpose descriptions for the following reasons: i) *third-party cookies are more likely to lead to privacy violations* [98]: by tracking users across websites, third parties can recreate a part of the user's browsing history which contains personal data. ii) *third-party cookies are usually not "strictly necessary" to the user visiting a website*: these cookies are usually related to a service that is distinct from the one that has been "explicitly requested" by the user [41]. As

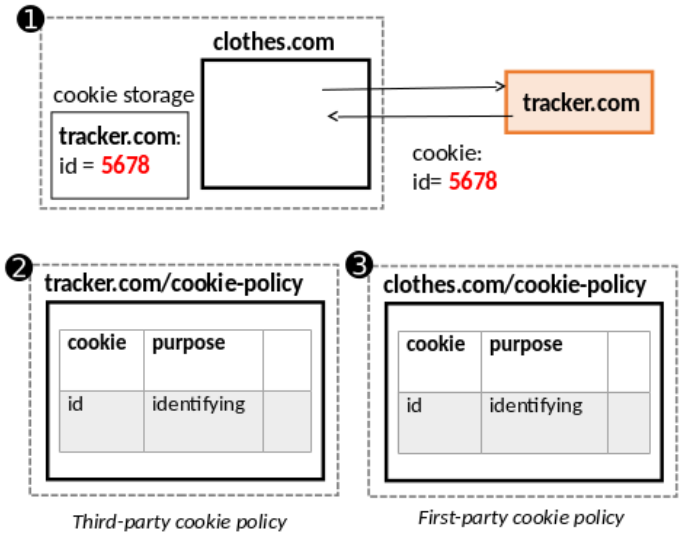


Figure 6.1: **Third-party cookies and first- and third-party cookie policies.**

a consequence, third party persistent cookies are far more likely to require the user's consent. iii) *Third-party cookies allow third parties to track the user even if she has never visited the corresponding third-party server directly.* By storing a unique identifier inside a third-party cookie, third parties are able to recognise the user without having a direct interaction with her through a third-party server.

Third party providers and legal responsibility Previous works have made large-scale measurements of the use of third-party cookies [226, 212]. However, the attribution of responsibility on the provision of information on purposes of third-party cookies was not explicitly determined yet. Since third party providers are *joint controllers* together with the first party website providers [43], Article 26 of the GDPR stipulates that *both* shall, in a transparent manner, determine their respective responsibilities for compliance with information obligations (referred to in Articles 13 and 14) which include the purposes of the processing and their legal basis. As such, we argue that *third parties are also bounded to respect the principle of transparency and list both the cookies and their purposes on their own websites.* The CJEU also established that third parties need to provide information of cookies and their purposes in their own policies [100, 101, 102].

Data collection Figure 6.2 summarizes all the steps of our data collection process. To collect third-party cookies for our experiment, we performed passive Web measurements using the Open Web Privacy Measurement (OpenWPM) platform [96]. While pretending to be a Web user, and maintaining the state of the browser, we automati-

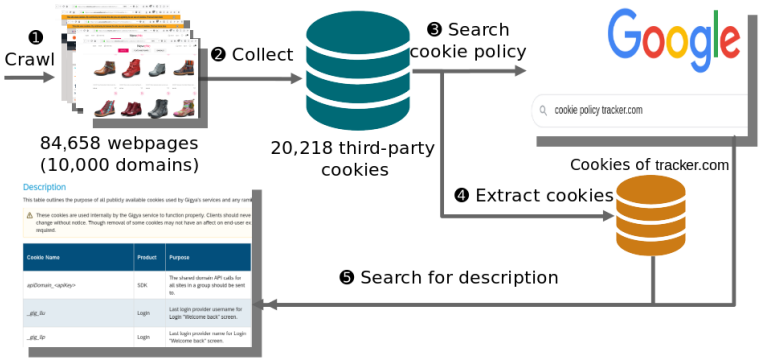


Figure 6.2: **Overview of the data collection process.**

cally visited the top $10,000$ domains according to Alexa ranking [32] in February 2019 from a server located in France (step ❶ in Figure 6.2). For each domain, we visited the home page and the first 10 links pointing to pages in the same domain, resulting in data collection from 84,658 pages. We recorded cookies set both by Javascript and via HTTP Responses (step ❷ in Figure 6.2). We consider a cookie to be *a third-party cookie* if it's set by a different domain than the visited one. By domain, we refer to the 2^{nd}-level TLD, such as google.com.

Extraction of cookie policies For each domain that sets a third-party cookie at least once, we make a Google search of the cookie policy of the domain. For example, if a third-party cookie id from Figure 6.1 is set in the user's browser by tracker.com, we will search for the cookie policy of tracker.com.

To automatize the search, we search for the domain name concatenated with "cookie policy" in the Google search engine. For example, we search for *"tracker.com cookie policy"* to get tracker.com's cookie policy (step ❸ in Figure 6.2). We then extract and store all the links L from the first page of the resulting search results. We extract the subset S of third-party cookies belonging to the same third-party domain from our crawling dataset. For example, we extract all the cookies set by tracker.com from all the pages we crawled (step ❹ in Figure 6.2). For each cookie in S, we search for the name of the cookie inside the rendered text of the extracted page of the cookie policy and save those where we found at least one mentioning of the cookie name (step ❺ in Figure 6.2).

For each third-party domain d that owns a cookie in our dataset (such as tracker.com in our example), we extract a set of cookie policy links L. We define two types of cookie policies derived from the set L:

1) *Third-party cookie policy:* for each link ℓ in the set L, we first check whether it has the same top-level domain as d or if they share the same *parents* organization. To extract the parent organization, we use the dataset built by Timothy Lib-

ert [220]. We call such link ℓ a *third-party cookie policy* because the cookie policy is directly provided by the owner of the third-party cookie. For example, step ❷ in Figure 6.1 shows the domain tracker.com that provides its third-party cookie policy tracker.com/cookie-policy with the list of cookies used by *tracker.com*.

2) *First-party cookie policy*: if no third-party cookie policy is found in the set L, we save all the cookie policy links that are hosted on other (first-party) websites. The cookie policy is hence provided by the first-party. For example, a cookie policy hosted on clothes.com/cookie-policy (see step ❸ in Figure 6.1) is a *first-party cookie policy* as it describes the cookie id set by tracker.com.

Extraction of cookie purposes To extract purposes of cookies, we analysed first and third party policies separately because they need a different treatment.

We automatized cookie purpose extraction from *third-party cookie policies* using the following approach:

1) *The cookie name appears inside of a table:* We only consider tables because its representation is machine-readable and can be adapted to large scale studies.

2) *The length of the text does not exceed 1500 characters:* We use this criteria to discard tables not used for cookies descriptions, but rather used as the webpage representation style.

3) *The length of the cookie name is bigger than 1:* Single characters can be used inside a description as propositions (examples: *I*, *A*). Hence, we discard these cookies to reduce false positives in our results.

4) *The cookie only appears once inside of a table:* When the cookie name reappears several times inside of a table, then either (1) the name of the cookie is a dictionary word in the language of the policy and so the description is not associated to a cookie, or (2) the cookie is referred in another cookie description, and in that case, we are not able to design which description defines the cookie purpose.

As to first-party cookie policies, we apply the same above approach and we further check that the domain name that set the cookie appears inside the description table as well. In fact, differently from the third-party policy directly provided by the cookie owner – where we are sure that the cookies in the description are those set by the third party– in case of first-party cookie policies, we need to check that both the domain that sets the cookie and the cookie name appear in the table. For example, in the first-party cookie policy of clothes.com in step ❸ of Figure 6.1, we search for the cookie name id together with the third party tracker.com that have set it.

Limitations To extract cookie policies, we use google queries and we analyze the links from the first page of the resulting search results in an automated way, which enables a large scale study. However, our exhaustive cookie policies extraction methodology may not return policies following different pattern. To extract cookies purposes, we search for cookie names inside of cookie policies, and then extract the correspond-

ing table row citing the cookie. When the cookie name belongs to the English dictionary, our exhaustive cookie description search algorithm may introduce some false positives. In such case, the description is using the English word and not providing a description of the giving cookie. We excluded all cookies with one character name to avoid introducing false positives. As a result, for these cookies we do not extract the cookie purpose descriptions even when they are available.

4 Evaluation of Cookie Purposes

In this section, we evaluate compliance of the extracted cookie purposes with the three requirements identified in Section 2. We explain the criteria adopted for each requirement and then we provide the analysis results. Notice that in this work we aim at *automated scalable auditing* and therefore we interpret legal requirements in terms of such auditing and compliance procedures. We thus take the position of a *website auditor*. Table 6.1 summarizes the results of this section.

Total number of cookies	20,218 (100%)
Cookies with available descriptions	2,598 (12,85%)
Cookies with explicit descriptions	997 (5%)

Table 6.1: **Proportion of cookies compliant with the availability and explicitness requirements.**

Criteria for availability We consider that a cookie is available if: i) a cookie policy exists; and ii) a cookie name is available in the cookie policy.
Results: Out of $20,218$ third-party cookies, only $2,598$ (12.85%) cookies satisfy the availability criteria: 423 of them are mentioned in a third-party cookie policy and $2,175$ of them in a first-party cookie policy. In the following, we consider all the $2,598$ cookies.

