Defining "Broken": User Experiences and Remediation Tactics When Ad-Blocking or Tracking-Protection Tools Break a Website's User Experience

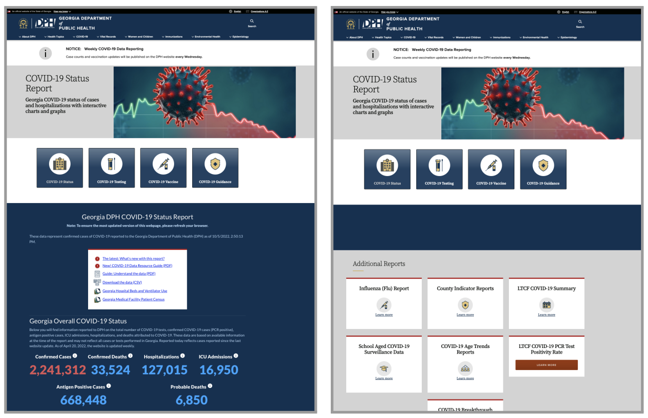
Alexandra Nisenoff †∗, Arthur Borem†, Madison Pickering†,
Grant Nakanishi†, Maya Thumpasery†, Blase Ur†
nisenoff@cmu.edu, {arthurborem,madisonp,gnakanishi,mthumpasery,blase}@uchicago.edu
† *University of Chicago,* ∗ *Carnegie Mellon University*

Abstract

To counteract the ads and third-party tracking ubiquitous on the web, users turn to blocking tools—ad-blocking and tracking-protection browser extensions and built-in features. Unfortunately, blocking tools can cause non-ad, non-tracking elements of a website to degrade or fail, a phenomenon termed *breakage*. Examples include missing images, non-functional buttons, and pages failing to load. While the literature frequently discusses breakage, prior work has not systematically mapped and disambiguated the spectrum of user experiences subsumed under "breakage," nor sought to understand how users experience, prioritize, and attempt to fix breakage. We fill these gaps. First, through qualitative analysis of 18,932 extension-store reviews and GitHub issue reports for ten popular blocking tools, we developed novel taxonomies of 38 specific types of breakage and 15 associated mitigation strategies. To understand subjective experiences of breakage, we then conducted a 95-participant survey. Nearly all participants had experienced various types of breakage, and they employed an array of strategies of variable effectiveness in response to specific types of breakage in specific contexts. Unfortunately, participants rarely notified anyone who could fix the root causes. We discuss how our taxonomies and results can improve the comprehensiveness and prioritization of ongoing attempts to automatically detect and fix breakage.

1 Introduction

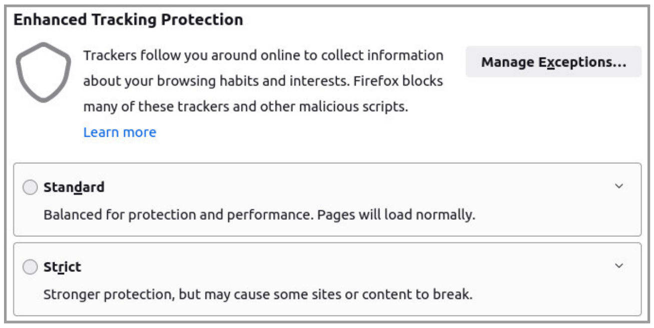
The modern web is filled with ads, enabling platforms and content creators to monetize websites whether or not users pay to access them. Furthermore, many of these ads are targeted to specific users based on inferences about their demographics, interests, and activities [42, 65, 66]. Enabling this ad targeting is an ecosystem in which users' activities online are frequently tracked using HTTP cookies, browser fingerprinting, and other mechanisms [24]. Because many users find online ads in general to be annoying [46] and the tracking underlying ad targeting to be creepy and problematic [7, 20, 64, 67], ad-blocking and tracking-protection tools have become popular [41, 49]. For example, the Adblock Plus [3] browser extension aims to block most ads, while the Ghostery [30] browser extension aims to block third-party tracking. Web browsers themselves increasingly include similar features, such as Mozilla Firefox's Enhanced Tracking Protection [44]. We use the term **blocking tools** to encapsulate all of the above.



(a) Normal functionality. (b) "Broken" functionality.

Figure 1: Many tracking-protection tools cause the status report (blue background) to disappear from this website [28].

These tools typically aim to stop ads and tracking by either blocking or modifying outgoing HTTP requests, or by either modifying or not loading the resources returned in response. Unfortunately, these processes can also degrade other parts of a website's user experience. Like prior work, we say a "website **breaks**" (and use the term **breakage** more generally) to refer to any non-ad and non-tracking aspect of a website's user experience no longer functioning because of a blocking tool. For example, as shown in Figure 1, Firefox's "strict" Enhanced Tracking Protection causes the interactive data (blue background) to disappear from Georgia's COVID-19 daily status report [28] even though, from the user perspective, this data is neither an ad nor tracking, but rather key functionality. Breakage may be inadvertent (e.g., a blocked tracking re-


Enhanced Tracking Protection

Trackers follow you around online to collect information about your browsing habits and interests. Firefox blocks many of these trackers and other malicious scripts.

Learn more

Manage Exceptions...

Standard

Balanced for protection and performance. Pages will load normally.

Strict

Stronger protection, but may cause some sites or content to break.

Figure 2: Firefox warns users that selecting "strict" tracking protection "may cause some sites or content to break."

source also defined non-tracking functionality) or intentional (e.g., encouraging users to uninstall blocking tools by purposefully tying essential functionality to ads or tracking) [4].

That blocking tools can break websites is a key barrier to user adoption [41] and consideration in tools' default settings [51, 54]. For example, as shown in Figure 2, while Mozilla Firefox's "strict" tracking protection setting improves user privacy beyond its "standard" setting, its interface warns users that the "strict" setting "may cause some sites or content to break" and that "if a site seems broken, you may want to turn off tracking protection." Furthermore, because experiences of breakage could cause users (who cannot see the typically hidden nature of improved web privacy) to switch browsers, the less-protective "standard" setting is the default. As a result, a number of recent efforts aim to detect and mitigate website breakage caused by blocking tools [4, 13, 33, 35, 37, 45, 51, 55–58].

Despite this interest in automatically mitigating breakage, prior work has neither systematically mapped the scope of breakage that users experience nor sought to understand how users react to the different failures subsumed under the concept of breakage. Without this understanding, researchers will be limited both in their ability to automatically detect and mitigate the full range of ways which users experience website breakage and in their ability to prioritize the types of breakage that matter most to users. In this paper, we fill those gaps.

First, to map the full spectrum of ways in which users experience website breakage, we analyzed public user reviews of blocking tools, as well as public issue reports sent to the developers of those tools. Specifically, we scraped recent negative and neutral reviews (those most likely to mention breakage) of ten popular blocking tools from the Chrome, Firefox, and Mac (Safari) extension stores, as well as the Github issues pages for the eight tools that used public issue tracking. We qualitatively coded the 18,932 reviews and issue reports (34,423 total posts and responses), developing the first taxonomy of types of website breakage users report experiencing. This taxonomy includes 38 specific types of breakage experiences in seven broad categories, encompassing experiences like pages being slow to load, authenticated sessions failing to persist, upload buttons not functioning, elements being mispositioned, and the page crashing. From the same process, we also develop the first taxonomy of how users report attempting to fix breakage. This taxonomy includes 15 specific strategies in four broad categories, including disabling the tool, modifying the block list, contacting the developer, and switching browsers.

To understand subjective reactions to breakage, we surveyed 46 users of dedicated blocking tools and 49 users of web browsers with built-in blocking functionality. While prior work [41] reported that "extensions only rarely break websites," we found the (not necessarily contradictory) result that nearly all participants had experienced multiple types of breakage from our taxonomy. Participants reported using an array of strategies to mitigate breakage, though many of those strategies were ineffective. Notably, few participants reported using tools' built-in mechanisms for notifying tool developers about breakage. We also asked participants their likelihood of trying to fix specific types of breakage in specific browsing contexts; we find missing images to be among the least important types of breakage to users and news websites to be among the least important browsing contexts when experiencing breakage. We conclude by discussing our contributions to mitigating breakage for better website user experiences.

2 Background and Related Work

Ads are ubiquitous on the web [49], and associated third-party tracking practices are widespread [16]. In 2020, Google's trackers were present on over 80% of the 1.96 billion pages surveyed [31]. Websites use cookies, browser fingerprinting, and account tracking to collect user data to target ads and otherwise personalize features [25, 31, 52]. This tracking creates concerns around security [41] and discrimination [18, 62]. Furthermore, users have expressed discomfort with these practices [14, 42, 61], finding them creepy [64] and annoying [49].

2.1 Web Tracking and Anti-Tracking Tools

In response, researchers and developers have created blocking tools. They are usually distributed as browser extensions, though in recent years blocking features have increasingly been incorporated into browsers [5, 37] like Firefox [44] and Brave [10]. Many blocking tools, such as uBlock Origin [32], Ghostery [30], and AdBlock [2], rely on frequently updated lists [26, 60] of known third-party tracking domains [58]. Other tools, such as Privacy Badger [23], rely in part on algorithmic methods to detect tracking and block resources.

While prior work found that blocking tools do not worsen page-loading times and actually may improve them [9, 11, 12, 33], there are opportunities for improving blocking tools' usability and performance. While blocking tools can be effective at protecting users from common tracking techniques, they have many blind spots, such as CNAME cloaking [17], stateless fingerprinting [43], and more [8, 16, 27, 29, 39]. In response, recent work has developed improved techniques for detecting potential tracking activities [4, 33, 35, 36, 55].

Other work has documented blocking tools' usability challenges. For instance, users frequently misunderstand the purpose and mechanism of popular blocking tools, often overestimating their protections [38, 61]. Confusing user interfaces might cause some of these misconceptions [1, 53, 68].

2.2 Breakage Caused by Blocking Tools

While most prior work has sought either to improve blocking tools' technical capabilities or to understand users' broad perceptions of those tools, recent work has increasingly considered how the user experience might be degraded by blocking tools breaking websites. Fifteen years ago, Krishnamurthy et al. [39] noted that privacy and website usability can often be at odds with each other; work from the past year by Jueckstock et al. measured these tensions on the modern web [37]. One reason breakage occurs is that website developers will often include code for tracking users and providing essential website functionality in the same scripts and files, either inadvertently or intentionally [4]. When a typical blocking tool encounters a script used in part for tracking, it will block it. Doing so protects the user's privacy, yet can also degrade or destroy the website's functionality.

