CSChecker: Revisiting GDPR and CCPA Compliance of Cookie Banners on the Web

Mingxue Zhang*
The State Key Laboratory of Blockchain and Data Security,
Zhejiang University
mxzhang97@zju.edu.cn

Wei Meng
The Chinese University of Hong Kong
wei@cse.cuhk.edu.hk

You Zhou
The State Key Laboratory of Blockchain and Data Security,
Zhejiang University
3170105739@zju.edu.cn

Kui Ren*
The State Key Laboratory of Blockchain and Data Security,
Zhejiang University
kuiren@zju.edu.cn

ABSTRACT

Privacy regulations like GDPR and CCPA have greatly affected online advertising and tracking strategies. To comply with the regulations, websites need to display consent management UIs (*i.e.*, cookie banners) implemented under the corresponding technical frameworks, allowing users to specify consents regarding their personal data processing. Although prior works have investigated the cookie banner compliance problems with GDPR, the technical specification has significantly changed. The compliance status under the latest framework remains unclear. There also lacks a systematic study of CCPA banner compliance. More importantly, most work have focused on detecting the regulation violations, whereas little is known about the possible culprits and causes.

In this paper, we develop CSChecker, a browser-based tool that monitors and records consent strings on websites. We use CSChecker to analyze the GDPR and CCPA cookie banners, and reveal previously unknown compliance problems under both frameworks. We also discover and analyze possible miscreants leading to the violations, *e.g.*, consent management providers that return wrong consent data. The comparison of the two frameworks inspires several suggestions about the design of cookie banners, the implementation of opt-out mechanisms, and the enforcement of user consent choices.

CCS CONCEPTS

• **Security and privacy → Privacy protections**.

KEYWORDS

Privacy Regulation, Compliance Analysis, GDPR, CCPA

ACM Reference Format:
Mingxue Zhang, Wei Meng, You Zhou, and Kui Ren. 2024. CSChecker: Revisiting GDPR and CCPA Compliance of Cookie Banners on the Web. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3597503.3639159

1 INTRODUCTION

Online advertisers commonly collect user data to facilitate web tracking and monetization. For years, such data collection has been silent, widely raising privacy concerns [20, 23, 53]. To provide users with transparent information and greater control over the use of their personal data, several regulations on online privacy have been proposed and taken effect, such as the European Union General Data Protection Regulation (GDPR) [1] and the California Consumer Privacy Act (CCPA) in the United States [6]. These regulations commonly require informed consents for data collection and use.

In response to GDPR and CCPA, the Interactive Advertising Bureau (IAB) Tech Lab [12] proposed two technical frameworks to help developers comply with the new regulations. The Transparency and Consent Framework (TCF) was released in April 2018 as a technical solution to support GDPR [3]. Correspondingly, the US Privacy String (USP) was released by IAB Tech Lab in November 2019 to support CCPA [5]. Both frameworks propose to collect user consents in dedicated graphic interfaces, which are usually known as the "cookie banners". The user consents are represented as consent strings that can be shared among different sites.

Existing works have discovered multiple violations of GDPR and CCPA [24, 31, 39]. However, there still lacks a systematic study of compliance problems under the latest frameworks. The TCF technical spec has been updated several times and introduced new features. It is unclear whether known compliance problems have been solved and whether new problems may occur. Similarly, the cookie banner compliance status under the USP framework needs to be systematically inspected. Meanwhile, the prior works detect the compliance problems without reasoning about the culprits, *e.g.*, the script that sets a consent cookie without obtaining a user's consent. The analysis of potential culprits would simplify the procedure for troubleshooting and provide supporting evidence for legal actions.

In this paper, we present a systematic study of cookie banner compliance status with GDPR and CCPA, under the latest frameworks. In addition to detecting the violations, we also aim to identify the potential culprits and analyze the possible causes. Note that we do not claim to automatically and accurately pinpoint miscreants

ICSE '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0217-4/24/04…$15.00
https://doi.org/10.1145/3597503.3639159

*Mingxue Zhang and Kui Ren are also with the ZJU-Hangzhou Global Scientific and Technological Innovation Center.

for *all* violations we detect, as it requires extensive inspection of interactions among different entities as well as the legal knowledge. Instead, we attempt to identify the *potential* culprits that are involved in the violations, to limit the search space and facilitate troubleshooting.

We face several challenges. Firstly, there is no existing definition of suspected violations of the latest compliance frameworks, *i.e.*, TCF v2.1 [1] and USP. We need to analyze the regulations and technical specifications to identify the potential problems. Second, the violations could be caused by multiple actors. To pinpoint the possible culprits, we need to attribute the use of a consent string to a specific party, which is difficult. Although prior works have designed several tools to study the compliance problems, they cannot readily be used in our study, as they are either designed based on the outdated TCF v1.1 [11], or rely on simple CSS rules to select an incomplete set of consent options [10]. They also do not provide detailed information about the registered consents, *e.g.*, which scripts generate the incorrect consent strings, which is needed for analyzing the possible causes of detected violations.

To overcome the above challenges, we first define 6 categories of violations (including new categories for each framework) based on the technical specifications, regulations and a prior study of TCF v1.1 [35]. We then develop CSChecker, a browser-based tool for analyzing GDPR and CCPA compliance. It records the consent strings used by real-world websites and the scripts that potentially craft the incorrect consent strings. The frameworks propose three ways to store and retrieve consent strings: (1) using cookies, (2) transmitting with network request URLs, and (3) calling specific JavaScript APIs. Therefore, we first hook the cookie setter method to detect consent strings in cookies, and identify the script writing the cookies by inspecting the JavaScript call stack. To detect initiators of network requests containing consent strings, we rely on the browser DevTools and Selenium webdriver performance logs to record all the network activities. This also allows us to detect the `Set-Cookie` headers in the network responses, which is another way to set consent cookies. We also inject JS code to invoke the APIs and parse the returned consent strings. We monitor the JavaScript write operations to locate the script that first defines the APIs, which should be responsible when incorrect consent strings are returned.

We use CSChecker to study the compliance with TCF v2.1 and USP on 469 real-world websites. The results demonstrate the known compliance problems with TCF v1.1 are not completely solved by the new release, and both TCF v2.1 and USP face previously unknown problems. For instance, websites adopting TCF v2.1 could opt in for special features (*e.g.*, use precise geolocation data) or legitimate interests without user awareness, and user opt-out choices for CCPA may not actually be respected. Our analysis of the potential miscreants revealed that advertising scripts frequently set consent cookies against user choices, and scripts could hardcode positive consent strings, leading to the violations, which, to the best of our knowledge, has not been systematically analyzed before. We compare the analysis results of the two frameworks, and suggest that the CMPs shall implement easy-to-notice cookie banners to improve user engagement, and that the publishers should avoid using CMP scripts that violate the regulation requirements. A more centralized in-browser user choice enforcement mechanism is in need to mitigate the violations.

In summary, we make the following contributions.

- We summarize and systematically detect 6 categories of suspected violations with TCF v2.1 and USP. We reveal that TCF v2.1 does not solve the problems with TCF v1.1, and find previously unknown compliance problems.
- We develop CSChecker, a browser-based analysis tool to aid the compliance analysis.
- To the best of our knowledge, we are the first to systematically identify and analyze the possible culprits of detected violations. We further discuss the possible causes, bringing deeper insights about the compliance problems.
- We compare the analysis results of TCF v2.1 and USP, and provide suggestions for improving the transparency of user data collection and processing.

The rest of this paper is organized as follows. We introduce the background of TCF and USP in §2 and §3, respectively. We formally define the violations we aim to detect in §4, and demonstrate our methodology to detect the violations in §5. Next, we present the violations we detected in §6. We then make a comparison between the two frameworks, and provide suggestions regarding the implementation of them in §7. In §8, we discuss the limitations of this work and our future work. Finally, we describe the related works in §9 and conclude in §10.