Criteria for explicitness As explained in Section 2, we suggest that a cookie purpose is explicit when described in a *structured table in the policy* because it is easier to identify the purpose for each specific cookie.
Results: Out of $2,598$ available cookies, only 997 (38.38%) cookies presented their description in an explicit way in a table (see Section 3 for details of our extraction algorithm). These 997 cookies correspond to only 5% of the total amount of $20,218$ third-party cookies we have collected demonstrating the illusion of the legal value of cookie policies.

Criteria for specificity We consider that a cookie is specific if its description provides a clear and precise information about the purpose.

Results We extracted 19,409 cookie descriptions from first- and third-party policies of the 997 cookies that have explicit purposes. Such high number derives from the fact that a single description can be repeated within first and third party policies. Out of 19,409 cookie descriptions, 6,428 are unique, however they describe 997 cookies. This situation can be caused either by: i) the diversity of languages in cookie policies and the false positives introduced by our extraction algorithm; or ii) inherent confusion in the specification of purposes. Nevertheless, we observed that some cookies have different descriptions in different policies. Table 6.2 presents the 10 most popular cookie

Row	Description	Occurrence	Specific
1	Pending Persistent HTML Local Storage	365(1.88%)	✗
2	Pending Session Pixel Tracker	267(1.38%)	✗
3	Pending Session HTTP Cookie	233(1.20%)	✗
4	Pending 1 year HTTP Cookie	220(1.13%)	✗
5	Purpose Expiry Type	216(1.11%)	✗
6	Stores the users video player preferences using embedded YouTube video Session HTML Local Storage	174(0.90%)	✓
7	Pending 1 day HTTP Cookie	156(0.80%)	✗
8	Registers anonymised user data, such as IP address, geographical location, visited websites, and what ads the user has clicked, with the purpose of optimising ad display based on the users movement on websites that use the same ad network. 1 year HTTP Cookie	125(0.64%)	✓
9	Used to present the visitor with relevant content and advertisement - The service is provided by third party advertisement hubs, which facilitate real-time bidding for advertisers. Session Pixel Tracker	108(0.56%)	✓
10	Registers a unique ID that identifies a returning users device. The ID is used for targeted ads. 1 year HTTP Cookie	107(0.55%)	✓

Table 6.2: **Top 10 cookies descriptions.** Occurrence: number of times the description is observed in a dataset of 19,409 cookie descriptions.

descriptions from a dataset of 19,409 cookie descriptions. These top-10 descriptions occur 1,971 times in our dataset, which constitutes 10% of all the descriptions. Surprisingly, the top-5 descriptions of purposes do not render any specification about the use of cookies because the only statement provided for these cookies refers to their life span (session or persistent). The description conveyed in 6 seems to refer to 'Session

Multimedia Content Player' cookies. Cookie 8 yields advertising purpose, for it refers to the collection of data with the purpose of optimizing ad display. Cookie 9 corresponds to the purpose of advertising to facilitate real-time-bidding. Cookie 10 refers also to advertising, since the data collected is used for targeted ads.

5 Recommendations and Observations

Our experimental results confirm the common conjecture that cookie purposes are not described in a legally compliant way. In this section, we provide recommendations to policy-makers on how to improve the specification of purposes for trackers per requirement.

How to improve specificity The top 5 cookie descriptions (see Table 6.2) show that purposes are rarely defined specifically. Purposes need to be pre-defined and modeled using ontologies that allow to reason about purposes inclusions, implications and generalisations. Such standardized approach would serve to minimize legal uncertainty [230]. Following our recent opinion [221] on the CNIL draft recommendation on cookies [118], *purposes should be defined in standardized taxonomies by the data protection authorities* to allow automatically reason about them.

The definition of purposes should be made with care because when users choose among many fine-grained purposes predefined in a system, they tend to opt for an open-ended "rest" category in which natural-language purpose descriptions are inserted [138].

How to improve explicitness We found that for the 2,598 cookies that have cookie policies, cookie descriptions are often mixed with other text, which makes it hard to extract them. Only 997 cookies came with descriptions in well-structured tables. Following our opinion [221] on the CNIL draft recommendation [118], we propose that *each cookie should have only one standard purpose and a legal basis applied to it.*

Such standard description of each cookie and its representation in a table enables automatic large scale auditing of trackers. The same standard can be used in the design of cookie banners requesting users' consent.

How to improve availability In Section 4 we found that only 2,598 (12.85%) out of 20,218 analysed cookies have a corresponding cookie policy wherein the cookie is mentioned. For the remaining 87.15% of cookies, no cookie policy was available. We suggest that cookie policies should be available on all websites to enable transparency of data processing purposes. We propose to use *a standard relative path on the server host, such as "/cookie-policy"* to enable its visibility. Similar self-declarative approaches are already used for websites: the declaration of access to crawlers in robots.txt file [192] and declaration of advertisers recently in ads.txt file [29].

6 Evaluation of SAR

When a data subject visits a website, she is interacting and being observed not only by the owner of the website, but also by numerous third party services included in those websites. In the recent years, researchers found that more than 90% of Alexa top 500 websites [193] contain third party tracking content, while some sites include as much as 34 distinct third party content [146].

Such third party content is often tracking users: third party tracking is the practice by which third parties recognize users across different websites as they browse the web. One of the most common and basic technology to track users is via *third-party cookies*. Such cookies, installed by the third party content when the user visits a website, usually contain a unique identifier and allows third parties to track the user across different websites, recreate part of her browsing history and collect data about her.

To examine the effectiveness of the access right set up by the GDPR in case of third party tracking services, we crawled the top $100,000$ websites according to Alexa ranking in October 2018 from a server located in France [32]. For each website, we visited the home page and other 10 webpages on the same website. Out of $100,000$ Alexa top websites, we successfully crawled 84,094 websites with a total of 829,349 webpages. We have identified the top 30 third parties that set third-party identifying cookies in the user's browser. We have then analyzed the privacy policies of these 30 third party trackers, and interacted with them via email when privacy policy page analysis was not sufficient to draw conclusions. As a result, we extracted information on the authentication procedures implemented by the third party tracking services integrated in websites, and whether it is possible to exercise the subject access rights with them based on identifiers stored in the browser.

6.1 *Evaluation criteria:*

To evaluate the data access procedure set up by third party tracking services,we considered two main criteria *authentication* and *simplicity*.

Authentication – Authenticating the user is one of the main requirements to allow the user to access her data. By using the online identifiers– that could be either a cookie in case of web access or a mobile ID in case of mobile, – third parties can uniquely identify the user. Notice that both identifiers stored in cookie or mobile ID are considered personal data according to the 29 Working Party Opinion 2/2010 on online behavioral advertising [12]. In some cases, the third parties require additional personal information , such as the name, email or even the ID document.

Simplicity – We evaluate simplicity by distinguishing how easy it is for the data subject to access her data collected by the third party trackers. Some third parties provide user-friendly access directly from the website, while for others the data subject

need to suffer from long email exchanges making the data access very difficult for the data subject.

6.2 *Results of our evaluation:*

Table 6.3 shows the results of our evaluation on the two main criteria described above. To simplify, we have grouped all the domains owned by Google (doubleclick.net, google.com, gstatic.com, youtube.com, google.fr, googlesyndication.com and 2mdn.net).

Impossible to start exercising SAR – Two companies, simpli.fi and casalemedia.com, were abusing identity check at the information extraction level. Simpli.fi refused to provide us with more information about the process unless we provide first and last name, address, phone number and email. Casalemedia.com did not explain how to exercise SAR on their website, and in order to ask a question we had to go through an online from, where we should provide additional personal data.

For four companies, teads.tv, baidu.com, innovid.com and serving-sys.com, we were not even able to start the SAR process. In their websites, teads.tv [217] and baidu.com [47] precise that data access is done upon request. We sent an email asking how we can access the third party data on December 6, 2018 and January 7, 2019 respectively but we have never received an answer as of March 18, 2019. We sent an email to innovid.com following the instruction on their website [134], but it appears that their domain isn't properly registered. Our message couldn't be delivered. The website of serving-sys.com is not accessible because of insecure connection error.

Denial of access – Three companies answered our emails within less than one month, but their answers did not help us exercise the SAR and get the third party data. Two tech giants that set identifier cookies, Google (that covers 7 distinct third party tracking domains) and facebook.com have not given us any indication on how to access the third party data. Instead, they pointed us to their documentation and how to access the data collected directly via their services as first parties. Nr-data.net owned by New relic did not ask for the cookie identifier but only told us that the email we are using to communicate with them is not linked to any data in their dataset.