Recent efforts have sought to develop new techniques for detecting and minimizing breakage. For example, Smith et al. developed methods that automatically replace tracking scripts with nearly identical code units that are unable to access the user data necessary to perform tracking [56]. The same group subsequently developed a tool for automatically detecting breakage due to filter list changes, with the goal of better understanding how breakage occurs and how it can be fixed [57]. Other researchers have studied the intricacies of JavaScript to prevent some types of breakage [13, 45], employing graph-based methods to model how web resources load. These efforts have enabled researchers to better understand blocking and breakage [35, 55], as well as how to disentangle functional resources from tracking-focused resources [4]. Researchers have also studied how blocking tools' (mostly manually curated) filter lists evolve both to increase tracking protection and to minimize breakage [51, 58].

While this recent work focuses on breakage, few studies have explored how users behave when confronted with breakage, or even what different types of degradations and failures are encompassed by the concept of breakage. Mathur et al. [41] most closely studied breakage from the user perspective by evaluating why users have—or have not—adopted blocking tools and how they perceive online tracking. The authors ask a few survey questions about how respondents encounter breakage, what types of breakage they encounter most frequently, and what they do in response. They report that breakage is fairly uncommon, though their survey relies on participants' own definitions and recollections of breakage.

Prior studies seem to assume that users react to different forms of breakage (e.g., pages not loading and text being misformatted) in the same way, and that user behaviors remain the same across website contexts. In contrast, we comprehensively unpack the nuances of how blocking tools break websites and how users respond in different contexts through large-scale analyses of reviews and issue reports, as well as a survey building on those analyses. Specifically, we present a novel taxonomy of breakage and describe how users respond to different forms of breakage on different websites, as well as which strategies they use to address breakage.

3 Tool Reviews & Issue Reports: Methods

To understand the many ways in which breakage manifests, we first analyzed user reviews and issue reports for ten popular blocking tools distributed as browser extensions: AdBlock [2], AdBlock Plus [3], Decentraleyes [50], Disconnect [19], DuckDuckGo Privacy Essentials [21], Ghostery [30], HTTPS Everywhere [22], NoScript [40], Privacy Badger [23], and uBlock Origin [32]. We selected these extensions because: they are popular, often having hundreds of thousands of users; they are available across browsers; they cover a variety of blocking methods and filter lists; or they are commonly studied in privacy research [9, 12, 41]. They also differ from each other in their technical approach to blocking, which could cause different issues to manifest. Abstractly, our methods were inspired by those of Huaman et al. [34], who also looked at reviews and issue reports to map problems users faced. While they looked at issues with websites that prevented password manager extensions from working, we looked at the reverse: extensions preventing websites from functioning.

3.1 Data Sources and Collection Procedures

Broadly, users can publicly report breakage by leaving a review that mentions breakage on an extension marketplace (e.g., the Chrome Web Store) or by raising an issue on a public issue tracker (e.g., on an extension's GitHub repository). We chose to draw from these sources because they are easily scrapeable through automated means, most extensions we studied are on these platforms, the review format was standardized, and the purpose of these platforms was for user feedback. Additionally, since the blocking tools we analyzed must be downloaded from extension marketplaces, we anticipated we would find reports from a greater variety of users compared to examining issue trackers exclusively.

Specifically, we collected data from three extension marketplaces (the Chrome Web Store, Firefox Browser Add-Ons, and Mac App Store) and one issue tracker (GitHub). While we considered other data sources [51], we chose not to use private issue trackers because the data is non-public (see Section 7), and we chose not to use proprietary forums or social media because the data is not in a standardized format and typically

Table 1: The number of reviews, support requests, and issue reports we ultimately analyzed.

		AdBlock	AdBlock Plus	Decentraleyes	Disconnect	DuckDuckGo Privacy Essentials	Ghostery	HTTPS Everywhere	NoScript	Privacy Badger	uBlock Origin
Chrome Web	Threads	4,994	3,557	4	214	130	377	240	21	151	627
Store Reviews	(Posts)	(6,569)	(4,163)	(4)	(240)	(196)	(543)	(321)	(28)	(221)	(891)
Chrome Web Store	Threads	-	-	18	-	404	431	-	-	-	2,462
Support Requests	(Posts)	(-)	(-)	(30)	(-)	(534)	(666)	(-)	(-)	(-)	(3,774)
Firefox Browser	Threads	161	434	7	55	91	210	48	150	4	237
Add-Ons	(Posts)	(282)	(548)	(7)	(72)	(95)	(348)	(51)	(158)	(5)	(246)
Mac App Store	Threads & Posts	338	225	-	112	23	67	-	-	-	-
GitHub	Threads	-	-	174	582	174	116	739	193	739	739
Issues	(Posts)	(-)	(-)	(901)	(1474)	(574)	(572)	(2,683)	(636)	(3,546)	(3,280)

represents more general discussions. We collected reviews and responses from the launch of each extension through January 1, 2022. Except for the Mac App Store, we collected data using custom Selenium [59] crawlers from a vantage point in the US. Below, we describe the data from each source:

Chrome Web Store: We collected reviews and support posts from the Chrome Web Store. Each review contained the original review and any additional comments. We collected the reviewer's username, their user rating (from 1 to 5), the review text, and the date of the post. Some extensions included a section separate from the reviews where users could submit problems, questions, or suggestions. For each post in this section, we collected the original text and any responses, the username of the reviewer, and the date the post was made.

Mac App Store: We collected reviews from the Mac App Store using the App Store Scraper Python library [15]. Reviews included the rating (from 1 to 5), username of the reviewer, the date the review was posted, and if it had been edited. Unlike our other sources, the Mac App Store tags reviews by country. We collected only reviews from the USA.

Firefox Browser Add-Ons: We collected reviews from the Firefox Browser Add-Ons website. For each review, we collected the reviewer's username, their user rating (from 1 to 5), the review text, and the date it was posted. We excluded any reviews that contained only a star rating without any text.

GitHub: We collected information about all open and closed issues for the eight (out of ten) blocking tools with public GitHub repository issue trackers. Some extensions had multiple repositories; all issues from each repository were collected. For every issue, we collected the title, the issue number, any tags (e.g., "broken site"), and the date the issue was closed (if applicable). We collected the date of the post, the username who posted the comment, and the post contents.

3.2 Filtering

To focus on the reviews most likely to discuss breakage, we filtered reviews as follows. Using the `spacy-langdetect` Python package [6], we removed reviews that were not in English. During qualitative coding, we flagged and excluded additional non-English reviews missed by this automated filtering. Because we anticipated that negative reviews were most likely to mention breakage, we kept only reviews that received three or fewer stars out of five.

GitHub issues often include repository-specific tags. Any issues that contained breakage-related tags (e.g., "broken site" from Privacy Badger or "Type: Broken Page" from Ghostery), tags that did not preclude the issue being about breakage (e. g. , "good first issue" from uBlock Origin), or no tags at all were included in our analysis. However, any issues that only included tags unrelated to breakage (e.g., "code-style" for HTTPS Everywhere) were excluded. Finally, since there were many more issues tracked for HTTPS Everywhere and uBlock Origin than for the other extensions, we randomly selected 739 of these posts (the number from the next largest issue tracker, Privacy Badger's) to analyze. Table 1 quantifies the reviews and issues included in our analysis after filtering.

3.3 Qualitative Coding

To systematically characterize the different user experience degradations entangled in the concept of breakage, we performed qualitative coding following a thematic analysis approach. Five members of the research team developed a codebook shared across all files via inductive coding. Specifically, each coder initially analyzed posts for a given file (e.g., all Privacy Badger reviews from the Chrome Web Store). In weekly meetings, any new codes created were discussed as a group. After a file was coded by two members of the research team, they cross-checked the assigned codes. When the codes did not match, the coders discussed the mismatch and came to an agreement on final codes for the response. For robustness, we aimed to assign distinct pairs of coders for each data source. In balancing the overall number of reviews each coder was assigned, we tried to make sure that all combinations of the five coders worked together. The full team discussed posts coded as "miscellaneous" to see if new codes should be created.

3.4 Tool Reviews & Issue Reports Limitations

We did not analyze all possible reviews from the ten tools, but only those rated three stars or fewer out of five. While this may have excluded breakage reported by more forgiving users, we prioritized exploring the more negative experiences related to breakage. We further excluded all non-English posts (a negligible amount) due to our team's language capabilities.

Posting reviews and creating GitHub issues is far from the only way users report issues with blocking tools. Some of these tools have built-in reporting features, email addresses specifically for reporting breakage, and social media accounts, all of which users may employ to report breakage. Our data only comes from users who use *public* avenues for communicating their experiences, excluding users with other communication preferences. In Section 6.4 we discuss the large gap in the percentage of users that have experienced breakage and that have reported it in any way. Notably, users who use GitHub issue trackers may be non-representative as familiarity with GitHub may indicate technical expertise. Furthermore, users may be biased towards reporting types of breakage that they find to be more detrimental to their browsing experience. Finally, the root causes and manifestations of breakage may have been reported inaccurately as they rely on the judgment of the review author, who may not have the technical expertise to accurately diagnose breakage.

4 Tool Reviews & Issue Reports: Results

In this section, we present the forms of breakage we identified in the reviews and issue reports, briefly describe each, then discuss the strategies users have applied to address these forms of breakage. We report on the frequency of specific forms of breakage and mitigation strategies in terms of threads (i.e., an original post with an optional list of comments), rather than posts and comments, since a single user would often mention a single instance of breakage in multiple posts within a thread. We also calculate the percentage of threads for each form of breakage and mitigation with respect to all threads that mention breakage or mitigation strategies, excluding threads that do not mention breakage. For example, issues with loading and responsiveness were mentioned at least once in 27.25% of all threads that mention breakage. Figures 7–8 in the appendix give examples of what constitutes a thread and a post.

4.1 Taxonomy of Breakage

We identify seven major classes of breakage and provide a breakdown of how these manifest to users in Table 2. The most prevalent forms of breakage concerned loading and responsiveness of websites and issues with loading third party resources and content (e.g., images and videos), accounting for around 27% and 24% of breakage threads, respectively. Interactions between blocking tools and websites or other extensions trailed behind, accounting for about 16% of breakage threads. More uncommon forms of breakage included issues with specific HTML elements malfunctioning or missing on screen (9%), failed attempts at authenticating and maintaining user sessions (5%), and issues manifesting on browsers (6%), such as the browser crashing. Many users, especially when leaving reviews rather than creating bug reports, were not precise about what issues they were experiencing, accounting for 25% of threads that mentioned breakage.

Table 2: Our taxonomy of breakage. For each category *C* we report: the percentage of threads that mention *C* across all that mention any form of breakage; the number of threads mentioning *C*; and the number of unique users mentioning *C*.