2 IAB TRANSPARENCY AND CONSENT FRAMEWORK

We visualize in Figure 1 the general workflow of consent collection and transmission in TCF. In the TCF, the vendor is a third-party chosen by the web publisher (*e.g.*, a website the user visits directly) to present the user with their contents, *e.g.*, advertisements. The vendors can also collect or receive the personal data of end users. Examples of vendors include Google Advertising Products, *etc.* The Consent Management Provider (CMP) is an entity that creates and manages the consent strings and communicates them with the vendors. Both CMPs and vendors need to register with IAB Europe.

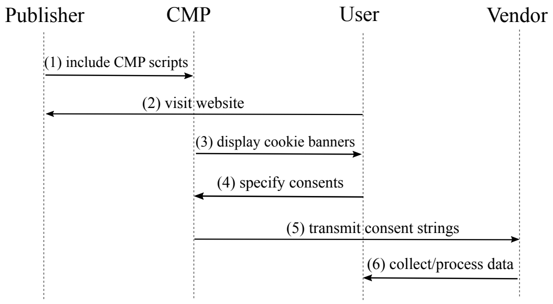

Figure 1: User consent collection and transmission in TCF.

To be compliant with the regulations, the publishers need to cooperate with CMPs, *e.g.*, by including the CMP scripts that implement the standard APIs for communicating user consents. When

[1] The latest version at the time of our experiments.

users visit the publishers' websites, the cookie banners would be displayed, through which the users specify their consents, *e.g.*, whether or not to consent to the sale of their personal data, *etc.*. The CMPs would then create the consent strings to store user consents. The vendors could query the CMPs for consent strings and process user data accordingly.

That said, the frameworks rely on CMPs to create consent strings that correctly reflect user choices. They also require the publishers and vendors to act as regulated. However, the behaviors of different parties are not monitored.

2.1 TCF v2.1

In this section, we focus on the technical support of TCF v2.1. We highlight the objectives and new features of TCF v2.1 in §2.2.

2.1.1 Standard APIs. In TCF v2.1, the CMPs must implement a global function `__tcfapi`, and an iframe that allows vendors to call `__tcfapi` named `__tcfapiLocator`. The API could be called by any vendor for obtaining the consent strings.

2.1.2 Consent strings. TCF v2.1 uses base64url-encoded bit strings as the consent strings. They encapsulate the consented purposes and vendors, a list of consented legitimate interests (both per-purpose and per-vendor), and a list of consented special features. Optionally, the consent string may also include other segment, *e.g.*, `publisherRestriction` [4]. In this work, we mainly focus on the consents granted to the purposes, vendors, legitimate interests and special features.

Consent sharing. Similar to TCF v1.1, the URL-based services could process the consent strings using the URL parameter `gdpr_consent`. A script can also get the consent strings by calling `__tcfapi` API. In terms of consent string storage, TCF v2.1 allows CMPs to freely choose the storage, including non-cookie ones. However, the IAB Europe did not provide any detailed instruction for the implementation. As our goal is not to detect all the possible violations, we focus on cookies in this work. In the future, there could be more websites using storage like localStorage to cope with the incoming bans of third-party cookies, which will be interesting to study in our future work.

2.2 Differences between TCF v1.1 and v2.1

We summarize the main differences between TCF v1.1 and v2.1 as follows. Detailed specifications can be found in [2].

More Purposes and Finer-grained Control on Vendors. TCF v2.1 defines more data collection purposes compared to v1.1. It also grants publishers more control on how vendors may process user personal data, *e.g.*, for which purpose vendors can process the data.

Disclosure of Special Features. TCF v2.1 defines 2 special features (*e.g.*, use precise geolocation data) for the processing of user data. It requires the adoption of special features to be disclosed to users, and the CMPs shall only signal an opt-in of special features after obtaining explicit user consents.

Right to Object to Legitimate Interests. TCF v2.1 requires data processing must base on a legal basis. Except for the "consent" legal basis, vendors may also declare legitimate interests as the legal basis. Users shall be provided appropriate information about the legitimate interests, and be able to communicate their rights to object to the data processing based on the legitimate interests.

3 IAB US PRIVACY STRING

CCPA requires websites to display a link named "Do Not Sell My Personal Information"[2], enabling customers to opt-out for sale of their personal data. According to the US Privacy String framework, digital property owners (*e.g.*, websites) are responsible to share the information with all parties that aim to exchange the data.

3.1 Standard APIs

Similar to the TCF, the USP consent strings are available to vendors through two standard APIs, a global function `__uspapi`, and an iframe named `__uspapilocator` that allows vendors to call `__uspapi` in an iframe.

3.2 Consent Strings

The CCPA consent strings follow a simple format, which only contains 4 characters. The first character indicates the version of a consent string. As the latest release is of version 1.0, the first character is always 1 at the current stage. The second character indicates whether users are provided explicit notices/opportunities to opt out for sale of their data. The third character represents user opt-out against the sale of their personal data. The final character indicates whether the transaction operates under the Limited Service Provider Agreement (LSPA). The possible values for the second to fourth characters include 'Y' (yes), 'N' (no) and '-' (not applicable).

Consent sharing. CCPA recommends that consent strings to be stored as a first-party cookie named `usprivacy`, and they can be retrieved by calling the standard API. For URL-based services to access the consent strings, CCPA allows the consent strings to be included in the URLs through the `us_privacy` parameter.

4 PROBLEM STATEMENT

In this section, we perform an analysis of the legal provision and technical specifications of GDPR and CCPA. We then formally define the regulation and spec violations we aim to detect[3]. Although we are not legal experts, the regulation and specifications are expected to be perspicuous in describing prohibited actions. Besides, it is not our main focus to comprehensively cover all possible violations. Rather, we aim to provide a clear definition of the violations we identified and perform a thorough analysis of them.

Violation #1: Positive consents registered before user actions. Consent strings indicating a non-empty list of consented TCF purposes and vendors are detected before users make choices on cookie banners. This has been detected under TCF v1.1 [24, 31, 35], However, it remains unclear whether the problem has been mitigated in TCF v2.1. We do not consider V1 for USP as it suggests *"when a sale of data may occur, the string should be created"* and *"a string can be created to indicate CCPA applies"* [17]. In other words, there is no explicit limit on when a USP consent string should be created.

[2] In this work, we use "cookie banners" to refer to the consent management UIs for both GDPR and CCPA.

[3] The detailed legal analysis of **V1** - **V4** for GDPR under TCF v1.1 can be found in [35].

Violation #2: Difficult or impossible to specify consents. The cookie banners are absent or difficult to find, or provide no way to opt out. This violates the requirement of CCPA: *"Provide a clear and conspicuous link ... to opt-out of the sale or sharing of the consumer's personal information"* (CIV 1798.135.(a)(1) [7]) and is also considered a violation under TCF v1.1 [35, 51]. We aim to study the presence of similar problems in TCF v2.1 and USP.

Violation #3: Pre-selected options. The cookie banners preselect one or more options, *e.g.*, using pre-ticked checkboxes, *etc.* This could result in ambiguous consents and violates the CCPA requirement of *"a clear and conspicuous link"* in [7]. Prior works have demonstrated the pre-selected options could nudge users to grant unintended consents [34, 35, 38, 50]. Note that under TCF v2.1, options of legitimate and special features may also be pre-selected.

Violation #4: Non-respect of user choices. Consent strings indicating positive consents (*e.g.*, consented purposes or vendors, or agreement to data selling) are still stored and transmitted, after users explicitly opt out. This violates CCPA 1798.135.(a)(5): *"For a consumer who has opted-out of the sale of the consumer's personal information, respect the consumer's decision to opt-out for at least 12 months..."* [7]. While the problem has been detected in TCF v1.1 [35], none of the existing works has investigated the problem with USP, and especially, who the potential culprits of the violations are.

Violation #5: Ambiguous consents of legitimate interests. Positive consents to legitimate interests are found after opting out for all. This violates the requirement *"The data subject shall have the right to object ... to processing of personal data concerning him or her which is based on point (e) or (f) of Article 6(1)"* (Article 14) [8] and TCF Chapter II 5(4) [2]. We detect the violation under TCF v2.1, which proposed the "right to object to legitimate interests".