Two companies, demdex.net and everesttech.net owned by Adobe also refused to provide us with the data collected from the third party context. In our experiments, we have observed that these companies use third party cookie identifiers that allow them to identify the data subject across websites. However, when we tried to exercise SAR, these companies stated that it's not possible to confirm that any information associated with the third party cookie relates to us. On a positive side, demdex.net and everesttech.net did not ask for addition personal information, but they didn't grant us access to the third party data. According to them, their practice is in line with GDPR, they quoted:

Third-party domain	Authentication					Simplicity	
	Online identifier		Other data			Direct access	email
	Cookies	Mobile ID	Name and surname	email	ID card		
simpli.fi [206]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
casalemedia.com [65]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
teads.tv [217]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
baidu.com [47]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
innovid.com [134]	_	_	_	_	_	_	_
serving-sys.com	_	_	_	_	_	_	_
Google domains	⊘	⊘	⊘	⊘	⊘	⊘	⊘
facebook.com [11]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
nr-data.net [170]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
demdex.net	⊘	⊘	⊘	⊘	⊘	⊘	⊘
everesttech.net	⊘	⊘	⊘	⊘	⊘	⊘	⊘
yandex.ru [241]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
openx.com [176]	⊘	⊘	⊘	⊘	⊘	⊘	⊘
pubmatic [182]	✓	✓	✓	✓	✓	✗	✓
mathtag.com [156]	✓	✓	✓	✓	✓	✗	✓
weborama.fr [234]	✓	✓	✓	✓	✓	✗	✓
criteo.com [80]	✓	✓	✓	✓	✓	✗	✓
scorecardresearch.com [205]	✓	✓	✓	✓	✗	✗	✓
adform.com [24]	✓	✓	✓	✓	✗	✗	✓
agkn.com	✓	✓	✗	✓	✗	✗	✓
smartadserver.com [207]	✓	✗	✗	✓	✗	✗	✓
adnxs.com [26]	✓	✓	✗	✗	✗	✓	✗
adsrvr.org [28]	✓	✓	✗	✗	✗	✓	✗
quantserve.com [184]	✓	✗	✗	✗	✗	✓	✗
spotxchange.com [214]	✓	✓	✗	✗	✗	✓	✗

Table 6.3: **Evaluation of the subject access right procedure of top 30 third parties**: " ⊘" means that the request is denied by the third party, while "-" means it's not technically accessible.

> *This is in line with the GDPR, which recognises both that the right to obtain a copy of personal data should not adversely affect others (art.15(4)) and that rights of access do not apply where an organisation is not able effectively to identify the data subject (art.11(2)).*

Two companies, yandex.ru and openx.com refused to process our request as well. These companies claim that they act as data processors on behalf of its publisher or developer partners. Hence, the subject access requests do not apply to them and they suggest us to contact the data controllers. Notice that such interpretation is not acceptable by the recent work of Mahieu et al. [153] and the CJEU decision of Wirtschaftsakademie [15] who state that both publishers and third parties are joint data controllers.

Abusive identity check – Third party domains are able to recognize the user across websites with a unique identifier, which we detected to be stored in the third party cookies. Such unique identifier is not related to the user's other personal information such as name or email. Therefore, any proof of user's name (such as the identity card) or email is not useful *to prove the ownership of the cookies.*

During our evaluation, we noticed that eight companies ask to provide not only the online identifier but other personal information as well. This practice allows the third parties to link the data subject's online identifier to her personal information. Therefore, a data subject is forced to reveal even more personal data to the third party in order to practice her access right. This results in an *abusive identity check.*

Eight companies, pubmatic.com, smartadserver.com, mathtag.com, scorecardresearch.com, agkn.com, weborama.fr, adform.com and criteo.com require additional information to authenticate the user such as the full name or even the ID document. In addition to the subject's ID document, pubamtic.com asks for the name and the ID document of a witness who signs the SAR form together with the data subject. Five out of eight companies (pubmatic.com, mathtag.com, adform.com, weborama.fr and criteo.com) ask the user to fill a form, print and sign it in order to validate that she is the owner of the online identifier and of the device associated to it. Interestingly, adform.com uses this form to acknowledge the user that the company will process the additional personal data provided in the signed form (such as signature and full name) and retain it for up to 10 years! To access her data, the user has no choice except to agree and sign this form.

Direct access without requesting additional data – Four companies, adnxs.com, adsrvr.com, quantserve.com and spotxchange.com provide direct access to third party data based on the data subject's third party cookie. To verify the identity of the user and prove the ownership of the cookie, adnxs.com and adsrvr.com add a verification step where the user confirms in an online form that she is the owner of the identifier.

7 Conclusion

In this chapter, we assessed the scope of the principle of purpose specification and analysed whether it is respected in case of web browser cookies and their cookie policies. We found out that 95% of cookies do not have an explicitly declared purpose and hence are impossible to audit for compliance. The identified issues are rooted in the fact that data controllers have no explicit obligations to describe cookie purposes in a well-defined form. Policy-makers need to converge on harmonized requirements regarding the definition of purposes for cookies and other tracking technologies in the line with the 29WP guidelines [40]. DPAs and Standard Committees should standardize types of purposes for different contexts – this would minimize legal uncertainty, and reduce a case-by-case examination.

We have also analyzed security aspects of the authentication procedures set up for subject access requests. We have discovered several issues: abusive identity checks, potential data breach or denial of access. These issues are the results of incorrect procedures or a lack of means. Data controllers need to enforce the proportionality principle when they authenticate the requests to avoid abusive identity checks.

Chapter 7

Conclusion

In 1989, when Tim Berners-Lee built the first web prototype on how to link information, he did not only connect researchers at CERN, with the help of other researchers, but he connected the world. Since then, the web never stopped evolving. With time it became richer and more dynamic. As a result, today the web became part of our daily life. To meet user's needs and to bring new functionalities, new technologies such as cookies were constantly added to the web, which opened the door to advanced tracking mechanisms.

Web tracking is rooted to the birth of the web, and it evolved with it. It is today the core of one of the most important ecosystems. Therefore, one can not expect to fix web tracking at breakneck speed. As long as the web, and thus, tracking techniques are evolving, new studies on detection and measurement of tracking will be always needed.

As we observed in this thesis, we have two main web tracking tracking techniques: statefull and stateless. These techniques can be used either to perform within-site tracking (allowing to track repeat visits), or cross-site tracking (allowing to recreate part of the user's history). The statefull tracking technique relies on storing an identifier on the user's browser that is then used to track her, while the stateless tracking technique relies on the user's browser and machine features to track the user. However, in practice, and as described throughout the thesis, trackers are deploying complex tracking behaviors, and often combining both techniques. Each update of the specifications related to the web, or regulations attempting to protect user's data from being illegally collected keep impacting the tracking deployed on the web, and result into the emergence of new tracking mechanisms. For instance, when the SOP was introduced, and limited the access to cookies, trackers deployed the cookie synchronization mechanism to exchange their identifiers stored in cookies.

Along with the technical efforts to prevent tracking, several regulations applied to online privacy ecosystem came forth. The GDPR and the ePrivacy directive are serious attempts to protect user's personal data. These regulations defined a number of rights for data subjects, and obligations for data controllers. However, these regulations often do not provide prescriptive requirements on how the data subject should exercise her rights, or how the data controller should implement regulatory requirements. This lack of concrete operational description undermines the practical effect of the GDPR and

ePrivacy directive. This is often the result of the breach between the legal and computer science communities. Therefore, we need interdisciplinary works that connect both communities, and hence provide recommendations and help the DPAs to enforce the current regulation.

The focus of this thesis is to help detecting tracking techniques, evaluate legal compliance, and provide recommendations to DPAs and policy makers. First, we designed a tracking detection methodology of the most prominent and basic tracking techniques, namely cookie based tracking techniques. Such study enabled us to evaluate cookie based tracking prevalence on the top visited websites, which allowed us to uncover different relationships between domains. The redirection process and the different behaviors that a domain can adopt are an evidence of the complexity of these relationships. We showed that even the most popular consumer protection lists and browser extensions fail to detect these complex behaviors. Therefore, behavior-based tracking detection should be more widely adopted.

Next, we assessed the prevalence of tracking techniques on health related websites, that are classified as sensitive websites. Such study allowed us to not only understand the prominence of such practices on these sensitive websites, but also to point out different legal violations on the studied health related websites. We have gleaned robust evidence of tracking technologies deployed on health-related websites (before user consent interaction, and also after accepting and rejecting).

Then, we studied how trackers can take advantage of the combination of both statefull and stateless tracking techniques. We designed a robust methodology to identify the dependency between cookies and browser and machine fingerprinting features. Such methodology allowed us to make the first study of cookie respawning with browser fingerprinting. We showed that this tracking technique lacks legal interpretation and its use, in practice, violates the GDPR and the ePrivacy Directive.

Finally, we analysed the legal requirements of the purpose specification principle, and derived how cookie purposes should be described to enable automatic auditing of websites at scale to ensure compliance. We found out that 95% of the studied cookies do not have explicitly declared purposes and hence are impossible to audit for compliance. Furthermore, we analyzed the authentication practices implemented in third-party tracking services to exercise the access right (SAR). We observed that some data controllers use unsafe or doubtful procedures to authenticate data subjects.

Future work

There is substantial research to be done, and in the following, we present examples of potential future work, perspectives and recommendations to improve transparency and legal compliance in the web.

Advanced auditing of websites

The methodology that we designed to detect the dependency between browser and machine features and cookies with a high certainty (see Figure 5.2), can then be deployed for advanced website auditing. Such methodology can help to evaluate websites transparency in two aspects:

- **Declaration of collected data:** Websites, as part of their accountability and transparency obligations [218], need to declare what data they do collect about the user. Our work could help to ensure transparency as follows. First, extracting the collected data used to generate cookies, by using our methodology described in Figure 5.2. Then, comparing what is detected as collected data, with what is actually mentioned in the privacy policy. This method can evaluate whether websites are transparent about the data that they collect.
- **Abusive data collection:** The minimization principle mandates that only adequate, relevant and proportional data should be processed according to declared purposes (Article 5(1) c) [218]. Our work helps to evaluate whether websites are compliant with this principle, or whether they are performing abusive data collection as follows. First, by extracting the data used to generate the cookie using our methodology. Then evaluating the need of such data regarding the declared cookie purpose.