Breakage category	**Breakage subcategory**	**% of Breakage Threads**	**# of Threads**	**# of Users**
Loading and responsiveness	Page fails to load	14.38	820	926
	Slow page load	9.63	549	569
	Unresponsive	2.37	135	145
	Page lag	1.93	110	114
	Automatic page reload	1.21	69	74
	Pop up does not appear	0.19	11	11
	Server failure	0.04	2	2
	Total	**27.25**	**1554**	**1699**
Resources and third party content	Video	10.03	572	657
	Images	6.89	393	414
	Formatting and style	4.37	249	270
	Embedded content	2.09	119	131
	Comment section and chat	1.77	101	104
	Sound	0.65	37	38
	Total	**23.50**	**1340**	**1475**
Extension detection and interaction	Extension detection	7.21	411	434
	Blocked access	4.23	244	259
	Extension interference	4.23	241	276
	Plugin	0.67	38	38
	Total	**16.22**	**925**	**1002**
HTML Elements	Link	3.16	180	212
	Button	2.42	138	145
	Upload	1.14	65	79
	Download	0.95	54	59
	Text field	0.75	43	43
	Form	0.63	36	43
	Menu	0.42	24	25
	Total	**9.00**	**513**	**584**
Browser level	Browser crash	2.31	132	138
	Webpage crash	0.96	55	57
	Functionality	0.82	47	48
	Scroll	0.61	35	35
	Full screen	0.54	31	40
	Vague crash	0.28	16	16
	Element position change	0.04	2	2
	Total	**5.54**	**316**	**334**
Authentication and sessions	Login	3.63	207	227
	CAPTCHA	0.84	48	70
	No session persistence	0.40	23	24
	Total	**4.72**	**269**	**312**
Vague	Generic	14.92	851	897
	Missing content	5.66	323	347
	Special cases	3.65	208	232
	Broken interaction	2.60	148	159
	Total	**25.40**	**1454**	**1563**

Loading and responsiveness: Several users mentioned trouble accessing websites with blocking tools installed. Some users experienced **slow page loads**, but were eventually able to open the page after waiting. A uBlock Origin user on Firefox waited *"3-5 minutes for any page to open,"* while an AdBlock user on Safari would *"rather sit through the ads"* than experience *"buffering every two seconds."*

Other users experienced **page lags** after the website finished loading. A Ghostery user on Safari noted that before disabling the application, *"search engines and web forms started to lag big time."* Lagging often occurred in conjunction with unresponsive websites, where either *"nothing happens"* after interacting with the websites by clicking buttons, for example, or a *"wait symbol"* appears and *"hangs forever,"* as a Privacy Badger reported on a GitHub issue. Some uBlock Origin and HTTPS Everywhere users experienced a less common phenomenon of **automatic page reloads**, where pages would *"reload over and over"* or *"auto-refresh... every second."*

Other users were simply **unable to load the page**. Several DuckDuckGo Privacy Essentials users on Firefox could not access their bank's website without *"completely disabling the add-on."* This issue not only manifested when trying to access a website by entering its URL, but also with **popups**. A few users were unable to access specific webpages and were prompted with messages attributing the connection failure to **server issues**. One AdBlock user, for example, encountered a blank page with a *"Waiting for AdBlock extension"* status connection message in Chrome. In two cases, users were shown explicit network or server error messages instead of being able to access a domain.

Resources and third party content: Sometimes, blocking tools may "correctly" block third-party content that is integral to website functionality. Over 10% of breakage threads mentioned issues with **videos**, making it the second most common form of breakage identified within this category. Both native and embedded videos would sometimes be missing, display error messages, *"run slow and freeze"* (Adblock Plus, Chrome), or have **sound** blocked.

Some users observed *"corrupted CSS"* (AdBlock, Chrome), *"fonts [being] blocked"* (Ghostery, Safari), *"misaligned, truncated"* elements (Adblock Plus, Chrome), and pages appearing as *"1990's text-only websites"* with no styling (uBlock Origin, Chrome). These and other **style and format** issues accounted for about 4% of threads.

Additional **embedded content**, such as tweets, Instagram posts, or **images** (almost 10% altogether), did not appear on pages. Finally, users reported that **comment sections** sometimes did not appear, that they were unable to log into third-party commenting services, or that comments they submitted would not be published (2%).

Extension detection and interaction: About 4% of the breakage threads expressed concern that websites were able to **block access** from their website to users of blocking tools. For example, a Chrome AdBlock user noted:

> *85% of sites are now aware of adblock extensions so they either lock you out of content, play black screen ads, or just ignore your adblock entirely.*

Users often reported experiences like this when interacting with news sites and streaming services.

Blocking tools also interfered with **extensions** (4%) and **plugins** (less than 1%). They found that videos (Disconnect, GitHub) and games (AdBlock, Chrome) that relied on plugins (e.g., Flash and Unity) often crashed or did not work. Users who had multiple blocking tools installed noted that the way **extensions interact** with each other sometimes caused at least one to stop working. For example, Chrome users with DuckDuckGo Privacy Essentials and Ghostery installed found that the former did not work unless the latter was uninstalled.

HTML elements: Around 9% of threads that discussed breakage mentioned malfunctioning or missing HTML elements. Developers and business owners were among the authors of posts in threads expressing concerns with broken third-party **forms** (less than 1%). Other users experienced issues with **text fields** (less than 1%), in which they *"can't type anything'* or websites' search features do not function.

The usability of certain websites was impaired due to blocking tools. In total, 3% of breakage threads mentioned faulty **links**, where clicking on a link had no apparent affect in the browser. Less than 1% of breakage threads noted **menus** were *"blank"* (Adblock, Chrome), *"drop down menus"* on websites were disabled (AdBlock Plus, Chrome), or *"navigation drop-down menu[s] [were] simply gone"* (uBlock Origin, Chrome).

Blocking tools also prevented **buttons** from working properly (2%). Buttons differ from links in that they more often execute JavaScript rather than simply navigating to a new page. Breakage in buttons sometimes overlapped with breakage in forms, resulting in failed authentication after *"clicking on sign in"* (HTTPS Everywhere, GitHub) or simply preventing users from submitting forms when buttons were unexpectedly disabled. Once again, breakage can have a concrete financial impact for users, such as a Safari Adblock user who became *"frustrated trying to* ***download*** *tax forms from a major financial investment web site."* About 1% of threads mentioned issues with **uploading** images and videos to Facebook (Disconnect, Chrome) and YouTube (Ghostery, Chrome).

Browser level: Around 6% of breakage threads reported issues that seemed to the user to manifest at the browser level, as opposed to the website level. Many AdBlock users on Chrome (in about 2% of threads) expressed frustration with the tool, noting *"Chrome crashes every time I enable [the extension]."* Users experienced **website crashes** (1%) in addition to **browser crashes**. A uBlock Origin user on Chrome, for example, noted that their page crashes *"every time if [they] upload a video to [social media] and trim it before posting."* In 16 threads, reviewers were less specific about whether their browser, webpage, extension, or even computer was crashing.

These issues were with **scrolling** (less than 1%), such as a DuckDuckGo Privacy Essentials user on Firefox who *"spent 15 minutes trying to find some support"* to understand why a website did *"not scroll with [the extension] installed."* Chrome Ghostery users who experienced issues with links and popups often found that videos *"on full screen [wouldn't] work."* Some users experienced additional **functionality** issues with browsers (less than 1%), particularly Chrome Ad-

Block Plus users whose back buttons and browsing history had *"10-20"* incorrect records of visits to the same URL.

Authentication and sessions: Many Privacy Badger users were affected by **login** issues (around 4%) where the blocking tool blocked Google or Facebook single-sign-on services, effectively blocking access to any websites using this form of authentication. Users were often directed to a *"blank screen"* after submitting their credentials or were unable to open the *"login popup"* to input their credentials. A Ghostery user on Chrome complained, *"I don't want to Disable Ghostery, but [I] must be able to do my banking."* One Privacy Badger user on Chrome reflected on the costs and benefits of blocking tools when faced with **CAPTCHA** issues:

> *...be prepared to spend 5-10 minutes a day playing the "ReCaptcha" clicking on the picture games for literally every site... Well that's 30-60 HOURS of [the] YEAR you are wasting with ReCaptcha because your user experience is indeed kinda broken. Some may say paying 30-60 hours of your LIFE away each year is a small price to pay for added privacy, but I think you are getting by just fine...*

While CAPTCHAs accounted for less than 1% of threads, the sentiment above was pervasive when dealing with most other forms of breakage.

Finally, a few users struggled with **session persistence**, which was mentioned in 23 threads. At least 10 uBlock Origin users on Chrome found that YouTube was not saving *"playback position[s] in videos."* In a Privacy Badger GitHub issue, one user reported authentication troubles on Google, noting *"it won't let you stay logged in."*

Vague: Around one fifth of threads had references to vague and imprecise forms of breakage. Some users were **generic** when describing breakage (15%), using language like *"this ad blocker breaks a lot of websites"* (AdBlock Plus, Safari). Others narrowed their problem to **missing content** (6%) but did not define this content, such as writing, *"It sometimes even causes most elements on the page to not show up"* (Ghostery, Chrome). Other reviews were similarly vague about a **broken interaction** (3%). For example, one Privacy Badger user on Chrome reported they were *"unable to open Google Contacts from my Gmail account."* Finally, a few **special case** incidents did not fit in any other categories (4%), such as reviews mentioning *"screen tearing"* (AdBlock Plus, Chrome) and *"rendering artifacts"* (AdBlock Plus, Chrome).

4.2 Taxonomy of Mitigation Strategies

We identified three high-level categories of actions users attempt to take when they encounter breakage. Table 3 details each of these high-level categories. By far, the most popular mitigation strategies involved limiting the functionality of the blocking tool in some way. Over 90% of threads that mentioned breakage included some mention of disabling the extension to some extent. Less common were strategies that involved taking some browser action (mentioned in just under 10% of threads), such as restarting the browser or clearing browser data, and performing page level actions (just under 5% of threads), such as reloading the page.

Table 3: Our taxonomy of strategies for fixing breakage. As in Table 2, for each category C we report: the percentage of threads that mention C across all that mention any form of breakage; the number of threads mentioning C; and the number of unique users mentioning C.

Mitigation	Mitigation subcategory	% of Breakage Threads	# of Threads	# of Users
Limiting functionality	Disable entirely	45.26	1202	1271
	Uninstall	32.53	864	962
	Modify block list	27.41	728	857
	Disable for page	6.59	175	184
	Reinstall	3.69	98	102
	Total	**90.21**	**2396**	**2834**
Browser level	Test alternate browser	5.23	139	143
	Restart browser	2.60	69	75
	Clear data	1.62	43	46
	Check developer console	0.98	26	28
	Reinstall browser	0.41	11	12
	Total	**9.98**	**265**	**281**
Page level	Reload page	3.01	80	82
	Test alternate website	1.13	30	30
	Abandon website	0.30	8	8
	Contact broken website	0.23	6	6
	Total	**4.63**	**123**	**125**
Vague	-	11.03	293	327

Limiting functionality: Some strategies users deployed limited the protections of the blocking tools, such as to block fewer trackers or ads. The least extreme case was **modifying the block list** of a given extension, reducing the number of domains or resources being blocked in hopes of fixing breakage. Figure 3 shows that this strategy is most often employed when the breakage is related to broken videos or when the page fails to load. This strategy reduces privacy less than many other approaches and was frequently mentioned (around 27%). More bluntly, **disabling the tool for the page** is often a blocking tool feature; users mentioned this strategy less (around 7%).

