Unintended Privacy Risks of Using Assistive Technology on Web Applications

Anonymous Author(s)

Abstract

Screen readers are essential for making the web accessible to Blind and Visually Impaired (BVI) users. After loading a website, BVI users typically navigate via the mouse cursor or keyboard (*e.g.,* Tab), triggering the screen reader to vocalize alternative descriptions of the focused elements. Modern websites are heavily saturated with advertisements (ads), which is a key revenue source for sustaining free internet, but also a pervasive avenue for involuntary user tracking. In the course of navigation, BVI users inevitably interact with ad elements, causing unintended gestures (*e.g.,* mouseover or focus) that can inadvertently trigger data sharing with tracking entities. Consequently, BVI users' privacy may be compromised simply by locating web elements that non-BVI users can easily avoid when passively viewing content.

We conduct the first large-scale analysis of its kind on web applications to measure the prevalence of this issue, identifying ads on popular websites that lead to privacy violations when accessed by BVI users through gestures like mouseover. We develop an automated framework that simulates BVI user gestures, triggers screen reader behaviors, and examines subsequent privacy-invasive activities. Analyzing 67K ad elements from the top 3.5K news websites, we find that gestures required for screen readers trigger an average of 5.2 privacy-invasive data-sharing events per website. In contrast, non-BVI users can avoid triggering such events altogether by simply refraining from interacting with ad elements. This option is not available to BVI users as they must engage with all focusable content, including ads, in the navigation path towards the content they desire to locate and access. As web developers bear responsibility for the interactions on their websites, we propose an automated, privacy-preserving overlay injection for ads for developers to use. This overlay maintains the web page's accessibility while preventing unintended privacy violations. Our experiments with 67K ads across 3.5K websites demonstrate that developers can dynamically inject these overlays, eliminating all tracking incidents (achieving a 100% reduction) for BVI users employing screen readers.

CCS Concepts

• **Human-centered computing → Accessibility systems and tools**; Empirical studies in accessibility; • **Security and privacy** → *Web application security*; • **Information systems** → Web measurement.

Conference acronym 'XX, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Keywords

web, privacy, accessibility

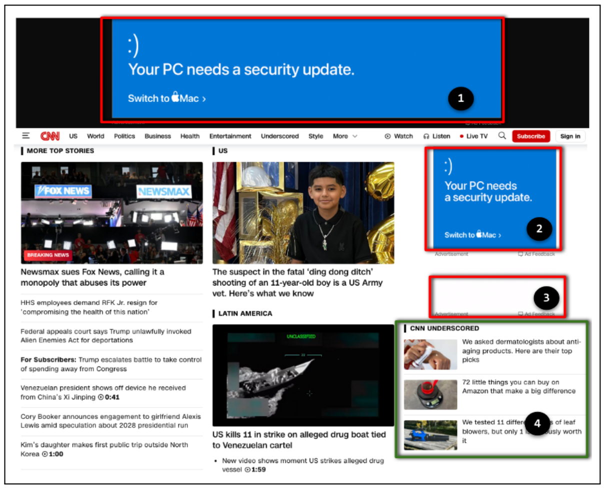
ACM Reference Format:
Anonymous Author(s). 2018. Unintended Privacy Risks of Using Assistive Technology on Web Applications. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

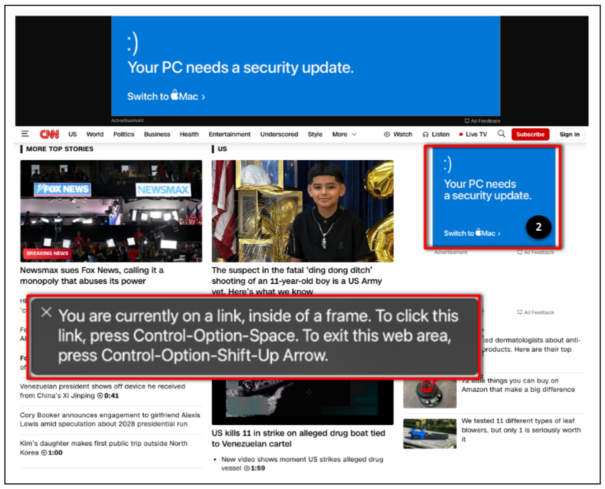
Blind and Visually Impaired (BVI) users rely on assistive technologies, such as screen readers, to navigate and interact with the internet. Using a screen reader during web browsing, BVI users build spatial and contextual awareness of the website's content by repetitively (1) pointing to (if using a mouse cursor) or (2) focusing on (if using the keyboard Tab key) individual web elements, signaling screen reader to vocalize the content of the focused element [14, 16, 21, 46]. Such mouse and keyboard gestures (*e.g.,*`mouseover`, `hover`, `mousemove`, `focus`) are key in reading selected content of the website, knowing what and where each web element is. Leading screen readers—including NVDA [14], VoiceOver [21], and Orca [16]—explicitly document support for these gestures as key interaction events. Reading content without the use of gestures causes spatial disorientation and context loss for BVI users [55, 75].

While these gestures are necessary for the effective use of screen readers, they may have unintended privacy consequences. Websites today are saturated with digital advertisements (ads), one of the few revenue sources that support free Internet, but also the primary instrument used by ad-services to track user browsing behavior, leading to user privacy violations. Any interaction with ads (*i.e.,* hover) will likely trigger network requests to a third-party service and deliver the user's personal identifier and related information [43, 47, 53]. Developers integrate ads by leasing space to ad-services (e.g., Taboola [19]) that dynamically deliver personalized content to visitors. In doing so, developers lose control over the ads and ads' behavior on the website.

Ads cover nearly 20% of the web visual surface area [36]. Thus, BVI users will inevitably reach an ad element on the website during exploration, performing mouseover or tab focus gestures to direct the screen reader to read the ads [45]. Users of screen readers such as Apple VoiceOver, NVDA, and Orca have reported difficulties [40, 41, 69] interacting with ad elements, which trigger unintended behavior. Prior work, including a BVI user study based on participant feedback [60], confirms that ads can significantly hinder accessibility and frequently trigger unintended behavior during screen reader navigation. There is over a decade of research showing that ads aggressively collect user information [44, 52, 61]. *We hypothesize that gestures required to use screen readers cause ads to transmit user data to third-party tracking entities. Unlike non-BVI users, who can glance at website content without generating detectable*


:)
Your PC needs a security update.
Switch to Mac ›
1
2
3
4
MORE TOP STORIES
US
LATIN AMERICA
CNN UNDERSCORED
Newsmax sues Fox News, calling it a monopoly that abuses its power
The suspect in the fatal 'ding dong ditch' shooting of an 11-year-old boy is a US Army vet. Here's what we know
US kills 11 in strike on alleged drug boat tied to Venezuelan cartel

(a) cnn.com with ad-slots (❶–❸) and podcast section (❹)


:)
Your PC needs a security update.
Switch to Mac ›
2
You are currently on a link, inside of a frame. To click this link, press Control-Option-Space. To exit this web area, press Control-Option-Shift-Up Arrow.

(b) MacBook VoiceOver on cnn.com interacting with the ad slot –❷

Figure 1: A case study demonstrating ads on CNN.com and the use of MacBook VoiceOver while interacting with one of the ads.

events, BVI users are forced to trigger hover or focus actions simply to navigate, as screen readers sequentially announce content through these events. Thus, while non-BVI users interact with ads by choice, BVI users are involuntarily exposed to tracking as a byproduct of using assistive technologies to access website content.

Contributions. We present the first large-scale empirical investigation into the prevalence and nature of privacy implications associated with using screen readers in web applications, as well as the responsibility of web developers in this context. To do so, we develop an end-to-end automated web evaluation framework that executes gestures necessary to trigger screen readers, captures the concise behavioral responses to those gestures, classifies behaviors as tracking or non-tracking, and, when tracking is detected, categorizes the nature of the privacy violation by analyzing the shared data. Each phase in the framework poses unique technical and research challenges.

First, to perform gestures that invoke screen readers, we must accurately identify the ad elements on a website among thousands of other benign elements. Since ads are integrated into the website by web developers, they use specific HTML tags to insert ads (*e.g.,*`iframe`). We utilize lists of HTML class names and IDs from community-maintained and verified resources such as EasyList [32] to detect ads with high precision, including the location, screenshots, and the corresponding HTML of ads for verification.

Second, our simulation of BVI user interactions is inspired by prior work demonstrating that screen reader navigation relies on analogous hover and focus events [45]. After performing these gestures on an ad element using Selenium [5], we must precisely identify the network request triggered by that gesture. This task is non-trivial because websites continuously issue hundreds of network requests to dynamically load content. Accurately isolating the one initiated by a specific gesture is challenging. Prior work demonstrates that dynamic tracing on real-world websites is prohibitively expensive and unreliable due to the event-driven, dynamic nature of JavaScript and HTML [70]. To address this, we design a call stack monitoring function that automatically overrides existing event listeners on ad elements, causing ads to invoke the injected function in the call stack whenever a network request results from a gesture event (*e.g.,* mouseover). Such instrumentation of event listeners, which appears in the call stack of a network request, enables accurate attribution of network requests to specific gestures.

Third, once a network request is attributed to a gesture, we must verify that the request causes a privacy violation by sending user information to third-party tracking entities. We leverage an existing verified database of known tracking entities *i.e.,* filter lists[7, 8], to verify if a request is made to a tracking entity. We further analyze the type of data shared.

