

A Privacy Measure Turned Upside Down? Investigating the Use of HTTP Client Hints on the Web

Stephan Wiefeling*
clienthints@swiefling.de
swiefling.de &

H-BRS University of Applied Sciences
Sankt Augustin, Germany

Marian Hönscheid
marian.hoenscheid@smail.inf.h-
brs.de

H-BRS University of Applied Sciences
Sankt Augustin, Germany

Luigi Lo Iacono
luigi.lo_iacono@h-brs.de
H-BRS University of Applied Sciences
Sankt Augustin, Germany

ABSTRACT

HTTP client hints are a set of standardized HTTP request headers designed to modernize and potentially replace the traditional user agent string. While the user agent string exposes a wide range of information about the client’s browser and device, client hints provide a controlled and structured approach for clients to selectively disclose their capabilities and preferences to servers. Essentially, client hints aim at more effective and privacy-friendly disclosure of browser or client properties than the user agent string.

We present a first long-term study of the use of HTTP client hints in the wild. We found that despite being implemented in almost all web browsers, server-side usage of client hints remains generally low. However, in the context of third-party websites, which are often linked to trackers, the adoption rate is significantly higher. This is concerning because client hints allow the retrieval of more data from the client than the user agent string provides, and there are currently no mechanisms for users to detect or control this potential data leakage. Our work provides valuable insights for web users, browser vendors, and researchers by exposing potential privacy violations via client hints and providing help in developing remediation strategies as well as further research.

CCS CONCEPTS

• **Networks** → **Network measurement**; • **Security and privacy** → **Web application security**; **Privacy protections**.

KEYWORDS

HTTP client hints, web measurement, privacy, tracking, security, risk-based authentication

ACM Reference Format:

Stephan Wiefeling, Marian Hönscheid, and Luigi Lo Iacono. 2024. A Privacy Measure Turned Upside Down? Investigating the Use of HTTP Client Hints on the Web. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024)*, July 30-August 2, 2024, Vienna, Austria. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3664476.3664478>

*Stephan Wiefeling started this research in March 2022 while working at H-BRS University of Applied Sciences. He was not involved in this project between July 25th and October 14th, 2022, where he worked at a Big Tech company. All opinions expressed are his own and not necessarily those of his current or former employers.

ARES 2024, July 30-August 2, 2024, Vienna, Austria

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *The 19th International Conference on Availability, Reliability and Security (ARES 2024)*, July 30-August 2, 2024, Vienna, Austria, <https://doi.org/10.1145/3664476.3664478>.

Table 1: Example UASs for different web browsers, devices, and OSs

Chrome 102 on Desktop PC with Windows 10 Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
Firefox 117 on iPhone with iOS 16 Mozilla/5.0 (iPhone; CPU iPhone OS 16_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) FxiOS/117.2 Mobile/15E148 Safari/605.1.15

1 INTRODUCTION

The user agent string (UAS) [47] has long played an important role for websites to obtain information about the client’s browser, operating system (OS) and device information. In web browsers, it is commonly a string with the format `Mozilla/5.0 (<system-information>) <platform> (<platform-details>) <extensions>` [43] (see Table 1). Websites use the UAS for security purposes to detect bots [61] or legitimate users [65], but also in a privacy-invasive way to track users across websites [20, 49]. As a countermeasure, initiatives by major web browser vendors recently started to deprecate the static and information-rich UAS and replacing it with Hypertext Transfer Protocol (HTTP) client hints (CHs) [46, 59, 62]. These aim to increase user privacy by default, submitting a low-entropy version of the UAS. To obtain high-entropy features—which are the default in the UAS—websites need to explicitly request them from a user’s web browser via a dedicated HTTP response header. Users will not directly notice this interaction, unless they inspect the HTTP responses in the browser’s developer tools. Still, this proactive approach reveals the type of information that websites aim to collect about users.

Apart from these essentially privacy-respecting innovations, however, HTTP CHs can provide much more information about the client than the UAS when explicitly requested by the server. This means that they can potentially be used to track users even better than with the UAS. It is therefore very important to understand whether HTTP CHs are misused for identification and tracking purposes in practice. If this is done without users’ consent, which would be the case when a user visits a website for the first time without sending active consent, this could potentially violate current privacy laws like the GDPR [21] and CCPA [56].

Besides that, there are also legitimate and legal purposes for requesting HTTP CH data, like for security measures to prevent unauthorized access [67]. Very relevant here is the case of risk-based authentication (RBA) [18, 41, 60, 63, 65, 66], which is recommended

by national authorities [5, 10, 25, 45] and often uses the UAS alongside the IP address to determine in a risk estimation whether the legitimate user is signing in. Since the UAS provides less information with its deprecation, we hypothesize that RBA-using websites might adopt HTTP CHs instead. As a major difference, the HTTP CH requests can now give indications of potential features used for the risk estimation, which was not possible with the UAS before. Learning this from other online services can help RBA system designers to focus on effective features to protect their users from attacks, such as credential stuffing [2] and password spraying [30].

Research Questions. HTTP CHs and their effects on online services and users in the Web have not been studied intensively in literature before. Understanding them is, however, very important, as their usage might have consequences for online users' privacy and security. To close this research gap, we formulated the following research questions.

RQ1: Adoption of HTTP CHs

- How do website start pages adjust to the transition from UAS to HTTP CHs?
- Do websites request different HTTP CHs on the start page and on the login page?
- How do login pages using RBA adopt to the transition from UAS to HTTP CHs?
- How wide-spread are HTTP CH requests on login pages in practice, also regarding requests from embedded third party domains and known web trackers?

RQ2: Requested HTTP CHs

- What HTTP CH data do websites and known web trackers request on login pages?
- What HTTP CH data do websites request on their login page when using RBA?
- Do geographical location and the Internet Service Provider (ISP) make a difference in how websites request HTTP CH data?
- What HTTP CHs do different categories of websites request and to what level of detail?
- What HTTP CHs do different RBA-instrumented websites request and to what level of detail?

RQ3: Impact

- How interconnected are HTTP CH requests by third party domains with different login pages?
- How much information do web browsers provide when receiving HTTP CH requests?

Contributions. By answering the research questions, we contribute the following to the current body of knowledge:

- **First historical overview of HTTP CH usage on the Web:** Using historical crawling data, we provide an overview of how the adoption of HTTP CHs on websites of the full Tranco list¹ with $\approx 8\text{M}$ of the most popular websites worldwide (Tranco 8M) increased since its first appearance in July 2017. Combined with our own crawling data of 327K identified login pages on the Tranco 8M Uniform Resource Locators (URLs), we also show that websites tend to request

more high-entropy HTTP CHs on login pages compared to their corresponding start page.

- **First large-scale analysis of HTTP CH usage on login pages worldwide:** Using our own crawling data, we show the type and amount of user information that 327K login pages request with HTTP CHs. We identify differences in requested level of detail across different website categories. We also show that websites known to use some sort of RBA request higher level of detail than those without.
- **HTTP CH analysis on RBA-using websites:** We show the type of HTTP CH features that RBA-using websites requested and potentially used for their RBA risk estimation.
- **Influence of third parties, trackers, regions and ISPs:** We provide an overview of how third parties and known trackers request HTTP CH data, and how they are connected to other websites. We show that third parties request a high to very high level of detail from the user. We also show that they are interconnected with 18.1% of the crawled 1,000 most popular (Tranco 1K) and 13.8% of the 5,000 most popular (Tranco 5K) websites, allowing user profiling across popular websites. Our results also show that geographical region and ISP have an effect on whether websites request HTTP CH data.
- **Open Data:** We provide the HTTP CH results from the crawled login pages as an open data set². This can be used to reproduce our results and to conduct own research on HTTP CHs. For researchers following ethical standards, we also provide our data set of login page URLs on request³.

Our findings support developers, and security and privacy engineers to gain insights on how HTTP CHs are used in practice and how they are used in relation to tracking and RBA. Researchers also get an overview of the popularity of HTTP CHs on the Web, and obtain an open data set to do more research on HTTP CHs in practice. Privacy experts and policy makers also gather information on whether HTTP CHs might be used to breach users' privacy, although they were intended and introduced as a privacy measure.

2 HTTP CLIENT HINTS (HTTP CHS)

Before introducing the study, we first give an overview of HTTP CHs and their implementation in web browsers.

2.1 Background

HTTP CHs [62] aim to be a privacy-preserving measure to request client information from a user's web browser. In contrast to the UAS, a web browser only provides more fine-grained browser, OS, and device information when a server sent an Accept-CH HTTP response header in a previous protocol message exchange. This header contains the type of information that the client should provide [62] (see Table 3 for the list of all possible HTTP CHs). The client will cache this header and remember its values. Therefore, in all subsequent client requests, the client's browser will always provide the requested information (e.g., the fine-grained browser

²Available at <https://github.com/das-group/http-client-hints-dataset>

³We decided not to publish this data openly to limit potential misuse by attackers (e.g., automated credential stuffing attacks), although it could be possible that smart attackers already own such URL data themselves.

