Understanding Website Compliance with Web Cookie Privacy Laws

Maxwell Lin
Duke University

Karen Wang
Duke University

Abstract

Websites use cookies for diverse purposes, such as authentication, targeted advertisement, and behavior tracking. Improper web cookie implementation can lead to security vulnerabilities and privacy violations. In this paper, we develop a framework to determine whether a given website's use of cookies is compliant with web cookie privacy laws such as GDPR. To automate this framework and easily measure website compliance at scale, we develop Chrome extensions and a Selenium-based web-crawler. From our preliminary study, we find that 100 of 255 (39%) of websites violate GDPR and CCPA by retaining tracking cookies after the reject button is clicked. Lastly, we also propose a method to verify cookie compliance for the OneTrust CMP.

1 Introduction

In many jurisdictions, websites are legally obligated to implement cookie notices that provide users the ability to consent to certain cookie types. For example, in the European Union, the General Data Protection Regulation (GDPR) [8] and ePrivacy Directive [7] require that websites obtain specific, informed, and unambiguous user consent before accessing or storing any user data that is not essential to website function. Pre-ticked check boxes or other forms of *opt-out* consent mechanisms violate GDPR since they do not satisfy the requirements of *unambiguous* consent. This means that all cookies that are not *Strictly Necessary* must be disabled until the user specifically opts in. Other laws such as the California Consumer Privacy Act (CCPA) [2] specify an *opt-out* approach where user data can be immediately accessed and stored until the user indicates otherwise.

Even in the absence of privacy laws, users themselves should have the right to control how their information is used on the Internet. It is often in the best interest of users to select the most privacy-preserving option in a cookie notice, often by rejecting all cookies that result in no direct benefit to the user (such as *Performance* or *Targeting/Advertising* cookies).

With this motivation, we seek to answer the following question:

When users select an option on a cookie banner, are their decisions being respected by the website?

In other words, we seek to verify that the appropriate types of cookies are active only if a user has consented to their usage. Websites can violate web privacy laws in other ways, for example, the use of *dark patterns* to influence users' choices [10, 22]; however, these violations are outside the scope of our study.

All code is made available on GitHub [16] for reproducibility.

2 Background and Related Work

In this section, we provide definitions for different types of cookies and cookie notices as well as evaluate related studies.

2.1 Cookie Classification

There are 4 different types of cookies defined by the UK International Chamber of Commerce [5]:

1. **Strictly Necessary Cookies:** Enable users to move around the website and use requested features, such as accessing secure areas of the website or adding items to a shopping cart. Since these cookies are essential, no consent is required.
2. **Performance Cookies:** Collect anonymized information about how visitors use a website (e.g., popular pages, error logs).
3. **Functionality Cookies:** Remember choices that users make (such as username, language, or region) to provide personalized features.
4. **Targeting/Advertising Cookies:** Collect information about users' browsing habits to deliver relevant advertisements.

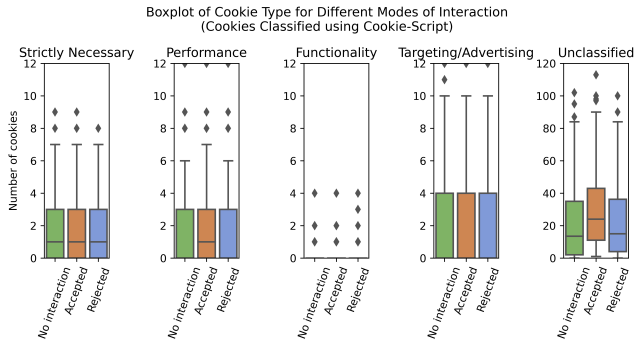


Figure 1: **Cookie-Script Boxplot**: Data generated using *BannerClick* [20]. Note that *Cookie-Script* was unable to classify the majority of the cookies collected—there is roughly ten times more *Unclassified* cookies than any of the 4 categorized cookie types. (Most websites had zero *Functionality* cookies; outliers are shown as diamonds.)

We note that *Targeting/Advertising* cookies (also known as "tracking cookies") pose the most serious privacy threat as they collect user data to transmit to third parties for commercial purposes. Thus, we first focus on verifying that tracking cookies are active only if the user has given explicit consent.