Identifiers respawned with a higher number of browser and machine features

In our work, we evaluated the cookie respawning with browser features (Chapter 5). However, related works showed that other browser storages are deployed to store the user's identifier [19]. As of today, these storages were less likely deployed for tracking purposes compared to cookies. However, the deprecation of cookies [199] can have an impact on the usage of such storages. To evaluate such practice, our study on cookie respawning with browser fingerprinting can be extended to the other storages used to save the user identifier such as the local storage.

Moreover, different features contain different amount of information and hence contribute differently to uniqueness of each user. Entropy can be used to evaluate how unique is a fingerprint. The higher the entropy, the more unique is the fingerprint. The study can be performed with a higher number of browser and machine features, which will help along with the calculation of entropy to evaluate the uniqueness of the respawned cookies.

Perspectives and recommendations to improve transparency in the web

Web tracking is intrusive for users privacy. Moreover, privacy implications of web tracking became more important when it is not transparent and happens without the user's consent. Studies on detection and measurement of web tracking are much needed

to shed the light on tracking organizations and the practices they are performing. These studies can help to detect the trackers, and thus help blocking them. However, *looking at a distant future, can we imagine a world without web tracking? Would the web be as useful if the advertising ecosystem vanishes? Would users be willing to pay for services to protect their privacy?* Despite privacy implications of web tracking, one can not ignore the fact that this tracking ecosystem has a main role in maintaining the usability of the web, which makes me strongly believe that the ultimate goal is not to block trackers, but to bring transparency, and give a meaningful choice to the user to ensure that if tracking is happening, then the user is aware of it, of its implications and freely gives consent to it.

Transparency can not be preserved without strong regulations in place, and clear recommendations on how to comply with it. Moreover, transparency will still be hard to achieve as long as the web is not designed in a way that ensures this transparency, and as long as we have a gap between the legal and computer science communities. To achieve the transparency, researchers in computer science should design joint recommendations, and regulators should account these throughout their legal and policy making initiatives.

To be transparent,purposes for processing should be made available, explicit, legitimate and specific [221].

Through the thesis, we proposed recommendations that can help automatic auditing of websites and that can bring more transparency to the web. In the following, we summarize the recommendations to browser vendors and policy-makers on how to improve the transparency in the web.

- **How to improve availability?** We suggest that cookie policies should be available on all websites to enable transparency of data processing purposes. We propose to use *a standard relative path on the server host, such as "/cookie-policy"* to enable its visibility. Similar self-declarative approaches are already used for websites: the declaration of access to crawlers in robots.txt file [192], and declaration of advertisers recently in ads.txt file [29].
- **How to improve explicitness?** Websites should explicitly describe the data they collect about the user, and the purpose of this data collection. We suggest to improve explicitness at the level of data collection and purposes of processing as follows.
  - *Collected data*: Websites need to precise the set of data used with each cookie.
  - *Processing purposes:* We propose that each cookie should have only one standard purpose and a legal basis applied to it.

  The description of the data used with each cookie and its purpose in a table will enable an automatic large scale auditing of trackers.
- **How to improve specificity?** Purpose of processing user's data should be clearly expressed, unambiguous and understood in the same way by everybody. Therefore, we suggest that the purposes should be predefined and standardized.

We further suggest to add standard logos with the description of collected data. Such practice will ease the understanding of the category of data collected by the websites and help map it with the purpose of processing.

To conclude, web tracking is continuously evolving, and to protect user's data, joint efforts between both data protection law and computer science communities are needed.

We hope that this work on legal and technical knowledge in online tracking can help to shed the light on multiple tracking practices, and bring more transparency to the web. We also hope that our work can serve as a foundation for improvement of the incoming ePrivacy regulation draft, currently under negotiation, and for other policy making endeavours from practically-minded parties (regulators, privacy NGOs, researchers, and computer scientists).

Chapter 8

Appendix

Appendix A

Detecting identifier sharing

GA sharing: Google-analytics serves invisible pixels on 69.89% of crawled domains as we show in Figure 8.1. By analyzing our data, we detect that the cookie set by google-analytics script is of the form GAX.Y.Z.C, while the *identifier cookies* sent in the parameter value to google-analytics is actually Z.C. This case is not detected by the previous cookie syncing detection techniques for two reasons. First, "." is not considered as a delimiter. Second, even if it was considered as a delimiter, it would create a set of values {GAX, Y, Z, C} which are still different than the real value Z.C used as an identifier by google-analytics.
Base64 sharing: When a domain wants to share its *identifier cookie* with doubleclick.net, it should encode it in base64 before sending it [5]. For example, when adnxs.com sends a request to doubleclick.net, it includes a random string into a URL parameter. This string is the base64 encoding of the value of the cookie set by adnxs.com in the user's browser.
Encrypted sharing: When doubleclick.net wants to share its *identifier cookie* with some other domain, it encrypts the cookie before sending it, which makes the detection of the identifier cookie sharing impossible. Instead, we rely on the semantic defined by doubleclick to share this identifier [5].

Assume that doubleclick.net is willing to share an identifier cookie with adnxs.com. To do so, Doubleclick requires that the content of adnxs.com includes an image tag, pointing to a URL that contains doubleclick.net as destination and a parameter *google_nid*. Using the value of the *google_nid* parameter, doubleclick.net get to know that adnxs.com was the initiator of this request. Upon receiving such request, doubleclick.net sends a redirection response pointing to a URL that contains adnxs.com as destination with encrypted doubleclick.net's cookies in the parameters. When the browser receives this response, it redirects to adnxs.com, who now receives encrypted doubleclick.net's cookie.

We detect such behavior by detecting requests to doubleclick.net with *google_nid* parameter and analysing the following redirection. If we notice that the redirection is

set to a concrete domain, for example adnxs.com, we conclude that doubleclick.net has shared its cookie with this domain.

Additional results

Figure 8.1 represents the Top 20 domains involved in invisible pixels inclusion in the 8,744 domains.

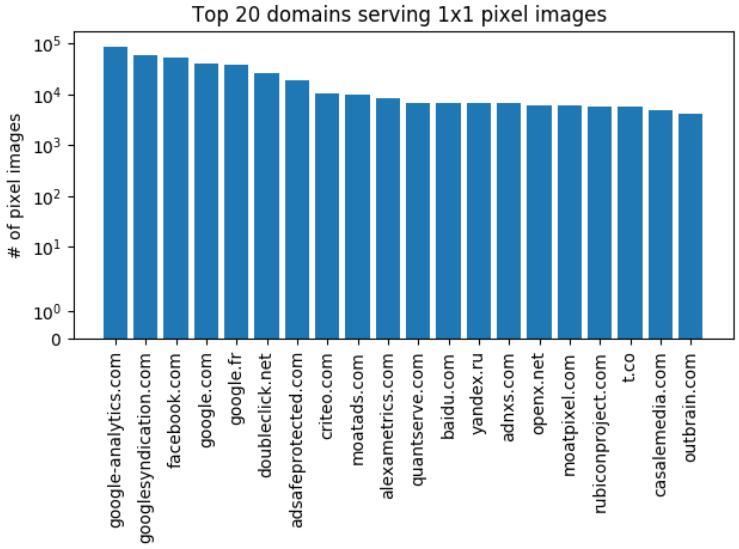


Figure 8.1: **Top 20 domains responsible for serving invisible pixels.**

Table 8.1 presents the top 10 domains using their cookie key to store the identifier.

Host	# cookies instances
lpsnmedia.net	583
i-mobile.co.jp	223
rubiconproject.com	83
justpremium.com	72
juicyads.com	64
kinoafisha.info	64
aktualne.cz	63
maximonline.ru	61
sexad.net	47
russian7.ru	45

Table 8.1: **Top 10 domains storing the identifier as key.**

Figure 8.2 represents the Top 15 third parties receiving the identifier cookies. Google-analytics is the top domain receiving identifiers in over 4% of the visited websites. Table 3.4 presents the top 10 third parties sharing their identifiers with google-analytics.

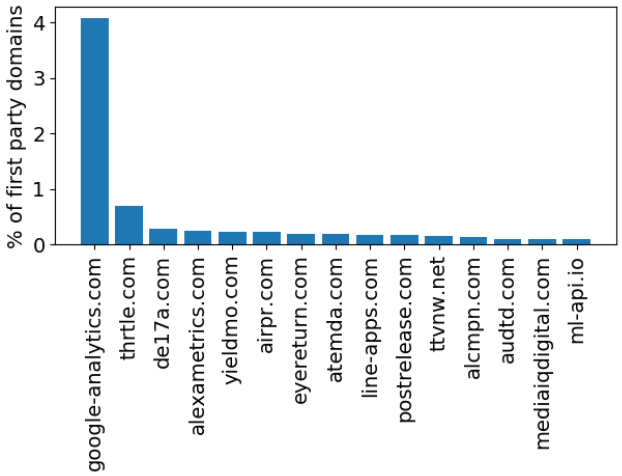


Figure 8.2: **Third party cookie forwarding**: Top 15 receivers in $8,744$ domains.