Findings. We based our empirical investigation on websites selected from the top-3.5K news and media websites listed on SimilarWeb, a recognized website ranking source [18]. Our automated analysis identified a total of 67,137 ad elements. Upon simulating gestures representative of BVI user interactions with these ads, we detected 345,498 distinct tracking network requests specifically triggered by gestures required to operate screen readers. Across all tested websites, each ad element generated an average of 7.83 tracking requests. We further analyzed third-party tracking endpoints receiving these requests. Google and Amazon emerged as the most frequently observed tracking entities and ad service providers, respectively. These entities track users through request parameters such as cookie, cust_params, gpic, to monitor BVI users' browsing behaviors and web interactions. Overall, the empirical and anecdotal findings strongly indicate that merely accessing website content using screen readers forces BVI users to involuntarily compromise their privacy–an issue not faced by non-BVI users–highlighting significant equity concerns.

Solution and Effectiveness. Blocking all web ads is not generally considered a sustainable solution as they disrupt revenue models for free web services [58], often break websites or suppress non-ad functionality [67]. Adblockers are also increasingly undermined by anti-ad-blocking technologies (e.g., Admiral, AdShield) that detect and neutralize blocking mechanisms. Prior studies show that some users perceive ads as valuable when relevant or informative [54].

Parameter	Description
anId	Advertiser identifier.
asId	Session or unique identifier for the ad instance.
tv	JSON-encoded telemetry values containing event timestamps, interaction events, and engagement signals.
slEvents	tracks scroll latency events, including timestamps, viewport dimensions, and interaction classifications.
rmeas, rend	Flags confirming rendering and measurement completion.
ph, sis	Parameters related to page load times or ad viewability metrics.
hov	hover interactions such as duration, number of hover events, and additional metrics indicating hover intensity or event classification.

Table 1: Parameters in tracking requests triggered by *hover gestures.*

Because web developers ultimately control what is served on their sites, we propose an automated code-injection approach to give them finer-grained control after ads are rendered. After a website and its ads load, the approach dynamically detects ads as they load in the browser and then automatically overrides their HTML with an invisible overlay. This overlay intercepts and discards any gesture events, enabling BVI users to safely interact with ads using gestures necessary for activating screen readers without triggering any tracking behavior. We evaluate the efficacy of this approach by re-conducting the experiments with overlay injection enabled. On 67,137 ads from top-3.5K websites, the custom overlay injection reduced the tracking request from gesture to zero, which is not surprising, as it programmatically intercepts gestures. In other words, the approach is 100% effective in completely eliminating privacy-violating data sharing. We make our measurement framework and empirical results publicly available to web developers and internet monitoring agencies to use for evaluating the state of the web.
Data Availability: Our source code and data is available at https://zenodo.org/records/15025412.

2 Motivating Example

This section presents a case study showing the consequence of performing gestures (*e.g.,*mouseover, mousemove, hover, focus, etc.) needed to invoke screen readers on ads when navigating a website. For brevity, we will call such gestures *hover gestures* from hereon. CNN (cnn.com [6]) is ranked 3rd in the top 5K websites in the news and media category based on web traffic and audience engagement [30]. This website, when loaded in a Chrome browser, features multiple ad placements: a banner ad at the top, two in the bottom-right, and another in the middle-right, as shown in Figure 1(a) - ❶-❸. These ad slots are exclusively filled by a single ad service, Google Ads[9]. Note that CNN leases space on their website (*i.e.,* top banner space and the right margin space of the page) to Google Ads by inserting Google Ads-provided (a) JS code snippets in their website's Javascript and (b) HTML element with an ad-specific HTML class tag into HTML body.
Non-BVI Users Interacting with Ads. If a non-BVI user wants to locate the CNN Underscored section (Figure 1(a) - ❹) on cnn.com, they can gaze over the presented content of the website to locate it on the bottom right without interacting with any element (ad or non-ad). While there are numerous ads scattered throughout the website, non-BVI users can consciously avoid these ads by reading the desired content directly without triggering any hover gestures (*e.g.,* mouseover, focus) or physical interaction (*e.g.,* click).
BVI Users Interacting with Ads. If a BVI user wants to locate and later interact with the CNN Underscored section, they must rely on assistive technologies like screen readers.

We use Apple's VoiceOver screen reader on a MacBook [11] to demonstrate how a BVI user navigates to the CNN Underscored section. One way for them to reach this section is by using the Tab key, which sequentially selects one element at a time, in a Tab Order [45], reading the alternate text for the element in focus, and vocalizing it for the user. After a few tab presses, the user reaches the CNN Underscored section, but they must tab over three ads along the way.

Another alternative is to use the mouse cursor to locate the CNN Underscored section. This approach requires the user to perform multiple hover gestures over different page components to develop spatial awareness of the page content until they identify the CNN underscored section. Each hover gesture triggers the screen reader to announce the alternative text of the underlying element. Given that nearly half of the visible page area is occupied by ads, it is highly probable that users will inadvertently hover over one or more ad components until they reach the desired underscored content.

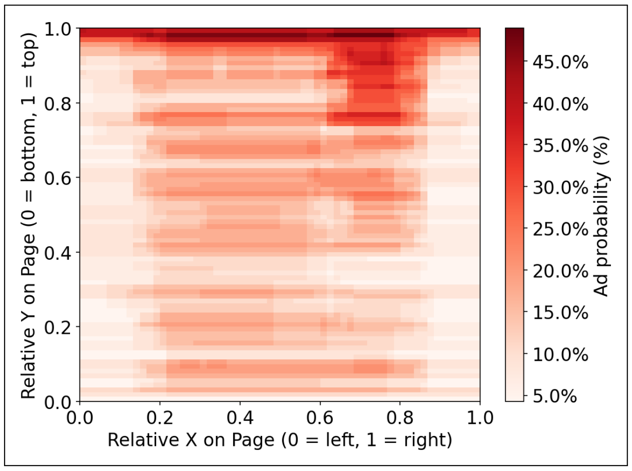
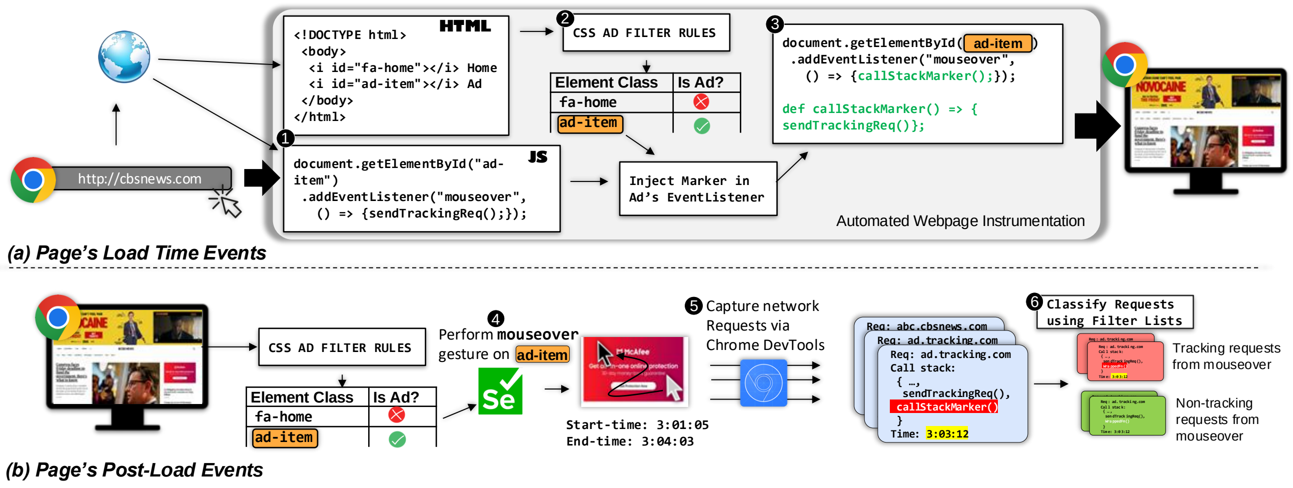


Figure 2: Heatmap of ad placement across the top 1K news media websites, showing dense clustering near the top and right edges of pages.

Privacy research has extensively analyzed network requests, parameters, and JavaScript behaviors on websites, showing how these elements may enable user tracking and violate user privacy [44, 52, 61, 65]. We leverage well-established tracking parameters curated by ad-blocking lists and privacy-focused browsers such as Brave to label parameters as tracking. We then investigate network requests and find that many query parameters in URLs contain user identifiers and additional metrics related to ad impressions and user–ad interactions, as shown in Table 1.

Upon monitoring the network traffic in Chrome, we find that mouseover gestures on the ad in Figure 1(b) send a network request to Integral Ad Science (IAS) [10] on domain dt.adsafeprotected.com, which is known to track user engagement.



```
<!DOCTYPE html>
 <body>
  <i id="fa-home"></i> Home
  <i id="ad-item"></i> Ad
 </body>
</html>
```

```
document.getElementById("ad-
item")
 .addEventListener("mouseover",
   () => {sendTrackingReq();});
```

Element Class	Is Ad?
fa-home	✗
ad-item	✓

```
document.getElementById(ad-item)
 .addEventListener("mouseover",
   () => {callStackMarker();});

def callStackMarker() => {
sendTrackingReq()};
```

```
Req: abc.cbsnews.com
Req: ad.tracking.com
Req: ad.tracking.com
Call stack:
 { …,
  sendTrackingReq(),
 callStackMarker()
 }
Time: 3:03:12
```

```
Start-time: 3:01:05
End-time: 3:04:03
```

HTML
JS
http://cbsnews.com
2 CSS AD FILTER RULES
1
3
Inject Marker in Ad's EventListener
Automated Webpage Instrumentation
(a) Page's Load Time Events
CSS AD FILTER RULES
4 Perform mouseover gesture on ad-item
5 Capture network Requests via Chrome DevTools
6 Classify Requests using Filter Lists
Tracking requests from mouseover
Non-tracking requests from mouseover
(b) Page's Post-Load Events

Figure 3: The end-to-end pipeline of our methodology encompasses crawling, detecting ads, injecting markers in event listeners, performing gestures, and capturing network request.

