Permission Rationales in the Web Ecosystem: An Exploration of Rationale Text and Design Patterns

Yusra Elbitar*
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
Saarland University
Saarbruecken, Germany
yusra.elbitar@cispa.de

Soheil Khodayari*
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
Saarland University
Saarbruecken, Germany
soheil.khodayari@cispa.de

Marian Harbach
Google
Munich, Germany
mharbach@google.com

Gianluca De Stefano
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
gianluca.de-stefano@cispa.de

Balazs Csaba Engedy
Google
Munich, Germany
engedy@google.com

Giancarlo Pellegrino
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
pellegrino@cispa.de

Sven Bugiel
CISPA Helmholtz Center for Information Security
Saarbruecken, Germany
bugiel@cispa.de

Abstract

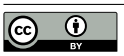
Modern web applications use features like camera and geolocation for personalized experiences, requiring user permission via browser prompts. To explain these requests, applications provide rationales—contextual information on why permissions are needed. Despite their importance, little is known about how often rationales appear on the web or their influence on user decisions.

This paper presents the first large-scale study of how the web ecosystem handles permission rationales, covering three areas: (i) identifying webpages that use permissions, (ii) detecting and classifying permission rationales, and (iii) analyzing their attributes to understand their impact on user decisions. We examined over 770K webpages from Chrome telemetry, finding 3.6K unique rationale texts and 749 rationale UIs across 85K pages. We extracted key rationale attributes and assessed their effect on user behavior by cross-referencing them with Chrome telemetry data. Our findings reveal nine key insights, providing the first evidence of how different rationales affect user decisions.

CCS Concepts

• **Information systems** → *Web applications*; • **Security and privacy** → **Usability in security and privacy**; • **Human-centered computing** → *Empirical studies in interaction design.*

*Both authors contributed equally to this research.



CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713547

Keywords

Web Measurement, Exploratory Analysis, Permissions, Rationales

ACM Reference Format:
Yusra Elbitar, Soheil Khodayari, Marian Harbach, Gianluca De Stefano, Balazs Csaba Engedy, Giancarlo Pellegrino, and Sven Bugiel. 2025. Permission Rationales in the Web Ecosystem: An Exploration of Rationale Text and Design Patterns. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 25 pages. https://doi.org/10.1145/3706598.3713547

1 Introduction

Modern web applications are becoming increasingly feature-rich and interactive, offering users a more dynamic and personalized online experience by utilizing device resources like cameras, microphones, push notifications, and geolocation data. However, to harness these capabilities, websites must often first ask users for permission through browser permissions prompts. While these permissions are critical to enable key features safely, they also introduce a significant burden for users. When confronted with permission prompts, users must make informed decisions about which capability accesses to allow and which to deny. Deciding wrongly can have negative security and privacy consequences, depending on the capability in question. In addition, prior work has shown that websites often ask for permissions in inopportune moments, making these requests annoying and lacking context [23].

Permission prompts on mobile platforms, particularly on Android, have been a frequent focus of security and privacy research studies over the years [2, 4, 8, 9, 11, 14, 16, 19, 20, 25, 31–38, 43–45, 54, 59–63, 66, 68, 71–73]. Unlike mobile apps, which are distributed through app stores with strict guidelines, websites are delivered dynamically via web browsers. Mobile app stores also

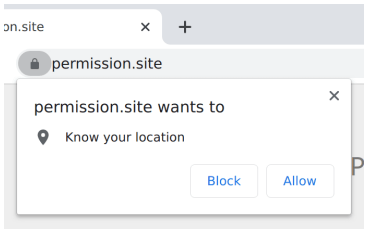


Figure 1: Example of a geolocation permission prompt in Chrome.

impose specific user experience (UX) requirements and guidelines for permission requests [5, 22], the web operates with significantly less centralized oversight. Web developers can trigger permission prompts at any point, even right after the page finished loading, and without following best practices [26].