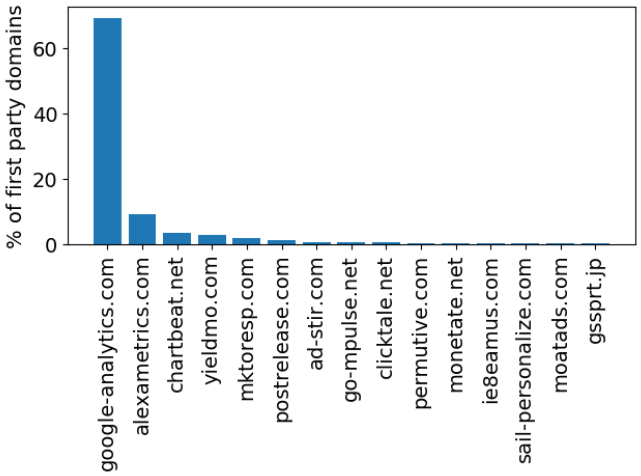


Figure 8.3: **Analytics: Top 15 receivers in the** $8,744$ **domains.**

Figure 8.3 presents the top 15 analytics domains in our dataset of 8,744 domains.

Table 8.2 presents the top 15 domains detected as trackers and missed by the filter lists. For each domain, we extract its category, owners and country of registration

Appendix B

Machines characteristics

Table 8.3 presents the characteristics of machine A and machine B used in our study.

Additional results

Table 8.4 presents the top first-party cookies respawned across websites. This practice is studied in Section 3.5.2.

Tracking enabled by a first party cookie						
Full domain	Prevalence of tracking in first-parties	Cookie name	Cookie expiration	Category	Company	Cou - ntry
code.jquery.com	756 (8.65 %)	__cfduid	1 years	Technology /Internet	jQuery Foundation	US
s3.amazonaws.com	412 (4.71 %)	s_fid	5 years	Content Servers	Amazon	US
ampcid.google.com	282 (3.23 %)	NID	6 months	Search Engines	Google LLC	US
cse.google.com	307 (3.51 %)	NID	1 year	Search Engines	Google LLC	US
use.fontawesome.com	221 (2.53 %)	__stripe_mid	1 years	Technology /Internet	WhoisGuard Protected	_
siteintercept. qualtrics.com	99 (1.13 %)	t_uid	100 years	Business /Economy	Qualtrics, LLC	US
push.zhanzhang .baidu.com	98 (1.12 %)	BAIDUID	68 years	Search Engines	Beijing Baidu Netcom Science Technology Co., Ltd.	CN
Tracking enabled in a third party context						
assets.adobedtm.com	427 (4.88 %)	_gd_visitor	20 years	Technology /Internet	Adobe Inc.	US
yastatic.net	303 (3.47 %)	cto_lwid	1 year	Technology /Internet	Yandex N.V.	RU
s.sspqns.com	278 (3.18 %)	tuuid	6 months	Web Ads/Analytics	HI-MEDIA	FR
tags.tiqcdn.com	276 (3.16 %)	utag_main	1 year	Content Servers	Tealium Inc	US
cdnjs.cloudflare.com	206 (2.36 %)	__cfduid	1 year	Content Servers	Cloudflare	US
static.quantcast.mgr. consensu.org	157 (1.80 %)	_cmpQc -3pChkKey	Session	Consent frameworks	IAB Europe	BE
a.twiago.com	133 (1.52 %)	deuxesse_uxid	1 month	Office/Business Applications	REDACTED FOR PRIVACY	_
g.alicdn.com	121 (1.38 %)	_uab_collina	10 years	Content Servers	Alibaba Cloud Computing Ltd.	CN

Table 8.2: **Top 15 domains missed by EL&EP and Disconnect but detected by BehaviorTrack to perform tracking.**

Characteristics	Machine A	Machine B
Date of the crawl	March 2021	March 2021
OS	Fedora 25	Fedora 31
Firefox version	68.0	45.0.1
Location	France	France
IP address	193.51.X.X	138.96.Y.Y
OpenWPM version	v0.9.0	v0.7.0
Language	English (en_US)	German (de_DE)
Time zone	CET	AKST
Geolocation	France	Alaska
Do not track	Null	True

Table 8.3: **Crawls Characteristics.** *All crawls were performed from machine A except user specific crawl that was done from machine B.*

Owner	Occurrence
wpbeaverbuilder.com	15
clarip.com	13
maricopa.gov	9
google-analytics.com	7

Table 8.4: Top first-party cookies respawned across websites. Every line in the table represents a cookie, hence the same owner can appear on multiple lines. Occurrence: presents the number of websites where the instance (owner,key,value) was respawned.

Bibliography

[1] Affection longue duree. https://www.ameli.fr/assure/droits-demarches/maladie-accident-hospitalisation/affection-longue-duree-ald/affection-longue-duree-ald.

[2] Alexa top sites. https://www.alexa.com/topsites.

[3] Alexa websites visited. https://www.dropbox.com/sh/nwjw7ggcx08o1x7/AACYrHqsxo7DcZjbVArE5Fxua?dl=0.

[4] Contextual identities. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/contextualIdentities.

[5] Cookie guide. https://developers.google.com/authorized-buyers/rtb/cookie-guide.

[6] Enahnced tracking protection in firefox for desktop. https://support.mozilla.org/en-US/kb/enhanced-tracking-protection-firefox-desktop.

[7] Google cookie types. https://policies.google.com/technologies/types.

[8] Googletakeout. https://takeout.google.com/?utm_source=pp&hl=en, accessed on 28 September 2018.

[9] List of websites visited. https://www.dropbox.com/sh/96pcfgj1qruow90/AABzQmH3CCLCMDYOBHaKxeG9a?dl=0.

[10] Pupeteer. https://github.com/puppeteer/puppeteer.

[11] Yourfacebookinformation. https://www.facebook.com/full_data_use_policy, accessed on 28 September 2018.

[12] Opinion 2/2010 on online behavioural advertising. Technical Report 171, 2010.

[13] WP29 Opinion 04/2012 on the Cookie Consent Exemption - ARTICLE 29 DATA PROTECTION WORKING PARTY, 2012. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2012/wp194_en.pdf.

[14] OECD guidelines on the protection of privacy and transborder flows of personal data, 2013.

[15] Case C-210/16 Wirtschaftsakademie Schleswig-Holstein. 2018. ECLI:EU:C:2018:388, http://curia.europa.eu/juris/document/document.jsf?docid=202543&doclang=EN.

[16] Judgment in Case C-673/17 Bundesverband der Verbraucherzentralen und Verbraucherverbände – Verbraucherzentrale Bundesverband eV v Planet49 GmbH, 2019. http://curia.europa.eu/juris/documents.jsf?num=C-673/17.

[17] 29 Working Party. https://edpb.europa.eu/our-work-tools/article-29-working-party_en.

[18] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juárez, Arvind Narayanan, and Claudia Díaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689, 2014.

[19] Gunes Acar, Marc Juárez, Nick Nikiforakis, Claudia Díaz, Seda F. Gürses, Frank Piessens, and Bart Preneel. Fpdetective: dusting the web for fingerprinters. In *2013 ACM SIGSAC Conference on Computer and Communications Security (CCS'13)*, pages 1129–1140, 2013.

[20] Acpm.fr website. https://www.acpm.fr/Adherer/Pourquoi-adherer-a-l-ACPM.

[21] Adblock offcial website. https://getadblock.com/.

[22] Adblock list parser. https://github.com/scrapinghub/adblockparser.

[23] Adblock Plus Official website. https://adblockplus.org/.

[24] Adform - privacy policy. https://site.adform.com/privacy-center/website-privacy/website-privacy-policy/.

[25] David Martin Adil Alsaid. Detecting web bugs with bugnosis: Privacy advocacy through education. In *Privacy Enhancing Technologies*, 2002.

[26] Adnxs – appnexus data subject rights. https://www.appnexus.com/data-subject-rights-policy.

[27] Adobedtm.com privacy policy. https://www.adobe.com/privacy/policy.html.

[28] Adsrvr. https://www.adsrvr.org/.

[29] Ads.txt specification. https://iabtechlab.com/ads-txt/.

[30] Guide on use of cookies, 2021. https://www.aepd.es/sites/default/files/2021-01/guia-cookies-en.pdf.

[31] Nasser Mohammed Al-Fannah, Wanpeng Li, and Chris J. Mitchell. Beyond cookie monster amnesia: Real world persistent online tracking. In Liqun Chen, Mark Manulis, and Steve A. Schneider, editors, *Information Security - 21st International Conference, ISC 2018, Guildford, UK, September 9-12, 2018, Proceedings*, volume 11060 of *Lecture Notes in Computer Science*, pages 481–501. Springer, 2018.

[32] Alexa official website. https://www.alexa.com/.

[33] Alexa websites. https://www.dropbox.com/sh/wnmugbzb2bfp7ca/AACUJbCbFM2iBcN7y2b-bqF-a?dl=0.

[34] Mshabab Alrizah, Sencun Zhu, Xinyu Xing, and Gang Wang. Errors, misunderstandings, and attacks: Analyzing the crowdsourcing process of ad-blocking systems. In *Proceedings of the Internet Measurement Conference*, IMC '19, page 230–244, New York, NY, USA, 2019. Association for Computing Machinery.

[35] Ameli.fr website. http://annuairesante.ameli.fr/professionnels-de-sante/recherche/fiche-detaillee-AbE1mjY2MDCw.html.

[36] Waleed Ammar, Shomir Wilson, Norman Sadeh, and Noah A. Smith. Automatic categorization of privacy policies: A pilot study. Technical report.

[37] Google-analytics service. https://developers.google.com/analytics/devguides/collection/protocol/v1/devguide.

[38] Article 3 of GDPR. https://gdpr-info.eu/art-3-gdpr/.