Violation #6: Ambiguous consents of special features. Positive consents to special features are found after users refuse all the consents. This violates the requirements in TCF Chapter II 5(5) *"A CMP must only generate a positive opt-in Signal for Special Features on the basis of a clear affirmative action taken by a user..."* [2]. Since the "disclosure of special features" is required by TCF v2.1, we aim to detect the violation only in TCF v2.1.

Comparison with prior works. Some of the above violations have been separately examined by prior works, *e.g.*, [43] and [51] detected **V2** of TCF. The closest to our work is [35], which studied TCF v1.1 and therefore ignored **V5** and **V6**. Some other studies of GDPR aimed to analyze the privacy policies to evaluate the compliance status [24, 41, 44], which have a different target from CSChecker. [39] revealed **V2** and **V3** of CCPA, whereas CSChecker aims to also identify other kinds of violations (**V4**).

Compared with prior works, we in particular attempt to pinpoint the potential culprits of the detected violations, *e.g.*, the misbehaving CMPs and scripts, which would greatly ease the burden of troubleshooting. We will present a detailed comparison in §9.

5 METHODOLOGY

We present our methodology in this section. We first describe the technical challenges (§5.1) and how we determine the adoption of the TCF and USP (§5.2). We then demonstrate how CSChecker detects the consent strings and locates the corresponding scripts that create or transmit the strings (§5.3). Next, we discuss how it identifies the CMP scripts, in order to pinpoint the possible culprits of detected violations (§5.4). Compared with existing works, the ability to locate the misbehaving scripts enables CSChecker to provide richer information about the violations and benefits troubleshooting. Finally, we describe how we detect the suspected violations defined in §4 (§5.5).

5.1 Technical Challenges

Tracking scripts/servers that set consent cookies. To attribute an incorrect consent cookie to a potential culprit, we need to track the writes to cookies. Especially, we need to know which script or server (using the `Set-Cookie` header) sets the cookie.

Identifying initiators of network requests. The network requests with incorrect consent strings could be sent out asynchronously. We need a way to precisely identify their initiators.

Locating CMP scripts. The violations can be caused by CMP scripts providing incorrect consent strings. Locating the CMP scripts, however, is not trivial. Intuitively, we can search for the definition of the standard APIs in §2 and §3. However, scripts may indirectly write to the global functions via different identifiers. A simple search for writes to the API names could be imprecise. Moreover, as scripts can also be dynamically loaded, we cannot identify the CMP scripts by statically analyzing the source code.

To overcome the above challenges, we build a browser-based framework CSChecker, by instrumenting the Chromium browser (version 88.0.4303.1). We identify the initiator of network requests and cookies in the HTML responses using the browser APIs. We further locate cookie initiator scripts by hooking the only relevant JavaScript API in browser. Meanwhile, we monitor all the JavaScript write operations to precisely locate the CMP scripts that define the standard TCF and USP APIs. Our browser-level monitoring ensures both the effectiveness and completeness on detecting CMP scripts and initiators of cookies and network requests.

5.2 Finding Websites Adopting TCF or USP

To determine the adoption of TCF and USP, we visit a website, wait for at most 2 minutes for a full page loading, and then inject a script through Selenium to call the global functions `__tcfapi` or `__uspapi`. We also search for iframes with the name `__tcfapiLocator` or `__uspapiLocator`. A website adopts the TCF or USP, if the injected script successfully identifies the APIs.

5.3 Detecting Consent Strings and the Initiators

We use Selenium webdriver and the DevTools to monitor and record the network requests and responses, along with their initiators. For comprehensiveness, CSChecker searches for consent strings in the query strings, headers and cookies. The network logs also allows us to find any response that specifies a `Set-Cookie` header, which can be used to write incorrect consents in cookies.

The shared cookies can also be set at the client side by JavaScript. As the only way to write to cookies in JavaScript is to access the `document.cookie` interface, CSChecker hooks the `Document::setCookie()` function, which implements the writes to the object `document.cookie`. Since TCF v2.1 does not specify a recommended name for the consent cookies, we record all

JavaScript writes to cookies. This also allows us to comprehensively detect any USP cookies with non-standard names. CSChecker then inspects the JavaScript call stack to locate the bottom frame, which corresponds to the script that initiates the write. Finally, the cookie write logs, which contain the cookie name and value, the initiating script URL and a timestamp, are dumped to files for further analysis. We leave it as a future work to investigate other storage mechanisms for consent strings.

Note that the consent strings could be captured before and after we make choices in the cookie banners through clicks. To differentiate between the consent strings used before and after our actions, we additionally hook the event dispatcher and record user clicks. For each click event, CSChecker checks if the click is created from the user agent (*i.e.*, a real user click, instead of a click generated by JavaScript code). If so, it logs the clicked frame URL and the corresponding timestamps. The consent strings can then be differentiated by comparing the timestamps.

5.4 Identifying the CMP Scripts

To precisely identify the original definitions of the standard APIs in CMP scripts, CSChecker monitors the JavaScript write operations at runtime. In the V8 engine, JavaScript variables and functions are represented as instances of the `Object` and `JSObject` classes. Therefore, by hooking the setter methods (*e.g.*, `Object::SetPropertyInternal`), CSChecker can obtain the receiver object, the property name and the value written to the target. It then logs the memory addresses of the receiver object and written value in V8 as a unique identifier. CSChecker further inspects the JavaScript call stack to identify the initiating script in the bottom stack frame, and maintains a list to record all scripts involved in the function definitions. When a function named `__tcfapi` or `__uspapi` is defined, CSChecker checks the list and marks the first script in it as the CMP script. For example, suppose script A assigns a function literal to variable `f`, which is assigned to `window.__tcfapi` by script B. CSChecker records two write operations in order: 1) `f = function(){...}, script A` and 2) `window.__tcfapi = f, script B`. Since the two operations correspond to the same written value (*i.e.*, memory address), CSChecker identifies script A as the one that first defines the standard APIs, *i.e.*, the CMP script. To ease the analysis, the TCF and USP CMP script IDs are also logged as HTML attributes and can be accessed via `document.tcfScriptID` and `document.uspScriptID`, respectively.

5.5 Detecting Suspected Violations

We now describe how we detect the violations with CSChecker. The procedure of our investigation is depicted in Figure 2.

We first conduct an automatic crawling of the Tranco top 100K websites using CSChecker. During the crawling, CSChecker searches for the standard APIs to detect websites that adopt TCF v2.1 and USP.

To detect violations, we visit the websites from IP addresses in Paris, France and California, US, respectively, as the regulations are enforced only in the specific countries/regions. We acknowledge that the websites may behave differently when visited from a different location, which we discuss in §8.

To detect **Violation #1**, CSChecker first identifies websites that adopt TCF v2.1 in the automated crawling, records the cookie writes, and extracts network activities with consent strings from the Selenium webdriver performance logs. It also automatically injects JavaScript code using Selenium to invoke `__tcfapi` and records the returned consent strings. We report a violation if a website used positive consent strings in this step.

Violation #2 - #6 are detected on a set of randomly sampled websites, because the number of websites adopting the frameworks is quite large. Details about the dataset can be found in §6. To detect **Violation #2 - #3**, we manually label the sampled websites on which: 1) it is "impossible/difficult to opt out" (**V2**), or 2) the cookie banner displays "pre-selected options" (**V3**).

To detect **Violation #4 - #6**, we launch CSChecker to visit the sampled websites, and deny all the consents and legitimate interests on the cookie banners. We do this manually, because it is difficult to automatically opt out on the various cookie banners. Existing tools like Consent-O-Matic [10] cannot reliably identify legitimate interest options due to the absence of corresponding CSS rules. Augmenting the tool with new rules also requires significant manual effort. We discuss in §8 possible ways to automate the procedure. After opting out, we wait for a full page loading, manually inject JS code to get API return values, and revisit the websites after 5 seconds. On the second visit, we rely on the DevTools to record network activities. The recorded user clicks help us identify consent strings used after opting out on the first visit, *e.g.*, those stored in cookies, as the cookies may be set only on the first visit. We analyze the recorded data to identify websites that "do not respect user choices" (**V4**), by detecting positive consent strings. We also decode the consent strings to find violations against the "right-to-object" to legitimate interests (**V5**) and the "disclosure of special features" (**V6**).