Other strategies were less privacy-preserving and often involved features external to a blocking tool. One strategy was **disabling the tool entirely**, usually via browser settings. This was by far the most popular mitigation strategy to fix breakage (45%). In addition to disabling a blocking tool, users can choose to **reinstall** the tool as a troubleshooting step or, in the most extreme case, **uninstall** the extension entirely.

Browser-level interventions: Browser-level interventions used settings within browsers to diagnose or fix breakage. Most popular was testing if the breakage occurred in **another browser** (5%). Many users, especially those who authored GitHub issues, mentioned using this strategy before posting about their issue, although developers also asked if users were experiencing breakage in other browsers or themselves at-

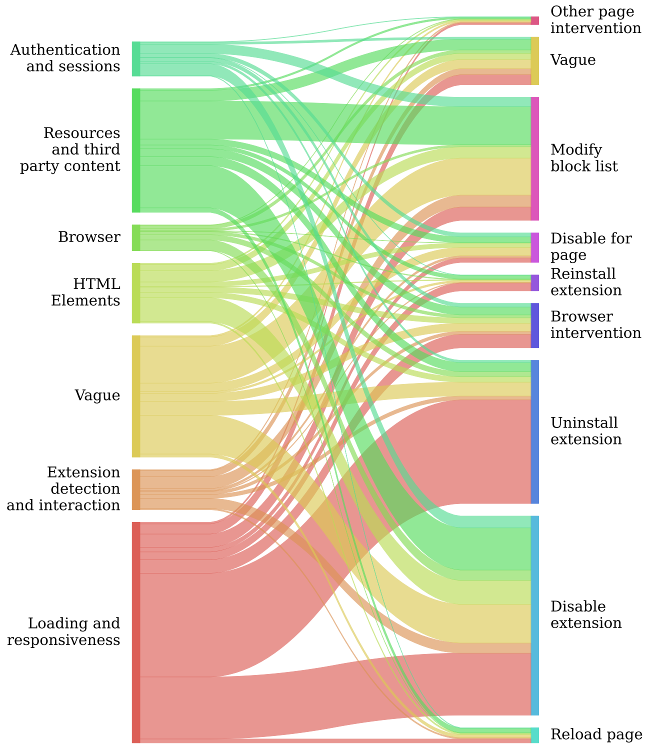


Figure 3: How often the most common mitigation strategies were used for the most common forms of breakage.

tempted to reproduce breakage across browsers. Another strategy mentioned primarily on GitHub was **checking the developer console** of a browser, either prompted by developers triaging issues or proactively included by the user. Developers from extensions like Privacy Badger that rely on user actions to detect trackers often advised users to **clear their browser or extension data**, including recreating user profiles. Similar strategies included **restarting** or **reinstalling** browsers in hopes of clearing even more data or refreshing states.

Page level interventions: Users infrequently mentioned **reloading** pages (3%) or **visiting other sites** (1%) to narrow down the source of the breakage. Less than 1% of threads mentioned **contacting the websites** that broke to seek help. Users and developers sometimes mentioned reaching out to broken websites when developers decided they would not fix the issue behind the breakage because websites seemed to be *"violating privacy"* (Privacy Badger, GitHub). Some users mentioned **abandoning** the website as a last resort, either temporarily or permanently.

Vague: We classified reviews that alluded to troubleshooting, diagnosis, or mitigation, yet were not specific, as **vague**. Some of these reviews were from employees of blocking tools linking to external resources, such as writing, *"If a website requires you to disable your ad blocker, we have some suggestions! Please see ... "* (AdBlock, Chrome). Others included instructions for the developers of sites affected by breakage and thus were not relevant for users who encounter breakage.

4.3 The Ecosystem of Breakage Diagnosis

While Figure 3 indicates that many users resorted to disabling or uninstalling blocking tools that cause breakage, it also shows that many users were not willing to give up on the tools. Additionally, the temporal aspect of troubleshooting breakage is not reflected in our mitigation strategies; a user who disables an extension momentarily may or may not enable it again. In fact, many users were explicit in saying they were only *"disabling for a while until this can be fixed"* (AdBlock, Chrome). A Chrome user of HTTPS Everywhere reflected:

> *love it, but uninstalled, it causes too many issues, spend half a day figuring out why [site] wasn't working in chrome (try it yourself), turned out to be this extension!.. disabled until bugfixed*

In contrast, another reviewer of HTTPS Everywhere wrote:

> *The costs outweighed the benefits and I have disabled this extension. I ran it for a few weeks and it culminated in chrome becoming exceptionally resource hungry and slow.*

Tolerance depended on the type of breakage, but also on the context. Users who uninstalled or disabled a blocking tool sometimes indicated the website they were using when breakage became unacceptable. A Chrome user said Disconnect *"messed up access to my online banking site - BIG NO NO - begone evil extension."* A uBlock Origin user on Chrome wrestled with a similar issue:

> *It is by far the best chrome extension I've used for comprehensively blocking all ads (even popups, chat box nonsense, etc). But it's also the most likely extension I've used (of any extension type) to create problems with functionality. It doesn't play well with many e-commerce sites, banking sites, etc.*

Users leveraged different strategies for different forms of breakage. For example, Figure 3 shows that **modifying the block list** was second only to **uninstalling the extension** for fixing **vague missing content**. In GitHub threads, long discussions about attempts at diagnosing or fixing breakage were frequent. For example, one thread in Privacy Badger's GitHub issue tracker started with a user reporting an issue logging into their banking site, followed by a developer requesting any domains that were blocked by the tool. The developer iteratively asked the user to attempt **blocking or unblocking specific domains**; the user responded with the outcome of each attempt. It was common for more than two users to participate in threads discussing potential solutions.

GitHub was a particularly fruitful forum for discussing breakage. In fact, a handful of users who were seemingly unaffiliated with blocking tools participated in several threads.

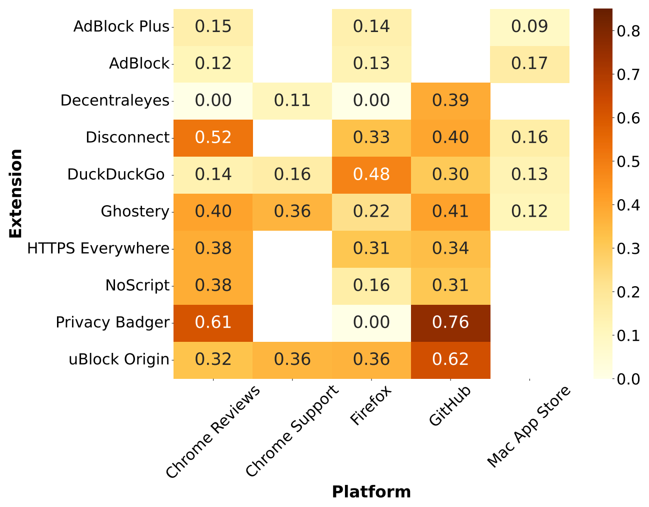


Figure 4: Proportion of threads mentioning breakage.

Developers from organizations external to the blocking tools we studied also contributed to GitHub issue trackers. Collaborations between tool developers, website owners whose sites broke, and users who experienced breakage were often successful at finding solutions.

5 Survey: Methods

To move beyond just what the users reported in the moment they wrote their review or issue report, as well as to reach users who are not inclined to post extension reviews or bug reports, we also conducted an online survey. In this survey we were able to ask users about all types of breakage they have experienced as well as well as gain a more subjective perspective on how users react to specific types of breakage in different contexts. We released two versions of the survey, one for users of blocking tools distributed as browser extensions and another for users of browsers with built-in privacy features. Both surveys followed the same basic structure, but the text of the questions and multiple choice options offered were tailored to the respective variant.

5.1 Recruitment

We recruited 100 participants (50 per survey variant) through the Prolific platform [48], though 5 participants failed an attention check and were subsequently excluded. We required participants be 18+ years old, from the United States, and have completed 20+ prior surveys on Prolific with a 95%+ approval rate. For the **extension variant**, we required that participants had ever installed and used one of the ten browser extensions included in our review analysis (Sections 3–4), even if they had subsequently uninstalled the extension. For the **browser variant**, we required that the participants used the Brave, Edge, Firefox, Opera, Safari, or Vivaldi browser at least once a week on a laptop or desktop computer. We excluded Chrome because it includes fewer blocking features. We compensated participants $6 USD via Prolific for completing the survey, which typically took about 20 minutes.

5.2 Survey Structure

The survey began by asking which browsers and which browser extensions the participant had ever used. Similar to prior work [61], we included fake options as an attention check. We then asked participants why they installed and, when applicable, uninstalled the extensions or browsers.

Next, we explained key types of breakage included in our taxonomy (Section 4) and asked what the participants would do if they experienced that breakage on common categories of websites (e.g., social media, news). Asking about the 38 distinct types of breakage in our taxonomy was infeasible, so we selected seven common, yet distinct, types of breakage to include in the survey: the page not loading; not being able to log into a website; a page loading slowly; an issue with fonts/style/layout/spacing; an image missing; a video not playing; and a message asking the user to disable their ad blocker (only in the browser-extension variant). The full descriptions of each type of breakage that were shown to the participants are given in Table 4. We also asked how much effort the participant would expend to try to fix each type of breakage, as well as the likelihood of them abandoning the website or extension if they could not fix it.

We then asked about the participant's past experiences of breakage. We first asked what types of breakage they remembered experiencing that they attributed to browser extensions or their browser (depending on the survey variant) from a list of the types of breakage in our taxonomy. We offered optional free-response boxes for describing experiences not covered by the options from the taxonomy. We also asked about their certainty that the breakage they experienced was caused by their extensions or browser. If the participant reported experiencing any type of breakage, we showed them the list of remediation techniques from our taxonomy and asked which actions they remembered taking, how successful those actions were, which ones they would take in similar situations, and whether they kept using extensions or browsers they believed caused breakage. We again provided free-responses boxes for strategies not covered by the taxonomy. If the participant reported that they had not experienced breakage, we asked similar questions posed as hypotheticals. The survey ended with a question about features participants wished existed for resolving breakage, as well as basic demographic questions.

5.3 Analysis and Metrics

We collected both quantitative and qualitative data. As with our analysis of reviews and issues, we conducted thematic analysis of participants' answers to free-response questions in

Table 4: Descriptions given of the seven types of breakage.