¹Available at <https://tranco-list.eu/list/GZP2K>

and OS version). Note that by means of this behavior, the website can recognize whether a particular user has already visited the website in the past. Beyond that, servers can also request device and network information that was not included in the UAS, like the client-measured round-trip time (RTT) or the display resolution. Note that some of this information could also be accessed via JavaScript functions. However, HTTP CHs do not require JavaScript, so preventing their data submission via deactivated JavaScript functionality is no longer possible.

With the roll out of HTTP CHs in web browsers, Chromium-based browsers started to deprecate the UAS and changed it to a low entropy version that makes it less distinguishable from other users [59]. Therefore, to obtain higher entropy information, servers have to request it via HTTP CHs.

The first draft of the idea was published in March 2013 [26]. It became an IETF draft in November 2015 [27] and then developed to an experimental RFC in February 2021 [28], but it is not an RFC standard yet. The specification of HTTP CHs was defined in a W3C draft community group report [62], but is not a W3C standard yet. Nevertheless, popular web browsers like Chrome and Edge already support HTTP CHs, which affects more than 75% of web users worldwide [12].

2.2 Implementation in Web Browsers

HTTP CHs have become a fundamental feature in contemporary web browsers. Chrome first implemented support for HTTP CHs in version 85 [58], which was released in August 2020 [14]. Microsoft Edge, built on the Chromium engine, also integrated support for HTTP CHs in alignment with Chrome. Similarly, Brave, also Chromium-based, synchronized its implementation with Chrome and Edge. Safari and Firefox stand alone among the major browsers in not offering support for HTTP CHs.

From this, it becomes apparent that HTTP CHs are supported by major browsers. The fact that this support is offered by the most widely used browsers with a desktop market share of 78% (see Table 2 and Section 6.2) suggests that widespread availability can be assumed in practice. However, this development remains opaque to users because none of the supporting browsers offer users the ability to control HTTP CHs. For example, there is no way to limit third-party requests to low-entropy hints or to disable them altogether. In principle, this poses a significant potential for abuse, as the HTTP CHs that are originally intended to enhance privacy can inadvertently and possibly even unlawfully be exploited to track web users. Therefore, it is important to understand how HTTP CHs are used in practice in order to take appropriate measures to enhance transparency and control for web users.

3 STUDYING HTTP CH USAGE

To investigate HTTP CHs in the wild, we used various data collection and analysis methods. We outline them in the following, including ethical and legal considerations regarding the work.

3.1 Extracting Historical Crawling Data

The introduction of HTTP CHs goes back to the year 2013, which was long before we decided to study this measure. To trace the history of HTTP CHs in the web, we used data from the HTTP

Archive [33]. The data contains monthly to half-monthly crawls of millions of popular start pages on the Internet as determined by the Chrome User Experience Report (CrUX) data set [13]⁴. The data used consisted of the URL, the timestamp of when it was crawled, and the received HTTP response headers for this crawl. A Chrome desktop browser with an empty browser cache crawled each URL from Google cloud instances inside the US [33]. The crawler then stored the results in the data set. As the data sets for each crawl are quite huge (approx. 1 TB per crawl), we queried the data using Google’s BigQuery [24]. In so doing, we extracted the first sent HTTP responses from all crawled websites that ever sent the Accept-CH HTTP response header. We focused on the first response to make sure that no HTTP CH headers were cached by the client at that time.

We started our data analysis with the December 2023 set and went back in time until we found no more HTTP CH headers in the data. As a result, we obtained the historical data of all start pages crawled by the HTTP Archive that ever used HTTP CHs.

3.2 Data Collection on Login Pages

The crawled historical data only included the start pages of websites. However, some websites may request other HTTP CH data during the login process, e.g. to prevent account takeover. Also, different privacy jurisdictions like GDPR [21] and CCPA [56] might limit the amount of data collected by online services, including HTTP CHs. For this reason, we additionally crawled HTTP CH header data from login pages of the Tranco list websites from four different regions on three different continents (North America: Johnstown, Ohio, USA; Europe: Frankfurt and Biere, Germany; Asia: Singapore) and two different ISPs, which were Amazon Web Services (AWS) and Deutsche Telekom (DT). We used this data to determine differences in the HTTP CH behavior compared to the start page, and across different regions and ISPs.

To determine login pages, we accessed the URLs from the Tranco 8M list from June 21st, 2022 and determined login page URLs using an automated process (see Section 3.2.1). With the final list of login page URLs, we started the crawling process (see Section 3.2.2).

For the whole login page detection and crawling process, we used the Chromium browser version 103, which is compatible with HTTP CHs [12, 59]. As websites might use bot detection mechanisms [37, 64], we automated the browser behavior with a patched version of the Puppeteer framework [64] to appear as a human-like user.

3.2.1 Identifying Login Pages. It is possible that websites request different client information in the login context. For instance, websites might not need the full browser version to generate basic traffic statistics. In case of RBA, e.g., a website might still be interested in the full browser version, as this might help to identify the legitimate user accessing an online account. Therefore, we decided to analyze both the start page and the login page of a website, to be able to spot differences.

The website start page appears when the URL of a Tranco URL entry is accessed. The login page is the web page where the login credentials (most commonly username and password) are requested

⁴The HTTP Archive integrated the CrUX URLs in January 2018. We did not find information on the URL data set used before that.

Table 2: Overview of HTTP CH support across different devices and Browsers.

Browser Platform	Chrome			Brave			Firefox			Edge			Samsung Internet	Safari	
User-Agent (Low Entropy)	●	○	●	●	○	○	○	○	○	●	○	●	○	○	○
OS (Low Entropy)	●	○	●	●	○	○	○	○	○	●	○	●	○	○	○
Prefers Mobile UX	●	○	●	●	○	○	○	○	○	●	○	●	○	○	○
User Agent (High Entropy)	●	○	●	○	○	○	○	○	○	●	○	●	○	○	○
User Agent Brand List	●	○	●	○	○	○	○	○	○	●	○	●	○	○	○
OS (High Entropy)	●	○	●	●	○	○	○	○	○	●	○	●	○	○	○
Platform Architecture	●	○	○	○	○	○	○	○	○	●	○	●	○	○	○
CPU Bitness	●	○	○	○	○	○	○	○	○	●	○	●	○	○	○
Device Model	●	○	●	○	○	○	○	○	○	●	○	●	○	○	○
Device Form Factor	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Prefers Reduced Data	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Viewport Width	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Client DPR	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Client's RAM	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Round-Trip Time	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Bandwidth	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Network Profile	●	○	●	○	○	○	○	○	○	●	○	●	●	○	○
Light/Dark Mode	●	○	○	○	○	○	○	○	○	●	○	●	●	○	○
Prefers Reduced Motion	○	○	○	○	○	○	○	○	○	●	○	●	●	○	○
Reduced Transparency	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Contrast Preference	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Forced Colors	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

: Desktop PC (macOS Catalina), : iOS 16 (Mobile), : Android 10 (Mobile)
 Versions: Chrome 116, Brave 116, Firefox 117 (iOS, Android) and 116 (macOS), Edge 116, Samsung Internet 111, Safari 605 (macOS) and 604 (iOS)

by the website. This login page can be located at the same or a different URL as the start page, whereby the latter URL is not included in the Tranco list. Manually inspecting all 8M sites to determine the login page URL would be rather impractical, and also dangerous as some of them might contain illegal or harmful content. Therefore, based on our observations on popular websites, we created a systematic approach to determine the login URLs of the websites.

For each domain of the Tranco list, we conducted a three step process, which we describe in the following. We explain the process for the domain `example.com`. We considered Transport Layer Security (TLS) mandatory for login pages, as it protects login credentials from interception. Therefore, we only sent HTTP requests to the `https://` prefix version of a website. We ran this process on the server infrastructure of our university using a university IP address (location: Sankt Augustin, Germany).

Step 1: Collecting Potential Login URLs. As a first step, we determined a list of URLs that could potentially be a login URL. We designed this procedure based on our previous experiences with websites and content crawling from them. We tested and improved this procedure for one month, to make it as accurate as possible. We first collected three types of potential login URLs and chose the best candidate afterwards. These types were the following:

(i) We opened `https://example.com`, waited until the page was fully loaded, and parsed the returned HyperText Markup Language

(HTML) [9] code to allow further analysis. We then added all URLs contained in HTML `<a>` tags together with the link text to our candidate list. We assume that in most cases the login URL should be among these links, when a website has a login page. (ii) We sent an HTTP request to `https://example.com/login`, as we observed this pattern often at login pages of websites. When we received the HTTP status code `200 OK`, which indicates that the URL points to an existing resource, we added the URL with the page title retrieved from the HTML `<title>` tag—as we do not have a link text here—to our candidate list. This should have further increased the probability that we found a login page. (iii) As a last resort, we opened the Google search engine and entered the search query `login site:example.com`. This query should only return pages containing the word `login` at the `example.com` website. We added the results with the link text to our candidate list as well.

Step 2: Scoring Login URLs. After Step 1, we had a candidate list of potential login URLs. To identify the best candidate, we scored the URLs by checking both URL and the text associated with it. We defined the criteria after a discussion of two researchers who made observations at various websites in the wild.