6 EVALUATION

In this section, we present our findings about privacy regulation compliance on the web. We firstly measure the adoption of both TCF v2.1 and the USP (§6.1). We then characterize the violations we detected (§6.2 and §6.3), and the scripts that may have caused the violations (§6.4). The code of CSChecker and our experiment data are released at https://doi.org/10.6084/m9.figshare.24943723.

6.1 Adoption of Consent Frameworks

We used a Selenium driven Chromium browser to crawl the Tranco top 100K websites in April, 2023. We waited 1 minute for a full page loading and successfully collected data from 82,624 (82.62%) websites within the timeout. Overall, 4,644 and 5,854 websites adopted TCF v2.1 and USP, respectively, and 2,302 websites adopted both. We further categorized these websites according to the language used by the web contents using CLD3 [9]. The most commonly used languages can be found in Table 1.

6.2 TCF v2.1 Violations

In this section, we categorize the detected violations of GDPR under TCF v2.1. The top ranked affected websites can be found in Table 2.

6.2.1 Positive consents before user choice (V1).

*6.2.1 Positive consents before user choice (**V1**).* We used CSChecker to automatically visit the 4,644 websites that adopted

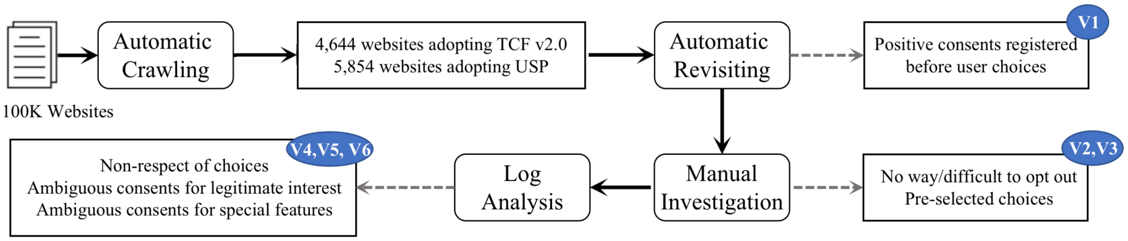
100K Websites
Automatic Crawling
4,644 websites adopting TCF v2.0
5,854 websites adopting USP
Automatic Revisiting
Positive consents registered before user choices
V1
Manual Investigation
No way/difficult to opt out
Pre-selected choices
V2,V3
Log Analysis
Non-respect of choices
Ambiguous consents for legitimate interest
Ambiguous consents for special features
V4,V5, V6

Figure 2: Our experiment procedure.

Table 1: Top 5 languages used by websites that adopted TCF and USP.

Language	#Websites	%Websites
English	2,273	48.95%
German	395	8.51%
French	279	6.01%
Polish	250	5.38%
Spanish	223	4.80%

(a) TCF Website Languages.

Language	#Websites	%Websites
English	4,231	72.28%
Maltese	227	3.88%
Polish	222	3.79%
Vietnamese	178	3.04%
German	172	2.94%

(b) USP Website Languages.

Table 2: Top ranked websites exhibiting TCF v2.1 violations.

Rank	Domain
258	slideshare.net
372	scribd.com
4,853	stltoday.com
6,548	buffalonews.com
7,461	nzz.ch

(a) Consents before choice.

Rank	Domain
4,808	tagesspiegel.de
7,494	mixi.media
8,108	haber7.com
10,095	sport1.de
10,374	sondakika.com

(b) No way/difficult to opt out.

Rank	Domain
733	vice.com
2,313	pcgamer.com
2,390	tomsguide.com
3,741	libero.it
4,024	fundingcircle.com

(c) Pre-selected options.

Rank	Domain
8,263	nintendolife.com
15,205	wetter.de
31,847	playtech.ro
39,878	bvb.de
54,495	thueringer-allgemeine.de

(d) Non-respect of choices.

Rank	Domain
3,741	libero.it
4,024	fundingcircle.com
4,073	prnt.sc
4,853	stltoday.com
6,548	buffalonews.com

(e) Ambiguous consent to legitimate interests.

Rank	Domain
55,210	resultados-futbol.com
-	-
-	-
-	-
-	-

(f) Ambiguous consent to Special features.

Table 3: Statistics about "consents before choice" violation cases.

Type	#Websites	#Cases	%Websites in all V1
URL	72	336	80.90%
Cookie	5	5	5.62%
API	84	84	94.38%
Unique Total	89	425	100%

TCF v2.1, under a Paris IP address. We waited with no interactions for at most 2 minutes for a full page loading and successfully collected data from 4,641 (99.94%) websites within the timeout. In total, 89 (1.92%) websites used positive consents (non-empty list of allowed vendors or purposes).

As shown in Table 3, 72 websites transmitted positive TCF v2.1 consents with 336 network requests before user actions. We matched the destination of these network requests against a public list of advertising servers and trackers[4], and found 324 (96.43%) of them were sent to advertising and tracking domains. One possible explanation is the requests forged positive consents for more profits by using the personal data. Additionally, 5 websites were found to write positive consents in cookies. The cookies were all set by scripts from CMP related domains, *e.g.*, a script from cmp.quantcast.com were found to set incorrect cookies on a website using CMP from quantcast.mgr.consensu.org. We think these scripts are misbehaving by forging positive consents without user awareness. By invoking the `__tcfapi` function, we detected 84 websites on which the API returned positive consent strings. As the APIs are defined by CMP scripts, they should be responsible for the violations.

*6.2.2 No way/difficult to opt out (**V2**).* We manually visited a set of 239 randomly sampled websites. Specifically, we checked all the 89 websites on which we find **V1**, and randomly sample 150 other websites without **V1** (*i.e.*, "clean websites"). We sampled more "clean websites" as there were also more such websites in the whole dataset. We divided all websites without **V1** into 3 groups based on their ranks in the Tranco list, and randomly sampled 50 in each group. This ensures we sampled both high ranked and less popular websites. The categorization results using WhoisXML API [18] further suggest our sampled dataset includes websites in 27 categories, *e.g.*, News and Politics, Pop Culture, Business and Finance, Technology & Computing *etc.*. Therefore, the sampled dataset is representative.

We detect **V2** on a website if it: (1) provided no reject option, or (2) displayed no cookie banners, or (3) automatically refreshed before we finished opting out. Each website was visited twice independently by two researchers. In case the researchers assign different labels to a website, they would double check to reach a consensus. Overall, 19 websites were not consistently accessible, so we excluded them to make sure we get reliable results. Out of the remaining 220 websites, we did not find any reject options on 8 websites. 3 websites did not display any cookie banners, and 4 consistently refreshed before we submitted consent choices. On the 15 websites, we think there is no way to reject the consents.

*6.2.3 Pre-selected options (**V3**).* We manually detected **V3** on the 239 sampled websites. Excluding the 19 websites that are not always accessible, we found in total 155 websites with **V3**. Out of them, 64 (41.29%) registered positive consents before user actions. We decoded the consent strings to identify CMPs on the 155 websites, and list the top 5 CMPs in Table 4. As shown, most websites displaying pre-selected options were cooperating with Osano and Quantcast.

[4] Available at https://github.com/anudeepND/blacklist

Table 4: Top 5 CMPs on websites that displayed pre-selected options in TCF cookie banners.

CMP	#Websites	%Websites in all V3
Osano, Inc.	67	43.23%
Quantcast International Limited	26	16.77%
Google LLC	18	11.61%
Sourcepoint Technologies, Inc.	12	7.74%
OneTrust LLC	7	4.52%

Table 5: Top 5 CMPs on websites that allowed legitimate interests after opting out.

CMP	#Websites	%Websites in all V5
Osano, Inc.	59	48.36%
Quantcast International Limited	26	21.31%
Google LLC	20	16.39%
iubenda	7	5.74%
consentmanager.net	5	4.10%

Osano cookie banners pre-selected all consent options for third-party vendors without providing a "reject all" option, making it time-consuming to opt out for all. Similarly, the OneTrust cookie banners also pre-selected all vendor consent options. Quantcast banners pre-selected many legitimate interests. Similar behaviors were also found on Google LLC and Sourcepoint cookie banners. The results suggest that new legal basis introduced by TCF v2.1, *i.e.*, legitimate interests, were often pre-selected. In these cases, the pre-selected options can be used to nudge users into giving consents. As many pre-selected options are only visible after users click other links/buttons first, they may be difficult for normal users to notice.