The research community has only recently begun exploring *web* permissions, primarily focusing on user experiences with permission prompts [23], in particular for push notifications [7, 24]. Previous studies have examined user perceptions of annoyance or interruption, the ease or difficulty of decision-making, and the presence of contextual information on desktop platforms. However, the way browsers display these prompts is just the tip of the iceberg. The broader context, including how websites present permission requests and the rationales provided to users before and after prompts, remains largely unexplored in terms of both scope and impact.

Permission rationales on the web are explanations added to webpages to clarify why certain capabilities are required, providing essential context for permission requests. Research has consistently shown that offering contextual information significantly impacts user interactions with permission prompts [14, 16, 23, 61, 63]. In the Android ecosystem, studies have highlighted not only the importance of rationales but also the diverse ways contextual information is presented, demonstrating their critical role in shaping user decisions [14, 16, 61, 63]. However, despite the acknowledged importance of rationales in shaping user responses to web permission prompts [23], we still lack detailed information on the variety of web rationale texts and designs, methods to automatically trigger and detect them, the prevalence of websites using rationales, and the effects of different rationale patterns and design choices on user decision-making regarding permission prompts.

In this paper, we conduct the first systematic and comprehensive study of web permission rationales on desktop platforms, a largely overlooked aspect of the web permission ecosystem. Our research focuses on (i) systematically exploring and collecting webpages that feature permission prompts, (ii) automatically detecting and classifying rationales, and (iii) thoroughly analyzing various text and UI attributes of these rationales to begin understanding their impact on user actions and sentiment toward permission prompts.

Starting with 770K URLs from Chrome telemetry, we performed automated, interactive web crawling. We collected snapshots of webpages that request the most common permissions-gated web APIs, i.e., notifications, geolocation, camera, and microphone. We considered both screenshots for rationale UIs and the DOM of the page [69] for rationale text. As a result, our crawler successfully captured snapshots for 739K reachable URLs and triggered permission prompts on over 20% of the visited webpages.

We detected and manually confirmed 3.6K unique text rationales using a robust machine-learning pipeline. In addition, we semi-automatically compiled a dataset of 749 distinct rationale UIs. We observed that 85K webpages in the wild use one of the 3.6K unique rationale instances. We found that the most prevalent rationales belong to 10 libraries. Then, we undertook a qualitative analysis to characterize rationales, considering various aspects, including message tone, encouragement, message content, functionality necessity for text and layout, position, elements, and timing for UI.

After analyzing rationales to extract their attributes, we conducted an exploratory analysis to study how these elements impact users' decisions to grant, deny, dismiss, or ignore permission prompts, again using Chrome telemetry and user sentiment data. We applied regression models to extract 10 key effects. Among others, we find that any rationale message, regardless of tone, significantly boosts grant rates and reduces dismiss and deny rates, with positive tones increasing grant rates by 18%. Additionally, we find that UI design elements can have an even higher impact. For example, overlays before or alongside a prompt had the most substantial impact on grant rates (+41%), followed by fullscreen rationales (+33%). When it comes to user sentiment, dialogs and text rationales were associated with increased user annoyance, particularly when shown before and after browser prompts.

Contributions. We make the following main contributions:

- We create the first (semi-)automated approach to detect and study web permission rationales at scale. We instantiate our approach on a set of 770K webpages, processing over 6M unique text snippets. As a result, we create a comprehensive dataset of 3.6K manually-vetted and unique rationale text and 749 rationale UIs on the desktop web.
- We estimate the prevalence of web permission rationales, focusing on push notifications, geolocation, camera and microphone permissions, identifying ~85K webpages that use a custom or library-provided rationale. We find 10 libraries that have the most prevalent rationales–with the top three being OneSignal [46], iZooto [30] and Smart Push [29]– and create 32 code signatures to detect their use on webpages.
- We conduct a qualitative analysis of permission rationales, examining both text and UI. For the rationale text, we extracted attributes across four dimensions: message tone, encouragement, content, and functional necessity. For the rationale UIs, we identified attributes spanning three dimensions including layout elements, position and timing. We used these attributes to characterize web rationales, identifying 18 common rationale text patterns and 8 common UI patterns.
- We study the impact of rationale attributes on user behavior and sentiment towards permission prompts, cross-referencing webpages with coded rationales against Chrome telemetry and user sentiment data, and extract nine key insights.