2.1.1 Automated Cookie Classification

There are many cookie databases that map cookie names to their ICC UK category. For example, when given a website, *Cookie-Script* [6] will categorize all present cookies as one of the four ICC UK categories or the *Unclassified* category if no database entry is found. By automating this process using Selenium [21], we created a database that can be used to map a cookie name to its category [23]. However, as shown by Figure 1, the majority of cookies collected from 255 websites are unable to be classified. Due to the ever-changing nature of the web, cookie databases will never be fully comprehensive.

To address this problem, Hu et al. [11] introduced *CookieMonster*, a machine learning model capable of categorizing cookies based on their name with an accuracy of 94%. However, machine learning techniques are susceptible to challenges such as overfitting and concept drift, which can compromise generalization and long-term accuracy.

Given these limitations, we propose classifying cookies using a behavioral approach. By matching a cookie's behavior to one of the 4 ICC UK definitions, it may be possible to classify a cookie as either *Strictly Necessary* or *Functionality*. On the other hand, *Targeting/Advertising* and *Performance* cookies are more difficult to categorize since they generally affect server-side behavior instead of client-side behavior. However, we can still obtain a lower-bound estimate for the number of *Targeting/Advertising* cookies by matching the cookie domain to a known tracker blocklist (see Section 5).

2.2 Cookie Notices and Automated Interaction

We split cookie notices into 3 types:

1. **Accept:** Notices where all cookie types are enabled by default and there is no option to reject.
2. **Accept/Reject:** Notices where users can explicitly give or deny consent to all non-necessary cookies.
3. **Accept/Settings:** Notices where users can choose more granular settings to consent to specific cookie types or vendors.

Examples of these cookie notice types are presented in Figure 2.

Previous studies have developed tools to automatically detect and interact with cookie notices. For example, *BannerClick* [20] uses a corpus of keywords to interact with *Accept/Reject* cookie banners with an accuracy of 97% and 87%, respectively. *CookieEnforcer* [13] uses a Text-To-Text Transfer Transformer (T5) model to automatically select the most privacy-preserving option in *Accept/Settings* cookie notices with an accuracy of 94%. Note that *Accept* cookie notices do not require interaction since all cookies are enabled by default and cannot be disabled.

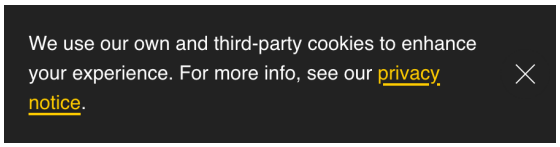
2.3 Web Cookie Compliance

Rasaii et al. [20] found that GDPR is successful at reducing third-party and tracking cookies while CCPA did not have a direct positive impact. In their study, Rasaii et al. counted the number of *set* cookies, either by JavaScript or by the `set-cookie` HTTP response header. In our study, we examine cookies that are sent in the `cookie` HTTP request header to count only the cookies that are actively being used. Additionally, we leverage the ICC UK classification (see Section 2.1). This is crucial for ensuring compliance at a granular level, especially since domains can track users across multiple websites using only first-party cookies [3, 9].

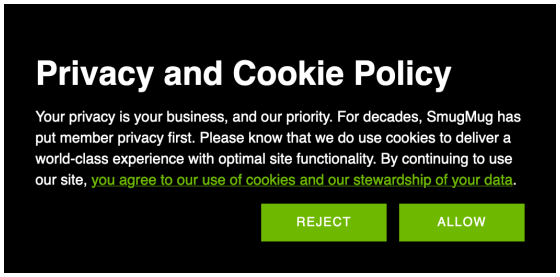
By intercepting CMP consent strings, Matte et al. [17] found that 9.9% websites store consent before choice and that 5.3% websites do not respect users' choice. While Matte et al. only read the consent string to verify compliance, our study modifies the consent string and examines whether websites react appropriately.