6.2.4 Non-respect of user choices (V4). To detect **V4**, we used CSChecker to investigate the 239 randomly sampled websites. We tried to reject all possible consent options, and recorded consent strings in cookies, network requests, and returned by the `__tcfapi`. We detected **V4**, if positive consents for vendors or purposes were found after we opted out. Overall, we successfully collected consent strings on 200 websites. Out of the rest 39 websites, 24 were skipped due to their unstable behaviors, *i.e.*, 19 were not consistently accessible, and 5 websites occasionally displayed cookie banners. On the rest 15 websites, we could not reject the consents (due to the absence of cookie banners *etc.*).

Overall, 10 (5.0%) websites still used positive consent strings after a manual opt-out. 7 websites were already using positive consent strings before user actions (**V1**). The top ranked websites are listed in Table 2d. Except for consents of vendors and purposes, we also found allowed legitimate interests and special features in the consent strings, which we discuss next.

6.2.5 New problems with TCF v2.1. Besides the violations existed for TCF v1.1, we detected multiple new problems with TCF v2.1.

Ambiguous consents of legitimate interests (V5). On the 200 websites that we successfully collected consent strings, 122 were found to send consent strings with allowed legitimate interests after we opted out. We further checked the CMPs on these websites, and listed them in Table 5. We found these websites mostly included two CMPs (*i.e.*, Osano and Quantcast), and therefore were using similar cookie banners. According to our observation, all the top 5 CMPs allowed users to opt out for legitimate interests on a per vendor basis, while the user choice was not respected. Although most websites disclosed legitimate interest usage to users, it was difficult to reject them on many websites. For example, on https://www.mycast.io/, we need to first click the "Vendors List" icon, then click to span every option, and finally scroll down to find the "Legitimate Interest Purposes" options. It would be difficult for normal users to notice such options. As the banner provided no "reject all" option, the users need to click every legitimate interest options to fully opt out, which is very time-consuming.

Ambiguous consents of special features (V6). We found 1 website was faking a positive consent to the employment of special features. On this website, the special feature usage was disclosed to users and can be rejected on a per vendor basis. However, the user choices were not respected.

Summary. Our manual investigation of 239 websites show that the violations of GDPR in TCF v1.1 still exist in TCF v2.1, *e.g.*, positive consent strings may still be used before users take any actions. Moreover, TCF v2.1 introduces new kinds of violations. We found consent strings indicating allowed special features and legitimate interests after we opted out. We also revealed that many websites displayed a huge number of legitimate interest options deep in the banners, making it difficult to opt out all.

6.3 USP Violations

6.3.1 No way/difficult to opt out (V2). We manually inspected another set of 239 websites randomly sampled for studying USP violations. For a fair comparison with TCF, we also randomly sampled 89 websites on which positive USP consent strings were used before user actions, although they are not considered as violations. We then sampled 150 other websites on which no positive consent strings are found. Similar to the study of TCF violations, we skipped 10 websites that were not consistently accessible to get reliable analysis results. We detected **V2**, if the website: (1) did not display any USP cookie banners, or (2) provided no reject options, or (3) displayed cookie banners that could not function, *e.g.*, covered by other elements, or cannot be clicked, or (4) provided hard-to-notice banners. Overall, we were not able to opt out on 78 websites. We also found hard-to-notice banner on 1 website.

The top-ranked **V2** websites are listed in Table 6a. We did not find any USP cookie banner on 66 websites, *i.e.*, the website did not provide a "Do Not Sell..." link either on the main page or in the privacy policy, and no other popup banners were found. 7 websites displayed cookie banners while provided no reject options. On the other 3 websites, the banner was covered by advertisements that cannot be dismissed. Additionally, we detected 2 banners that did not function, which we discuss below. On the above 78 websites, there was no way for users to specify their consents.

Erroneous Implementation. We observed on 2 websites erroneous implementation of the USP opt-out mechanism. https://swimswam.com/ invoked `window.__uspapi('displayUspUi');` to display the cookie banner. However, the API `__uspapi` was incorrectly implemented to infinitely invoke itself. Consequently, the request cannot be handled and no banner was displayed. Another similar example

Table 6: Top ranked websites exhibiting USP violations.

Rank	Domain
1,516	sapo.pt
2,102	wowhead.com
3,321	suntimes.com
5,238	spiceworks.com
6,237	folkd.com

(a) No way/difficult to opt out.

Rank	Domain
795	hbo.com
2,670	eonline.com
2,888	azcentral.com
3,888	oneindia.com
4,377	humblebundle.com

(b) Pre-selected options.

Rank	Domain
2,670	eonline.com
4,377	humblebundle.com
4,705	simplecast.com
4,716	newrepublic.com
4,853	stltoday.com

(c) Non-respect of choices.

was found on https://www.jacksonville.com/, which included a "Do Not Sell" link while clicks on that click had no effect.

Hard-to-notice Cookie Banner. We also found 1 case where the cookie banner was extremely difficult to find. Specifically, on https://www.flickeringmyth.com/, the "Do Not Sell My Data" link was displayed at the bottom of the page, in the same color as the footer. Therefore, it is very difficult for normal users to notice.

*6.3.2 Pre-selected options (**V3**).* We manually detected **V3** on the 239 sampled websites and skipped 10 occasionally inaccessible websites. Out of the rest 229 websites, we found pre-selected options (sale of personal data allowed by default) in USP cookie banners on 54 websites. We list the top ranked websites in Table 6b.

6.3.3 New problems with USP. We present below the previously unknown compliance problem with USP.

Non-respect of choices (V4). To detect **V4**, we used CSChecker to visit the 239 randomly sampled websites. Except for the 10 occasionally inaccessible websites, we also skipped the 78 websites on which we could not opt out (**V2**), and the other 3 websites that did not consistently displayed cookie banners. Additionally, we skipped 7 websites that required users to contact developers (*e.g.*, by filling request forms) for opting out, as the effect of our opt out may not be immediately observable. Overall, we collected consent strings on 141 websites.

In total, we observed positive USP consents after a manual opt-out on 24 (17.02%) unique websites. 10 website was found to write positive consents into cookies, and 15 websites sent positive consents with network requests. On 9 websites, the `__uspapi` returned positive consent strings. We list the top ranked websites in Table 6c.

Summary. Our investigation of 239 randomly sampled websites show that user choices made on USP cookie banners may not be truthfully reflected in consent strings. Many websites were still using positive consent strings after we manually opted out. We were not able to specify consent choices on 78 (32.64%) websites, and found websites may display hard-to-notice banners. Moreover, some websites adopted an erroneous implementation of the USP, leading to the violations.

6.4 Scripts Involved in the Violations

In this section, we aim to analyze the scripts that may have caused the detected violations, which, to the best of our knowledge, has not been systematically investigated in prior works. We identify the scripts that: 1) set incorrect consent cookies (§6.4.1); 2) define the standard CMP APIs, *i.e.*, the CMP scripts (§6.4.2); and 3) initiate the network requests that contain wrong consent strings (§6.4.3).

We acknowledge that this is not a very accurate way to identify the miscreants. For example, an incorrect consent string could also be included in network requests by a script different from the request initiator or the CMP scripts. Nonetheless, it is non-trivial to thoroughly analyze the behaviors of all involved scripts. By pinpointing the possible culprits, we are still able to provide auxiliary information to facilitate troubleshooting. We discuss this in detail in §8, and leave it as a future work to thoroughly analyze the behaviors of affected websites to precisely locate the culprits.

6.4.1 Scripts Setting Consent Cookies. As mentioned in §6.2.1, 4 scripts on 5 websites were found to set positive TCF consent cookie before user actions. Three of them were loaded from subdomains under consentmanager.net, a registered CMP domain. We checked source code of the scripts, and found they read consents pre-configured by another script on the same website, and then wrote the encoded consent strings to cookies. The other script was from cmp.quantcast.com that pre-configured the consents itself. In particular, one website https://gamemonetize.com/ did not display any cookie banners. We believe it is a design flaw to write consent cookies before users make choices, and the scripts should not decide user consents without providing consent options.