Type of Breakage	Survey Description
Page Fails to Load	In this scenario you attempt to access a website. When the website is functioning **correctly**, the page loads. When the website is functioning **incorrectly**, you only see a blank page.
Broken Login	In this scenario, you open a website. When the website is functioning **correctly**, you are able to log in to your account. When the website is functioning **incorrectly**, you are unable to log into your account.
Slow Page Load	In this scenario, you attempt to access a website. When the website is functioning **correctly**, the page loads almost instantly. When the website is functioning **incorrectly**, you experience a much slower load time, but eventually see the page you expect.
Formatting & Style	In this scenario, you open a website. When the website is functioning **correctly**, the page looks normal. When the website is functioning **incorrectly**, you see issues with fonts, style, layout, and spacing of elements.
Missing Images	In this scenario, you open a website. When the website is functioning **correctly**, you see an image on the page, but when it is functioning **incorrectly**, the image is missing.
Broken Video	In this scenario, you open a website. When the website is functioning **correctly**, you are able to play a video on the page. When the website is functioning **incorrectly**, the video does not play even after you click the play button.
Extension Detected	In this scenario, you attempt to access a website. When the website is functioning **correctly**, you can interact with and view the entire page. When the website is functioning **incorrectly**, a popup appears that tells you to disable your adblocker to use the website.

the survey. One member of the research team iteratively developed a codebook for each free-response question while coding all responses. A second member of the research team coded the same responses using the codebook created by the primary coder. At that point, we calculated inter-rater reliability. If the agreement was high (Cohen's $\kappa > 0.7$), we considered the codebook verified and used the primary coder's codes. Otherwise, the two coders met to discuss the codebook, repeating the process until high agreement was reached. We do not report the exact number of participants that mentioned a particular code for a given question, so reaching exact agreement between reviewers was not necessary.

5.4 Survey Limitations

The conclusions we draw from analyzing the survey data are limited by the accuracy of our participants' memory and recollections. While Tang et al. have shown that Prolific provides representative results for questions about user perceptions and experiences, this is not the case for questions about security and privacy knowledge [63]. Additionally, we mentioned in our Prolific task title and description that the study is related to blocking tools, which may bias our sample towards privacy-aware users. Finally, since we did not conduct an initial analysis of breakage-related reviews of web browsers as we did for extensions, survey questions that rely on the taxonomy we developed are skewed toward people's experiences with breakage due to extensions, not browsers. We do not aim to make any generalizable numerical claims with our survey data. Rather, our main goal with the survey was to explore participants' subjective experiences with breakage, mitigation, and reporting. The sample size of our survey prevents us from identifying any differences in experiences, behaviors, or opinions from subsets of the survey participants.

Similarly to the authors of the reviews, survey participants may have misattributed the breakage they remembered experiencing to the extension. We attempted to control for this by including examples and scenarios. While some participants' definitions of specific types of breakage might not be aligned with ours, we preferred that individuals be generous with what they consider breakage to minimize the risk of not describing a previously unreported type of breakage.

6 Survey: Results

In this section, we present the results of our survey. We removed the five responses that selected any of the fake browsers or extensions we included as an attention check. This left us with 46 participants in the extension variant of the survey and 49 in the browser variant.

Our sample skewed young and educated, as is typical of Prolific studies. Approximately 60% of our participants identified as male, whereas typical Prolific samples skew slightly female. One-third of our participants reported having a background in a technical field, which is perhaps an artifact of our recruitment criteria (e.g., having installed an extension). Table 8 in the appendix details participants' demographics.

AdBlock, AdBlock Plus, and uBlock Origin were the extensions participants most commonly used. The median number of the ten browser extension-based blocking tools participants reported having used was 2 (mean 2.3). Among participants, 57% reported having uninstalled a blocking extension, most commonly AdBlock or AdBlock Plus. A full breakdown of the extensions and browsers that participants used can be found in Figure 9 in Appendix A.

When asked *why* they uninstalled them, participants mentioned considering them unnecessary, switching to a different extension, privacy concerns, performance (e.g., failing to block ads), and that they caused websites to break. Note that at this point, the survey had not yet mentioned breakage.

Chrome was the most commonly used browser. While we required that participants used Brave, Edge, Firefox, Opera, Safari, or Vivaldi at least once a week, we did not exclude individuals who also used Chrome. The median number of browsers participants used at least weekly was 2 (mean 2.5). Only 11% of participants had ever stopped using or uninstalled one of their browsers. Despite not requiring it, 82% of the participants in the browser study also used at least one of the ten extensions we analyzed.

6.1 Breakage Types and Contexts

The surveys asked participants what they would do in response to encountering specific types of breakage on different types of websites. Figure 5a shows responses for the extension variant of the survey. For example, 83% of participants said they would attempt to fix their primary streaming service if it did not load, followed closely by encountering a broken login on their primary banking website (80%). On the other hand, only 22% of participants said they would attempt to restore a missing image or attempt to fix a slow loading time on their primary news site. In fact, across every type of breakage, news websites had the lowest fraction of participants stating that they would try to fix the breakage.

In terms of breakage type, missing images consistently had the smallest fraction of participants reporting that they would try to fix the breakage. Conversely, participants reported being far more likely to try to fix pages not loading, broken logins, and the extension being detected (e.g., a popup banner from a website imploring the user to disable their ad blocker).

Figure 5b shows parallel information for the browser version of the survey. Interestingly, we observed less variation in participants' intent to fix breakage than in the extension variant. We also observed some fluctuation in participant priorities between surveys. A participant's primary news site loading slowly was again the least likely for participants to attempt to fix (18%). However, pages loading slowly was the issue that the fewest participants would try to fix across most types of websites, a key difference from the extension variant.

We observed that participants in the extension variant who would attempt to fix breakage in at least one category of website were generally unlikely to permanently abandon a website, regardless of whether they were able to fix the breakage (Figure 10a in Appendix A). The notable deviation was that participants were more likely to do so if the website detected that a blocking extension was installed and asked them to disable it. We observed no parallel patterns in responses for the browser variant of the survey, though overall a higher percentage of people stated they were likely to abandon the website if their mitigation attempts failed (Figure 10b in Appendix A).

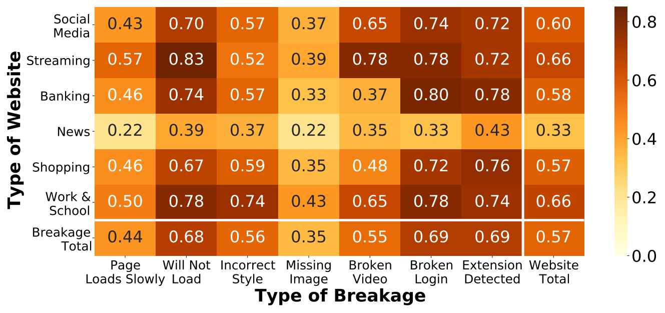
Compared to Figure 10a, Figure 11 in Appendix A shows that the same group of participants—those unable to fix the breakage they encountered—were more likely to report they would stop using the extension they believed was causing the breakage. In fact, around 50% of survey participants said they were either extremely likely or likely to abandon the extension they believed was causing the breakage if they could not find a way to fix it. This finding emphasizes the importance of tracking protection tools and filter list maintainers making sure that users can resolve breakage they encounter.

Table 5: Breakage types participants recalled experiencing.

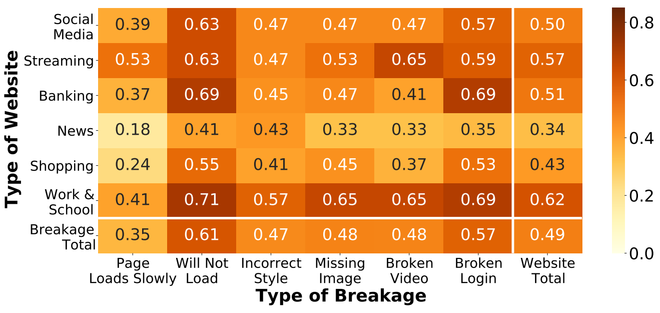
Type of Breakage	Extension		Browser	
Missing Popup	23	(50%)	16	(33%)
Extension Detected	22	(48%)	-	-
Page Doesn't Load	20	(43%)	22	(45%)
Missing Image	17	(37%)	16	(32%)
Page Loads Slowly	15	(32%)	30	(61%)
Broken Video	14	(30%)	17	(35%)
Formatting Issue	11	(24%)	19	(39%)
Missing Embedded Content	10	(22%)	13	(27%)
Unresponsive Page	9	(20%)	27	(55%)
Page Lags	9	(20%)	17	(35%)
Broken Link	8	(17%)	13	(27%)
Broken Download	8	(17%)	9	(18%)
Broken CAPTCHA	7	(15%)	19	(39%)
Broken or Missing Button	7	(15%)	6	(12%)
Broken Login	6	(13%)	12	(24%)
Browser Crashes	5	(11%)	33	(67%)
Broken Form	5	(11%)	13	(27%)
Page Does Not Save Previous Interaction	5	(11%)	9	(18%)
Broken or Missing Menu	3	(7%)	9	(18%)
Broken Audio	3	(7%)	9	(18%)
Breaks Another Extension	3	(7%)	-	-
Broken or Missing Comment Section	2	(4%)	10	(20%)
Broken Upload	2	(4%)	6	(12%)
Page Reloads by Itself	1	(2%)	12	(24%)
Broken Scroll	1	(2%)	8	(16%)
Broken Text Entry Field	1	(2%)	7	(14%)
Page Reports Issue Connecting to the Internet	0	(0%)	13	(27%)
Broken Browser Button	-	-	13	(27%)
None of the above	4	(9%)	0	(0%)

6.2 Past Experiences of Breakage

Nearly all participants (91% in the extension survey and 100% in the browser variant) reported that they had previously experienced breakage that they attributed to their respective tools. These percentages appear far higher than in prior work by Mathur et al. [41]; only 37.5% of their participants reported experiencing such breakage. We asked participants to select types of breakage they had experienced from a list we provided based on the taxonomy we developed, rather than as free-response questions, which could partially explain these differences. Using a list of options allowed us to use a standard frame of reference across participants, and it may have prompted them to remember breakage they might have otherwise forgotten. Participants in our study also reported attempting to fix breakage more often than Mathur et al.'s participants. Again, this may be due to the prescriptive framing of our question regarding troubleshooting and mitigation. These diverging findings are an opportunity for future work to investigate the different ways that users define, conceptualize, and attribute breakage.



(a) Extension.



(b) Browser.

Figure 5: The proportion of participants that stated they would attempt to fix the type of breakage for a given browsing context.