3 Methodology

We further investigate the network request and find that the query parameters in the URL contain user identifiers and additional metrics related to ad impression and user-ad interaction data, as shown in Table 1. The `asId` parameter tracks users by assigning a unique session or identifier to an ad instance, enabling persistent tracking across ad interactions. Whereas other parameters, such as `tv` and `slEvents`, collect telemetry data and behavioral fingerprints by capturing event timestamps, interaction events, scroll latency, and viewport dimensions. IAS is capturing a user behavioral profile based on hover interactions, indicating that the user engaged with this ad. This lets third-party entities know that the user visited CNN, which eventually helps them build the users' browsing and interests profile. These parameters are authoritatively established as privacy-violating parameters by prior work [63, 71]. Thus, for merely locating the CNN underscored section on the website, BVI users have no choice but to compromise their personal data and browsing history. In contrast, non-BVIs can do the same without any privacy violation.

Where do ads appear on the web and why they matter for BVI users? We find that ads are strategically positioned on the websites where it becomes difficult for BVI users navigating with a screen reader to avoid them. Because screen readers move sequentially across the accessibility tree, users must pass over ad regions to reach desired content. The heatmap in Figure 2, generated from measurements of the top 1K news and media websites, illustrates this placement. Large portions of the page canvas, especially near the top and right edges, are dominated by ads. As a result, BVI users are effectively forced to interact with ad elements through hover or focus gestures to reach the actual web content. Note that these ad interactions are not by choice but are a result of an unavoidable consequence of how ads are embedded on webpages.

Problem statement. Website developers struggle to anticipate the privacy risks that advertisements pose for BVI users. As shown in the `cnn.com` case study, these ads create an unequal web experience, disproportionately exposing BVI users to unintended data collection. This raises critical questions: *Is this issue prevalent across all website ads? Which ad-serving technologies are involved, and what data is being leaked? Most importantly, what solutions can website developers implement to mitigate these risks?*

4 Research Questions

Given the prevalence of ads on popular websites, their unintended privacy impact on BVI users, and the lack of awareness among developers, we aim to address the following research questions:

(1) **RQ1:** How prevalent are the privacy risks for BVI users due to tracking when gestures required to enable screen readers are performed on ads?
(2) **RQ2:** Which ad services actively track and collect data from hover gestures?
(3) **RQ3:** What specific types of data do ad services track when hover gestures are performed on ads to invoke screen readers?
(4) **RQ4:** How can website developers mitigate such unintended data collection from BVI users when building spatial awareness of web content via screen reader?

To address these research questions, we develop a fully automated end-to-end framework that simulates BVI user's gestures (*e.g.,* mouseover) on ad elements needed to trigger screen readers. We design this framework with Selenium, a custom-built Chrome extension, and Chrome DevTool to automate browsing behavior, detect advertisements, and perform gestures needed for screen reader usage. We build a custom Chrome extension that, when enabled, intercepts requests triggered by these gestures, analyzes the request to classify them as tracking or not, and extracts fine-grained request parameters to examine the nature of tracking. Figure 3 shows the overview of our methodology. It starts with instrumenting the website right before loading it in the browser (Figure 3-a) and then

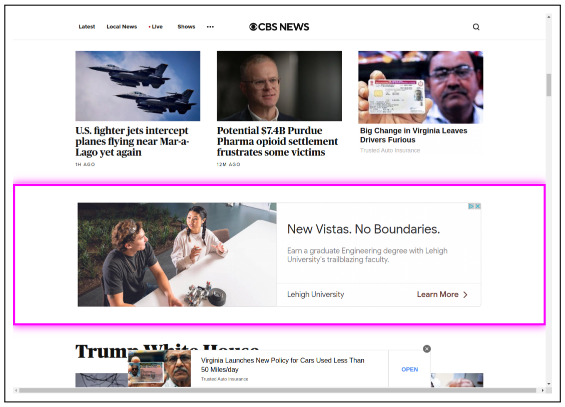

Latest
Local News
• Live
Shows
CBS NEWS
U.S. fighter jets intercept planes flying near Mar-a-Lago yet again
1H AGO
Potential $7.4B Purdue Pharma opioid settlement frustrates some victims
12M AGO
Big Change in Virginia Leaves Drivers Furious
Trusted Auto Insurance
New Vistas. No Boundaries.
Earn a graduate Engineering degree with Lehigh University's trailblazing faculty.
Lehigh University
Learn More
Virginia Launches New Policy for Cars Used Less Than 50 Miles/day
Trusted Auto Insurance
OPEN

Figure 4: The ad detected on cbsnews.com, highlighted with a purple border, indicates that our instrumentation is actively running to detect and interact with the ad.

simulating the BVI user gesture and monitoring the corresponding website events (Figure 3-b).

4.1 Website Crawling

We crawl the top-3.5K news and media websites from SimilarWeb [18] using the Chrome browser (version 132.0.6834.159) with Selenium (version 4.28.1) and our custom-built Chrome extension. Choosing news and media websites for ads-related analysis is standard practice[57, 74, 76], as such websites provide the most opportunity to interact with ads and are mainly unrestricted, free to access compared to websites offering services like `zoom.com` [38] or `gmail.com` [35]. Our crawl captures landing pages, limiting interaction to ten scrolls to prevent infinite scrolling on dynamically loading websites. Some websites detect Selenium as a bot, often requiring CAPTCHAs for human verification, which prevents us from interacting with them as we perform ethical web crawling with `robot.txt` enabled. These websites are excluded from our analysis; however, their numbers are reported in Section 5.

4.2 Ad Detection via CSS Filter Rules

To detect ads on websites during website load, we employ a CSS filter-based detection mechanism that leverages general hiding style rules [32], as shown in Figure 3-❷. These rules are commonly used by ad blockers to filter ads. Once the DOM `onload` event is triggered, we initiate a systematic scan of each element on the website, evaluating its attributes such as class names and IDs against predefined filter rules to detect potential ads. If an element matches a rule, it is detected as an ad. Filter list rules are industry standard to detect ad elements, offering nearly 0% false positives rate *i.e.,* if an HTML elements class name matches one of the rules in the list, it is guaranteed to be an ad. However, filter lists may suffer from completeness. Today, all popular ad blockers use this list, vouching for its accuracy. To prevent tracking pixels — tiny, invisible images used to track user activity — we ensure that the ad is visibly rendered by verifying that its width and height exceed a minimum threshold of 2 pixels. Such tracking pixels are different from ads, and both BVI and non-BVI are vulnerable to it. For each detected ad, we capture a screenshot and visually highlight it for verification. For example, Figure 4 shows an ad detected on CBS News using the filter rule

Attribute	Value
`Website URL`	www.cbsnews.com/
`ad_ID`	Ad_ID_1
`Screen_shot path`	/cbs/screenshots/Ad_ID_1
`CSS Detection rule`	id=ad-leader-plus-top
`mouse-over start_time`	2025-03-06T03:37:36.833Z
`mouse-over end_time`	2025-03-06T03:37:40.106Z

Table 2: Metadata collected from our methodology for ad interaction.

Attribute	Value
`URL`	cbsnews.com/ad-request?id=12345
`Time`	2025-03-10T14:30:15Z
`Headers`	{ "User-Agent": "Chrome/120.0"}
`Call Stack`	`{ "function_1": "sendTrackReq", "function_2": "WrappedFn" }`

Table 3: Metadata collected from our methodology for network request logs.

`.ad-leader-plus-top`, highlighted with a purple border in the captured screenshot. This approach ensures precise ad detection while maintaining a structured record of identified elements for further analysis.

4.3 Injecting JS Marker and Capture Network Requests

The dynamic nature of websites makes it challenging to associate a network request made by the website to a user event. It is common for a website to continuously load dynamic content and communicate with the first-party server, which leads to recurrent network traffic (network requests and responses) even when no actions are performed on the website. Thus, it is difficult to verify whether the gesture (e.g., `mouseover`) resulted in a request.

We use the Chrome DevTools API [4] to fetch network data, which is also accessible through the Chrome DevTools window. It uses fine-grained monitoring to capture all information related to each network request, including a snapshot of the function call stack of when the request was issued and the request's query parameters. In Chrome, users can view this data by right-clicking, selecting `Inspect`, and navigating to the `Network` tab. Clicking on a request reveals detailed information, including the call stack. Table 3 shows a sample call stack against a network request. We devise a method to inject a *marker* in the call stack of the network request issued precisely due to a hover gesture (*e.g.,* mouseover or focus), distinguishing itself from background network requests.