2 Background

Modern web applications rely on permission-protected APIs, such as the `MediaDevices` API for accessing the camera and microphone [64], the `Geolocation` API for location data [41], and the `Notifications` and `Push` APIs for notifications [65, 70]. These requests trigger permission prompts within the browser. Users

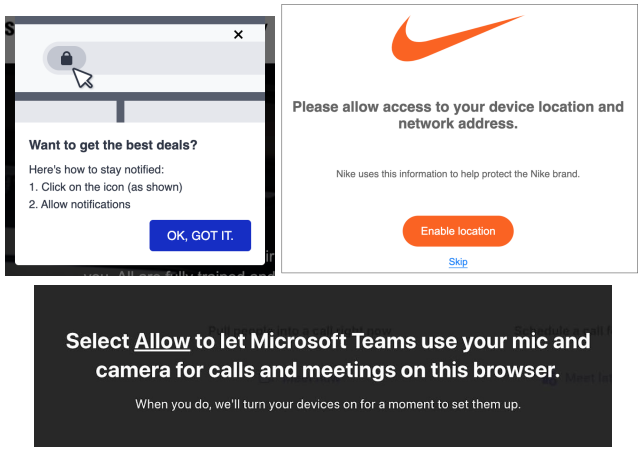


Figure 2: Examples of permission rationales in real websites. *Left:* notification rationale in samsung.com. *Right:* geolocation rationale in nike.com. *Bottom:* camera and microphone rationale in teams.microsoft.com.

have the option to *grant*, *deny*, or *ignore* these permission prompts, which typically appear near the browser's address bar and are often non-modal, meaning they do not block further interaction with the website. Many popular browsers also offer a temporary block option: in Chrome and Edge, this is done by clicking the *"X"* to dismiss the prompt, while in Firefox, users can click *"Block"* without selecting *"Remember this decision."* In this work, we refer to this action as *dismiss*, consistent with terminology used in prior research [23]. An example of these options is shown in Figure 1.

Websites can offer rationales to explain why permissions are needed, as shown in Figure 2. These rationales may appear as UI elements before the prompt, alongside it, or after the user has ignored, dismissed, or denied the request (i.e., not granted the request). When designed effectively, these explanations can build user trust and enhance decision-making. However, not all websites that request permissions provide rationales. In this work, we explore the current state of web permission rationales, including their types, components, and UI design choices.

3 Related Work

We present the previous research related to our study, focusing on both mobile (§3.1) and web (§3.2) permissions.

3.1 Mobile Permissions

Research on permissions has primarily centered on mobile applications, particularly within the Android ecosystem. Earlier research has shown that users face difficulties in understanding requested permissions, their purposes, and the potential risks associated with granting them [19, 20, 31, 33]. This often resulted in users' confusion and unmet expectations [8, 9, 11, 20, 31, 59, 71].

To address this, previous research has explored predicting users' responses to permission requests based on privacy profiles [32, 34, 35, 45, 72], or using privacy nudges to encourage informed choices [4, 73]. Additionally, fine-grained permission managers have been proposed to give users greater control over their data [62]. Moreover, previous research suggested that permission requests should be tied directly to user actions within the app, as users are more likely to understand and accept these requests when they occur in response to something they have done [38].

3.1.1 Contextualizing Permission Requests. Researchers have shown that the timing and context of permission requests significantly influence users' decisions. Studies highlight the importance of contextualizing permission requests, where developers decide when to prompt users [11, 16, 61, 63]. For instance, prior research [63] found that the context in which data is shared—specifically when, why, and with whom—significantly influences users' access decisions. Similarly, an online study [16] demonstrated that timing and rationales significantly impact permission decisions.

3.1.2 Mobile Rationales. Additionally, apart from the timing and context of permission requests, offering well-defined rationales is essential for helping users understand the purpose of these requests. However, research has shown that developers often underutilize them, and when they do provide rationales, the content is frequently vague or ineffective [14, 16, 36, 60]. Earlier research has shown that users benefit from receiving additional information in permission requests [25, 54, 66, 73]. Specifically, research has indicated that including rationales increases the likelihood of users granting permissions [11, 14, 16, 60]. For instance, related work [11] observed that including a rationale string with permission requests reduced the denial rate by half. Additionally, recent research [14] conducted a comprehensive analysis of Android permission rationales, focusing on their design and phrasing. Their findings emphasize that the phrasing of rationales affects users' permission decisions.