[39] Article 83 of GDPR. https://gdpr-info.eu/art-83-gdpr/.

[40] Article 29 Working Party. Opinion 03/2013 on purpose limitation (WP203).

[41] Article 29 Working Party. Opinion 04/2012 on Cookie Consent Exemption (WP 194).

[42] Article 29 Working Party. Opinion 16/2011 on EASA/IAB Best Practice Recommendation on Online Behavioural Advertising.

[43] Article 29 Working Party. Opinion 2/2010 on online behavioural advertising, (wp 171).

[44] Article 29 Working Party. Working Document 02/2013 providing guidance on obtaining consent for cookies', (WP208).

[45] Article 29 Working Party. Guidelines on transparency under Regulation 2016/679, (WP260), 2018.

[46] Mika D Ayenson, Dietrich James Wambach, Ashkan Soltani, Nathan Good, and Chris Jay Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. Technical report, Available at SSRN: https://ssrn.com/abstract=1898390orhttp://dx.doi.org/10.2139/ssrn.1898390, 2011.

[47] Baidu - privacy policy. http://usa.baidu.com/privacy/.

[48] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. A longitudinal analysis of the ads. txt standard. In *Proceedings of the Internet Measurement Conference*, pages 294–307. ACM, 2019.

[49] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William K. Robertson, and Christo Wilson. How tracking companies circumvented ad blockers using websockets. In *Internet Measurement Conference 2018*, pages 471–477, 2018.

[50] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In *Proceedings of the 25th USENIX Security Symposium*, 2016.

[51] Muhammad Ahmad Bashir and Christo Wilson. Diffusion of User Tracking Data in the Online Advertising Ecosystem. In *Proceedings on Privacy Enhancing Technologies (PETS 2018)*, 2018.

[52] David A. Basin, Søren Debois, and Thomas T. Hildebrandt. On purpose and by necessity: Compliance under the gdpr. In *International Conference on Financial Cryptography and Data Security*, 2018.

[53] Belgium DPA. Guidance on cookies and other tracking technologies, 2020.

[54] Betweendigital.Com privacy policy. https://betweenx.com/.

[55] Bitmedia.io privacy policy. https://bitmedia.io/cookie-policy.

[56] Bizible.com privacy policy. https://documents.marketo.com/legal/privacy.

[57] Károly Boda, Ádám Máté Földes, Gábor György Gulyás, and Sándor Imre. User tracking on the web via cross-browser fingerprinting. In *16th Nordic Conference on Secure IT Systems, NordSec 2011*, pages 31–46, 2011.

[58] Coline Boniface, Imane Fouad, Nataliia Bielova, Cédric Lauradoux, and Cristiana Santos. Security analysis of subject access request procedures - how to authenticate data subjects safely when they request for their data. In *7th Annual Privacy Forum, APF*, volume 11498 of *Lecture Notes in Computer Science*, pages 182–209. Springer, 2019.

[59] Brave. Europe's governments are failing the gdpr – brave's 2020 report on the enforcement capacity of data protection authorities. https://brave.com/wp-content/uploads/2020/04/Brave-2020-DPA-Report.pdf, 2020.

[60] Carolyn A. Brodie, Clare-Marie Karat, and John Karat. An empirical study of natural language parsing of privacy policy rules using the sparcle policy workbench. In *SOUPS*, 2006.

[61] Bundesbeauftragter für den Datenschutz und die Informationsfreiheit (BfDI, German DPA). Guidance from german authorities for telemedia providers (translation). https://deutschland.taylorwessing.com/de/documents/get/1820/guidance-from-german-authorities-for-telemedia-providers-partial-translation.PDF_show_on_screen, accessed on 2020.01.21.

[62] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. An empirical study of web cookies. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 891–901. ACM, 2016.

[63] Canvas defender. https://addons.mozilla.org/en-US/firefox/addon/no-canvas-fingerprinting/.

[64] Yinzhi Cao, Song Li, and Erik Wijmans. (cross-)browser fingerprinting via os and hardware level features. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, 26 February - 1 March, 2017*, 2017.

[65] Casalemedia - privacy policy. http://casalemedia.com/.

[66] Casalmedia privacy policy . https://sugru.com/cookies.

[67] Symantec categorization service. http://sitereview.bluecoat.com/#/.

[68] Chameleon extension. https://addons.mozilla.org/en-US/firefox/addon/chameleon-ext/.

[69] Number of chrome users. https://www.statista.com/statistics/543218/worldwide-internet-users-by-browser/.

[70] ookies : les outils pour les maîtriser. https://www.cnil.fr/fr/cookies-les-outils-pour-les-maitriser.

[71] Cnil: Violation de données de santé. https://www.cnil.fr/fr/violation-de-donnees-de-sante-la-cnil-rappelle-les-obligations-des-organismes-la-suite-dune-fuite-de.

[72] Code de la santé publique, version in effect as of february 27, 2021. https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000036515027/. Translated with DeepL https://www.deepl.com on February 27, 2021.

[73] Commission Nationale de l'Informatique et des Libertés (French DPA). French guidelines on cookies: Deliberation No 2020-091 of September 17, 2020 adopting guidelines relating to the application of article 82 of the law of January 6, 1978 amended to read and write operations in a user's terminal (in particular to "cookies and other tracers"), 2020. https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000042388179.

[74] Consent-O-Matic browser extension. matic/mdjildafknihdffpkfmmpnpoiajfjnjd.

[75] John Cook, Rishab Nithyanand, and Zubair Shafiq. Inferring tracker-advertiser relationships in the online advertising ecosystem using header bidding. *Proc. Priv. Enhancing Technol.*, 2020(1):65–82, 2020.

[76] HTTP cookie standard. https://tools.ietf.org/html/rfc6265.

[77] Cookiebot. Cookie scanner for gdpr/epr and ccpa compliance. https://www.cookiebot.com/en/cookie-scanner/.

[78] Cookiepedia Official website. https://cookiepedia.co.uk/.

[79] Lorrie Faith Cranor. Necessary but not sufficient: Standardized mechanisms for privacy notice and choice. *JTHTL*, 10:273–308, 2012.

[80] access right criteo. https://www.criteo.com/privacy/.

[81] Anupam Das, Gunes Acar, Nikita Borisov, and Amogh Pradeep. The web's sixth sense: A study of scripts accessing smartphone sensors. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1515–1532. ACM, 2018.

[82] Data. https://www.dropbox.com/sh/voi7levu2qgq9m3/AAC2SQ5iQ3Eu022BKK5HBwVla?dl=0.

[83] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies: Measuring the gdpr's impact on web privacy.

[84] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy ... now take some cookies: Measuring the GDPR's impact on web privacy. In *NDSS*, 2019.

[85] Disconnect. Disconnect. https://disconnect.me/.

[86] Disconnect Official website. https://disconnect.me/.

[87] Disconnect. disconnect-tracking-protection. https://github.com/disconnectme/disconnect-tracking-protection, accessed on 2019.07.16.

[88] Jaromir Dobias. Privacy effects of web bugs amplified by web 2.0. In *Privacy and Identity Management for Life - 6th IFIP WG 9.2, 9.6/11.7, 11.4, 11.6/PrimeLife International Summer School, Helsingborg, Sweden, August 2-6, 2010, Revised Selected Papers*, pages 244–257, 2010.

[89] Guidance note on the use of cookies and other tracking technologies, 2020. https://www.dataprotection.ie/sites/default/files/uploads/2020-04/Guidance\%20note\%20on\%20cookies\%20and\%20other\%20tracking\%20technologies.pdf.

[90] EasyList filter lists. https://easylist.to/.

[91] EasyPrivacy filter lists. https://easylist.to/easylist/easyprivacy.txt.

[92] Peter Eckersley. How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 1–18. Springer-Verlag, 2010.

[93] Guidelines on the protection of personal data processed through web services provided by EU institutions, November, 2016. https://edps.europa.eu/sites/edp/files/publication/16-11-07_guidelines_web_services_en.pdf.

[94] EFF. Privacy badger. https://privacybadger.org/.

[95] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. I never signed up for this! privacy implications of email tracking. In *Privacy Enhancing Technologies*, 2018.

[96] 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 ACM CCS*, pages 1388–1401, 2016.

[97] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of WWW 2015*, pages 289–299, 2015.

[98] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan R. Mayer, Arvind Narayanan, and Edward W. Felten. Cookies that

give you away: The surveillance implications of web tracking. In *WWW*. ACM, 2015.

[99] Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex\%3A32009L0136, accessed on 2019.10.31.

[100] European Court of Justice. Case C-40/17 Fashion ID GmbH & Co.KG v Verbraucherzentrale NRW eV, ECLI:EU:C:2019:629.

[101] European Court of Justice. Case C-673/17 Verbraucherzentrale Bundesverband v. Planet49, ecli:eu:c:2019:801.

[102] European Court of Justice. Case C-210/16 Wirtschaftsakademie Schleswig-Holstein, ECLI:EU:C:2018:388.

[103] European Data Protection Board. Opinion 15/2011 on the definition of consent (WP 187), adopted on 13 july 2011. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2011/wp187_en.pdf.

[104] European Data Protection Board. Opinion 4/2007 on the concept of personal data (WP 136), adopted on 20.06.2007. https://ec.europa.eu/justice/article-29/documentation/opinionrecommendation/files/2007/wp136_en.pdf.

[105] European Data Protection Board. Working document 02/2013 providing guidance on obtaining consent for cookies, adopted on 2 october 2013. https://www.pdpjournals.com/docs/88135.pdf.