In addition to crawling, we developed a custom Chrome extension to override and intercept `mouseover, mousemove, hover, focus` events, as shown in Figure 3-❸. The extension systematically uses hooks on JavaScript `addEventListener` calls to detect and modify `mouseover` event handlers using the extension's content script. Specifically, it identifies all `mouseover` event listeners, checks if they are attached to ad elements, detected using CSS rule-based filtering, and wraps the original event handler inside a custom-defined identity function, `callStackMarker`. This approach ensures that the original functionality remains unchanged while allowing us to identify the precise chain of function calls, eventually leading to network requests, if any, whenever a hover

Data	Value
Total Sites Crawled	3,500
Sites with CAPTCHAs	208
Sites after Instrumentation	2,352
Total Ad Interactions	67,137
Total Requests	456,133
Total Tracking Requests	345,498
Total Non-Tracking Requests	110,635

Table 4: Overall statistics of web crawling and data collection.

gesture is performed. By doing so, the `callStackMarker` function appears in the callstack whenever a `mouseover` event occurs on an ad.

Simultaneously, we capture the complete network request data, including request parameters, timestamps, and the call stack containing functions involved in initiating the request, as shown in Figure 3-❺. If `callStackMarker` is present in the call stack, it indicates that the request was triggered by a `mouseover` event on an ad. Table 3 shows the data collected for each network request. Our approach does not interrupt any threads to capture this data, as it is natively exposed through the API.

4.4 Emulating Screen-Reader Gestures

During the page load time, our Chrome extension identifies all ad elements on the website as well as injects `callStackMarker` in the events listeners of ads elements only against all the gestures that BVI users use to access element content via screen reader.

Once this instrumentation is complete, the website load is complete. We then simulate BVI user gestures on the ad elements identified previously on the website, shown in Figure 3-(b). While doing so, we use the Chrome Extension to capture all network requests via Chrome DevTools [4], excluding the background noise *i.e.,* requests not issued by the simulated hover gestures. We use selenium to simulate user gesture by performing a `mouseover` action on the detected ad element, as shown in Figure 3-❹. The following code demonstrates the implementation in Selenium:

```
# Record start time of mouse-over action
mouseover_start_time = time.time()
# Perform mouse-over action on the ad element
# using Selenium Actions
actions.move_to_element(ad_element).perform()
# Record end time of mouse-over gesture
mouseover_end_time = time.time()
```

Listing 1: Selenium's Mouse-Over Action on Ad Element

We first record the timestamp, marking the start of the mouse-over action. Using Selenium's `move_to_element` function, we then simulate a mouse-over on the detected ad element. After completing the interaction, we capture the end timestamp. This will encompass all relevant hover gestures, including mouseover, mousemove, CSS-based hover, and focus (when the element is focused). Table 2 presents the data collected from the `mouseover` interaction.

How does our automation simulate BVI user behavior? BVI users typically navigate a page's accessibility tree through keyboard commands provided by screen readers, progressing linearly across focusable elements. Borodin et al. surveyed these strategies [45], showing that pressing Tab, Shift+Tab, or performing a mouse move shifts the virtual cursor sequentially between interactive items, while semantic shortcuts (e.g., `H` for headings, `L` for links)

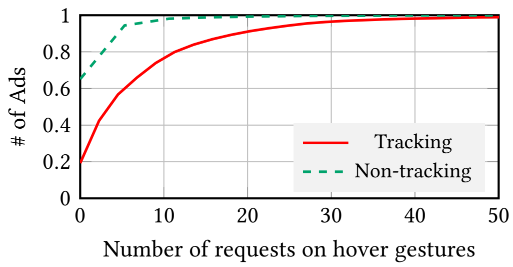


Figure 5: Cumulative distribution plot of ads based on the number of requests triggered by hover gesture interactions: tracking vs. non-tracking.

enable users to jump directly to structural landmarks in the content. They also note that these navigation commands trigger underlying DOM events, including focus and mouse-related interactions, as part of surfacing content. Our simulation is grounded in this prior work [45], programmatically emulating these mechanisms on landing pages, starting from the top and capturing the same event flow that occurs during real screen reader usage.

How do we filter network request data to link it with ad interactions? To filter network request data linked gestures performed on ads, we first examine only those requests whose call stacks include our `callStackMarker` function, as shown in Figure 3-❺. Second, to associate an interaction request with a specific ad, we further use the timestamp. This ensures that the request is triggered by user interaction, with the function confirming its origin and the timestamp mapping it back to the corresponding ad. Additionally, to prevent any timestamp overlaps, we do not run parallel crawls, ensuring accurate request-to-ad mapping.

4.5 Classifying Tracking Requests

To classify the captured network requests as tracking or non-tracking, we apply widely used filter lists, such as EasyList [7] and EasyPrivacy [8], as shown in Figure 3-❻. These lists are widely used by ad blockers to prevent network requests associated with ad tracking. They consist of regex-based rules that match request URLs against predefined patterns. If a request URL matches a rule, it is classified as tracking; otherwise, it is considered non-tracking. Notably, these lists are highly comprehensive and optimized to minimize false positives, ensuring minimal impact on website functionality. Thus, if a request is classified as tracking, it can be reliably considered as such. Overall, our evaluation framework takes in a list of websites and user gestures to perform on the website and outputs a detailed trace of the corresponding network requests, request parameters, and whether the requests were to tracking endpoints or not.

5 Results

This section presents the results of the empirical study. Table 4 summarizes our investigation's key statistics and parameters. We analyzed a total of the top 3.5K news and media websites, adhering to ethical crawling procedure by presenting `robot.txt` to the website when visited by our framework. As a result, approximately 6% of the websites disabled crawler access or displayed CAPTCHAs that could not be solved by the crawler. Among 3,288 successful visits, we were able to interact with and override event listeners on 2,352 websites, achieving a 71.54% success rate for running our

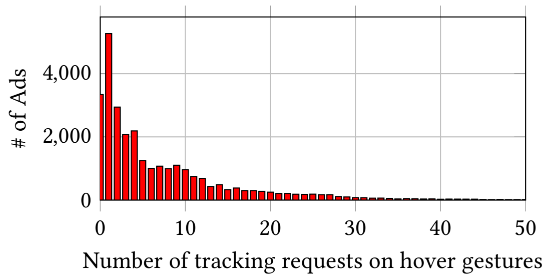


Figure 6: Distribution of ads by the number of tracking requests triggered via hover gesture interactions.

instrumentation, exceeding the success rate achieved by recent work on web content tracing and instrumentation. [76]. On these websites, we detect 67,137 ads and use our framework to perform the hover gestures that invoke screen readers. Note that this is a computationally intensive process, where performing all steps on a single website takes three minutes on average. In our analysis, we collect over 134 GB of ad-related data generated by the hover gestures. Below, we analyze the collected data to find empirical evidence to answer each research question.

5.1 RQ1: Prevalence of Privacy Risks for BVI Users from Screen Reader Gesture

In this research question, we first find the extent to which hover gestures triggered privacy violations per ad and website in our experiments. For each hover gesture simulated by our framework on an ad on a website, we inspect the corresponding requests collected in the trace.

Figure 5 presents the cumulative distribution plot, where the x-axis represents the number of requests triggered by hover gestures, and the y-axis shows the proportion of ads. The red line represents the tracking requests, whereas the green-dashed line represents the non-tracking requests. For instance, we find that 80% of the total ads have one or more tracking requests when hover gestures like mouseover are performed, sharing user data with a third-party tracking entity. In contrast, only 35% of the ads issued one or more non-tracking requests upon hover gestures.

The gap between the two lines clearly indicates that tracking requests outnumber non-tracking requests, meaning that ads frequently cause involuntary privacy violations when BVI users use gestures to invoke screen readers on web content. The comparison also demonstrates that if our methodology struggles to capture a certain proportion of requests, the tracking request still remains higher than non-tracking. Concretely, the average number of tracking requests triggered by gestures on ads is 7.83, whereas the average number of non-tracking requests is 2.11. Overall, 80.8% of ads have at least one tracking request triggered by gestures required for screen readers, as shown in Figure 6.

Additionally, we find the average number of tracking requests triggered by hover gestures on ads per website is 6.22, whereas the average number of non-tracking requests is 3.10. Figure 7 visualizes the results of this analysis. Even when ads are grouped per website, we find tracking requests from hover gestures to remain consistently higher than non-tracking requests, demonstrating ads' active involvement in privacy-violation data sharing with tracking

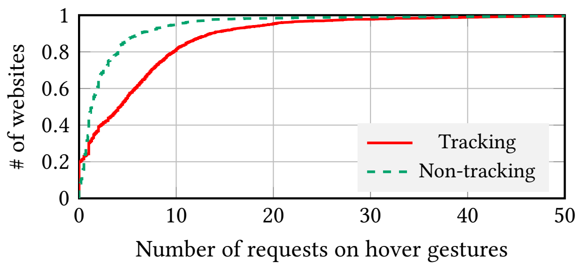


Figure 7: Cumulative distribution plot of ads on websites based on the number of requests triggered by hover gesture interactions: tracking vs. non-tracking.

endpoints. Across both per website and per ad, we observe a smaller number of non-tracking requests than tracking requests. Ads are third-party web content. A web developer leases space on their website to an ad service, which fills the leased space with ads when a user requests a website. These ads are tailored to the user's profile to maximize impressions and impact. Loading a website triggers a chain of ad bidding events, culminating in an ad being placed in the leased space. To measure ad effectiveness and refine user profiles, ad services continuously monitor user behavior on ads. Consequently, hover gestures are likely to generate requests to ad services, which inherently involve tracking and, thus, cause a higher proportion of tracking requests. Most non-tracking requests are sent directly to website servers and are not blocked by filter lists, as doing so might disrupt website functionality. Consequently, these requests occur in lower numbers. Web developers are completely unaware of the characteristics of the user-specific ads served on their website by ad services. As a result, BVI users relying on assistive technologies may be subject to unwanted data collection without their consent. Identifying the ad services involved in such tracking can help web developers understand the potential tracking on their website as a result of hover gestures.