3 System Overview

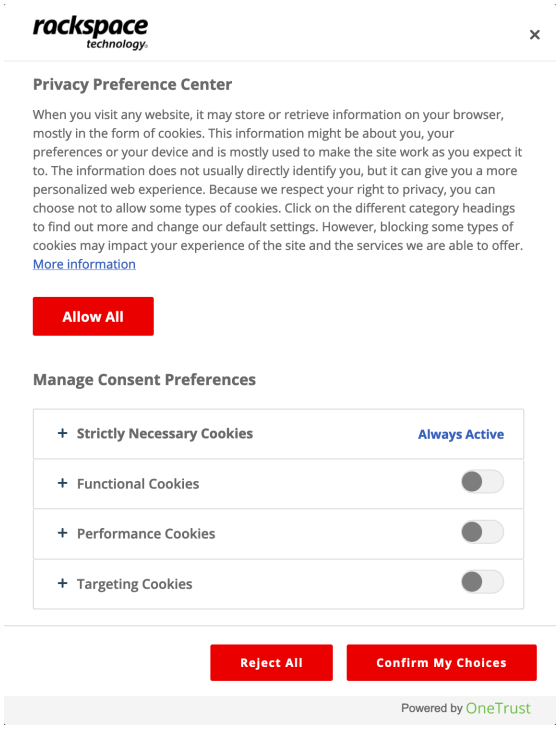
The high-level algorithm for determining whether a given website is compliant with web cookie privacy laws is given in Figure 3. Our primary contribution is detailed in Section 4, where we check whether websites with an *Accept/Reject* cookie notice continue to send tracking cookies after the reject button is clicked. We also propose a method for verifying the compliance of the *OneTrust* CMP in Section 7.2. As



(a) **Accept:** A cookie notice on https://www.nationalgeographic.org. Note that there is no explicit accept button—continued use of the website implies consent to all cookies. Typically, this type of cookie notice violates GDPR since only *Strictly Necessary* cookies can be enabled by default [8].



(b) **Accept/Reject:** A cookie notice on https://www.smugmug.com.



(c) **Accept/Settings:** A cookie notice on https://www.rackspace.com that uses the *OneTrust* CMP. Note that OneTrust offers many different types of cookie notices with varying interfaces and cookie categories.

Figure 2: **Examples of Different Types of Cookie Notices**

OneTrust is the most popular CMP, this method would verify compliance for many *Accept/Settings* cookie notices.

4 Selenium Crawler

Our crawler uses Selenium [21] to drive a headless Firefox instance. Given a starting URL, the crawler collects all anchor elements (i.e., hyperlinks) present and records their depth d. The crawler then recursively repeats this algorithm for each recorded inner page (ignoring duplicates). For each inner page, we also intercept the `referer` HTTP request header to contain the address of the page that pointed to this inner page. Thus, we mimic a real user navigating the inner pages of a website. In our implementation, we zero-index depth so that a $d = 0$ crawl would only crawl the starting URL.

We ignore URLs that redirect to a different domain than the starting URL. Additionally, we only use the host and path when checking for duplicates and ignore other elements such as the scheme and query string.

Lastly, we incorporate the open-source code *BannerClick* [20] to automate clicking the accept and reject buttons in cookie notices. Out of 218 accessible websites that had a cookie notice, we found that *BannerClick* could click both the accept and reject buttons on 180 websites. This 83% accuracy rate demonstrates that *BannerClick* is suitable for our study.

There are 4 high-level steps in the web crawling process:

1. **First run:** Crawl each site (and its inner pages if specified) to load cookies into the browser.
2. **"Normal" run:** Crawl each site (and its inner pages if specified), saving session data into a HTTP Archive (HAR) file.
3. **Banner Interaction:** Use *BannerClick* to click the reject button.
4. **"After Reject" run:** Crawl each site (and its inner pages if specified), saving session data into a HAR file.

In our analysis, we compare the tracking cookies collected between the "Normal" run and the "After Reject" run.

5 Analysis

An aggregated list of known tracking domains was created from 4 domain-only filter lists obtained from JustDomains [12]. First, we looked at the HTTP response entries in the collected HAR files, and if a cookie's domain was found to be in the blocklist, that cookie's name, value, and domain was recorded as a tracking cookie. Then, we looked at the HTTP request entries and recorded all cookies that had appeared in the HTTP response list of tracking cookies. Thus,

Figure 3: **Website Cookie Compliance Flowchart**

a tracking cookie is a cookie sent in a HTTP request, which was detected as a tracker in a HTTP response based on the cookie's domain.

We only count unique tracking cookies—i.e., if a cookie with the same name, value, and domain appeared multiple times, it was only counted once.

For crawls with $d > 0$, we also calculated the average number of trackers per page by dividing the total number of trackers by the total number of inner pages.