We had a list of positive indicators, which consisted of different variations of the words `“login”` and `“sign in”` in German and English. We checked both languages, as we browsed from a German IP address, so language variations were possible. Based on the matches

with those indicators, we gave different scores: Matches of both text and URL received three points, matching the URL but not the text received two points, and matching the text but not the URL received one point. We found the URL a better indicator than the link text for rare cases where websites displayed different languages than our tested ones.

We also found negative indicators. These were when the words “help”, “premium”, “pro”, or “forgot password” appeared in a link text or URL, with the URL being a higher indicator. We experienced that such URLs rather opened a registration, password recovery, or help desk page than a login page. We assumed that a higher number of negative indicators decreased the probability of a crawled URL being a login URL. Therefore, based on the matches with these negative indicators, we reduced the following values from the current scores using a ranking-based approach: Matches of both text and URL lost three points, matching URL but not the text lost two points, and matching text but not the URL lost one point.

Step 3: Determining Best Candidate. After the scoring process, we determined the best candidate for the login URL as the URL with the highest score. In case of multiple candidates with the highest score, we only focused on the one that appeared first in the list. We assume that links to login pages appear on the top area of a website, so this link would be the first in our list based on how we ordered it with our process.

With this approach, we determined 419K login page candidates on June 22nd, 2022, which we used for the crawling process. We verified the validity of our approach by reading the most popular URLs of the determined login pages, comparing them with our previous experiences on these websites, and checking a small subset of them whether they lead to the login page.

3.2.2 Crawling Process. We crawled the login page candidates on a monthly basis, from August 7th, 2022 to December 21st, 2023⁵. We assumed that a monthly time interval should be sufficient to identify changes in the adoption of HTTP CHs. Also, as crawling a huge number of URLs produces a large amount of data, we had to limit our data collection to a reasonable level.

For each login page URL, our crawler started a new Chromium browser session with empty cache and storage. We did this to make sure that no HTTP CH requests were cached. Then, Chromium initiated an HTTP request to the URL as a human-like Internet user, and the crawler recorded the Accept-CH value, if present, of the server’s HTTP response. We also recorded all browser-initiated third party requests during the page loading, and recorded a present Accept-CH response as well.

To mitigate being detected as a bot, we crawled the data using a cluster of six different servers with six different IP addresses located in three different zones in the same data center. We furthermore limited the network throughput to 5 MBit/s and crawled the URLs in random order. Each server had 12 GB SSD, 4 GB RAM and one CPU core. Crawling the URLs took around three to four days per iteration.

3.2.3 Filtering Login Pages. We aimed to investigate which websites request HTTP CHs on their own login page. Websites might

⁵For technical reasons, we had crawling gaps of one (October 2022) and two months (October/November 2023). However, the impact should be minimal (see Section 8).

change over time. To ensure a high data quality, we applied additional filtering after the crawling. Therefore, we made sure that the domain name of the login URL matched the domain name of the Tranco list entry. In so doing, e.g., `https://login.example.com` was considered the valid login URL of `https://example.com` while `https://login.example.org` was not.

3.2.4 Extracting Third Party Domains. We extracted the domain names from all crawled URLs and all their resources that were requested. To identify third party domains, we extracted the domain names from all URLs and then kept all URLs which did not belong to the same domain. We did this to only include URLs that are external resources that are likely not directly related to the original website (e.g., a subdomain pointing to an internal content delivery network).

3.3 Data Analysis

We used the following methodology to analyze our collected data.

3.3.1 Aligning the Two Data Sets for Comparison. To allow comparisons between the historical crawling and login pages crawling data sets, we only considered the domains in the analysis that were present in both of them. We took the data from December 2023 from both HTTP Archive and our login pages crawl, to ensure that we have the same state of crawling data. The crawling from the HTTP Archive was from December 13th to December 23rd, 2023 and the login pages crawling from December 16th to December 21st, 2023. As the total amount of HTTP CH websites did not change to a large degree in that month, we assume that the changes of websites using HTTP CHs were minimal. We then extracted a union set of websites based on the domain names which were present in both data sets. We took the domain names as some websites have a dedicated subdomain for logins (e.g., `accounts.google.com` for `google.com`). Then, we compared the requested HTTP CHs for the start page and the login page. Our union subset contained 1,938 websites that used HTTP CHs.

3.3.2 Identifying HTTP CHs. We processed each Accept-CH HTTP response header sent by the websites. This header included the list of HTTP CHs requested by the server. The RFC [29] and W3C draft [62] classified HTTP CHs into *valid*, *deprecated*, and *experimental* ones (see Figure 4). We took their classification and considered HTTP CHs *not valid* when they did not appear in these documents. HTTP CHs classified as *experimental* also belonged to the *valid* ones.

3.3.3 Determining Level of Detail. To analyze the user trackability on websites, we rated each of the available valid and deprecated HTTP CHs by their level of detail that they potentially revealed about a user. Based on related work by Alaca and van Oorschot [3], and Wiefeling et al. [63], we classified the HTTP CHs on a scale with the ratings *very low*, *low*, *medium*, *high*, and *very high* (see Table 3). In cases in which both publications did not give a rating on a corresponding feature, we made the decision based on related features with similar expected entropy. Based on the classifications of all requested HTTP CHs, we determined the maximum level of distinguishing information per website.

Table 3: Classification of the level of detail for each available HTTP CH (as defined in the RFC [29] and specification [62])

HTTP CH	Header	Level of Detail	Source	Explanation
User Agent				
User Agent (High Entropy)	Sec-CH-UA-Full-Version	Very High	[63]	User agent’s full semantic version string
User Agent Brand List	Sec-CH-UA-Full-Version-List	Very High	[63]	Full version for each brand in the user agent’s brand list
OS (High Entropy)	Sec-CH-UA-Platform-Version	High	[3, 63]	Operating system version
Device Model	Sec-CH-UA-Model	Medium	[3]	Device model
OS (Low Entropy)	Sec-CH-UA-Platform	Low	-	Operating system (low entropy hint)
User Agent (Low Entropy)	Sec-CH-UA	Low	[63]	Branding and major version
CPU Bitness	Sec-CH-UA-Bitness	Very Low	-	CPU architecture bitness (mostly 32 or 64 bit)
Device Form Factor	Sec-CH-UA-Form-Factor	Very Low	[63]	Form factor of device (Automotive, Mobile, Tablet, TV, VR, XR, Unknown)
Is Windows64	Sec-CH-UA-WoW64	Very Low	-	Binary runs on 64-bit Windows
Platform Architecture	Sec-CH-UA-Arch	Very Low	-	Platform architecture (mostly x86 or ARM)
Prefers Mobile UX	Sec-CH-UA-Mobile	Very Low	-	Prefers a mobile user experience (boolean value)
User Preference Media				
Contrast Preference	Sec-CH-Prefers-Contrast	Very Low	-	Preference for contrast
Forced Colors	Sec-CH-Forced-Colors	Very Low	-	Forces a color scheme
Light/Dark Mode	Sec-CH-Prefers-Color-Scheme	Very Low	-	Prefers light or dark color scheme
Prefers Reduced Motion	Sec-CH-Prefers-Reduced-Motion	Very Low	-	Reduced motion preference setting (either no-preference or reduced)
Reduced Transparency	Sec-CH-Prefers-Reduced-Transparency	Very Low	-	Prefers reduced transparency
Device Information				
Viewport Width	Viewport-Width	Very High	[63]	Layout viewport width
Width	Width	Very High	[63]	Desired resource width
Client DPR	Content-DPR	Low	[63]	Image device pixel ratio
Client’s RAM	Device-Memory	Low	[3, 63]	Approximate amount of available client RAM memory
Image DPR	DPR	Low	[63]	Client device pixel ratio
Network				
Bandwidth	Downlink	High	[3]	Approximate bandwidth of the client’s connection to the server
Network Profile	ECT	High	[3]	The effective connection type (“network profile”) that best matches the connection’s latency and bandwidth
Round-Trip Time	RTT	High	[63]	Application layer round trip time in milliseconds
Prefers Reduced Data	Save-Data	Very Low	-	Preference for reduced data usage (boolean value)

3.3.4 Website Classification. We classified the different HTTP CH websites into different categories. Inspired by related work [23], we queried the McAfee URL classification API [42] for all the websites that used HTTP CHs.

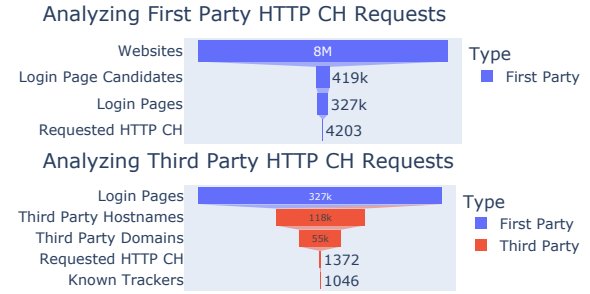
Gavazzi et al. [23] studied more than 200 popular websites inside the Tranco 5K regarding their RBA usage. We obtained their data set with permission, and used their results to classify the crawled websites from the Tranco 5K into those that used some form of RBA and those that likely did not use RBA.