RQ1 Finding

Our analysis shows that hover gestures required to enable screen readers frequently trigger tracking requests, with **80%** of ads containing at least one such request. This underscores the need for website developers to be aware that ad services may inadvertently track BVI users, raising privacy concerns.

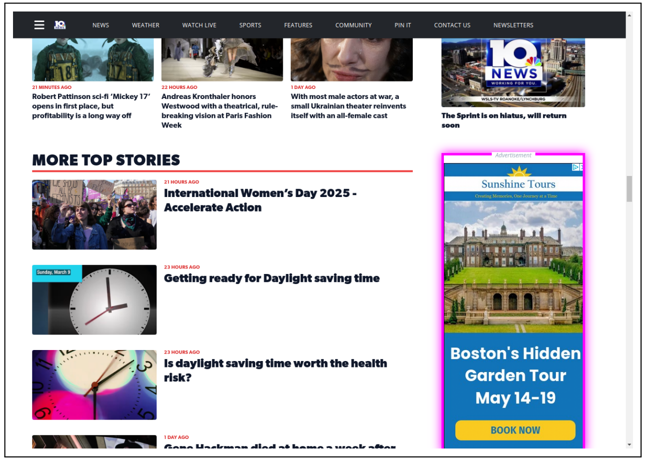
5.2 RQ2: Ad Services Involved in Tracking BVI User Data

We conduct a fine-grained analysis of tracking requests triggered by simulated hover gestures on ads to extract the destination domains, enabling the identification of the ad service. This analysis helps website developers identify these ad services and consider mitigation strategies, as discussed in Section 6, for the ad spaces they lease. Additionally, it may encourage ad services to refrain from triggering tracking requests when a screen reader is detected, ensuring an equitable web experience for BVI users comparable to non-BVI users who do not need to perform hover gestures to access web content. Table 5 presents the most prevalent ad-serving domains along with their respective owners.

securepubads.g.doubleclick.net is a Google-owned ad service primarily responsible for managing and delivering targeted ads across the web [31]. It facilitates real-time bidding, ad placement,

Domain	Owner	# of Ads	# of Websites
securepubads.g.doubleclick.net	Google	10,024	1,338
aax.amazon-adsystem.com	Amazon	9,935	1,013
pagead2.googlesyndication.com	Google	7,139	1,245
pb-ing-minutemedia.ccgateway.net	Minute Media	6,397	348
htlb.casalemedia.com	Casale Media	4,798	517
hbopenbid.pubmatic.com	PubMatic	4,732	582
prebid-server.rubiconproject.com	Rubicon Project	4,615	400
targeting.unrulymedia.com	Unruly	4,256	434
btlr.sharethrough.com	Sharethrough	3,652	335
tpc.googlesyndication.com	Google	3,606	811
tlx.3lift.com	TripleLift	3,600	353
dt.adsafeprotected.com	Integral Ad Science	3,342	469
prg.smartadserver.com	Smart AdServer	3,303	334
trc.taboola.com	Taboola	3,221	426
bidder.criteo.com	Criteo	3,145	314
ib.adnxs.com	Xandr	2,858	472
hb.undertone.com	Undertone	2,497	216
analytics.google.com	Google	2,394	545
protected-by.clarium.io	Clarium	2,159	256
sb.scorecardresearch.com	Comscore	1,585	309

Table 5: Top-20 most prevalent ad services, their owners, and the number of ads tracking user interactions via the hover gesture.



21 MINUTES AGO
Robert Pattinson sci-fi 'Mickey 17' opens in first place, but profitability is a long way off

22 HOURS AGO
Andreas Kronthaler honors Westwood with a theatrical, rule-breaking vision at Paris Fashion Week

1 DAY AGO
With most male actors at war, a small Ukrainian theater reinvents itself with an all-female cast

The Sprint is on hiatus, will return soon

MORE TOP STORIES

21 HOURS AGO
International Women's Day 2025 - Accelerate Action

23 HOURS AGO
Getting ready for Daylight saving time

23 HOURS AGO
Is daylight saving time worth the health risk?

1 DAY AGO

Figure 8: The ad (highlighted with a purple border) on the right side of `wsls.com` is detected, and a hover gesture is performed.

and user tracking to optimize ad delivery. This service plays a central role in ad distribution, leveraging cookies, device identifiers, and browsing behavior to serve personalized ads. In our analysis, `securepubads.g.doubleclick.net` was responsible for tracking requests associated with 10,024 ads across 56.8% of the websites.

For example, on `wsls.com` [37], we identify the ad-serving domain `securepubads.g.doubleclick.net` actively delivering ads. As shown in Figure 8, when a hover gesture is performed over one of these ads, a request is triggered to this domain, notifying Google's ad-tracking system of the user's engagement with the ad. This request contains a series of encrypted parameters that transmit details such as session data, security tokens, and user agent information, ensuring the event is logged accurately and securely. For example, session data is a common parameter in tracking requests, used to identify and persist user activity across visits, enabling ad services to track behavior and personalize content. The data collected through this process enables ad services to analyze ad engagement and refine ad placement strategies for better ad impact. However, this also raises privacy concerns, particularly for BVI users relying on screen readers, as their gestures to locate web content may be tracked without explicit consent.

aax.amazon-adsystem.com is an Amazon-owned ad service primarily responsible for delivering targeted advertisements and tracking user interactions across the web [26]. It facilitates programmatic ad bidding, personalized ad placements, and behavioral tracking to optimize ad performance for advertisers. This service leverages Amazon's extensive user data, including shopping behavior and browsing patterns, to serve relevant ads across partnered websites and applications. In our analysis, `aax.amazon-adsystem.com` was responsible for tracking requests associated with 9,935 ads across 43.0% of the websites.

For example, on `accuweather.com` [22], we identified the ad-serving domain `aax.amazon-adsystem.com` actively delivering ads. As shown in Figure 9, when a hover gesture is performed over one of these ads, a request is triggered to this domain, notifying Amazon's ad-tracking system of user's hover engagement with the ad. This URL is a bid request sent to Amazon's ad system, triggered specifically by a gesture on the ad. It contains parameters that specify details like the source website (Accuweather), publisher ID, ad slot characteristics (including size and media type), and display dimensions, as well as timing and version information. This gesture-triggered request ensures that user engagement is captured accurately, which can influence the real-time bidding process for ad delivery. Together, the captured information can be used to identify user browsing history. Non-BVI users do not need such hover gestures to locate, access, and interpret the web content and, thus, are less susceptible to such tracking.

RQ2 Finding

Ad services owned by entities like Google, Amazon, and PubMatic are actively tracking hover gestures required for screen readers. These ad-serving domains are responsible for a significant number of tracking requests across a large percentage of websites, raising privacy concerns for BVI users who may be unknowingly monitored for simply using screen readers.

5.3 RQ3: Types of Data Tracked by Ad Services from Screen Reader Interactions

For both the BVI users and the web developer, it is important to know what category and type of user data is collected from the website in response to hover gestures, as certain data sharing will require the web developer to solicit user consent depending on regional regulations like GDPR [34] and CCPA [29]. This analysis also helps in understanding which type of data ad services collect from hover gestures. With this knowledge, developers can take proactive steps, such as (1) implementing stricter content security policies, (2) working with privacy-focused ad services, or mitigating unintended tracking of BVI users, as discussed in Section 6 .

We examine the tracking requests triggered by hover gestures, extract parameters from the URLs, and compare them with known tracking parameters identified in prior research [66]. Extensive research has been done to build framework-based web graphs to classify such parameters, which are also adopted by ad blockers to sanitize URLs [28, 33, 66]. By leveraging these heuristics, we identify how these parameters can be used to track BVI users. Table

6 presents the most prevalent domains and their associated tracking URL parameters found in gesture-triggered requests, ranked by the number of websites on which they appear. Notably, these parameters are often combined within a single request to track users on the website. We explain a sample of parameters found on tacking requests triggered by hover gestures on ads.

adxs and adys parameters, observed in requests from 9,334 ads on 51.4% of websites, are used by `securepubads.g.doubleclick.net` to capture the ad's position relative to the viewport. This allows ad services to monitor which ads are in view, how long they remain visible, and whether users interact with them, enabling more precise ad targeting, impression tracking, and engagement measurement.

cookie, cust_params, and gpic are used by `securepubads.g.doubleclick.net` to facilitate user tracking by transmitting data from browser storage. `cookie` is observed in requests from 9,176 ads on 48.7% of websites. It provides a unique user identifier and timestamps to track individual sessions and behaviors. `cust_params` is found on request from 8,860 ads on 46.7% of websites, whereas `gpic` is found in request from 9,174 ads from 48.7% of websites. They include details such as browser type, keywords, and page context. This information helps refine targeting based on inferred user interests or demographics, with prior research [65] indicating that these parameters often leverage stored browser data for tracking purposes.

u parameter is used by `aax.amazon-adsystem.com` and observed in requests from 9,633 ads on 42.1% of websites. It acts as a unique user identifier to recognize and track users across sessions and page visits. This enables Amazon to maintain a persistent user profile and deliver relevant ads over time.