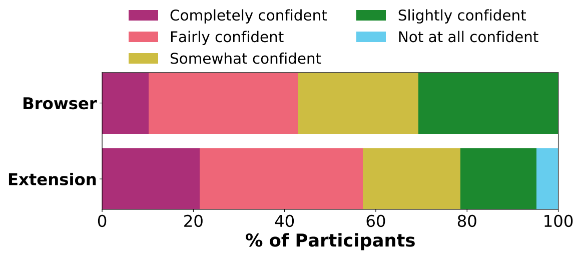


Figure 6: Participants' reported confidence in correctly attributing breakage to their extension or browser.

As shown in Table 5, the percentages of participants who recalled experiencing each type of breakage varied between extensions and browsers. The complexity of browsers may explain why the most common type of breakage that participants remembered encountering in the browser was the *browser itself* crashing (67%), compared to only 11% of participants reporting that their browser had crashed ($\Delta = 56\%$) in the extension version of the survey. Other key differences between the extension and browser variant include the "unresponsive page" ($\Delta = 35\%$), "page loads slowly" ($\Delta = 29\%$), "page reports issue connecting to the internet" ($\Delta = 27\%$), and "broken CAPTCHA" ($\Delta = 24\%$) categories.

Figure 6 shows participants' confidence in attributing breakage they had experienced to their extension or browser. Internet connection issues and problems with websites themselves were often cited as root causes of breakage that participants did not attribute to their browser or extension.

6.3 Past Mitigation Strategies

Every participant who reported experiencing breakage also reported taking some action to try to fix it. We provided participants with options from our taxonomy based on analyzing extension reviews, augmented with browser-specific options. Table 6 summarizes the fraction of participants who reported having employed each mitigation strategy in the past. While we again found a much higher fraction of participants reporting attempts to fix breakage compared to prior work [41], we note that we took a fairly inclusive definition of actions.

Table 6: Participants' mitigation strategies. Dashes indicate the option was not available for that survey variant.

Mitigation Strategy	Extension		Browser	
Reloaded Page	29	(69%)	45	(91%)
Disabled for Specific Website	29	(69%)	10	(20%)
Cleared Browser Data	17	(40%)	34	(69%)
Restarted Browser	16	(38%)	40	(82%)
Incognito Mode	14	(33%)	21	(43%)
Switched Browser	12	(29%)	33	(67%)
Updated Extension	11	(26%)	-	-
Restarted Computer	9	(21%)	27	(55%)
Modified Blocklist	9	(21%)	-	-
Disabled Extension Entirely	8	(19%)	-	-
Reinstalled Extension	7	(17%)	-	-
Uninstalled Extension	7	(17%)	-	-
Updated Browser	6	(14%)	28	(57%)
Asked Family/Friend/Peer	1	(2%)	5	(10%)
Changed Tracking Protection Level	-	-	8	(16%)
Disabled "use HTTPS by default"	-	-	3	(6%)
None of the above	0	(0%)	0	(0%)

In the extension survey, reloading the page and disabling the extension for the broken site were by far the most common actions participants reported (69% of participants reported having taken each). Conversely, only 17% of participants said that uninstalling an extension was a step they took to solve breakage. Over 3.5 times as many participants said they had disabled an extension for the entire page as opposed to modifying the blocklist for the page. This result suggests that when users encounter breakage, to fix the issue they often choose to take an action that would expose them to all potential tracking, rather than following a more complex process to allow just the resources necessary to fix the breakage.

Many participants reported that they kept using their extensions or browsers despite experiencing breakage. The reasons that they gave included the relative rareness of breakage, that they were often able to fix the issues, and that they still wanted the functionality that the extensions provided. Many participants in the browser variant mentioned believing that the breakage would be fixed and the problems would go away without the need for any intervention on their part. For example, participant B24 stated that they kept using a browser they thought to be responsible for breakage they were experiencing "because I think they are temporary problems, sooner or later they are corrected with an update."

Comparing our analysis of extension reviews with our survey results, we observed several key differences in the actions users reported taking. Few reviews mentioned refreshing the page, but that was one of the most common actions reported in the survey. This could be because users did not think it was relevant to include this action in their reviews, especially if it did not solve the problem, highlighting the importance of our approach to understanding breakage through multiple complementary lenses.

Conversely, one of the most common mitigation strategies mentioned in reviews was uninstalling an extension, which was not nearly as common in the survey. This difference may speak to the context of reporting this information. Since we looked at reviews with three or fewer stars, as well as support requests and GitHub issues, we were potentially seeing a disproportionate number of people upset about breakage and willing to take more drastic steps, such as completely uninstalling the extension they believed to be at fault.

Another difference we observed was in users disabling the extension generally, versus disabling it only for a specific site. In the survey, three times as many participants reported disabling the extension for a page rather than globally. In analyzing reviews, we had to rely on sometimes ambiguous language; many posts simply mentioned disabling the extension without specifying if it was for all pages or just one.

6.4 Reporting Breakage

One way users can address breakage is by reporting specific instances to the developers of blocking tools so they can diagnose and fix those issues. Unfortunately, as shown in Table 7, 67% of participants in the extension variant and 57% in the browser variant who recalled experiencing breakage said that they had never taken any action to report any of the breakage they had experienced. Furthermore, Table 7 potentially represents an overestimation of users' reporting behavior due to the perceived social desirability of taking such actions. Regardless, only looking at reports sent to tool maintainers is thus likely to miss a large portion of the breakage that users experience. Among participants who recalled reporting breakage, the most common method was through features built into either the extension or browser. The prevalence of user reports is generally not public. However, a recent talk reported that Brave receives over 2,000 reports of breakage per day through their "report a broken site" feature [51].

6.5 Desired Features

Finally, we asked participants to describe features they wished existed to help them handle breakage they might encounter. While most participants could not think of any features they would be interested in having, the most common request from those who did boiled down to wanting instructions on how to fix the breakage they were experiencing. A few participants also wanted some way of figuring out what was causing the breakage, which ties back to the fact that some participants were unsure what was causing the breakage they had experienced even if they suspected it was their extension. Interestingly, several participants mentioned wanting some form of crowdsourcing for understanding if other people had experienced breakage on a particular site, as well as what steps they took to resolve it. Some participants also mentioned live support chats and dedicated breakage forums.

Table 7: Participants' recollections of reporting breakage (among those who had experienced breakage).

Reporting Method	**Extension**		**Browser**	
Never Reported	28	(67%)	28	(57%)
Feature in Tool	8	(19%)	12	(25%)
Email	3	(7%)	2	(4%)
Internet Forum	2	(5%)	8	(16%)
Social Media	2	(5%)	2	(4%)
Online Reviews	1	(2%)	5	(10%)
Website for Tool	1	(2%)	3	(6%)
Filter List	1	(2%)	2	(4%)
GitHub	1	(2%)	1	(2%)
Other	0	(0%)	2	(4%)

7 Ethics

All reviews, posts, and issues we analyzed are publicly available. While we do store information about the author of a post, we do not report the names or usernames of authors in this paper, nor include post excerpts that contain identifiable information. Knowing a post's author was necessary during the coding process to distinguish between different discussants on a thread, as well as between users and developers.

Both our survey and review analysis procedures were approved by our Institutional Review Board (IRB). Participants recruited for the survey portion of this study had to provide their explicit consent to participate. None of the questions in the survey asked users to disclose any personally identifiable information beyond general demographics.

8 Discussion and Conclusions

In this paper, we used the complementary lenses of qualitative analysis of publicly posted information for browser extension-based blocking tools and a 95-participant online survey to understand blocking tool-related website breakage far more comprehensively and systematically than in prior work. Specifically, we developed novel taxonomies of the specific issues users experience as part of a "website breaking," as well as user strategies for attempting to fix those issues. In this section, we conclude by discussing our work's lessons for various stakeholders in the tracking protection tool ecosystem and directions for future work.

8.1 Key Takeaways for Tool Developers

Prior work typically discussed breakage as an abstract, monolithic concept or through a few chosen examples. For instance, Mathur et al. specifically mentioned five types of breakage [41]. In contrast, we identified 38 distinct types of breakage that users have reported experiencing due to blocking tools. The developers of tracking-protection tools could use the breakage taxonomy to create internal benchmarks and logs to measure the frequency of types of breakage associated with their tool. Additionally, our work highlights particular types of breakage, such as pages failing to load, which are strongly associated with users uninstalling a given extension. Extension developers should pay particular attention to types of breakage that are associated with such drastic actions.

Current methods of reporting breakage require significant effort for end users, which may lead to breakage being underreported. Less than half of the participants in our extension survey said they had notified tool developers of breakage even though 91% said they had experienced breakage. Not all browsers and extensions are equipped with this capability. Even those that are may not make this reporting functionality easily accessible or apparent. **Tracking protection tools may receive more reports sooner if they provided users with an easy-to-find and low-effort way of reporting breakage**.

We observed that user reports of breakage typically did not include enough information to give a developer any chance of fixing the issue without additional follow-up. Having reporting templates that would require minimal effort or technical knowledge (e.g., accessing the console for the extension) for users to fill out could help streamline the process. Privacy Badger, for example, requests that users answer very specific questions before creating a new GitHub Issue, which reduces back-and-forth by providing developers and other users with key information needed to provide support.

Through our review analysis, we found that the developers of tracking protection tools who interacted with users who experienced breakage were often successful at finding solutions. This suggests that maintaining an active presence on these platforms can be beneficial both to the users that report breakage in this way and to the developers. In the absence of "proper channels" for reporting breakage, users expressed interest in crowdsourced solutions. In particular, survey respondents requested a method to **connect with other users to see if they had experienced breakage on a given site and, if so, learn how those to fix it**. This could perhaps take a form analogous to Downdetector [47], a website that provides a degree of crowd-sourced troubleshooting.

We also noticed users would sometimes post reviews to different tools in waves that seemed to be about the same underlying issue—frequently a filter list issue rather than a blocking tool issue. **Blocking tool developers could collaborate to build common and centralized reporting processes for issues related to these filter lists.** This could potentially alleviate swaths of users who are reporting the same issue to multiple venues by having fixes be identified and rolled out in the filter list rather than through the tracking protection tool.

Armed with the knowledge of users' mitigation strategies, **blocking tool developers can invest in features that leverage common strategies and prioritize types of breakage that users find the most intolerable**. While recommending fine-grained fixes that preserve user privacy while addressing the breakage would be ideal, a first step could be highlighting simple remediation actions that are less drastic than disabling or uninstalling the tool completely. Temporarily disabling the extension on a per-site basis and undoing previous blocking actions should be clear and easy actions for end users to take. Further, by leveraging our taxonomy of mitigation strategies, **browsers could detect user behavior that indicates breakage is occurring** (e.g., excessive reloading or disabling of extensions) and prompt users to report the broken page or provide suggestions for solving their issue.

Quantifying the frequency and types of breakage that end users experience would be a significant contribution to this area of research. While our survey reveals that many users who experience breakage do not report it, the most common method of reporting breakage was through a feature built into a tracking protection tool. Unlike the reviews we analyzed, these reports are not publicly available. Developers of these tracking protection tools have unique access to list of sites on which users reported experiencing breakage and the state of the tracking protection tool at the time the breakage was reported at the very least. Analyzing this more detailed data set could build on our findings and increase the granularity of the taxonomies we present in this work.