From the third party requests, we identified known trackers using the EasyPrivacy list [19] that is commonly integrated in adblockers.

3.3.5 Comparisons. To identify significant differences between the HTTP CH usage of different website categories and types in RQ2, we used Pearson’s Chi-squared test for contingency table analysis (χ^2). We considered $p < 0.05$ as significant. For pairwise comparisons, we adjusted the p-values using Bonferroni correction and used $\alpha = 0.05$ as the start level.

3.4 Ethical and Legal Considerations

We do not have a formal institutional review board process at our university. Nevertheless, this study was carefully reviewed and approved by the university’s data protection officer. Furthermore, we made sure to minimize potential harm by complying with the

**Figure 1: Overview of the collected data and the results of our data analysis, based on the data from December 2023.**

ethics code of the German Sociological Association (DGS) and the standards of good scientific practice of the German Research Foundation (DFG). We also made sure to comply with the GDPR [21].

We used an external API to classify websites. Since the number of URLs was much lower than the original 8M URLs used for crawling, we kept the traffic to this external API to a minimum. Furthermore, we only sent one HTTP request to each website in the crawling process to keep the traffic as low as possible. We did not record any response data beyond the Accept-CH HTTP header, to make sure that no personally identifiable information was in the collected data. We also identified the login page URLs with an automated process to avoid that a person was accidentally confronted with harmful or illegal content.

3.5 Results

In the following sections, we show the results of our study ordered by the research questions. Unless otherwise noted, we used the results from the European region using the DT ISP. Figure 1 shows the amount of data we collected and the identified websites and third parties that requested HTTP CHs during our study.

4 ADOPTION OF HTTP CHS (RQ1)

We first describe the results regarding the HTTP CH adoption on website start pages and login pages, also those using RBA.

4.1 Website Start Pages (RQ1a)

The adoption of HTTP CHs on website start pages has increased since its introduction in 2013, with some notable peaks in 2020 and 2022 (see Figure 2). An HTTP CHs request first appeared on the punknews.org website in July 2017, and this was the only website using HTTP CHs for a long time. From July 2018, more websites started adopting HTTP CHs, and this was also the first time they appeared inside the Tranco 5K. The Tranco 1K followed a year after in September 2019.

In June 2020, Google wrote a blog article about HTTP CHs with the aim of soon replacing UAS with HTTP CHs in the Chrome browser [44]. After that, a first large peak in HTTP CH adoption was noticeable. At the end of August 2022, the Google Chrome team also warned developers that the UAS would change to a reduced version in October 2022 and that websites would have to switch to HTTP CHs to gather more user information again [46]. At this point, we observed the second big increase in HTTP CH adoption

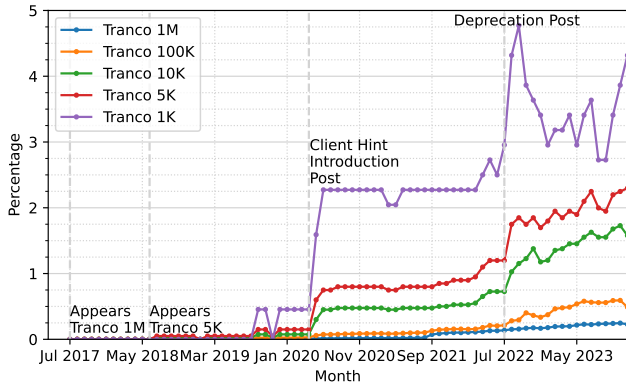


Figure 2: Observed HTTP CH adoption rates on start pages over time, grouped by rankings inside the Tranco list. The data is taken from the HTTP Archive crawling data, that crawled the whole Web each month.

on the Web. Overall, the use of HTTP CHs has remained at a very marginal level since its introduction.

4.2 Difference Between Login Page and Start Page (RQ1b)

For the crawled websites that use HTTP CHs, 54% behaved differently on start and login pages. The majority of them (90.6%) did not request HTTP CHs on their login page while requesting HTTP CHs on the start page. However, the login pages requested more HTTP CHs ($\bar{X} = 4.5$; $SD = 2.86$) than the start pages ($\bar{X} = 1$; $SD = 3.43$).

The behavior was not constant based on the Tranco ranking (see Figure 3a). While the majority of the 10 most popular websites did not show a different behavior, few websites until Tranco 100 requested different HTTP CHs. Websites between Tranco 100 and Tranco 1K then often requested different HTTP CHs. After that, we observed mixed behavior until between Tranco 10K and Tranco 100K. After that, most websites requested the same HTTP CHs on login page and start page again.

As the HTTP CHs requested by login pages seemed to be more detailed, we used our login pages data for the subsequent analyses. This allowed us to gain a larger and probably more realistic view of the HTTP CH adoption on the Web.

4.3 Login Pages Using RBA (RQ1c)

The amount of RBA websites that used HTTP CHs during our observation was low. Out of 73 websites with potential RBA usage, 12 (16.4%) sent an HTTP CH header. In terms of different online service providers, these were nine, while there were multiple domains belonging to the online service Amazon. We show more details on these services and their requested HTTP CHs in Section 5.4.

4.4 Distribution in Practice (RQ1d)

Aside from very popular online services, the overall HTTP CH adoption on login pages on the Web remained very low. While the the most popular websites from the Tranco list started using HTTP CHs, the adoption rate constantly declined with a lower rank on the

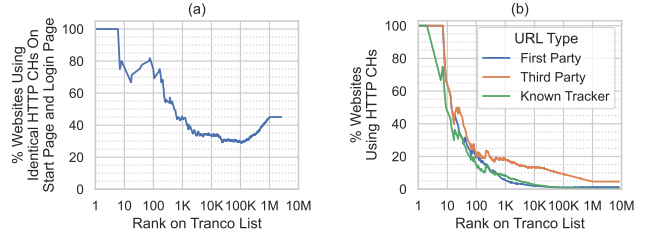


Figure 3: (a) Percentage of websites showing identical HTTP CH behavior on their login page compared to the start page. (b) Percentage of websites that use HTTP CHs based on their rankings inside the Tranco list. We calculated the usage based on the data from December 2023.

Tranco list (see Figure 3b). However, many of the websites included requests to third party domains, some of which also requested HTTP CHs. When taking these into account, HTTP CH data was collected by third parties on 18.1% of websites inside the Tranco 1K. About half of them were also known trackers at this point.

4.5 Discussion

Websites tended to requested more HTTP CHs on their login pages than on their start pages. It could be possible that they only requested additional information when it was necessary for their service (e.g., preventing attacks by malicious actors). However, it could also be possible that they hid their real HTTP CH usage from crawlers browsing the start pages only. Nevertheless, this suggests that the typical crawl of a website’s start page may not reflect real-world HTTP CH usage and that the collected user information could be more detailed. Following this, future research should include more URLs besides a website’s start page to obtain more realistic usage statistics on HTTP CHs.

HTTP CHs seemed not to be used by a lot of websites in practice. However, the websites that used HTTP CHs were very influential on people’s daily lives and have a large amount of traffic (e.g., Google, Facebook, and Amazon). Therefore, they could potentially still collect a lot of user data although the HTTP CHs were introduced as a measure to protect “users’ privacy [...] against covert tracking methods” [44]. This is also highlighted by the large amount of embedded third party resources requesting HTTP CHs. We did not identify all of them as trackers. Nevertheless, we assume that this is only a lower bound as trackers try to circumvent detection [39].

5 REQUESTED HTTP CHS (RQ2)

Requested HTTP CHs can vary largely in practice. In the following, we show the differences between the various website types and client attributes, as well as the level of detail that websites requested.

5.1 Used HTTP CHs (RQ2a, b)

The usage of HTTP CHs on Tranco 5K websites significantly differed from the full Tranco 8M crawled (see Figure 4b and c). There were also significant differences between Tranco 5K websites with and without RBA usage. Especially the *high entropy user agent* feature was significantly less used on the Tranco 8M than on the

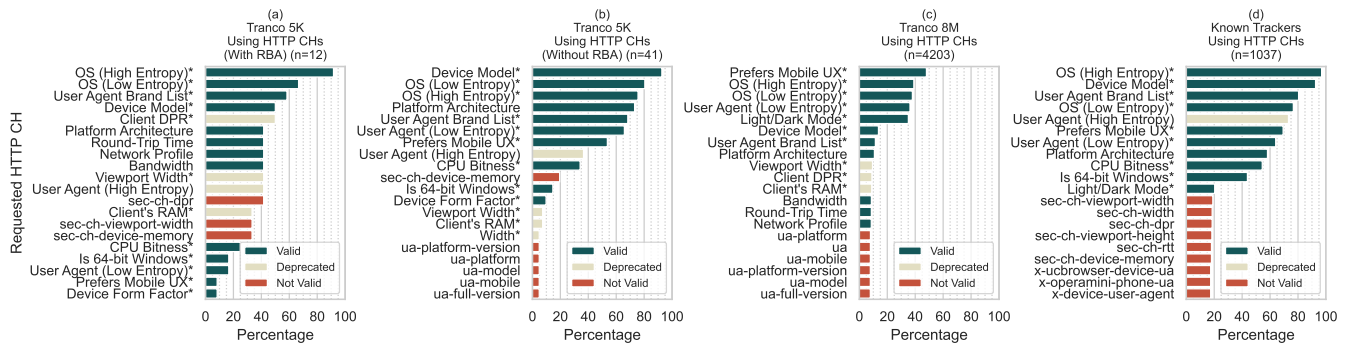


Figure 4: Top 20 HTTP CHs used among all websites and trackers that used HTTP CHs in the study. (*: Experimental HTTP CH)

Table 4: Significant χ^2 results and pairwise comparison p -values for valid HTTP CHs. We omitted p -values greater than 0.05.