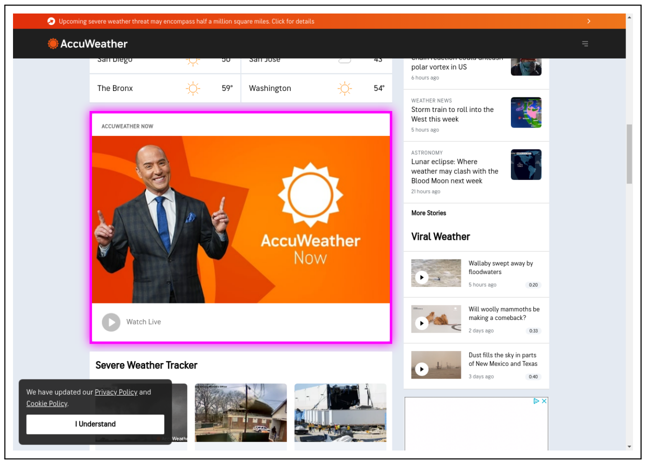
r parameter is used by `pagead2.googlesyndication.com` and observed in requests form 2,857 ads on 24.7% of websites. It serves as a referrer or request identifier, indicating the source of the ad request or the page context in which it occurred. By linking requests to their sources, Google can track user activity across multiple websites and enhance its targeting strategies.

RQ3 Finding

Tracking URL parameters, such as `cookie`, `cust_params`, `gpic`, `adxs`, and `adys`, are used by ad services to track BVI users' browsing behaviors and web gestures when using screen readers.

5.4 RQ4: Assisting Website Developers in Mitigating BVI User Tracking

In this section, we evaluate a potential solution that website developers can implement to prevent involuntary tracking for BVI users when using screen readers. One trivial solution is to use an ad blocker (*e.g.,* Adblock Plus [2] and uBlock [3]) to remove all ads from a website. While ad blockers are likely to be highly effective against such tracking as they block ads from loading on the website in the first place, they significantly disrupt the revenue streams of free websites [23]. As a result, an increasing number of websites are using anti-ad blockers (*e.g.,* Ad Shield [24] and Admiral [25]), rendering ad blockers useless. Second, blanket blocking of all ads creates additional equity problems as BVI users are deprived of the same web experience as non-BVI users.



Upcoming severe weather threat may encompass half a million square miles. Click for details
AccuWeather
The Bronx 59°
Washington 54°
ACCUWEATHER NOW
AccuWeather Now
Watch Live
Severe Weather Tracker
We have updated our Privacy Policy and Cookie Policy.
I Understand
Storm train to roll into the West this week
Lunar eclipse: Where weather may clash with the Blood Moon next week
More Stories
Viral Weather
Wallaby swept away by floodwaters
Will woolly mammoths be making a comeback?
Dust fills the sky in parts of New Mexico and Texas

Figure 9: The ad (highlighted with a purple border) in the middle of `accuweather.com` **is detected, with a hover gesture performed.**

```html
<!-- Overlay wrapping the Ad Container -->
<div class="ad-overlay" role="presentation" aria-hidden="
    true"
    style="position:relative;width:300px;height:250px;"
    ondblclick="enableAdInteraction(this)">
  <div class="blocking-layer"
      style="position:absolute;top:0;left:0;width:100%;
    height:100%;
            background:rgba(255,255,255,0);z-index
    :9999;pointer-events:auto;"
      onmouseover="event.stopImmediatePropagation();
    event.preventDefault();">
  </div>
  <div id="AD_300" style="position:relative;width:100%;
    height:100%;"
      onmouseover="callStackMarker(org_function())">
    <iframe src="https://example-ad.com" width="300"
    height="250"
          style="border:none;display:block;">
    </iframe>
  </div>
</div>
```

Listing 2: HTML overlay wrapping an ad to intercept interactions

We propose an **overlay-based** approach that prevents ads from sending tracking request on hover gestures required to use screen readers effectively without entirely blocking ads. This approach involves inserting an overlay, an invisible layer positioned on top of ad iframes, that intercept all gestures performed on an ad and stop them from executing the Javascript code tied to the ad responsible for sending the tracking requests. Thus, it enables the screen reader to still access the overlay and alternate text in child DOM elements but prevents the hover gesture-triggered event from reaching the ad's event listener. Since we do not own any of the top websites, for evaluation purposes, we demonstrate this solution using a Chrome extension that injects the overlay dynamically. However, website developers can implement this natively by detecting screen reader users and automatically applying an overlay on ads via appropriate implementation in their website's JS, effectively blocking tracking events while maintaining accessibility.

We utilize a Chrome extension content script `content.js` to dynamically inject an overlay on top of all ads displayed on a website.

Domain	Owner	Parameter	Description	Ads	Websites
securepubads.g.doubleclick.net	Google	idt	Unique tracking identifier	9334	1210
		prev_iu_szs	Previous ad unit sizes		
		dt	Document load or timing stamp		
		ucis	User click/session info		
		adys	Tracks ad vertical position		
		vrg	Visitor region or variation group		
		adks	Ad keywords or segments		
		msz	Measured ad slot size		
		adxs	Tracks ad horizontal position		
		psz	Ad size in pixels		
		url	Requesting page URL		
		scr_y	Vertical scroll position		
		correlator	Anti-caching correlator		
		pvsid	Page view session ID		
		iu_parts	Inventory unit segmentation		
		eid	Event or experiment identifier		
		dlt	Download/load time measurement		
		lmt	Limits ad frequency	9156	1169
		prev_scp	Previous ad scope tracking	9231	1162
		cookie	Associates session via cookie	9176	1147
		gpic	Tracks graphical asset parameters	9174	1146
		cust_params	Custom ad parameters	8860	1099
		sig	Digital signature for validation	5187	1051
		psts	Post statistics tracking	8686	1018
		a3p	Third-party ad integration marker	8006	954
		didk	Device identifier tracking	5448	653
aax.amazon-adsystem.com	Amazon	slots	Identifies ad slot positions	9633	991
		u	Unique user identifier		
		gdprl	GDPR compliance flag		
		pid	Publisher or product ID		
		pj	Ad campaign project ID	9602	984
		vm	Version or view mode info	9374	886
pagead2.googlesyndication.com	Google	xai	External ad interaction ID	2965	604
		sig	Request signature for integrity		
		v	Version parameter	2857	581
		adk	Ad campaign key		
		rst	Reset status for session tracking		
		r	Referrer or request identifier		
		rpt	Repetition or report count		
		mcvt	Conversion metric variable		

Table 6: Most prevalent tracking URL parameters, associated domains, parent companies, and the number of ads tracking user interactions via the hover gesture.

Listing 2 shows the example overlay called `ad-overlay` on an ad element `AD_300`. The overlay employs `event.stopImmediatePropagation()` and `event.preventDefault()`, ensuring that gestures required for screen reader navigation do not unintentionally trigger tracking mechanisms attached to the ad element underneath. However, to preserve intended gesture, we implement an functionality-friendly mechanism: if a BVI user double-clicks on the overlay, the ad becomes fully accessible. This design prevents disruption of genuine engagement—a double-click signals intentional ad interaction. By balancing privacy and accessibility, our approach mitigates tracking risks while preserving seamless ad engagement. The following code in Listing 2 demonstrates how our overlay works in real time.

We deploy this approach on all 67,137 ads across 2,352 websites. We then repeat the standard methodology of overriding gesture-related functions on ads, detecting ads, performing screen reader activation gestures, and measuring the tracking and non-tracking requests initiated solely due to performed hover gesture on ads. Surprisingly, we observe zero tracking requests issued by the ads with overlay across all websites. This translates into a 100% success in mitigating tracking from hover gestures on ads, demonstrating the potential of this approach to prevent unintended tracking of BVI users. This method provides website developers with a viable strategy–detecting screen readers and dynamically applying overlays–to enhance privacy while maintaining accessibility for BVI users.

RQ4 Finding

Our findings show that enabling an overlay on ads results in **100%** success in mitigating tracking from hover gestures on ads needed for screen readers.

6 Discussion and Takeaways

Based on the findings of this empirical study, we discuss three perspectives from which the ad interaction process in website and the website serving pipeline can be augmented by different stakeholders to address involuntary tracking of BVI users when using screen readers.

Website's Design Perspective. The overlay solution proposed earlier offers an effective remediation strategy that addresses the problem after loading the website. However, during web development, a developer can consciously make accessible and privacy-preserving design choices in their website design. Website developers can adopt an alternative solution by detecting when a screen reader is active and then proactively announcing the positions of ads on the page to enhance spatial awareness for BVI users. This detection can be achieved using browser accessibility APIs or by monitoring specific user agent behaviors that indicate the presence of assistive technologies. Once a screen reader is identified, developers can use ARIA live regions [27, 72, 73] to insert audible notifications just before an ad element, for example, an announcement stating "The next element is an advertisement. Press [Skip] to bypass it if you do not wish to interact." This approach empowers BVI users by giving them control over whether to engage with the ad content, thus preventing unintentional interactions that could lead to unwanted tracking.

Screen Reader's Design Perspective. Instead of interacting directly with the live website, an LLM-based approach can capture a screenshot of the website–similar to how we capture ad screenshots using Selenium in our methodology–while maintaining a precise mapping to its original interactive elements. In this solution, the screenshot serves as an intermediary representation that an LLM processes to generate spoken content describing the layout and details of the page. This "speak-over" functionality allows BVI users to receive audio feedback about the content, ensuring they are aware of the spatial arrangement of elements, including advertisements and navigation components, without triggering unintended interactions. Once the user indicates their intent, the system leverages the pre-established mapping to accurately translate the user's input into the corresponding event on the original website. This method minimizes privacy risks by avoiding direct interaction with potentially tracking elements and enhances accessibility by providing a controlled and descriptive user experience powered by advanced language models.