In total, we found on 8 websites that incorrect TCF consent cookies were stored as cookies after we opt out. On 7 websites, the cookies were set by scripts from three domains, consentmanager.mgr.consensu.org, consentmanager.net, and cookiepro.com, which were either registered CMP domains, or claimed to provide consent management services. On 1 website, the cookie was written by a first-party script. Additionally, we found incorrect USP consent cookies on 10 websites after opting out. Most such cookies were set by scripts from mediavine.com, which is an advertising domain.

6.4.2 CMP Scripts. One possible reason for the use of incorrect consents is that the CMP scripts provided wrong consent strings. Therefore, we analyzed the write operation logs collected by CSChecker to identify the scripts that defined the global functions `__tcfapi` and `__uspapi`. Overall, we found on 84 websites that the `__tcfapi` returned positive consent strings before user actions, and got positive TCF consents after opting out on 7 websites. Additionally, the CMP scripts on 9 websites returned positive USP consent strings after we opt out. We list these CMP domains in Table 7 and Table 8.

Notably, the CMP domain cmp.osano.com supplied 5 different scripts to 73 websites, on which the scripts returned positive consent strings before user actions. As the CMP scripts are expected to return consent strings that correctly reflect user choices, we believe the "consents before choice" and "non-respect of user choices" violations on these websites can be attributed to the CMP scripts.

6.4.3 Scripts Sending Network Requests. We analyzed the network logs recorded by CSChecker during our automatic crawling, to locate the initiators of network requests with wrong consent strings.

Table 7: Domains serving CMP scripts on TCF "consents before choice" websites.

Script Domain	#Websites	#Scripts
cmp.osano.com	73	5
cdn.consentmanager.net	7	1
cdn.consentmanager.mgr.consensu.org	2	1
quantcast.mgr.consensu.org	2	1

In total, 72 websites transmitted positive consent strings before users take actions. We analyzed the initiator scripts of these network requests and list the top script domains in Table 9[5]. The top 4 of the 5 most commonly detected TCF initiator script domains are advertising/tracking domains. Meanwhile, 96.24% of the requests were also sent to advertising and tracking domains.

We analyzed the network logs on websites that still used positive consent strings after we opted out. 9 websites transmitted positive TCF consent strings, and 15 websites sent positive USP consent strings. The top initiator script domains are listed in Table 10. The request initiator scripts were mostly loaded from advertising/tracking domains (except for cdn.consentmanager.mgr.consensu.org and cdn.consentmanager.net, which were all consent management scripts). Note that the consent strings in network requests were not necessarily crafted by the initiator scripts, *e.g.*, they can be obtained from the CMP scripts included on the same website. Nevertheless, the analysis results demonstrate that advertisers and trackers might violate the policies by crafting positive consent strings.

Summary. We found that many advertising scripts wrote positive consent cookies before user actions. The TCF libraries could also incorrectly set consent cookies, and some CMP scripts were found to violate the requirements of GDPR by implementing standard APIs that return wrong consent strings. Meanwhile, the requests transmitting wrong consent strings were mostly initiated by advertising and tracking scripts. This demonstrate that these advertisers and trackers might be violating the policies by forging positive consents.

7 COMPARISON AND DISCUSSION

We compare the violations between TCF and USP (§7.1), and provide suggestions regarding the design and implementation of the consent collection frameworks and the privacy policies (§7.2).

7.1 Comparison Between Two Frameworks

We summarize the quantitative comparison results in Table 11.

Compared with the USP, we found "cannot/difficult to opt out" **V2** on much fewer websites when they adopt TCF v2.1. The TCF cookie banners were mostly implemented as popups, which were easy to notice. Another reason is much more websites referred users to third-party platforms for opting out when they adopt USP. Therefore, many websites did not display a USP cookie banner at all. We discuss in detail in §7.2. Similarly, the TCF banners were more likely to respect user consent choices, as we also discovered less **V4** for TCF v2.1. This may result from the more standardized implementation of the TCF.

In contrast, we detected fewer "pre-selected options" **(V3)** violations of USP. One possible reason is the relatively simple design of the USP, where only one option needs to be disclosed to customers, leaving little space for the violations.

7.2 Discussion and Suggestions

Blocking third-party cookies. TCF v2.1 suggests that the consent strings should not be stored as third-party cookies from September 1st, 2021. Similarly, USP recommends the consent strings to be stored as first-party cookies. Indeed, during our experiment, we found cases where consent strings were stored in first-party cookies. Therefore, blocking the third-party cookies cannot fully mitigate the violations. In particular, both frameworks allow CMPs to use non-cookie storage while no detailed specification is provided. This calls for a refinement in the standards, *e.g.*, how exactly are CMPs expected to store the consent strings.

Centralized strategies for opting out. Compared with the TCF v2.1, much more websites relied on third-party platforms to implement the opt-out mechanism for USP. Especially, many websites provided multiple choices of such platforms when visited using a California IP address. The inconsistent and custom UIs for opting out greatly increases the user's cost on exercising the rights to opt out. Further, some opt-out options only apply to selected vendors, such as Google and Amazon, which requires more user actions to opt-out for all the vendors. We believe a more centralized opt-out strategy, *i.e.*, a single control panel for managing the consents to all the advertising actors, would help mitigate the problem.

Design of cookie banners. We found most CCPA banners were "Do Not Sell ..." link at the bottom of the main pages. In contrast, the TCF cookie banners are mostly implemented as popups that were easier to find. As also discussed in [39], users are more likely to interact with the popup banners instead of the simple links. To improve user engagement, we believe the CCPA or USP should provide more detailed instructions on the design and implementation of the cookie banners as in [2]. On the other hand, we also observed websites using TCF banners with many pre-selected options, while no "deny all" option was provided. The "non-standard stacks" feature in TCF v2.1 might be a good starting point, which allows to group multiple data collection purposes into one. We believe the TCF cookie banners shall be carefully designed to avoid introducing too much burden to users.

Enforcement of user choices. Preventing the use of incorrect consent strings is a complex task. On the one hand, websites adopt various designs of cookie banners, making it hard to record and enforce user choices in a general way. On the other hand, the consent strings are expected to be created and stored by CMPs, whose implementation could be diverse and incorrect. To solve the problems, the publishers and CMPs could implement the cookie banners based on the templates of web development frameworks (*e.g.*, Angular) for a more uniformed design. As the behaviors of publisher or CMP scripts are inevitably subject to the manipulation of other scripts, we believe a more centralized mechanism should be implemented by the browser for enforcing user choices. For example, the browser could implement the cookie banners in a uniformed design across different websites, which cannot be forged or manipulated. The browser will then be responsible for collecting

[5] We ignore the cases that we cannot determine the domain name, *e.g.*, the initiator URL is about:srcdoc, or no URL is present in the initiator field.

Table 8: Domains serving CMP scripts on "non-respect of choices" websites.

Script Domain	#TCF Websites	#TCF CMP Scripts	Script Domain	#USP Websites	#USP CMP Scripts
cdn.consentmanager.net	3	1	cmp.osano.com	3	1
cookie-cdn.cookiepro.com	2	2	consent.cookiebot.com	3	2
cdn.consentmanager.net	2	1	cdn.ziffstatic.com	1	1
-	-	-	htlbid.com	1	1
-	-	-	cdn.cookielaw.org	1	1

Table 9: Top domains serving scripts that initiated suspicious network requests on "store before choice" websites.

Script Domain	#Websites	#Scripts
securepubads.g.doubleclick.net	64	5
c.amazon-adsystem.com	64	1
f.h12-media.com	1	1
widgets.outbrain.com	1	1
gum.criteo.com	1	1

user consents and blocking the use of incorrect consent strings. This allows for the enforcement of user choices in a more standard way and cannot be bypassed.