HTTP CH	Level of Detail	p	χ^2	Tranco 8M /		
				5K with RBA/ 5K without RBA	5K with RBA	5K without RBA
Viewport Width	Very High	0.0006	14.8	0.0411	0.0027	-
User Agent Brand List	Very High	<0.0001	146.9	-	<0.0001	<0.0001
User Agent (HE)	Very High	<0.0001	104.4	-	<0.0001	<0.0001
Bandwidth	High	0.0001	18.6	0.0034	0.0012	-
OS (HE)	High	<0.0001	36.4	-	0.0018	<0.0001
Network Profile	High	0.0001	18.7	0.0034	0.0011	-
Round-Trip Time	High	0.0001	18.6	0.0034	0.0012	-
Device Model	Medium	<0.0001	216.1	0.0074	0.0038	<0.0001
OS (LE)	Low	<0.0001	35.1	-	-	<0.0001
Client's RAM	Low	0.0113	9.0	-	0.0412	-
User Agent (LE)	Low	0.0002	17.4	0.0220	-	0.0005
Client DPR	Low	<0.0001	26.7	0.0004	<0.0001	-
Device Form Factor	Very Low	<0.0001	110.6	-	0.0474	<0.0001
Platform Architecture	Very Low	<0.0001	166.6	-	0.0090	<0.0001
Prefers Mobile UX	Very Low	0.0175	8.1	0.0422	0.0417	-
Light/Dark Mode	Very Low	<0.0001	22.8	-	-	0.0001
Prefers Reduced Data	Very Low	0.0007	14.5	-	-	-
Is 64-bit Windows	Very Low	<0.0001	57.2	-	0.0074	<0.0001
CPU Bitness	Very Low	<0.0001	79.4	-	0.0336	<0.0001

HE: High Entropy, LE: Low Entropy

Tranco 5K. Table 4 shows the HTTP CHs that were significantly different among the website types, with the Tranco 5K collecting more high-entropy user data than the Tranco 8M. Some websites also requested invalid HTTP CHs, mostly because of typos or potential misinterpretation of the HTTP CH values (see Section 5.5).

Some HTTP CHs were significantly less popular among known trackers ($p < 0.0001$, see Figure 4c and d). These were (*Viewport Width*, *Bandwidth*, *Network Profile*, *Round-Trip Time*, *Image DPR*, *Client DPR*, *Prefers Reduced Data*, and *Device Form Factor*). The other HTTP CHs were not significantly different from the Tranco 8M.

5.2 Geographical Location and ISP (RQ2c)

We observed differences between regions. While our crawlings from North America and Asia both identified 4,077 websites requesting HTTP CHs, this was different in our Europe crawls. The crawl using the AWS ISP revealed 4,089 websites and the crawl with the DT ISP identified 4,203 websites. The ranking of requested HTTP CHs did not change between the different crawls.

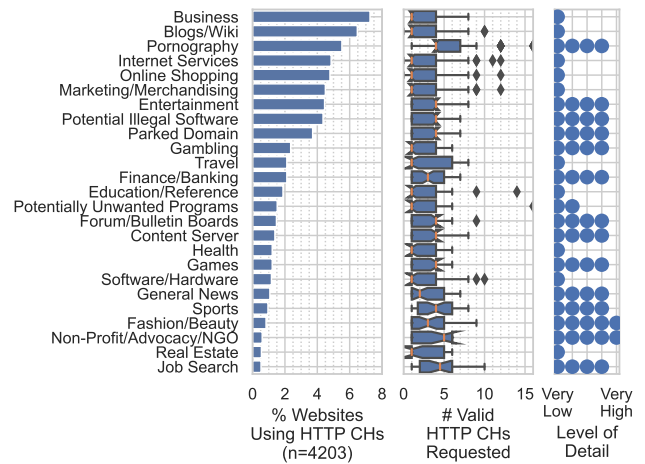


Figure 5: Top 25 categories of the Tranco 8M websites that used HTTP CHs, their percentage, their number of valid HTTP CHs requested, and the median level of detail they requested with the HTTP CHs. There is a maximum of 19 possible HTTP CHs. We also counted deprecated HTTP CHs as valid here, as they can still be valid in some browsers.

5.3 Level of Detail Requested (RQ2d)

Figure 5 shows the median requested level of detail for each website category. Many of the crawled website categories requested a high level of detail, like the fine-grained browser or OS name and version. These were also the ones that requested more than the median of one HTTP CH.

5.4 RBA Websites (RQ2e)

Figure 4a shows the most requested HTTP CHs from the websites that used RBA. Compared to the Tranco 5K without RBA, they significantly requested more HTTP CHs with high level of detail (see Table 4).

Table 5 lists the identified RBA-using websites and the information they requested from the user with the HTTP CHs. Except for Amazon, all websites requested user agent information. Amazon

Table 5: Overview of RBA websites that use HTTP CHs, and the HTTP CHs they requested (sorted from very high to very low level of detail).

Website	Classification	(Very) High	Medium	(Very) Low
amazon.[com, in, de, co.uk]	Online Shopping	OS (High Entropy)* Viewport Width Round-Trip Time Bandwidth Network Profile		OS (Low Entropy)* Client DPR Client's RAM
bedbathandbeyond.com	Online Shopping	OS (High Entropy) User Agent Brand List	Device Model	Platform Architecture
etsy.com	Online Shopping	OS (High Entropy) User Agent Brand List Round-Trip Time Bandwidth Network Profile		Prefers Mobile UX User Agent (Low Entropy) Platform Architecture CPU Bitness OS (Low Entropy) Client DPR Prefers Reduced Data Light/Dark Mode Client DPR
facebook.com	Social Networking	OS (High Entropy) User Agent Brand List Viewport Width	Device Model	Is 64-bit Windows Platform Architecture CPU Bitness OS (Low Entropy) Device Form Factor OS (Low Entropy)
google.com	Internet Services	OS (High Entropy) User Agent Brand List User Agent (High Entropy)	Device Model	
mail.ru	Portal Sites	OS (High Entropy) User Agent Brand List User Agent (High Entropy)	Device Model	OS (Low Entropy)
microsoft.com	Business	OS (High Entropy) User Agent (High Entropy)	-	OS (Low Entropy)
paypal.com	Finance/ Banking	OS (High Entropy) User Agent Brand List User Agent (High Entropy)	Device Model	Is 64-bit Windows Platform Architecture CPU Bitness User Agent (Low Entropy) Platform Architecture OS (Low Entropy)
p****b.com	Pornography	OS (High Entropy) User Agent Brand List User Agent (High Entropy)	Device Model	

*: Not on amazon.com

and Etsy, however, requested network information like the bandwidth and the client-originated RTT. These were also the RBA-using websites that requested the highest level of detail.

5.5 Discussion

Most websites tended to collect low entropy data, except for the OS where high and low entropy versions were requested at the same level. We assume that strong privacy regulations [21, 56] might had an impact here, so that online services avoided collecting high entropy data without explicit user consent. This could be an improvement of privacy compared to the UAS. Nevertheless, Tranco 5K websites tended to collect more fine-grained user data than the Tranco 8M. That is probably because their business models often depend mainly on collected user data [69] and that they have enough money and power to defend themselves against privacy lawsuits compared to smaller websites [68]. Also, websites might still be able to gather user information from third parties although they do not collect this information themselves, e.g. by matching it with the full IP address that can still be recorded.

In contrast to the other websites, Amazon did not seem to rely on the high-entropy user agent for their RBA algorithm. Instead, they collected network information like the RTT. A more reliable version would be to measure the RTT from the server side, which is possible, e.g., with WebSockets [63] or token exchanges [31]. Nevertheless, it could be possible that the server-side RTT is also measured, and then compared with the client-submitted version to detect whether the client tried to spoof some values.