Ethical Perspective. In the ad ecosystem, every user interaction with an advertisement typically counts as an impression–data that ad services use to bill advertisers and determine ad revenue. However, for BVI users who depend on screen readers, their gestures (like mouseover or tab focus) are essential for accessing web content rather than deliberately engaging with the ad. This inadvertent triggering of ad impressions creates a misrepresentation in ad data, where impressions are recorded without the user interacting with or viewing the ad. Ethically, this raises a critical issue: advertisers are charged for interactions that do not represent meaningful engagement, and BVI users are unfairly caught in a system that misinterprets their necessary accessibility behaviors as deliberate ad interactions. Therefore, it becomes mandatory from an ethical standpoint to implement solutions, such as overlay mechanisms, that accurately distinguish between genuine user interactions and those triggered by assistive technologies.

7 Threat to validity

Internal validity. The study relies on automated tools (*e.g.,* Selenium) to simulate user gestures. However, if these simulated gestures differ from those used by screen readers, they may not accurately capture the full range of real BVI user interactions, leading to underestimating tracking and non-tracking events. To mitigate this, we confirmed that the three most popular screen readers employ either mouse-based interactions or keyboard tabbing for content reading. Injecting a marker to override event listeners (*e.g.,* via the `callStackMarker`) may fail to override all event listeners, particularly when code is dynamically loaded using `eval`. However, in such cases, we might miss the detection of a small portion of tracking requests triggered by the gestures.

External validity. We conducted our experiments using the Chrome browser with a custom Chrome extension. Tracking behaviors may vary across browsers because of unique built-in features; for example, Safari automatically blocks certain ads [17] and Mozilla Firefox includes an integrated ad blocker [12]. While these browser-specific characteristics could influence tracking outcomes, our focus on Chrome provides a consistent testing environment and a broadly applicable baseline due to the popularity of Chrome.

Construct validity. Our analysis simulates key hover user gestures (*e.g.,* mouseover and focus) to measure tracking behaviors triggered by screen readers. However, some tracking requests may be missed because websites can implement custom event handlers. Despite this, our approach establishes a robust lower bound on tracking, suggesting that the actual prevalence of screen reader-triggered tracking might be even higher. The lists [1] used to classify requests effectively identify known trackers; they might yield false negatives if specific tracking endpoints are not yet documented. Importantly, since these filters do not generate false positives, any request flagged as tracking is reliably classified.

8 Related Work

Underlying workflow of assistive technologies. Assistive technologies such as screen readers enable BVI users to interact with web content through input methods like the mouse, trackpad, and keyboard. For example, macOS's VoiceOver [20] supports both mouse and trackpad gestures for navigating and reading webpage content [21], while NVDA [13], a widely used Windows screen reader, similarly uses mouse movements for interaction. Orca [15], the Linux-based screen reader, also relies on mouse-based events to review website content [16]. Across these systems, mouse actions serve as a common mechanism for triggering reading functions and advancing through content. Borodin et al. [45] show that navigation commands such as Tab, Shift+Tab, and mouse moves shift the virtual cursor sequentially between interactive elements, while semantic shortcuts (e.g., H for headings, L for links) allow users to jump directly to structural landmarks. These navigation behaviors inherently trigger DOM events—including focus and mouse-related events—as part of surfacing content to the user. Grounded in this prior work [45], our simulation programmatically emulates these mechanisms on landing pages, reproducing the same event flow that occurs during real screen-reader usage. Rather than relying on a single screen reader in our methodology, we emulate the underlying interaction model common across assistive technologies, consistent with prior findings that screen readers internally generate mouse and focus events to surface interactive elements.

Challenges in web ad navigation for BVI-users. Prior work [60] includes BVI-focused user studies showing that visually impaired users can be misled by ads or content that blends into the surrounding page during screen-reader navigation. These studies also highlight that many websites are designed in ways that impose limitations on BVI users when using screen readers, potentially leading to privacy issues [48, 59]. Clarke et al. [48] highlight the challenges faced by visually impaired users, showing that cookie notices are often inaccessible, making them invisible, unreadable, or inaudible for screen reader users. Khan et al.[59] investigate phishing threats in the email browsing experience of visually impaired individuals, highlighting how the reliance on visual cues for phishing detection creates security challenges for this user group. Because these behavioral insights are well-established, we focus instead on the technical mechanisms that create such risks.

Privacy issues in assistive technologies. Prior work [49, 56, 64] has shown that assistive technology and privacy issues are closely intertwined, as these systems often rely on user data to enhance functionality while introducing potential risks. Crawford et al.[49] analyze the privacy policies of assistive technologies, revealing significant challenges in how data collection and processing are communicated to users, including inconsistencies, a lack of transparency, and insufficient protections for individuals with disabilities. Similarly, Hamidi et al. [56] examine the privacy tradeoffs in adaptive assistive technologies, emphasizing that while these systems enhance usability by collecting user data, they also introduce privacy risks. Both studies highlight the need to balance functionality and privacy, focusing on these challenges from an assistive technology design perspective.

Beyond assistive technologies, prior research has extensively studied ads on websites and mobile applications in the context of privacy-invasive targeted advertising mechanisms (*e.g.,* [42, 50, 51]), malicious ads spreading malware or facilitating click fraud and phishing attacks (*e.g.,* [62, 68]), and deceptive or manipulative content (*e.g.,* [76]). Moreover, Amjad et al. [39] manually interacted with a selection of web ads to demonstrate that tracking occurs. However, their analysis is heavily manual and lacks the background noise present in real-world network traffic. We address this limitation by instrumenting JavaScript event listeners to capture a more comprehensive view of user interactions and tracking behaviors.

9 Conclusion

BVI users rely on assistive technologies to access web content. Yet, the very gestures required to use an assistive technology, such as mouseover and keyboard focus, unexpectedly activate ad tracking mechanisms, exposing them to significant privacy risks. This means that each gesture intended to build spatial and contextual awareness actually initiates tracking requests, exposing personal data to third-party entities. Such a privacy breach is not typically encountered by non-BVI users. Our large-scale empirical study across top websites revealed that each ad element generates, on average, 7.83 tracking requests, with major ad serving platforms like Google and Amazon frequently serving such ads. To counteract this issue, we developed an overlay injection approach that intercepts and discards gesture events before they can trigger tracking. This solution proved to be 100% effective in eliminating privacy-violating data sharing. In light of these findings, our work underscores not only the critical privacy challenges for BVI users but also the urgent need for greater developer accountability and enhanced accessibility measures that safeguard their online experience.

References

[1] [n. d.]. EasyList Forum. https://forums.lanik.us. https://forums.lanik.us
[2] 2016. Adblock Plus, Chrome web store. https://chrome.google.com/webstore/detail/adblock-plus/cfhdojbkjhnklbpkdaibdccddilifddb. https://chrome.google.com/webstore/detail/adblock-plus/cfhdojbkjhnklbpkdaibdccddilifddb
[3] 2020. gorhill/uBlock: uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. https://github.com/gorhill/uBlock.
[4] 2021. Extending DevTools. https://developer.chrome.com/docs/extensions/mv3/devtools/
[5] 2021. Selenium. http://docs.seleniumhq.org/. http://docs.seleniumhq.org/
[6] 2024. CNN. https://www.cnn.com/.
[7] 2024. easylist. https://github.com/easylist/easylist.
[8] 2024. easyprivacy. https://github.com/easylist/easylist/tree/master/easyprivacy.
[9] 2024. Google-ads. https://ads.google.com/home/.
[10] 2024. IAS. https://integralads.com/insider/marketers-stop-relying-hover-rate/.
[11] 2024. Macbook-voicerover. https://support.apple.com/guide/voiceover/welcome/mac.
[12] 2024. mozila. https://www.mozilla.org/en-US/firefox/features/adblocker/?utm_source=chatgpt.com.
[13] 2024. Nvda. https://www.nvaccess.org/download/.
[14] 2024. NVDA-mouse. https://download.nvaccess.org/documentation/userGuide.html#NavigatingWithTheMouse.
[15] 2024. orca. https://help.gnome.org/users/orca/stable/index.html.en.
[16] 2024. orca-mouse. https://help.gnome.org/users/orca/stable/howto_mouse_review.html.en.
[17] 2024. Safari-ITP. https://clearcode.cc/blog/intelligent-tracking-prevention/.
[18] 2024. Similar-web. https://www.similarweb.com/top-websites/news-and-media/.
[19] 2024. Taboola. https://www.taboola.com/.
[20] 2024. voicover. https://support.apple.com/guide/voiceover/welcome/10/mac.
[21] 2024. voicover-mouse. https://support.apple.com/guide/voiceover/use-the-trackpad-gestures-rotor-vo28035/10/mac/15.0.
[22] 2025. accuweather. https://www.accuweather.com/.
[23] 2025. ad-block-problem. https://www.admonsters.com/ad-blocking-a-54b-problem-for-publishers-in-2024/.
[24] 2025. ad-shield. https://www.ad-shield.io/.
[25] 2025. admiral. https://www.getadmiral.com/.
[26] 2025. amazon-adsystem. https://aax-us-east.amazon-adsystem.com/.
[27] 2025. aria-live-region. https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Guides/Live_regions.
[28] 2025. brave-param. https://web.archive.org/web/20240207231358/https://raw.githubusercontent.com/brave/brave-core/master/components/query_filter/utils.cc.
[29] 2025. CCPA. https://oag.ca.gov/privacy/ccpa.
[30] 2025. CNN-rank. https://www.similarweb.com/website/cnn.com/#overview.
[31] 2025. doubleclick. https://m.doubleclick.net/.
[32] 2025. Easylist-general-hide. https://github.com/easylist/easylist/blob/master/easylist/easylist_general_hide.txt.
[33] 2025. firefox-param. https://web.archive.org/web/20230714022136/https://firefox.settings.services.mozilla.com/v1/buckets/main/collections/query-stripping/records.
[34] 2025. GDPR. https://gdpr-info.eu/.
[35] 2025. gmail. https://www.gmail.com/.
[36] 2025. Prop-ads. https://shorturl.at/ga3BK.
[37] 2025. wsls. https://www.wsls.com/.
[38] 2025. zoom. https://www.zoom.com/.
[39] Abdul Haddi Amjad, Muhammad Danish, Bless Jah, and Muhammad Ali Gulzar. 2024. Accessibility Issues in Ad-Driven Web Applications. *arXiv preprint arXiv:2409.18590* (2024).
[40] AppleVis Forum Users. 2017. VoiceOver having trouble navigating ads in Safari on iOS 11.2. https://www.applevis.com/forum/ios-ipados/voice-over-having-trouble-navigating-ads-safari-ios-11-2. https://www.applevis.com/forum/ios-ipados/voice-over-having-trouble-navigating-ads-safari-ios-11-2 Accessed: 2025-05-30.
[41] AppleVis Forum Users. 2017. VoiceOver having trouble navigating ads in Safari on iOS 11.2. https://www.applevis.com/forum/ios-ipados/voice-over-having-trouble-navigating-ads-safari-ios-11-2. https://www.applevis.com/forum/ios-