6 Results

We base our initial analysis on 255 websites that have cookie notices. Due to resource constraints, we were only able to fully complete a $d = 0$ crawl. Overall, we detect 1,475 unique tracking cookies during the "Normal" run and 1,108 unique tracking cookies during the "After Reject" run suggesting that many websites have an *opt-out* cookie consent mechanism.

Our primary result is that 100 out of 255 (39%) domains persistently send the same tracking cookies during both the "Normal" and "After Reject" runs. In these websites, the reject button fails to appropriately deactivate all tracking cookies. There are only 13 domains with a correct *opt-out* cookie implementation where all tracking cookies during the "Normal" run were disabled during the "After Reject" run. Note that these *opt-out* cookie notices are compliant with CCPA but still violate GDPR (See Section 1).

Additionally, 122 (48%) of websites use tracking cookies during the "Normal" run. This violates GDPR since only *Strictly Necessary* cookies are allowed to be active before the user provides consent [8].

7 CMP Compliance

Many websites implement cookie notices using a third-party Consent Management Provider (CMP). For example, a *OneTrust* CMP cookie notice is shown in Figure 2c. Typically, CMPs expose a JavaScript API to communicate the user's consent settings (e.g., IAB Europe's `__tcfapi` [1] or OneTrust's `OneTrust` API [18]). By targeting the APIs of popular CMPs, we can verify the compliance of thousands of diverse websites even if they have different types of cookie notices.

7.1 TCF Consent Management Platform API

IAB Europe's Transparency and Consent Framework (TCF) specifies an API that CMPs can use to comply with GDPR [1]. The current API (CMP API v2) uses a Transparency and Consent (TC) string to communicate the user's consent choice with vendors. To access the TC string, vendors must call the `__tcfapi` JavaScript function with the `addEventListener` command and a callback. The callback is then invoked whenever the TC string changes, usually as a result of user interaction with a cookie notice.

We develop a Chrome extension [15] that hooks the JavaScript `__tcfapi` function to inject custom cookie consent settings. Upon `document_start` [1], our extension immediately executes a script that defines the `set` property descriptor

[1]This results in our extension loading as the first element inside the `<html>` tag, ensuring that all calls to `__tcfapi` are intercepted. See https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/manifest.json/content_scripts.

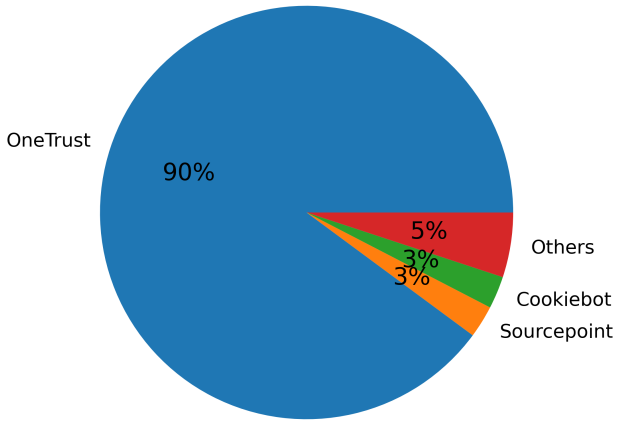


Figure 4: **Distribution of CMPs Used by Websites:** *OneTrust* is responsible for about 90% of observed CMP banners during our crawl of 255 websites. To detect OneTrust, we check whether the `OneTrust` API [18] is defined after page load.

OptanonConsent:Object
isGpcEnabled:"0"
datestamp:"Fri+Aug+18+2023+11:53:27+GMT-0700+(Pacific+Daylight+Time)"
version:"6.21.0"
isIABGlobal:"false"
hosts:""
consentId:"28b99caf-fedd-47ec-bc1f-208c2420cbe9"
interactionCount:"1"
landingPath:"NotLandingPage"
groups:"C0004:0,C0003:0,C0002:0,C0001:1"

Figure 5: **OptanonConsent groups Field:** Upon user consent, OneTrust sets an `OptanonConsent` cookie that contains a serialized JavaScript object. This object encodes user consent within its `groups` field. In this example, only the "C0001" Cookie Group ID is enabled, and the other 3 groups are disabled.

of `__tcfapi` to wrap itself whenever it is defined. This wrapper function modifies the original callback by intercepting the legitimate TC string and injecting custom consent settings. Thus, we can programmatically provide varying degrees of consent without having to interact with complex cookie notices.