Our crawlings revealed that some websites change their HTTP CH behavior based on ISP and region. Especially the different ISP revealed more than 100 HTTP CH using websites than before. This likely means that some websites detected AWS instances and

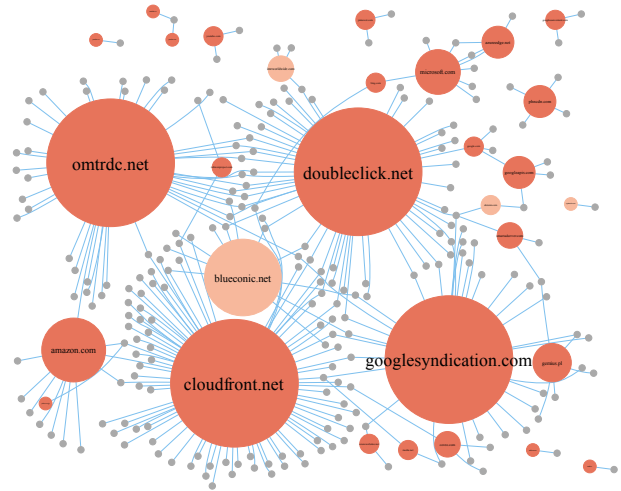


Figure 6: Network of third party domains that request HTTP CH data at the login pages of Tranco 5K websites (dark gray dots). The color at the third party domains shows their requested level of detail (light red: high, dark red: very high). None of the third party domains requested a medium or lower level of detail from the user

changed their behavior based on it. Therefore, we assume that our results using the different ISP better reflect the real-world HTTP CH behavior on the Web. Also, we were able to detect more websites with HTTP CHs when residing in the EU. One reason for this could be that we identified the login pages from a server inside one country and that few websites applied some kind of geoblocking to clients outside of that country. Based on our crawled data, however, we cannot fully test this hypothesis as the access to the server-side logic would have been required.

HTTP CHs are not consistently worded. When related to the UAS, they start with sec-ch-. In all the other cases, for device and network information, they do not have this prepending string. We had the impression that some website developers seemed to get this wrong, even those belonging to high-traffic websites inside the Tranco 5K. This can be seen by the fact that the HTTP CH for client's RAM (Device-Memory) was often spelled wrong (Sec-CH-Device-Memory, see Figure 4a, b, and d). Further research should investigate whether this is a common usability issue for developers and how HTTP CHs should be designed to avoid such pitfalls.

6 IMPACT (RQ3)

To better understand the impact of our results, we also studied the interconnection of the different third party domains and the browser support in practice.

6.1 Interconnection of Third Party Domains (RQ3a)

Some crawled login pages implicitly submitted HTTP CH data to large tracking and content delivery networks by referencing

third parties, who were integrated on other login pages as well (see Figure 6). These third party domains were interconnected with popular websites, many of which did not use HTTP CHs themselves. These covered 18.1% of the crawled Tranco 1K, 13.8% of Tranco 5K, 13.3% of Tranco 10K, and 9.1% of Tranco 100K websites (see Figure 3b).

6.2 Browser Support (RQ3b)

To estimate the real-world impact of HTTP CHs, we measured how browsers on three different devices (desktop, iOS, Android) interpret the different HTTP CHs. Therefore, we opened a website that sent an Accept-CH HTTP header containing all possible values from the RFC [29] and specification [62]. Then we measured the response of the corresponding browser. We included the most popular browsers based on their market share in desktop and mobile devices [57]. As a privacy-baseline besides Firefox, we also included Brave, which is a well-known privacy-oriented browser [40].

The results show that web browsers Chrome and Edge support a wide range of HTTP CHs in the desktop and Android browser versions (see Table 2 in Section 2.2). Brave on desktop and Samsung Internet for Android only support a subset of them. All mobile browsers on iOS as well as the Safari and Firefox browsers on all devices did not support any HTTP CH variants.

6.3 Discussion

Although the amount of third parties requesting HTTP CHs was low compared to the first parties, their influence on tracking users across websites is huge.

Most of the third parties requested a very high level of information using HTTP CHs, potentially allowing them to track users across websites. Some involved third parties were online services, which were also interconnected with trackers (e.g., amazon.com, microsoft.com, google.com). With a successful login on these online services, we assume that the involved third parties might be able to associate the tracking data with a user account and eventually also a real-world identity. This could be done with approaches like cookie respawning [22], which aim to restore cookie data with the help of browser fingerprinting attributes. This affects the majority of web users. Following the market shares of Chrome (65% desktop, 65% mobile), Edge (13% desktop, 0.3% mobile), and Samsung Internet (4% mobile) [57], we can assume that at least 78% of desktop and 69% of mobile users are trackable with HTTP CHs.

Interestingly, browsers on iOS do not seem to send HTTP CH responses. One reason for this could be that app developers have to meet high data protection requirements in order to get into Apple’s App Store [4].

7 RELATED WORK

The invasion of web user privacy by third parties while browsing the Web is ubiquitous today. Over the years, many tracking technologies have evolved, with cookies being the first and still widely used tracking technology [11, 22, 50, 54]. With the many countermeasures developed and implemented in web browsers, either as an integral part or as available add-ons [51, 52], new approaches to track web users evolved. These include user sessions [1, 16], client memory and cache [55], domain name system entries [15, 17], and

fingerprinting [38]. Some of them are even rather persistent tracking mechanisms [1, 22]. The UAS has always been part of this tracking mixture. However, the UAS can also be used for good, for example in RBA where it is considered a useful feature to detect and prevent malicious account takeovers [63, 65]. HTTP CHs were introduced to replace the UAS.

Senol and Acar [53] measured the usage of HTTP CHs for tracking purposes on the top 100K sites from the Chrome User Experience Report list in June 2023. They found that third-party domains frequently requested high entropy HTTP CHs from the user. Our study confirms their observations. Our results go beyond that, and provide a timely overview of HTTP CH usage on the Web, investigated the HTTP CHs usage on (RBA-instrumented) login pages, and identified differences in HTTP CH usage between first and third party domains, start pages and login pages, geographical user locations, and used ISPs.

Intumwayase et al. [35] crawled a rank-sliced randomized sample of 12K domains belonging to the Tranco 1M to study the usage of UASs on the Web. They found that only few websites use the user agent to adjust the website contents. Following the results, they suggested to replace the UAS with HTTP CHs to reduce trackability by online services. Our analysis showed, however, that trackers seem to adjust to this change and obtain even more fine-grained user information via HTTP CHs as well.

Related work also measured the usage of tracking mechanisms on the web. Papadogiannakis et al. [48] crawled one million websites to study whether they continue tracking users that rejected a cookie consent banner. They found that more than 14K of them tracked their users even when they did not consent to cookie-based tracking. Fouad et al. [22] crawled the top 30K websites and found that more than 1K of them use browser fingerprinting to restore cookies deleted by a user. Especially the IP address and user agent features were commonly used to achieve that. In contrast to our work, all studies did not include HTTP CHs. Our work furthermore shows that HTTP CHs are commonly used by known trackers, and therefore potentially used to improve the user tracking abilities.

Iqbal et al. [36] measured the usage of browser fingerprinting on the Alexa 100K websites. Bahrami et al. [6] did a similar approach with long-term measurement by crawling historical data from the Wayback Machine [34]. Both works proposed tooling to detect usage of different JavaScript API components for tracking. However, as HTTP headers were out of scope, they did not mention HTTP CHs in their work. Also, the methodology using the Wayback Machine has flaws as it crawls from different locations and different time periods [32]. Using our own crawling, we showed that HTTP CHs are probably used by trackers for tracking purposes.

8 LIMITATIONS

Our results are limited to the points in time where we crawled the data. It is possible that some websites requested HTTP CHs in a time period that was shorter than one month. Judging from the observed overall tendencies regarding HTTP CH adoption over time, however, we assume that the impact of this should be minimal.

We did not interact with the login pages and only recorded the first HTTP response. It is still possible that websites would have requested HTTP-CH data after submitting the login form.

Nevertheless, our data still provides indications that websites show a different HTTP CH behavior on login pages compared to their start pages.

Our goal was to measure how web browsers with deactivated JavaScript (e.g., privacy-savvy users) could be tracked. It is still possible that online services used JavaScript to extract sensitive user information [53]. Therefore, our study results rather represent a lower baseline of websites that track their users.

We analyzed the HTTP CHs that were sent using the Accept-CH HTTP header. An expired IETF draft suggested that HTTP CHs could also be sent in the TLS 1.3 handshake when using 0-Round-Trip-Time and the TLS Application-Layer Protocol Settings (ALPS) Extension [7]. However, we found no indications that this was used in practice. We crawled the TLS handshake of the websites and did not see any response where this was included. Furthermore, it shall be noted that also the IETF draft for TLS ALPS has expired [8].

9 CONCLUSION

HTTP CHs were introduced to replace the UAS as a privacy measure in almost all major browsers, but they can be abused to collect even more data from users than was originally possible. In this paper, we present the first long-term study of the use of HTTP CHs in the wild. We found that RBA-instrumented websites tend to collect more detailed user data than those without RBA. Nevertheless, the use of HTTP CHs remains generally low despite their implementation in almost all web browsers. However, in the context of third-party websites, which are often linked to trackers, the usage rate is significantly higher. This is concerning because HTTP CHs allow the retrieval of more data from the client than the UAS provides for, and there are currently no mechanisms in place for users to detect or control this potential data leak. To protect against these practices, browsers should incorporate countermeasures that allow users to control what information they choose to reveal via HTTP CHs, and they should have reasonable default settings that maximize privacy.