8.2 Key Takeaways for Website Developers

Understanding the ways websites can break could allow web engineers to check that popular blocking tools do not interfere with the functionality users expect from their sites. The taxonomy provided in this paper could be used to create a test suite for checking for the presence of breakage.

We found in both our review analysis and survey that users engage in different actions depending on the type of breakage they experience, as well as the importance or relevance of the website on which they experience breakage. Table 5 could be used to prioritize which breakage to address based on the type of website. In the case where the breakage cannot be fixed by website developers, providing users with instructions for fixing the issue through the tool could be beneficial so they know the cause of the problem. That being said, we found in our survey that around half of participants were either likely or very likely to abandon a website if it alerted a user, similar to our proposal above, that an extension had been detected and the user was unable to circumvent the alert. Therefore, these notices of instructions would need to be carefully considered to prevent users from abandoning the site or tool.

8.3 Key Takeaways for Researchers

With breakage resulting in users giving up the protection of their blocking tools, breakage must be central when evaluating new tools. For task-driven manual analysis of breakage, the tasks should relate to what users would actually attempt to fix in the context of the site. Additionally, the types of actions that need to be taken to circumvent the issues require varying degrees of effort for a user without technical expertise.

Due to the small scale of our surveys, sub-dividing the data to look for patterns was not possible and could be explored in future work. For example, analyzing whether actions users had taken in the past and their reactions to the scenarios had any connection to demographics could reveal how different groups of users respond to breakage.

Our survey asked how users would react to seven types of breakage we had identified. Future work could expand this line of questioning to include other types of breakage in different contexts or ask about the actions users would or would not take at finer granularity. While both the reviews and surveys give insights into the actions that users take to mitigate breakage, a more specific investigation of the thought process and ordered steps users take could prove useful for identifying pain points in the breakage mitigation process.

Acknowledgments

We thank Julia Hanson for her assistance. This material is based upon work supported by the National Science Foundation under Grant No. CNS-2047827 and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2140739.

References

[1] Ruba Abu-Salma and Benjamin Livshits. Evaluating the end-user experience of private browsing mode. In *Proc. CHI*, 2020.

[2] AdBlock. https://getadblock.com/en/.

[3] Adblock Plus. https://adblockplus.org/.

[4] Abdul Haddi Amjad, Danial Saleem, Muhammad Ali Gulzar, Zubair Shafiq, and Fareed Zaffar. TrackerSift: Untangling mixed tracking and functional web resources. In *Proc. IMC*, 2021.

[5] Andrés Arrieta. The browser privacy arms race: Which browsers actually protect your privacy? In *Proc. USENIX Engima*, 2020.

[6] Abhijit Balaji. Spacy-langdetect: Fully customizable language detection pipeline for spaCy. https://github.com/Abhijit-2592/spacy-langdetect.

[7] Natã M. Barbosa, Gang Wang, Blase Ur, and Yang Wang. Who am I? A design probe exploring real-time transparency about online and offline user profiling underlying targeted ads. *PACM IMWUT*, 5(3), 2021.

[8] Muhammad Ahmad Bashir and Christo Wilson. Diffusion of user tracking data in the online advertising ecosystem. *PoPETs*, 2018(4):85–103, 2018.

[9] Kevin Borgolte and Nick Feamster. Understanding the performance costs and benefits of privacy-focused browser extensions. In *Proc. WWW*, 2020.

[10] Brave Browser. https://brave.com/.

[11] Ismael Castell-Uroz, Rubén Sanz-García, Josep Solé-Pareta, and Pere Barlet-Ros. Demystifying content-blockers: Measuring their impact on performance and quality of experience. *IEEE TNSM*, 19(3):3562–3573, 2022.

[12] Ismael Castell-Uroz, Josep Solé-Pareta, and Pere Barlet-Ros. Demystifying content-blockers: A large-scale study of actual performance gains. In *Proc. CNSM*, 2020.

[13] Quan Chen, Peter Snyder, Ben Livshits, and Alexandros Kapravelos. Detecting filter list evasion with event-loop-turn granularity JavaScript signatures. In *Proc. IEEE S&P*, 2021.

[14] Kovila P. L. Coopamootoo, Maryam Mehrnezhad, and Ehsan Toreini. "I feel invaded, annoyed, anxious and I may protect myself": Individuals' feelings about online tracking and their protective behaviour across gender and country. In *Proc. USENIX Security*, 2022.

[15] cowboy-bebug. App-store-scraper: Single API App Store Review Scraper. https://pypi.org/project/app-store-scraper/.

[16] Savino Dambra, Iskander Sanchez-Rola, Leyla Bilge, and Davide Balzarotti. When Sally met trackers: Web tracking from the users' perspective. In *Proc. USENIX Security*, 2022.

[17] Ha Dao, Johan Mazel, and Kensuke Fukuda. CNAME cloaking-based tracking on the web: Characterization, detection, and protection. *IEEE TNSM*, 18(3):3873–3888, 2021.

[18] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *PoPETs*, 2015(1):92–112, 2015.

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

[20] Claire Dolin, Ben Weinshel, Shawn Shan, Chang Min Hahn, Euirim Choi, Michelle L. Mazurek, and Blase Ur. Unpacking perceptions of data-driven inferences underlying online targeting and personalization. In *Proc. CHI*, 2018.

[21] DuckDuckGo. DuckDuckGo Privacy Essentials. https://duckduckgo.com/app.

[22] Electronic Frontier Foundation. HTTPS Everywhere. https://www.eff.org/https-everywhere.

[23] Electronic Frontier Foundation. Privacy Badger. https://privacybadger.org/.

[24] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proc. CCS*, 2016.

[25] Tatiana Ermakova, Benjamin Fabian, Benedict Bender, and Kerstin Klimek. Web tracking - A literature review on the state of research. In *Proc. HICSS*, 2018.

[26] Fanboy, MonztA, Famlam, and Khrin. EasyList. https://easylist.to/.

[27] Kiran Garimella, Orestis Kostakis, and Michael Mathioudakis. Ad-blocking: A study on performance, privacy and counter-measures. In *Proc. WebSci*, 2017.

[28] Georgia Department of Public Health. COVID-19 status report. https://dph.georgia.gov/covid-19-status-report.

[29] Arthur Gervais, Alexandros Filios, Vincent Lenders, and Srdjan Capkun. Quantifying web adblocker privacy. In *Proc. ESORICS*. 2017.

[30] Ghostery. https://www.ghostery.com/.

[31] Ghostery Editorial. Tracking the trackers 2020: Web tracking's opaque business model of selling users, 2020. https://www.ghostery.com/blog/tracking-the-trackers-2020-web-trackings-opaque-business-model-of-selling-users.

[32] Raymond Hill and Nik Rolls. uBlock Origin. https://ublockorigin.com/.

[33] Panchakshari N. Hiremath, Jack Armentrout, Son Vu, Tu N. Nguyen, Quang Tran Minh, and Phu H. Phung. MyWebGuard: Toward a user-oriented tool for security and privacy protection on the web. In *Proc. FDSE*, 2019.

[34] Nicolas Huaman, Sabrina Amft, Marten Oltrogge, Yasemin Acar, and Sascha Fahl. They would do better if they worked together: The case of interaction problems between password managers and websites. In *Proc. IEEE S&P*, 2021.

[35] Umar Iqbal, Peter Snyder, Shitong Zhu, Benjamin Livshits, Zhiyun Qian, and Zubair Shafiq. AdGraph: A graph-based approach to ad and tracker blocking. In *Proc. IEEE S&P*, 2020.

[36] Umar Iqbal, Charlie Wolfe, Charles Nguyen, Steven Englehardt, and Zubair Shafiq. Khaleesi: Breaker of advertising and tracking request chains. In *Proc. USENIX Security*, 2022.

[37] Jordan Jueckstock, Peter Snyder, Shaown Sarker, Alexandros Kapravelos, and Benjamin Livshits. Measuring the privacy vs. compatibility trade-off in preventing third-party stateful tracking. In *Proc. WWW*, 2022.

[38] Ankit Kariryaa, Gian-Luca Savino, Carolin Stellmacher, and Johannes Schöning. Understanding users' knowledge about the privacy and security of browser extensions. In *Proc. SOUPS*, 2021.

[39] Balachander Krishnamurthy, Delfina Malandrino, and Craig E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. In *Proc. SOUPS*, 2007.

[40] Giorgio Maone. NoScript. https://noscript.net/.

[41] Arunesh Mathur, Jessica Vitak, Arvind Narayanan, and Marshini Chetty. Characterizing the use of browser-based blocking extensions to prevent online tracking. In *Proc. SOUPS*, 2018.

[42] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *Proc. IEEE S&P*, 2012.

[43] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *Proc. EuroS&P*, 2017.

[44] Mozilla. Firefox. https://www.mozilla.org/en-US/firefox/.

[45] Miguel Müller. Self-healing Javascript errors caused by the browser extension Privacy Badger. Master's thesis, KTH Royal Institute of Technology, June 2021.

[46] Katie O'Donnell and Henriette Cramer. People's perceptions of personalized ads. In *Proc. WWW*, 2015.

[47] Ookla. Downdetector. https://downdetector.com.

[48] Prolific. https://www.prolific.co/.

[49] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proc. IMC*, 2015.

[50] Raymond Rientjes. Decentraleyes. https://decentraleyes.org/.

[51] Shivan Kaul Sahib and Anton Lazarev. Bringing content blocking to the masses: Dealing with filter list development, maintenance, and compatibility for 50 million users. In *Proc. USENIX PEPR*, 2022.

[52] John Sammons and Michael Cross. Chapter 2 - Before connecting to the Internet. In John Sammons and Michael Cross, editors, *The Basics of Cyber Safety*, pages 29–52. Syngress, Boston, January 2017.

[53] Florian Schaub, Aditya Marella, Pranshu Kalvani, Blase Ur, Chao Pan, Emily Forney, and Lorrie Faith Cranor. Watching them watching me: Browser extensions' impact on user privacy awareness and concern. In *Proc. USEC*, 2016.

[54] Justin Schuh, Tanvi Vyas, Yan Zhu, and Eric Lawrence. Browser privacy: Opportunities and tradeoffs. In *Proc. USENIX Enigma*, 2020.

[55] Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. WebGraph: Capturing advertising and tracking information flows for robust blocking. In *Proc. USENIX Security*, 2022.

[56] Michael Smith, Pete Snyder, Benjamin Livshits, and Deian Stefan. SugarCoat: Programmatically generating privacy-preserving, web-compatible resource replacements for content blocking. In *Proc. CCS*, 2021.