This extension has been thoroughly tested using the TCF *CMP Validator* Chrome extension [4] on a variety of different CMPs. However, we find that only 9 of 255 (3.5%) websites actually implement the `__tcfapi`. Therefore, we also target the *OneTrust* API.

7.2 OneTrust Implementation

Through our crawl of 255 websites, we find that *OneTrust* is responsible for 90% of observed CMP banners (See Figure 4). Thus, by targeting the OneTrust CMP, we can verify compliance for *most* websites that use CMPs.

OneTrust stores user consent in the form of an `OptanonConsent` cookie. Specifically, the `groups` field encodes the categories of cookies that the user has consented to. An example is shown in Figure 5. Note that each group in the `groups` field is encoded by an alphanumeric ID (e.g., "C0001").

These *Cookie Group IDs* each map to a cookie type (e.g., "C0001" may map to *Strictly Necessary* cookies) [19]. While OneTrust does not provide an API to determine these ID to category pairs, we find that all OneTrust cookie notices

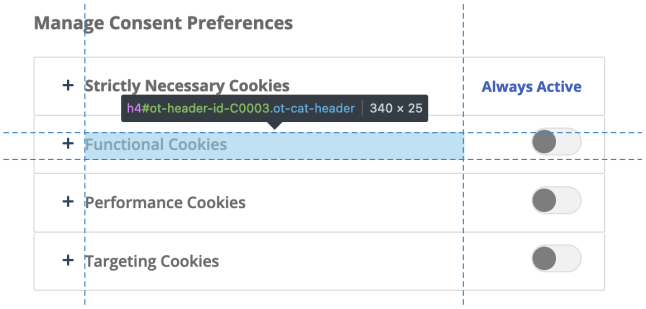


Figure 6: **Mapping Cookie Group ID to Cookie Type:** Consent labels in OneTrust cookie consent interfaces are identified with the HTML `id` attribute (e.g., "ot-header-id-C0003"). In this example, the "C0003" Cookie Group ID maps to the "Functional Cookies" category.

have the same underlying structure that can be used to determine this mapping. Specifically, we first select all HTML elements with an `id` that starts with "ot-header-id-". The text following this prefix is the cookie group ID. (For example, "ot-header-id-C0001" has the cookie group ID "C0001"). The category string can then be extracted with the `innerText` property (See Figure 6). We develop a JavaScript snippet that can be executed within the browser to easily find these mappings [14].

Thus, by decoding, modifying, and re-encoding the `OptanonConsent` cookie, we can simulate more granular consent options beyond simply accepting or rejecting all cookies. The immediate next step will be to simulate a user accepting all cookies *except* for tracking cookies; then, check whether any tracking cookies are sent in subsequent HTTP requests. If tracking cookies are sent, then the website violates GDPR for not respecting the users' choice.

The full algorithm is as follows:

1. Get mappings of cookie group IDs to cookie categories.
2. Decode, modify, and re-encode the `groups` field of the `OptanonConsent` cookie to accept all cookies except for tracking cookies.
3. Refresh the page for our changes to take effect and save HAR file.

After data collection, the HAR files can be analyzed for tracking cookies. This same technique can also be applied for other cookie types; however, an automated cookie classification method is needed in order to verify the type of each cookie.

8 Discussion

Current results are obtained from a small sample of 255 websites. Therefore, we plan to extend our analysis to a larger sample size for a more comprehensive measurement study. Additionally, to best measure GDPR compliance, crawls should be conducted from an EU member state as some websites show different banners depending on the geographical location of the user.

Currently, *Accept/Settings* websites can only be verified if they implement either `__tcfapi` or *OneTrust*. To verify compliance for all *Accept/Settings* websites, a more general solution must be created that can accurately click through desired settings of cookie notices. For example, we can supplement or replace our use of *BannerClick* with *CookieEnforcer* [13] to directly interact with a wider variety of cookie notices. This will significantly increase the number of websites we can verify since about 79% of websites require multiple clicks to opt-out of all cookies [13].

Lastly, our results so far only indicate whether a website's use of tracking cookies is in violation of web cookie privacy laws. For a fully comprehensive method to determine compliance, a behavioral cookie classification algorithm must be created (See Section 2). We hypothesize that such a classifier would be able to categorize *Strictly Necessary* and *Functionality* cookies as these cookies affect the visible behavior of a website [5].