ipados/voice-over-having-trouble-navigating-ads-safari-ios-11-2 Forum discussion on VoiceOver interaction issues with ads in Safari. Accessed: 2025-05-30.

[42] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. 2016. Tracing information flows between ad exchanges using retargeted ads. In *25th USENIX Security Symposium (USENIX Security 16)*. 481–496.

[43] Nataliia Bielova. 2017. Web tracking technologies and protection mechanisms. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2607–2609.

[44] Sophie C Boerman and Edith G Smit. 2023. Advertising and privacy: An overview of past research and a research agenda. *International Journal of Advertising* 42, 1 (2023), 60–68.

[45] Yevgen Borodin, Jeffrey P Bigham, Glenn Dausch, and IV Ramakrishnan. 2010. More than meets the eye: a survey of screen-reader browsing strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*. 1–10.

[46] Carl Brown. 1992. Assistive technology computers and persons with disabilities. *Commun. ACM* 35, 5 (1992), 36–45.

[47] Tomasz Bujlow, Valentín Carela-Español, Josep Sole-Pareta, and Pere Barlet-Ros. 2017. A survey on web tracking: Mechanisms, implications, and defenses. *Proc. IEEE* 105, 8 (2017), 1476–1510.

[48] James M Clarke, Maryam Mehrnezhad, and Ehsan Toreini. 2024. Invisible, unreadable, and inaudible cookie notices: An evaluation of cookie notices for users with visual impairments. *ACM transactions on accessible computing* 17, 1 (2024), 1–39.

[49] Kirk Crawford, Yi Xuan Khoo, Asha Kumar, Helena Mentis, and Foad Hamidi. 2024. Decoding the Privacy Policies of Assistive Technologies. In *Proceedings of the 21st International Web for All Conference*. 87–95.

[50] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 1388–1401.

[51] Motahhare Eslami, Sneha R Krishna Kumaran, Christian Sandvig, and Karrie Karahalios. 2018. Communicating algorithmic process in online behavioral advertising. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–13.

[52] José Estrada-Jiménez, Javier Parra-Arnau, Ana Rodríguez-Hoyos, and Jordi Forné. 2017. Online advertising: Analysis of privacy threats and protection approaches. *Computer Communications* 100 (2017), 32–51.

[53] Gian M Fulgoni and M Morn. 2008. How online advertising works: Whither the click. *comScore. com Whitepaper* (2008).

[54] Avi Goldfarb and Catherine E. Tucker. 2009. Online Display Advertising: Targeting and Obtrusiveness. *Journal of Economic Perspectives* 23, 3 (2009), 37–60. doi:10.1257/jep.23.3.37

[55] Michael Andrew Gresty, John Foster Golding, Huy Le, and Kelly Nightingale. 2008. Cognitive impairment by spatial disorientation. *Aviation, space, and environmental medicine* 79, 2 (2008), 105–111.

[56] Foad Hamidi, Kellie Poneres, Aaron Massey, and Amy Hurst. 2018. Who Should Have Access to my Pointing Data? Privacy Tradeoffs of Adaptive Assistive Technologies. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (Galway, Ireland) *(ASSETS '18)*. Association for Computing Machinery, New York, NY, USA, 203–216. doi:10.1145/3234695.3239331

[57] Ziyao He, Syed Fatiul Huq, and Sam Malek. 2024. " I tend to view ads almost like a pestilence": On the Accessibility Implications of Mobile Ads for Blind Users. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13.

[58] Elina Ikuinen and Minna Ruckenstein. 2021. Adblockers and the Imaginary of the Free and Open Internet. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM. doi:10.1145/3411764.3445459

[59] Emaan Bilal Khan, Emaan Atique, and Mobin Javed. 2024. Investigating Phishing Threats in the Email Browsing Experience of Visually Impaired Individuals. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–11.

[60] Satwik Ram Kodandaram, Mohan Sunkara, Sampath Jayarathna, and Vikas Ashok. 2023. Detecting Deceptive Dark-Pattern Web Advertisements for Blind Screen-Reader Users. *Journal of Imaging* 9, 11 (2023), 239. doi:10.3390/jimaging9110239

[61] Aleksandra Korolova. 2010. Privacy violations using microtargeted ads: A case study. In *2010 IEEE international conference on data mining workshops*. IEEE, 474–482.

[62] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 674–686.

[63] Francois Marier. 2025. Query String Filter. https://github.com/brave/brave-browser/wiki/Query-String-Filter. Accessed: 2025-09-11.

[64] Z Müftüoğlu, MA Kızrak, and T Yıldırım. 2021. Privacy-preserving mechanisms with explainability in assistive AI technologies. In *Advances in Assistive Technologies: Selected Papers in Honour of Professor Nikolaos G. Bourbakis–Vol. 3*. Springer, 287–309.

[65] Shaoor Munir, Patrick Lee, Umar Iqbal, Zubair Shafiq, and Sandra Siby. 2024. PURL: Safe and Effective Sanitization of Link Decoration. In *33rd USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, 4103–4120. https://www.usenix.org/conference/usenixsecurity24/presentation/munir

[66] Shaoor Munir, Patrick Lee, Umar Iqbal, Zubair Shafiq, and Sandra Siby. 2024. {PURL}: Safe and Effective Sanitization of Link Decoration. In *33rd USENIX Security Symposium (USENIX Security 24)*. 4103–4120.

[67] Sam Nisenoff, Oleksii Starov, and Alexandros Kapravelos. 2023. The Web is Broken: Studying the Effect of Third-Party Content on Website Breakage. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, 3887–3904. https://www.usenix.org/conference/usenixsecurity23/presentation/nisenoff

[68] Vaibhav Rastogi, Rui Shao, Yan Chen, Xiang Pan, Shihong Zou, and Ryan D Riley. 2016. Are these Ads Safe: Detecting Hidden Attacks through the Mobile App-Web Interfaces.. In *NDSS*.

[69] Reddit user u/No-Telephone3975. 2024. Help! Advertising is taking over VoiceOver! https://www.reddit.com/r/Blind/comments/1ebzpuv/help_advertising_is_taking_over_voiceover/. https://www.reddit.com/r/Blind/comments/1ebzpuv/help_advertising_is_taking_over_voiceover/ Reddit discussion in r/Blind on accessibility issues due to intrusive advertising. Accessed: 2025-05-30.

[70] Gregor Richards, Sylvain Lebresne, Brian Burg, and Jan Vitek. 2010. An analysis of the dynamic behavior of JavaScript programs. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation*. 1–12.

[71] Brave Privacy Team. 2020. Grab bag: query stripping, referrer policy, and reporting API. https://brave.com/privacy-updates/5-grab-bag/. Accessed: 2025-09-11.

[72] Peter Thiessen. 2011. WAI-ARIA live regions and HTML5. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility*. 1–4.

[73] Peter Thiessen and Charles L Chen. 2009. ARIA Live Regions: An introduction to channels. *Journal of Access Services* 6, 1-2 (2009), 215–230.

[74] David Thompson and Birgit Wassmuth. 2001. Accessibility of online advertising: a content analysis of alternative text for banner ad images in online newspapers. *Disability Studies Quarterly* 21, 2 (2001).

[75] P Turriziani, GA Carlesimo, R Perri, F Tomaiuolo, and C Caltagirone. 2003. Loss of spatial learning in a patient with topographical disorientation in new environments. *Journal of Neurology, Neurosurgery & Psychiatry* 74, 1 (2003), 61–69.

[76] Eric Zeng, Tadayoshi Kohno, and Franziska Roesner. 2020. Bad news: Clickbait and deceptive ads on news and misinformation websites. In *Workshop on Technology and Consumer Protection*. 1–11.