8 LIMITATIONS AND FUTURE WORK

Vantage Point. We visited the websites from a Paris (resp. Californian) IP address for analyzing the compliance with GDPR (resp. CCPA). The behaviors of websites could change when visited from a different location. We leave it as a future work to measure the compliance problems with websites when visited in other regions.

Automated Interaction with Cookie Banners. We require manual efforts to deny all the consents due to the diverse design of cookie banners. The existing tool Consent-O-Matic [10] does not apply well to TCF v2.1 banners, because many of its CSS rules only select simple "purpose consent" options. It requires significant efforts to extend the tool by writing new rules. To automate the procedure, we could analyze the structure of cookie banners of different CMPs, and deploy CSChecker to automatically interact with the cookie banners accordingly. We leave this as a future work.

Identification of Culprits. The detected violations can be caused by multiple miscreants in various ways. For example, a network request containing an incorrect consent string could be constructed by one script and sent out by another. Consequently, we were not able to accurately attribute all detected violations to a specific script. Dynamic information flow tracking could help reason about the origin of a consent string. However, as the consent strings could be constructed from any source (*e.g.*, any JavaScript variable), it requires to track all the possible information flows, which is extremely expensive if not impossible. Therefore, we pinpoint the possible culprits, which greatly limit the search space for troubleshooting. We leave it as a future work to thoroughly analyze the behaviors of affected websites to precisely locate the culprits.

9 RELATED WORK

Impact of and Compliance with GDPR. Previous works have studied the impact of GDPR in various aspects. Degeling *et al.* [24] found the majority of top EU websites had updated their privacy policies in response to GDPR. Libert *et al.* [32] found after GDPR went into effect, less third-party contents and cookies were present. Urban *et al.* [49] demonstrated the effectiveness of GDPR in restricting data sharing was quite limited. Similarly, Sorensen *et al.* [46] conducted a long-term analysis and found after GDPR, more third-party contents were present on many websites. Sanchez *et al.* [43] revealed that although GDPR indeed reduced tracking, most websites still used cookies for tracking web users.

Many works have measured the compliance with GDPR on the web. Matte *et al.* [35] studied 4 categories of user choice violations of TCF v1.1 on over 28K European websites. Sakamoto *et al.* [42] demonstrated that websites commonly continue tracking after users opt out. Nouwens *et al.* [38] revealed the design of 5 most popular CMPs left space for implicit and ambiguous consents. Degeling *et al.* [24] demonstrated that many consent libraries violated the requirement of GDPR, *e.g.*, by forcing an opt-in. and 32% of them provided no option for opting out. Bollinger *et al.* [21] found that many banners declared incorrect cookie usage purposes and expiration date. Similar works include [22, 26, 31, 33, 48, 51]. In this work, we develop a framework to study the compliance with the latest GDPR technical standard—TCF v2.1, and especially compare it with CCPA. We showed that the problems identified in previous works are not solved by the new release, and TCF v2.1 introduced new categories of violations. Robol *et al.* [40] proposed a formal consent framework expressed in Description Logic to verify the compliance status, which has a different focus from this work.

Some other works focused on compliance problems of Android apps. Nguyen *et al.* [36] discovered that many apps have already shared tracking IDs with third-parties before user actions. Similar findings were also presented by Kollnig *et al.* [29]. Nguyen *et al.* [37] further revealed similar violations as in [35] on Android apps. These works have a different target from ours. Although some work have attempted to automate cookie banner identification, they cannot easily apply to websites in various languages.

Studies of CCPA. To the best of our knowledge, there have been few research works that focus on the CCPA. Baik *et al.* [19] revealed the views of CCPA from corporate speakers and consumers differed in various aspects, *e.g.*, the definition of "personal information" *etc.* Veys *et al.* [52] discovered that the downloaded personal data copies did not provide sufficient transparency to users. These works have different targets from ours. O' Connor *et al.* [39] found that only 35.8% websites implemented the CCPA banners or links for opting out. Siebel *et al.* [45] conducted user study with 54 participants and suggested CCPA banners to be more conspicuous and standardized to improve user engagement. In addition to the known violations, we also found that consent strings stored on websites may not truthfully reflect user choices. We also demonstrate erroneous implementation can lead to the violations.

Implementation of Privacy Notices. Eijk *et al.* [25] found the top level domain of websites greatly influenced the presence of cookie

Table 10: Top domains serving scripts that initiated suspicious network requests on "non-respect of choice" websites.

Script Domain	#TCF Websites	#TCF Scripts	Script Domain	#USP Websites	#USP Scripts
securepubads.g.doubleclick.net	3	3	securepubads.g.doubleclick.net	5	2
cdn.consentmanager.mgr.consensu.org	3	2	scripts.mediavine.com	3	3
cdn.consentmanager.net	3	2	exchange.mediavine.com	3	1
c.amazon-adsystem.com	2	1	eus.rubiconproject.com	3	1
aponet.adspirit.de	1	4	ads.pubmatic.com	2	2

Table 11: Quantitative comparison between violations of TCF and USP. Numbers in the table represent the number of websites that fall in the corresponding category.

Category	TCF	USP
Cannot/difficult to opt-out (V2)	**15/220 (6.82%)**	79/229 (34.50%)
Pre-selected options (V3)	155/220 (70.45%)	**54/229 (23.58%)**
Non-respect of choices (V4)	**10/200 (5.0%)**	24/141 (17.02%)

banners. Utz *et al.* [50] evaluated how the design of cookie banners may affect user engagement. It was also proved when provided with multiple choices, users would be more likely to give more consents than intended [30, 34]. Another branch of research revealed that the privacy notices might be designed to nudge users to grant consents, *e.g.*, by pre-selecting certain options [35, 38]. In this work, we also found different kinds of violations, *i.e.*, the explicit violation of user choices, and the consents registered before user actions.

Analysis of Privacy Policies. Prior works leveraged natural language processing and machine learning techniques to interpret and annotate privacy policies [28, 41, 44, 47, 55]. Some projects aggregate the evaluation results of privacy policies using a crowdsourcing method [13–16]. [54] evaluated the risk level of a given privacy policy by classifying the privacy policy text. [27] studied the impact of the length of privacy policies on user awareness. These works are orthogonal to our work, which detects consent strings to investigate the violations of the corresponding regulations.

10 CONCLUSION

In this paper, we developed a browser-based analysis tool, CSChecker, to study the cookie banner compliance with GDPR and CCPA under the latest consent frameworks, *i.e.*, TCF v2.1 and USP. With CSChecker, we revealed multiple previously unknown compliance problems under both TCF v2.1 and USP. It also helped locate the potential culprits of the detected violations to facilitate troubleshooting. We compared the two frameworks and made recommendations to help design better consent frameworks. We believe that it takes the CMPs, publishers and browser vendors together to help improve the compliance and respect user choices.