ACKNOWLEDGMENTS

We would like to thank Rudolf Berrendorf and Javed Razzaq for providing us a huge amount of computational power for our big data analysis. The Platform for Scientific Computing was supported by the German Ministry for Education and Research, and the Ministry for Culture and Science of the state North Rhine-Westphalia (research grant 13FH156IN6).

REFERENCES

- [1] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) (CCS '14). ACM, 674–689. <https://doi.org/10.1145/2660267.2660347>
- [2] Akamai. 2020. Loyalty for Sale – Retail and Hospitality Fraud. [state of the internet] / security 6, 3 (Oct. 2020). <https://web.archive.org/web/20201101013317/https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-loyalty-for-sale-retail-and-hospitality-fraud-report-2020.pdf>
- [3] Furkan Alaca and Paul C. van Oorschot. 2016. Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods. In *32nd Annual Computer Security Applications Conference* (Los Angeles, CA, USA) (ACSAC '16). ACM, 289–301. <https://doi.org/10.1145/2991079.2991091>
- [4] Apple. 2024. User Privacy and Data Use. <https://web.archive.org/web/20240503143642/https://developer.apple.com/app-store/user-privacy-and-data-use/>
- [5] Australian Cyber Security Centre. 2021. *Australian Government Information Security Manual*. Technical Report. [https://web.archive.org/web/20210830131917/https://www.cyber.gov.au/sites/default/files/2021-06/01.%20ISM%20-%20Using%20the%20Australian%20Government%20Information%20Security%20Manual%20\(June%202021\).pdf](https://web.archive.org/web/20210830131917/https://www.cyber.gov.au/sites/default/files/2021-06/01.%20ISM%20-%20Using%20the%20Australian%20Government%20Information%20Security%20Manual%20(June%202021).pdf)
- [6] Pouneh Nikkhhah Bahrami, Umar Iqbal, and Zubair Shafiq. 2022. FP-Radar: Longitudinal Measurement and Early Detection of Browser Fingerprinting. *Proceedings on Privacy Enhancing Technologies* 2022, 2 (April 2022), 557–577. <https://doi.org/10.2478/popets-2022-0056>
- [7] David Benjamin. 2021. *Client Hint Reliability*. Internet-Draft. Internet Engineering Task Force. <https://web.archive.org/web/20230322052424/https://datatracker.ietf.org/doc/draft-davidben-http-client-hint-reliability/03/> Work in Progress.
- [8] David Benjamin and Victor Vasiliev. 2020. *TLS Application-Layer Protocol Settings Extension*. Internet-Draft. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/draft-vvv-tls-alps/01/> Work in Progress.
- [9] Tim Berners-Lee and Daniel W. Connolly. 1995. Hypertext Markup Language - 2.0. <https://doi.org/10.17487/RFC1866>
- [10] Joseph R. Biden Jr. 2021. Executive Order on Improving the Nation's Cybersecurity. *The White House* (May 2021). <https://web.archive.org/web/20220323200621/https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>
- [11] Tomasz Bujlow, Valentín Carela-Español, Josep Solé-Pareta, and Pere Barlet-Ros. 2017. A Survey on Web Tracking: Mechanisms, Implications, and Defenses. *Proc. IEEE* 105, 8 (2017), 1476–1510. <https://doi.org/10.1109/JPROC.2016.2637878>
- [12] caniuse.com. 2024. Can I use Client Hint? <https://caniuse.com/?search=client-hint>
- [13] Chrome Developers. 2023. Chrome UX Report. <https://web.archive.org/web/20240506101055/https://developer.chrome.com/docs/crux/>
- [14] Chrome for Developers. 2020. New in Chrome 85. <https://web.archive.org/web/20240324075831/https://developer.chrome.com/blog/new-in-chrome-85/>
- [15] Ha Dao, Johan Mazel, and Kensuke Fukuda. 2020. Characterizing CNAME Cloaking-Based Tracking on the Web. In *Network Traffic Measurement and Analysis Conference 2020* (Berlin, Germany) (TMA '20). IFIP Open Digital Library. <https://web.archive.org/web/20221122113517/https://dl.ifip.org/db/conf/tma/tma2020-tma2020-camera-paper66.pdf>
- [16] Ha Dao, Johan Mazel, and Kensuke Fukuda. 2021. CNAME Cloaking-Based Tracking on the Web: Characterization, Detection, and Protection. *IEEE Transactions on Network and Service Management* 18, 3 (2021), 3873–3888. <https://doi.org/10.1109/TNSM.2021.3072874>
- [17] Nurullah Demir, Daniel Theis, Tobias Urban, and Norbert Pohlmann. 2022. Towards Understanding First-Party Cookie Tracking in the Field. In *Sicherheit 2022 (Lecture Notes in Informatics)*. Gesellschaft für Informatik, 19–34. https://doi.org/10.18420/sicherheit2022_01
- [18] Periwinkle Doerfler, Kurt Thomas, Maija Marincenko, Juri Ranieri, Yu Jiang, Angelika Moscicki, and Damon McCoy. 2019. Evaluating Login Challenges As a Defense Against Account Takeover. In *The World Wide Web Conference 2019* (San Francisco, CA, USA) (WWW '19). ACM, 372–382. <https://doi.org/10.1145/3308558.3313481>
- [19] easylist. 2024. EasyList / EasyPrivacy / Fanboy Lists. <https://web.archive.org/web/20240427112250/https://github.com/easylist/easylist>
- [20] Peter Eckersley. 2010. How Unique Is Your Web Browser? In *Privacy Enhancing Technologies*. Vol. 6205. Springer. https://doi.org/10.1007/978-3-642-14527-8_1
- [21] European Union. 2016. General Data Protection Regulation. (May 2016). <https://web.archive.org/web/20220317074247/https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04/Regulation%28EU%292016679>
- [22] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. 2022. My Cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting. *Proceedings on Privacy Enhancing Technologies* 2022, 3 (July 2022), 79–98. <https://doi.org/10.56553/popets-2022-0063>
- [23] Anthony Gavazzi, Ryan Williams, Engin Kirda, Long Lu, Andre King, Andy Davis, and Tim Leek. 2023. A Study of Multi-Factor and Risk-Based Authentication Availability. In *32nd USENIX Security Symposium* (Anaheim, CA, USA) (USENIX Security '23). USENIX Association, 2043–2060. <https://www.usenix.org/conference/usenixsecurity23/presentation/gavazzi>
- [24] Google Cloud. 2024. BigQuery. <https://web.archive.org/web/20240504103158/https://cloud.google.com/bigquery?hl=en>
- [25] Paul A. Grassi, James L. Fenton, Elaine M. Newton, Ray A. Perlner, Andrew R. Regenscheid, William E. Burr, Justin P. Richer, Naomi B. Lefkowitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, and Mary F. Theofanos. 2017. *Digital Identity Guidelines: Authentication and Lifecycle Management*. Technical Report NIST SP 800-63b. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-63b>
- [26] Ilya Grigorik. 2013. *HTTP Client Hints*. Internet-Draft. Internet Engineering Task Force. <https://web.archive.org/web/20230528025111/https://datatracker.ietf.org/doc/draft-grigorik-http-client-hints/00/> Work in Progress.
- [27] Ilya Grigorik. 2015. *HTTP Client Hints*. Internet-Draft. Internet Engineering Task Force. <https://web.archive.org/web/20231004041902/https://datatracker.ietf.org/>