[57] Michael Smith, Peter Snyder, Moritz Haller, Benjamin Livshits, Deian Stefan, and Hamed Haddadi. Blocked or broken? Automatically detecting when privacy interventions break websites. *PoPETs*, 2022(4):6–23, 2022.

[58] Peter Snyder, Antoine Vastel, and Ben Livshits. Who filters the filters: Understanding the growth, usefulness and efficiency of crowdsourced ad blocking. In *Proc. SIGMETRICS*, 2020.

[59] Software Freedom Conservancy. Selenium. https://www.selenium.dev/.

[60] Spam404. https://www.spam404.com/.

[61] Peter Story, Daniel Smullen, Yaxing Yao, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. Awareness, Adoption, and Misconceptions of Web Privacy Tools. *PoPETs*, 2021(3):308–333, 2021.

[62] Latanya Sweeney. Discrimination in online ad delivery. *CACM*, 56(5):44–54, May 2013.

[63] Jenny Tang, Eleanor Birrell, and Ada Lerner. Replication: How well do my results generalize now? The external validity of online privacy and security surveys. In *Proc. SOUPS*, 2022.

[64] Blase Ur, Pedro Giovanni Leon, Lorrie Faith Cranor, Richard Shay, and Yang Wang. Smart, useful, scary, creepy: Perceptions of online behavioral advertising. In *Proc. SOUPS*, 2012.

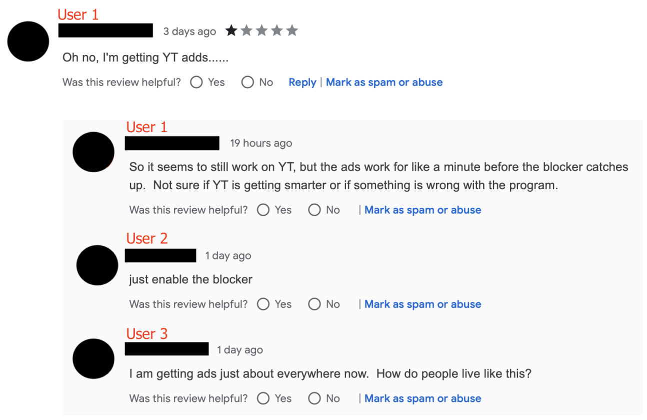
[65] Sophie Veys, Daniel Serrano, Madison Stamos, Margot Herman, Nathan Reitinger, Michelle L. Mazurek, and Blase Ur. Pursuing usable and useful data downloads under GDPR/CCPA access rights via co-design. In *Proc. SOUPS*, 2021.

[66] Miranda Wei, Madison Stamos, Sophie Veys, Nathan Reitinger, Justin Goodman, Margot Herman, Dorota Filipczuk, Ben Weinshel, Michelle L. Mazurek, and Blase Ur. What Twitter knows: Characterizing ad targeting practices, user perceptions, and ad explanations through users' own Twitter data. In *Proc. USENIX Security*, 2020.

[67] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L. Mazurek, and Blase Ur. Oh, the places you've been! User reactions to longitudinal transparency about third-party web tracking and inferencing. In *Proc. CCS*, 2019.

[68] Yuxi Wu, Panya Gupta, Miranda Wei, Yasemin Acar, Sascha Fahl, and Blase Ur. Your secrets are safe: How browsers' explanations impact misconceptions about private browsing mode. In *Proc. WWW*, 2018.

A Additional Figures

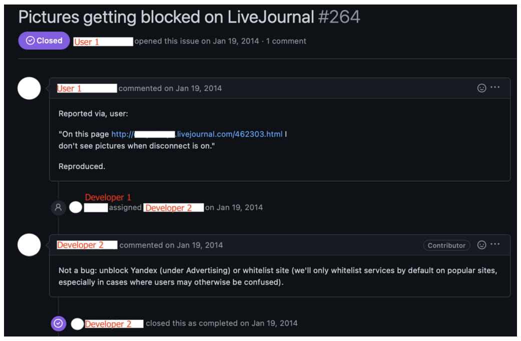

User 1
[redacted] 3 days ago ★☆☆☆☆
Oh no, I'm getting YT adds......
Was this review helpful? ○ Yes ○ No Reply | Mark as spam or abuse

User 1
[redacted] 19 hours ago
So it seems to still work on YT, but the ads work for like a minute before the blocker catches up. Not sure if YT is getting smarter or if something is wrong with the program.
Was this review helpful? ○ Yes ○ No | Mark as spam or abuse

User 2
[redacted] 1 day ago
just enable the blocker
Was this review helpful? ○ Yes ○ No | Mark as spam or abuse

User 3
[redacted] 1 day ago
I am getting ads just about everywhere now. How do people live like this?
Was this review helpful? ○ Yes ○ No | Mark as spam or abuse

Figure 7: A single thread from the review section for uBlock Origin in the Chrome Web Store with 4 posts in it, two being from the same user.



Pictures getting blocked on LiveJournal #264

Closed User 1 [redacted] opened this issue on Jan 19, 2014 · 1 comment

User 1 [redacted] commented on Jan 19, 2014

Reported via, user:

"On this page http://[redacted].livejournal.com/462303.html I don't see pictures when disconnect is on."

Reproduced.

Developer 1 [redacted] assigned Developer 2 [redacted] on Jan 19, 2014

Developer 2 [redacted] commented on Jan 19, 2014 Contributor

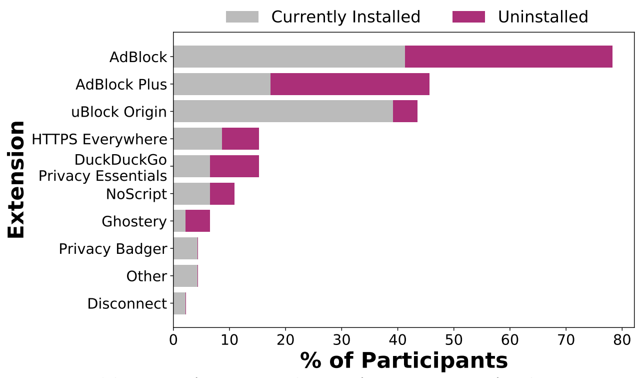
Not a bug: unblock Yandex (under Advertising) or whitelist site (we'll only whitelist services by default on popular sites, especially in cases where users may otherwise be confused).

Developer 2 [redacted] closed this as completed on Jan 19, 2014

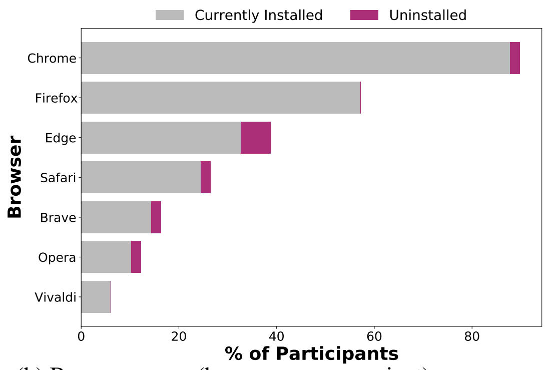
Figure 8: A single thread (issue) for Disconnect on GitHub with 2 posts in it from two unique users, one of which (Developer 2) is a developer for the extension.

Table 8: Survey participants' demographics.

Gender	Extension	Browser	Education	Extension	Browser
Male	29	30	High school	13	10
Female	17	19	Some college	5	6
Non-Binary	0	0	Trade / vocational	3	5
Prefer to self-describe	0	0	Associate's degree	3	1
Prefer not to answer	0	0	Bachelor's degree	20	23
Technical Experience			Master's degree	2	3
No	32	29	Master's degree	2	3
Yes	14	20	Doctorate	0	1

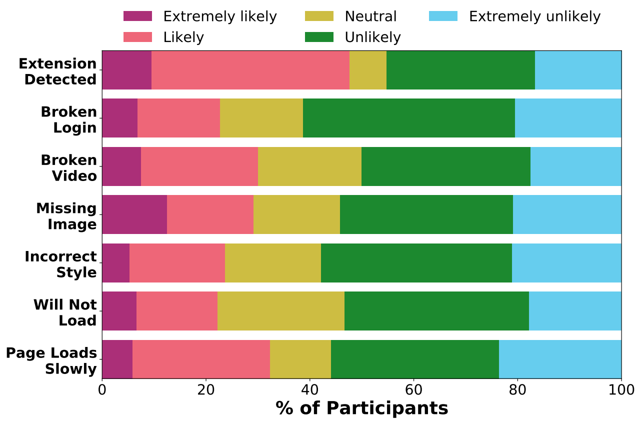


(a) Extension usage (extension survey variant).

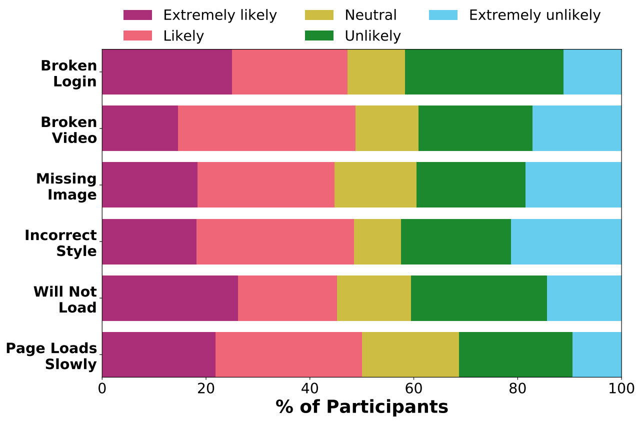


(b) Browser usage (browser survey variant).

Figure 9: The percentage of participants who used each extension or browser currently (gray) or previously (magenta).



(a) Abandon website (extension survey variant).



(b) Abandon website (browser survey variant).

Figure 10: Participants' reported likelihood to abandon the website if their attempt to fix the breakage failed.

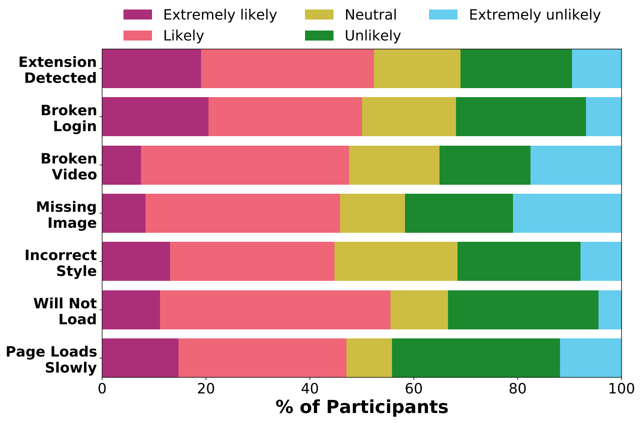


Figure 11: Participants' reported likelihood to abandon the extension if their attempt to fix extension-related breakage failed.