[106] European Data Protection Board. Guidelines 05/2020 on consent, Version 1.1, adopted on 4 May 2020, 2020. https://edpb.europa.eu/sites/edpb/files/files/file1/edpb_guidelines_202005_consent_en.pdf.

[107] European Data Protection Board (EDPB). Guidelines 05/2020 on consent under regulation 2016/679, 2020.

[108] EZIGDPR. Gdpr website compliance check. https://www.ezigdpr.com/products/gdpr-website-compliance-checker.

[109] Finnish DPA. Guidelines on confidential communications, 2020.

[110] The new Firefox. Fast for good. https://www.mozilla.org/en-US/firefox/new/.

[111] Firefox Now Available with Enhanced Tracking Protection by Default Plus Updates to Facebook Container, Firefox Monitor and Lockwise . https://blog.mozilla.org/blog/2019/06/04/firefox-now-available-with-enhanced-tracking-protection-by-default/.

[112] Number of Firefox users. https://consent.yahoo.com/v2/collectConsent?sessionId=3_cc-session_85b2e8b2-05db-4bd0-8ace-208266886510.

[113] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2020, 2020. Published online: 08 May 2020, https://doi.org/10.2478/popets-2020-0038.

[114] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. volume 2020, 2020.

[115] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. In *PoPETs*, 2020. Accepted for publication.

[116] Imane Fouad, Cristiana Santos, Feras Al Kassar, Nataliia Bielova, and Stefano Calzavara. On Compliance of Cookie Purposes with the Purpose Specification Principle. In *IWPE 2020 - International Workshop on Privacy Engineering*, pages 1–8, Genova, Italy, September 2020.

[117] Imane Fouad, Cristiana Santos, Feras Al Kassar, Nataliia Bielova, and Stefano Calzavara. On Compliance of Cookie Purposes with the Purpose Specification Principle. In *2020 International Workshop on Privacy Engineering, IWPE*, 2020. https://hal.inria.fr/hal-02567022.

[118] French DPA. Recommendation on "cookies and other trackers", 2020.

[119] Ghostery. Ghostery. https://www.ghostery.com/.

[120] Ghostery Official website. https://www.ghostery.com/.

[121] Alejandro Gómez-Boix, Pierre Laperdrix, and Benoit Baudry. Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale. In *WWW2018 - TheWebConf 2018 : 27th International World Wide Web Conference*, pages 1–10, Lyon, France, April 2018.

[122] Google analytics solutions. https://www.google.com/analytics.

[123] Google.com cookie usage. https://developers.google.com/analytics/devguides/collection/analyticsjs/cookie-usage.

[124] Colin Gray, Cristiana Santos, Nataliia Bielova, Michael Toth, and Damien Clifford. Dark patterns and the legal requirements of consent banners: An interaction criticism perspective. In *ACM CHI 2021*, 2020. https://arxiv.org/abs/2009.10194.

[125] Greek DPA. Guidelines on cookies and trackers, 2020.

[126] Greek DPA (HDPA). Guidelines on Cookies and Trackers, 2020. http://www.dpa.gr/APDPXPortlets/htdocs/documentSDisplay.jsp?docid=84,221,176,170,98,24,72,223.

[127] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *USENIX Security*, 2018.

[128] Harward Business Review. What patients like — and dislike — about telemedicine. https://hbr.org/2020/12/what-patients-like-and-dislike-about-telemedicine accessed on 27 February 2021.

[129] Raymond Hill and Contributors. ublock origin. https://github.com/gorhill/uBlock/.

[130] RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1. https://tools.ietf.org/html/rfc6585.

[131] IAB. Openrtb (real-time bidding). https://www.iab.com/guidelines/real-time-bidding-rtb-project/, accessed on 2019.09.16.

[132] Muhammad Ikram, Hassan Jameel Asghar, Mohamed Ali Kaafar, Anirban Mahanti, and Balachandar Krishnamurthy. Towards seamless tracking-free web: Improved detection of trackers via one-class learning. In *Privacy Enhancing Technologies*, 2017.

[133] Information Commissioner's Office. Guidance on the use of cookies and similar technologies, 2019. https://ico.org.uk/media/for-organisations/guide-to-pecr/guidance-on-the-use-of-cookies-and-similar-technologies-1-0.pdf.

[134] access right innovid. https://www.innovid.com/privacy-policy/.

[135] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolaos Laoutaris. Tracing cross border web tracking. In *ACM Internet Measurement Conference (IMC)*, 2018.

[136] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. In *IEEE Symposium on Security & Privacy*, 2021.

[137] JS cookies. https://www.w3schools.com/js/js_cookies.asp.

[138] Bert-Jaap Koops. The (in) flexibility of techno-regulation and the case of purpose-binding. *Legisprudence*, 5(2):171–194, 2011.

[139] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)*, 14(2):8:1–8:33, 2020. https://dl.acm.org/doi/10.1145/3386040.

[140] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints. In *37th IEEE Symposium on Security and Privacy (S&P 2016)*, 2016.

[141] Tobias Lauinger, Abdelberi Chaabane, Sajjad Arshad, William Robertson, Christo Wilson, and Engin Kirda. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. In *Network and Distributed System Security Symposium, NDSS*, 2017.

[142] lefigaro.fr website. https://annuaire.lefigaro.fr/annuaire/ville/marseille-1er-arrondissement-13/endocrinologue.

[143] lefigaro.fr privacy policy. http://mentions-legales.lefigaro.fr/page/infos-cookies.

[144] W. Davis. KISSmetrics Finalizes Supercookies Settlement. http://www.mediapost.com/publications/article/191409/kissmetrics-finalizes-supercookies-settlement.html,2013..

[145] R. Singel. Online TrackingFirm Settles Suit Over Undeletable Cookies. http://www.wired.com/2010/12/zombie-cookie-settlement/.

[146] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, 2016.

[147] Letour.fr privacy policy. https://www.letour.fr/en/privacy-policy.

[148] Timothy Libert. An automated approach to auditing disclosure of third-party data collection in website privacy policies. In *WWW*, 2018.

[149] Timothy Libert, Lucas Graves, and Rasmus Kleis Nielsen. Changes in third-party content on european news websites after gdpr„ 2018. https://timlibert.me/pdf/Libert_et_al-2018-Changes_in_Third-Party_Content_on_EU_News_After_GDPR.pdf.

[150] Timothy Libert and Rasmus Kleis Nielsen. Third-party web content on eu news sites: Potential challenges and paths to privacy improvement, 2018. https://timlibert.me/pdf/Libert_Nielsen-2018-Third_Party_Content_EU_News_GDPR.pdf.

[151] LINC. Cookieviz 2: new features to observe hidden web practices. https://linc.cnil.fr/fr/cookieviz-2-new-features-observe-hidden-web-practices.

[152] Logicrdv.fr visited website. https://www.logicrdv.fr/cardiologue/13006-marseille-6eme.html.

[153] Rene Mahieu, Joris van Hoboken, and Hadi Asghari. Responsibility for data protection in a networked world – on the question of the controller, "effective and complete protection" and its application to data access rights in europe. 2019. Available at https://ssrn.com/abstract=3256743.

[154] Maricopa.gov privacy policy. https://www.maricopacountyparks.net/privacysecurity-policies/.

[155] David Martin, Hailin Wu, and Adil Alsaid. Hidden surveillance by web sites: Web bugs in contemporary use. 2003.

[156] Mathtag - privacy policy. http://www.mediamath.com/privacy-policy/#Section-11.

[157] 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 *IEEE Symposium on Security and Privacy (IEEE S&P 2020)*, 2020.

[158] McAfee categorization service. https://www.trustedsource.org/.

[159] Description of McAfee categories. https://www.trustedsource.org/download/ts_wd_reference_guide.pdf.

[160] Aleecia M. McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *Journal of Law and Policy for the Information Society*, 2009.

[161] Using http cookies, MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#define_where_cookies_are_sent.

[162] N. Mohamed. http://www.wired.com/2009/08/you-deleted-your-cookies-think-again/.

[163] J. Leyden. Sites pulling sneaky flash cookie-snoop. http://www.theregister.co.uk/2009/08/19/flashcookies/.

[164] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *2nd IEEE European Symposium on Security and Privacy*, Paris, France, 2017. To appear.

[165] Vikas Mishra, Pierre Laperdrix, Antoine Vastel, Walter Rudametkin, Romain Rouvoy, and Martin Lopatka. Don't count me out: On the relevance of IP address in the tracking ecosystem. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 808–815. ACM / IW3C2, 2020.

[166] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W. Felten, Prateek Mittal, and Arvind Narayanan. Watching you watch: The tracking ecosystem of over-the-top tv streaming devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 131–147, New York, NY, USA, 2019. Association for Computing Machinery.

[167] Victor Morel and Raul Pardo. Three dimensions of privacy policies. Working paper.

[168] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In Matt Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.

[169] Mozilla. Lightbeam 3.0. https://addons.mozilla.org/en-GB/firefox/addon/lightbeam-3-0/.

[170] New relic - privacy policy. https://www.simpli.fi/site-privacy-policy/.

[171] Steve Nichols. Big brother is watching: An update on web bugs. In *SANS Institute*, 2001.

[172] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *IEEE Symposium on Security and Privacy, SP 2013*, pages 541–555, 2013.

[173] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. In *CHI*, 2020.