- doc/draft-ietf-httpbis-client-hints/00/ Work in Progress.
- [28] Ilya Grigorik and Yoav Weiss. 2021. HTTP Client Hints. RFC 8942. <https://doi.org/10.17487/RFC8942>
- [29] I. Grigorik and Y. Weiss. 2021. *HTTP Client Hints*. Technical Report RFC8942. <https://doi.org/10.17487/RFC8942>
- [30] Morey J. Haber. 2020. Attack Vectors. In *Privileged Attack Vectors: Building Effective Cyber-Defense Strategies to Protect Organizations*. Apress, Berkeley, CA, USA, 65–85. https://doi.org/10.1007/978-1-4842-5914-6_4
- [31] Alex Heunhe Han and Dong Hoon Lee. 2023. Detecting Risky Authentication Using the OpenID Connect Token Exchange Time. *Sensors* 23, 19 (2023). <https://doi.org/10.3390/s23198256>
- [32] Florian Hantke, Stefano Calzavara, Moritz Wilhelm, Alvis Rabitti, and Ben Stock. 2023. You Call This Archaeology? Evaluating Web Archives for Reproducible Web Security Measurements. In *2023 ACM SIGSAC Conference on Computer and Communications Security (Copenhagen, Denmark) (CCS '23)*. ACM, 3168–3182. <https://doi.org/10.1145/3576915.3616688>
- [33] HTTP Archive. 2023. Frequently Asked Questions. <https://web.archive.org/web/20240506061519/https://httparchive.org/faq>
- [34] Internet Archive. 2023. Wayback Machine. <https://web.archive.org>
- [35] Jean Luc Intumwayase, Imane Fouad, Pierre Laperdrix, and Romain Rouvoy. 2023. UA-Radar: Exploring the Impact of User Agents on the Web. In *22nd Workshop on Privacy in the Electronic Society (Copenhagen, Denmark) (WPES '23)*. ACM, 31–43. <https://doi.org/10.1145/3603216.3624958>
- [36] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *2021 IEEE Symposium on Security and Privacy (San Francisco, CA, USA) (SP '21)*. IEEE, 1143–1161. <https://doi.org/10.1109/SP40001.2021.00017>
- [37] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. 2019. Fingerprint Surface-Based Detection of Web Bot Detectors. In *ESORICS '19*. Springer, 586–605. https://doi.org/10.1007/978-3-030-29962-0_28
- [38] Pierre Laperdrix, Natalia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser Fingerprinting: A Survey. *ACM Trans. Web* 14, 2, Article 8 (apr 2020), 33 pages. <https://doi.org/10.1145/3386040>
- [39] Hieu Le, Athina Markopoulou, and Zubair Shafiq. 2021. CV-Inspector: Towards Automating Detection of Adblock Circumvention. In *2021 Network and Distributed System Security Symposium (Virtual) (NDSS '21)*. Internet Society. <https://doi.org/10.14722/ndss.2021.24055>
- [40] Douglas J. Leith. 2021. Web Browser Privacy: What Do Browsers Say When They Phone Home? *IEEE Access* 9 (2021), 41615–41627. <https://doi.org/10.1109/ACCESS.2021.3065243>
- [41] Jan-Phillip Makowski and Daniela Pöhn. 2023. Evaluation of Real-World Risk-Based Authentication at Online Services Revisited: Complexity Wins. In *18th International Workshop on Frontiers in Availability, Reliability and Security (Benevento, Italy) (FARES '23)*. ACM. <https://doi.org/10.1145/3600160.3605024>
- [42] McAfee. 2023. Check Single URL. <https://sitelookup.mcafee.com/>
- [43] mdn web docs. 2024. User-Agent. <https://web.archive.org/web/20240224150202/https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>
- [44] Rowan Merewood and Yoav Weiss. 2020. Improving user privacy and developer experience with User-Agent Client Hints. <https://web.archive.org/web/20221208053222/https://developer.chrome.com/articles/user-agent-client-hints/>
- [45] National Cyber Security Centre. 2022. *Cloud Security Guidance: Principle 10: Identity and Authentication*. Technical Report. <https://web.archive.org/web/20220518230012/https://www.ncsc.gov.uk/collection/cloud/the-cloud-security-principles/principle-10-identity-and-authentication>
- [46] Heremy Ney. 2022. Prepare for User-Agent Reduction changes in October. <https://web.archive.org/web/20221003134709/https://developer.chrome.com/blog/user-agent-reduction-oct-2022-updates/>
- [47] Henrik Nielsen, Roy T. Fielding, and Tim Berners-Lee. 1996. Hypertext Transfer Protocol – HTTP/1.0. <https://doi.org/10.17487/RFC1945>
- [48] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2021. User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users. In *The Web Conference 2021 (Ljubljana, Slovenia) (WWW '21)*. ACM, 2130–2141. <https://doi.org/10.1145/3442381.3450056>
- [49] Gaston Pugliese, Christian Riess, Freya Gassmann, and Zinaida Benenson. 2020. Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 558–577. <https://doi.org/10.2478/popets-2020-0041>
- [50] 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 *2019 ACM Asia Conference on Computer and Communications Security (Auckland, New Zealand) (AsiaCCS '19)*. ACM, 340–351. <https://doi.org/10.1145/3321705.3329806>
- [51] Iskander Sanchez-Rola, Xabier Ugarte-Pedrero, Igor Santos, and Pablo G. Bringas. 2017. The web is watching you: A comprehensive review of web-tracking techniques and countermeasures. *Logic Journal of IGPL* 25, 1 (Feb. 2017), 18–29. <https://doi.org/10.1093/jigpal/jzw041>
- [52] Lorin Schöni, Karel Kubicek, and Verena Zimmermann. 2024. Block Cookies, Not Websites: Analysing Mental Models and Usability of the Privacy-Preserving Browser Extension CookieBlock. *Proceedings on Privacy Enhancing Technologies* 2024, 1 (Jan. 2024), 192–216. <https://doi.org/10.56553/popets-2024-0012>
- [53] Asuman Senol and Gunes Acar. 2023. Unveiling the Impact of User-Agent Reduction and Client Hints: A Measurement Study. In *22nd Workshop on Privacy in the Electronic Society (Copenhagen, Denmark) (WPES '23)*. ACM, 91–106. <https://doi.org/10.1145/3603216.3624965>
- [54] Konstantinos Solomos, Panagiotis Ilia, Sotiris Ioannidis, and Nicolas Kourtellis. 2020. Clash of the Trackers: Measuring the Evolution of the Online Tracking Ecosystem. In *Network Traffic Measurement and Analysis Conference 2020 (Berlin, Germany)*. IFIP Open Digital Library. <https://web.archive.org/web/20201127171719/https://tma.ifip.org/2020/wp-content/uploads/sites/9/2020/06/tma2020-camera-paper36.pdf>
- [55] Konstantinos Solomos, John Kristoff, Chris Kanich, and Jason Polakis. 2021. Tales of favicons and caches: Persistent tracking in modern browsers. In *Network and Distributed System Security Symposium (NDSS '21)*. <https://doi.org/10.14722/ndss.2021.24202>
- [56] State of California. 2018. California Consumer Privacy Act. (June 2018). https://web.archive.org/web/20220323195500f_/https://leginfo.ca.gov/faces/billTextClient.xhtml?bill_id=201702180AB375 Assembly Bill No. 375.
- [57] Statcounter. 2024. Browser Market Share Worldwide. <https://gs.statcounter.com/>
- [58] The Chromium Projects. 2021. User-Agent Client-Hints? https://web.archive.org/web/20240000000000*/https://www.chromium.org/updates/ua-ch
- [59] The Chromium Projects. 2023. User-Agent Reduction? <https://web.archive.org/web/20240412234718/https://www.chromium.org/updates/ua-reduction/>
- [60] Vincent Unsel, Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. 2023. Risk-Based Authentication for OpenStack: A Fully Functional Implementation and Guiding Example. In *13th ACM Conference on Data and Application Security and Privacy (Charlotte, NC, USA) (CODASPY '23)*. ACM, 237–243. <https://doi.org/10.1145/3577923.3583634>
- [61] Antoine Vastel, Walter Rudametkin, Romain Rouvoy, and Xavier Blanc. 2020. FP-Crawlers: Studying the Resilience of Browser Fingerprinting to Block Crawlers. In *Proceedings 2020 Workshop on Measurements, Attacks, and Defenses for the Web (San Diego, CA)*. Internet Society. <https://doi.org/10.14722/madweb.2020.23010>
- [62] Yoav Weiss. 2023. *Client Hints Infrastructure*. Technical Report. <https://wicg.github.io/client-hints-infrastructure/> Draft Community Group Report, 14 July 2023.
- [63] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. 2021. What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics. In *25th International Conference on Financial Cryptography and Data Security (Grenada) (FC '21)*. Springer, 361–381. https://doi.org/10.1007/978-3-662-64331-0_19
- [64] Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. 2019. Even Turing Should Sometimes Not Be Able To Tell: Mimicking Humanoid Usage Behavior for Exploratory Studies of Online Services. In *24th Nordic Conference on Secure IT Systems (Aalborg, Denmark) (NordSec '19)*. Springer, 188–203. https://doi.org/10.1007/978-3-030-35055-0_12
- [65] Stephan Wiefling, Paul René Jørgensen, Sigurd Thunem, and Luigi Lo Iacono. 2023. Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service. *ACM Transactions on Privacy and Security* 26, 1, Article 6 (Feb. 2023). <https://doi.org/10.1145/3546069>
- [66] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. 2019. Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. In *34th IFIP TC-11 International Conference on Information Security and Privacy Protection (Lisbon, Portugal) (IFIP SEC '19)*. Springer, 134–148. https://doi.org/10.1007/978-3-030-22312-0_10
- [67] Stephan Wiefling, Jan Tolsdorf, and Luigi Lo Iacono. 2021. Privacy Considerations for Risk-Based Authentication Systems. In *2021 International Workshop on Privacy Engineering (Vienna, Austria) (IWPE '21)*. IEEE, 320–327. <https://doi.org/10.1109/EuroSPW54576.2021.00040>
- [68] Stephan Wiefling, Jan Tolsdorf, and Luigi Lo Iacono. 2023. Data Protection Officers' Perspectives on Privacy Challenges in Digital Ecosystems. In *4th Workshop on Security, Privacy, Organizations, and Systems Engineering (Copenhagen, Denmark) (SPOSE '22)*. Springer. https://doi.org/10.1007/978-3-031-25460-4_13
- [69] Shoshana Zuboff. 2019. Surveillance Capitalism and the Challenge of Collective Action. *New Labor Forum* 28, 1 (2019), 10–29. <https://doi.org/10.1177/1095796018819461>

All URLs were last accessed on May 9th, 2024.