9 Conclusion

In this paper, we present a framework to verify whether a given website's use of cookies is compliant with current privacy laws. To automate this framework we develop a Selenium-based web-crawler that interacts with *Accept/Reject* cookie notices and a Chrome extension which intercepts the `__tcfapi` function. We also propose an algorithm to verify compliance for the OneTrust CMP. We find that 100 of 255 (39%) of websites violate GDPR and CCPA by retaining tracking cookies after the reject button is clicked.

References

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

[2] California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa, October 2018.

[3] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. Cookie Swap Party: Abusing First-Party Cookies for Web Tracking. In *Proceedings of the Web Conference 2021*, WWW '21, pages 2117–2129, New York, NY, USA, June 2021. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/3442381.3449837.

[4] CMP Validator. https://chrome.google.com/webstore/detail/cmp-validator/ffhhjklgcfabkpholngojpkijlafjooc.

[5] ICC UK Cookie Guide. https://www.cookielaw.org/wp-content/uploads/2019/12/icc_uk_cookiesguide_revnov.pdf, November 2012.

[6] Cookie-Script. https://cookie-script.com/.

[7] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). http://data.europa.eu/eli/dir/2002/58/oj/eng, July 2002.

[8] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj.

[9] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. Did I delete my cookies? Cookies respawning with browser fingerprinting. http://arxiv.org/abs/2105.04381, May 2021.

[10] Hana Habib, Megan Li, Ellie Young, and Lorrie Cranor. "Okay, whatever": An Evaluation of Cookie Consent Interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, pages 1–27, New York, NY, USA, April 2022. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/3491102.3501985.

[11] Xuehui Hu, Nishanth Sastry, and Mainack Mondal. CCCC: Corralling Cookies into Categories with CookieMonster. In *Proceedings of the 13th ACM Web Science Conference 2021*, WebSci '21, pages 234–242, New York, NY, USA, June 2021. Association for Computing Machinery. https://dl.acm.org/doi/10.1145/3447535.3462509.

[12] JustDomains Blocklist. JustDomains, August 2023. https://github.com/justdomains/blocklists.

[13] Rishabh Khandelwal, Asmit Nayak, Hamza Harkous, and Kassem Fawaz. Automated Cookie Notice Analysis and Enforcement. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1109–1126, 2023. https://www.usenix.org/conference/usenixsecurity23/presentation/khandelwal.

[14] Maxwell Lin. OneTrust ID Mapping. https://gist.github.com/maxwellmlin/6ad1f179b477f741aa8f01e06283b76a, July 2023.

[15] Maxwell Lin. __tcfapi Hook. https://github.com/maxwellmlin/tcfapi-hook, July 2023.

[16] Maxwell Lin and Karen Wang. Artifacts for "Understanding Website Compliance with Web Cookie Privacy Laws". https://github.com/stars/maxwellmlin/lists/cookie-compliance.

[17] 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. http://arxiv.org/abs/1911.09964, February 2020.

[18] CookiePro Knowledge: Banner SDK JavaScript API. https://community.cookiepro.com/s/article/UUID-d8291f61-aa31-813a-ef16-3f6dec73d643?language=en_US.

[19] Article: Finding the Cookie Group IDs. https://my.onetrust.com/articles/en_US/Knowledge/UUID-8102e851-d860-d465-d8d6-b1d636d68eb9.

[20] Ali Rasaii, Shivani Singh, Devashish Gosain, and Oliver Gasser. Exploring the Cookieverse: A Multi-Perspective Analysis of Web Cookies. In Anna Brunstrom, Marcel Flores, and Marco Fiore, editors, *Passive and Active Measurement*, volume 13882, pages 623–651. Springer Nature Switzerland, Cham, 2023. https://link.springer.com/10.1007/978-3-031-28486-1_26.

[21] Selenium. https://www.selenium.dev/.

[22] Than Htut Soe, Cristiana Teixeira Santos, and Marija Slavkovik. Automated detection of dark patterns in cookie banners: How to do it poorly and why it is hard to do it any other way. http://arxiv.org/abs/2204.11836, April 2022.

[23] Helen Wu. Cookie-Script Database. https://github.com/ReLCS/cookie-compliance/blob/main/NCSAS/results.json, March 2023.