[174] Lukasz Olejnik, Minh-Dung Tran, and Claude Castelluccia. Selling off user privacy at auction. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[175] Information stored by openwpm. https://github.com/mozilla/OpenWPM.

[176] Openx - privacy policy. https://www.openx.com/legal/privacy-policy/.

[177] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass gdpr consent to track users. In *Proceedings of WWW 2021*, 2021. https://arxiv.org/abs/2102.08779.

[178] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1432–1442, 2019.

[179] Panagiotis Papadopoulos, Pablo Rodríguez Rodríguez, Nicolas Kourtellis, and Nikolaos Laoutaris. If you are not paying for it, you are the product: how much do advertisers pay to reach you? In *Internet Measurement Conference, IMC*, pages 142–156, 2017.

[180] Piwik. Free online cookie scanner. https://piwik.pro/cookie-scanner/.

[181] Privacy Badger Official website - Electronic Frontier Foundation. https://www.eff.org/privacybadger.

[182] Data subject rights notice, pubmatic. https://pubmatic.com/legal/eea-data-subject-rights-notice/.

[183] Pubmatic privacy policy . https://pubmatic.com/legal/platform-cookie-policy/.

[184] Quantserve - privacy policy. https://www.quantcast.com/privacy/.

[185] Ramsaygds.fr website. https://hopital-prive-residence-du-parc-marseille.ramsaygds.fr/vous-etes-patient-pourquoi-choisir-notre-etablissement/urologie-22.

[186] Ramsaygds.fr privacy policy. https://ramsaygds.fr/mentions-l%C3%A9gales.

[187] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Network and Distributed System Security Symposium, NDSS*, 2018.

[188] HTTP Redirection. https://developer.mozilla.org/en-US/docs/Web/HTTP/Redirections.

[189] Joel R. Reidenberg, N. Cameron Russell, Alexander J. Callen, Sophia Qasir, and Thomas B. Norton. Privacy harms and the effectiveness of the notice and choice framework. In *Research Conference on Communication, Information and Internet Policy*, 2014.

[190] J.R. Reidenberg, J. Bhatia, and T.D. Breaux. Automated comparisons of ambiguity in privacy policies and the impact of regulation. *J Legal Studies*, 2017.

[191] RFC 2616 - Hypertext Transfer Protocol. HTTP/1.1.https://tools.ietf.org/html/rfc2616.

[192] https://developers.google.com/search/reference/robots_txt.

[193] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012*, pages 155–168, 2012.

[194] Rubicon privacy policy . https://rubiconproject.com/rubicon-project-advertising-technology-privacy-policy/platform-cookie-statement/.

[195] Jukka Ruohonen and Ville Leppänen. Invisible pixels are dead, long live invisible pixels! In *Workshop on Privacy in the Electronic Society, WPES@CCS*, pages 28–32, 2018.

[196] Number of Safari users. https://www.statista.com/statistics/543218/worldwide-internet-users-by-browser/.

[197] Same Origin Policy. https://www.w3.org/Security/wiki/Same_Origin_Policy.

[198] Iskander Sánchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. Can I Opt Out Yet?: GDPR and the Global Illusion of Cookie Control. In *Proceedings of the ACM Asia Conference Computer and Communications Security*, pages 340–351, 2019.

[199] Google Privacy Sandbox. https://www.chromium.org/Home/chromium-privacy/privacy-sandbox.

[200] Cristiana Santos, Nataliia Bielova, and Célestin Matte. Are cookie banners indeed compliant with the law? deciphering eu legal requirements on consent and technical means to verify compliance of cookie banners. https://arxiv.org/abs/1912.07144, 2019.

[201] Cristiana Santos, Nataliia Bielova, and Célestin Matte. Are cookie banners indeed compliant with the law? deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners. *Technology and Regulation*, pages 91–135, 2020.

[202] Cristiana Santos, Aldo Gangemi, and Mehwish Alam. Detecting and editing privacy policy pitfalls on the web. In *TERECOM@JURIX*, 2017.

[203] You should install two browsers. http://www.compukiss.com/internet-and-security/you-should-install-two-browsers.html.

[204] Florian Schaub, Rebecca Balebako, Adam L. Durity, and Lorrie Faith Cranor. A design space for effective privacy notices. In *SOUPS 2015*, 2015.

[205] access right scorecardresearch. https://www.scorecardresearch.com/privacy.aspx.

[206] Simpli - privacy policy. https://www.simpli.fi/site-privacy-policy/.

[207] Smartadserver privacy policy . https://www.sublime.xyz/en/legal-mentions.

[208] Wall Street Journal's "What They Know" Series. https://ashkansoltani.org/work/what-they-know/.

[209] Konstantinos Solomos, John Kristoff, Chris Kanich, and Jason Polakis. Tales of favicons and caches: Persistent tracking in modern browsers. In *NDSS*, 2021.

[210] Ashkan Soltani, Shannon Canty, Quentin Mayo, Lauren Thomas, and Chris Jay Hoofnagle. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[211] Jannick Kirk Sørensen and Sokol Kosta. Before and after GDPR: the changes in third party presence at public and private european websites. In Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia, editors, *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1590–1600. ACM, 2019.

[212] Jannick Kirk Sørensen and Sokol Kosta. Before and after GDPR: the changes in third party presence at public and private european websites. In *WWW*. ACM, 2019.

[213] Ove Sörensen. Zombie-cookies: Case studies and mitigation. In *8th International Conference for Internet Technology and Secured Transactions, ICITST 2013, London, United Kingdom, December 9-12, 2013*, pages 321–326. IEEE, 2013.

[214] Spotxchange - privacy policy. https://www.spotx.tv/privacy-policy/.

[215] Matic Srdjan, Iordanou Costas, Smaragdakis Georgios, and Nikolaos Laoutaris. Identifying sensitive urls at web-scale. In *ACM Internet Measurement Conference (ACM IMC 2020)*, 2020.

[216] European Data Protection Supervisor. Edps inspection software. https://edps.europa.eu/press-publications/edps-inspection-software_en.

[217] Teads - privacy policy. https://www.teads.tv/privacy-policy/.

[218] The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016.

[219] The Bug in your PC is a smart cookie. . https://archive.org/details/FinancialTimes1996UKEnglish/Apr%2001%201996%2C%20Financial%20Times%2C%20%231%2C%20UK%20%28en%29/page/n29/mode/2up?view=theater.

[220] Timlib domains ownership. https://github.com/timlib/webXray_Domain_Owner_List.

[221] Michael Toth, Nataliia Bielova, Cristiana Santos, Vincent Roca, and Célestin Matte. Contribution to the public consultation on the CNIL's draft recommendation on "cookies and other trackers", 2020.

[222] "Guidelines on transparency under Regulation 2016/679, WP260 rev.01. https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=622227.

[223] uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. https://github.com/gorhill/uBlock.

[224] UK DPA. Guidance on the rules on use of cookies and similar technologies', 2020.

[225] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. Beyond the front page: Measuring third party dynamics in the field. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1275–1286. ACM / IW3C2, 2020.

[226] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. Beyond the front page: Measuring third party dynamics in the field. In *WWW*, 2020.

[227] Securing your web browser. https://www.us-cert.gov/publications/securing-your-web-browser.

[228] Pelayo Vallina, Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, and Antonio Fernández Anta. Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem. In *Proceedings of the Internet Measurement Conference*, pages 245–258, 2019.

[229] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. FP-STALKER: tracking browser fingerprint evolutions. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 728–741. IEEE Computer Society, 2018.

[230] Max von Grafenstein. *The Principle of Purpose Limitation in Data Protection Laws: The Risk-based Approach, Principles, and Private Standards as Elements for Regulating Innovation.* Nomos Verlagsgesellschaft mbH, 1 edition, 2018.

[231] Max von Grafenstein. *Regulation as a Facilitator of Startup Innovation: The Purpose Limitation Principle and Data Privacy.* 2018.

[232] Browsec vpn. https://addons.mozilla.org/en-US/firefox/addon/browsec/.

[233] Webcookies. Web cookies scanner. https://webcookies.org/.

[234] Weborama - privacy policy. https://weborama.com/weborama-privacy-commitment/.

[235] Whoer website. https://whoer.net.

[236] Whois library. https://pypi.org/project/whois/.

[237] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard H. Hovy, Joel R. Reidenberg, and Norman M. Sadeh. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1*, 2016.

[238] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard H. Hovy, Joel R. Reidenberg, and Norman M. Sadeh. The creation and analysis of a website privacy policy corpus. In *ACL*, 2016.

[239] wpbeaverbuilder.com privacy policy. https://www.wpbeaverbuilder.com/privacy-policy/.

[240] Xiti.com website. https://www.xiti.com/en/.

[241] Yandex.ru - privacy policy. https://yandex.com/legal/privacy/.

[242] Zhiju Yang and Chuan Yue. A comparative measurement study of web tracking on mobile and desktop environments. *Proc. Priv. Enhancing Technol.*, 2020(2):24–44, 2020.

[243] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M. Pujol. Tracking the trackers. In *International Conference on World Wide Web, WWW*, pages 121–132, 2016.

[244] David Zeber, Sarah Bird, Camila Oliveira, Walter Rudametkin, Ilana Segall, Fredrik Wollsén, and Martin Lopatka. The representativeness of automated web crawls as a surrogate for human browsing. In *Proceedings of The Web Conference 2020*, pages 167–178, 2020.