REFERENCES

[1] 2018. General Data Protection Regulation (GDPR). https://gdpr-info.eu/.
[2] 2018. IAB Europe Transparency and Consent Framework policies (v2.1). https://iabeurope.eu/iab-europe-transparency-consent-framework-policies/.
[3] 2018. Transparency and Consent Framework (TCF). https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework.
[4] 2019. Consent string and vendor list format v2. https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework/blob/master/TCFv2/IAB%20Tech%20Lab%20-%20Consent%20string%20and%20vendor%20list%20formats%20v2.md.
[5] 2019. U.S. Privacy String. https://github.com/InteractiveAdvertisingBureau/USPrivacy/blob/master/CCPA/US%20Privacy%20String.md.
[6] 2020. California Consumer Privacy Act (CCPA). https://en.wikipedia.org/wiki/California_Consumer_Privacy_Act.
[7] 2020. California Consumer Privacy Act of 2018 [1798.135]. https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?sectionNum=1798.135.&nodeTreePath=8.4.45&lawCode=CIV.
[8] 2020. GDPR Article 14: Information to be provided where personal data have not been obtained from the data subject. https://gdpr-info.eu/art-14-gdpr/.
[9] 2023. CLD3. https://github.com/google/cld3.
[10] 2023. Consent-O-Matic. https://consentomatic.au.dk/.
[11] 2023. Cookie Glasses. https://chrome.google.com/webstore/detail/cookie-glasses/gncnjghkclkhpkfhghcbobednpchjifk.
[12] 2023. IAB Tech Lab. https://iabtechlab.com/.
[13] 2023. privacychoice. http://www.privacychoice.org..
[14] 2023. Terms of Service; Didn't Read (ToS;DR). https://tosdr.org..
[15] 2023. TOSBack. http://tosback.org..
[16] 2023. TOSBack2. https://github.com/pde/tosback2..
[17] 2023. US Privacy String. https://github.com/InteractiveAdvertisingBureau/USPrivacy/blob/master/CCPA/US%20Privacy%20String.md.
[18] 2023. WhoisXMLAPI. https://website-categorization.whoisxmlapi.com/api.
[19] Jeeyun Sophia Baik. 2020. Data privacy against innovation or against discrimination?: The case of the California Consumer Privacy Act (CCPA). *Telematics and Informatics* 52 (2020).
[20] Ghazaleh Beigi, Ruocheng Guo, Alexander Nou, Yanchao Zhang, and Huan Liu. 2019. Protecting user privacy: An approach for untraceable web browsing history and unambiguous user profiles. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 213–221.
[21] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. 2022. Automating Cookie Consent and {GDPR} Violation Detection. In *Proceedings of the 31st USENIX Security Symposium (Security)*. Boston, MA, USA.
[22] Caudio Carpineto, Davide Lo Re, and Giovanni Romano. 2016. Automatic assessment of website compliance to the european cookie law with coolcheck. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*. 135–138.
[23] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *Proceedings on privacy enhancing technologies* 2015, 1 (2015), 92–112.
[24] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. 2022. We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy. In *Proceedings of the 2022 Annual Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, USA.
[25] Rob van Eijk, Hadi Asghari, Philipp Winter, and Arvind Narayanan. 2019. The impact of user location on cookie notices (inside and outside of the European union). In *Workshop on Technology and Consumer Protection (ConPro'19)*.
[26] Papadogiannakis Emmanouil, Papadopoulos Panagiotis, Kourtellis Nicolas, and Markatos Evangelos. 2021. User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users. In *Proceedings of the Web Conference (WWW)*. Ljubljana, Slovenia.
[27] Joshua Gluck, Florian Schaub, Amy Friedman, Hana Habib, Norman Sadeh, Lorrie Faith Cranor, and Yuvraj Agarwal. 2016. How short is too short? Implications of length and framing on the effectiveness of privacy notices. In *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*. 321–340.
[28] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 531–548.
[29] Konrad Kollnig and Ge Wang. 2021. A Fait Accompli? An Empirical Study into the Absence of Consent to Third- Party Tracking in Android Apps. In *Proceedings of the 7th Symposium on Usable Privacy and Security (SOUPS 2021)*. Proceedings of the 7th Symposium on Usable Privacy and Security (SOUPS 2021).
[30] Stefan Korff and Rainer Böhme. 2014. Too much choice: End-user privacy decisions in the context of choice proliferation. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*. 69–87.
[31] Ronald Leenes and Eleni Kosta. 2015. Taming the cookie monster with dutch law–a tale of regulatory failure. *Computer Law & Security Review* 31, 3 (2015), 317–335.
[32] Timothy Libert, Lucas Graves, and Rasmus Kleis Nielsen. 2018. Changes in third-party content on European News Websites after GDPR. (2018).

[33] Chaoyi Lu, Baojun Liu, Yiming Zhang, Zhou Li, Fenglu Zhang, Haixin Duan, Ying Liu, Joann Qiongna Chen, Jinjin Liang, Zaifeng Zhang, et al. 2021. From WHOIS to WHOWAS: A Large-Scale Measurement Study of Domain Registration Privacy under the GDPR. In *Proceedings of the 2021 Annual Network and Distributed System Security Symposium (NDSS)*. San Diego, CA, USA.

[34] Dominique Machuletz and Rainer Böhme. 2020. Multiple purposes, multiple problems: A user study of consent dialogs after GDPR. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 481–498.

[35] Célestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do Cookie Banners Respect my Choice?: Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 791–809.

[36] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. 2021. Share First, Ask Later (or Never?)-Studying Violations of GDPR's Explicit Consent in Android Apps. In *Proceedings of the 30th USENIX Security Symposium (Security)*. Virtual Event.

[37] Trung Tin Nguyen, Michael Backes, and Ben Stock. 2022. Freely Given Consent? Studying Consent Notice of Third-Party Tracking and Its Violations of GDPR in Android Apps. In *Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS)*. Los Angeles, CA, USA.

[38] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark patterns after the GDPR: Scraping consent pop-ups and demonstrating their influence. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[39] Sean O'Connor, Ryan Nurwono, and Eleanor Birrell. 2020. (Un) clear and (In) conspicuous: The right to opt-out of sale under CCPA. *arXiv preprint arXiv:2009.07884* (2020).

[40] Marco Robol, Travis D Breaux, Elda Paja, and Paolo Giorgini. 2023. Consent verification monitoring. *ACM Transactions on Software Engineering and Methodology* 32, 1 (2023), 1–33.

[41] Amos Ryan, Acar Gunes, Lucherini Elena, Kshirsagar Mihir, Narayanan Arvind, and Mayer Jonathan. 2021. Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset. In *Proceedings of the Web Conference (WWW)*. Ljubljana, Slovenia.

[42] Takahito Sakamoto and Masahiro Matsunaga. 2019. After GDPR, Still Tracking or Not? Understanding Opt-Out States for Online Behavioral Advertising. In *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 92–99.

[43] Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. 2019. Can i opt out yet? gdpr and the global illusion of cookie control. In *Proceedings of the 2019 ACM Asia conference on computer and communications security*. 340–351.

[44] Liu Shuang, Zhao Baiyang, Guo Renjie, Meng Guozhu, Zhang Fan, and Zhang Meishan. 2021. Have You been Properly Notified? Automatic Compliance Analysis of Privacy Policy Text with GDPR Article 13. In *Proceedings of the Web Conference (WWW)*. Ljubljana, Slovenia.

[45] Aden Siebel and Eleanor Birrell. 2022. The Impact of Visibility on the Right to Opt-out of Sale under CCPA. *arXiv preprint arXiv:2206.10545* (2022).

[46] Jannick Sørensen and Sokol Kosta. 2019. Before and after gdpr: The changes in third party presence at public and private european websites. In *Proceedings of the Web Conference (WWW)*. San Francisco, CA, USA.

[47] Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. 2018. PrivacyGuide: towards an implementation of the EU GDPR on internet privacy policy evaluation. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*. 15–21.

[48] Martino Trevisan, Stefano Traverso, Eleonora Bassi, and Marco Mellia. 2019. 4 years of EU cookie law: Results and lessons learned. *Proceedings on Privacy Enhancing Technologies* 2019, 2 (2019), 126–145.

[49] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Measuring the impact of the gdpr on data sharing in ad networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. 222–235.

[50] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (un) informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 26th ACM Conference on Computer and Communications Security (CCS)*. London, UK.

[51] Pelayo Vallina, Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, and Antonio Fernández Anta. 2019. Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem. In *Proceedings of the Internet Measurement Conference*. 245–258.

[52] Sophie Veys, Daniel Serrano, Madison Stamos, Margot Herman, Nathan Reitinger, Michelle L Mazurek, and Blase Ur. 2021. Pursuing Usable and Useful Data Downloads Under GDPR/CCPA Access Rights via Co-Design. In *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS) 2021*.

[53] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L Mazurek, and Blase Ur. 2019. Oh, the Places You've Been! User Reactions to Longitudinal Transparency About Third-Party Web Tracking and Inferencing. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 149–166.

[54] Razieh Nokhbeh Zaeem, Rachel L German, and K Suzanne Barber. 2018. Privacycheck: Automatic summarization of privacy policies using data mining. *ACM Transactions on Internet Technology (TOIT)* 18, 4 (2018), 1–18.

[55] Sebastian Zimmeck and Steven M Bellovin. 2014. Privee: An architecture for automatically analyzing web privacy policies. In *Proceedings of the 23rd USENIX Security Symposium (Security)*. San Diego, CA, USA.