3.2 Web Permissions

Research on web permissions has been limited, with most studies focusing on permission prompts and APIs [12] rather than rationales. A significant portion of this work has centered around push notifications, particularly their potential for misuse. Studies have highlighted how unethical content providers exploit these notifications on both mobile and desktop platforms, sending irrelevant or abusive messages to drive traffic [6, 57]. To address these issues, browser vendors such as Firefox [42], Edge [39], and Chrome [7] have implemented features to minimize unwanted interruptions. For example, Chrome experiments showed that quieter prompts, which are less visually prominent, significantly reduce interruptions while maintaining similar grant rates [7].

To reduce frequent and disruptive permission prompts, recent research [24] developed a machine learning-based solution in Chrome to predict when users are unlikely to grant permissions, thus reducing unnecessary prompts. Other research [27] explored vulnerabilities like click-jacking attacks on webcam access, highlighting the need for stronger user protections.

Finally, recent research [23] analyzed web permission interactions across 100 million Chrome installations, finding that geolocation and notification prompts are often ignored, while contextual information increases users' likelihood of granting permissions. However, no research has yet thoroughly explored the use of rationales for web permissions, particularly on desktop platforms, marking a gap in our understanding of how to best support users in making informed permission decisions.

4 Problem Statement

This paper aims to answer the following research questions:

RQ1: Rationale Detection, Prevalence, and Catalog. Since permission rationales on the web often appear within main page content, the first part of our paper focuses on developing a machine learning-based approach to identify these rationales within crawler results. To achieve this, we must first build a ground-truth dataset of web permission rationales tailored for ML-based solutions. Additionally, we require a detection method capable of conducting large-scale measurements of the prevalence of rationales for different permission types. This will allow us to create a comprehensive catalog of online permission rationales.

RQ2: Rationale Design Patterns and their Effects. The second part of our paper analyzes how permission rationales are presented in the web ecosystem. Using samples from the rationale catalog in RQ1, we identify common patterns of rationales found in the wild. By cross-referencing these patterns with permission prompt action rates from aggregated Chrome Telemetry and user sentiment data, we aim to gain insights into the impact of these rationales on user behavior and perception.

5 Methodology

To address our research questions, we follow the methodology outlined in Figure 3. We begin with *"Web Crawling & Prompt Detection"*, where we use a JavaScript-enabled web crawler to navigate a list of seed URLs. During each visit, the crawler captures webpage snapshots and screenshots, while also monitoring the page to collect any permission prompts that appear. Next, in *"Rationale Identification,"* we extract distinct text snippets from webpages. These snippets are processed through Large Language Models (LLMs) to identify those that pertain to permission-protected concepts, such as access to a camera or microphone. Using this filtered data, we manually curate a ground-truth dataset for permission rationales. This dataset is then used to train a BERT classifier to identify rationales. As a result, we build a comprehensive catalog of rationales. Finally, in *"Analysis of Rationale Text & UI patterns,"* we apply both automated clustering and manual coding to identify rationale patterns. This step incorporates both quantitative analysis of textual content and qualitative examination of UI elements of rationales. Finally, in *"Exploring the Effect of Rationales on User Decision-Making,"* we evaluate the impact of these rationale patterns by comparing permission grant and deny rates based on user activity data from Chrome telemetry.

5.1 Web Crawling & Prompt Detection

To answer RQ1, we developed a Chromium-based web crawler using Puppeteer and the DevTools Protocol (CDP) to capture webpage snapshots, simulating a desktop browser. The crawler loads an initialization script that modifies JavaScript permission APIs, enabling it to monitor permission prompts in real-time. For each webpage, the crawler waits up to 30 seconds for the page to fully load, then collects the client-side code, a Document Object Model (DOM) snapshot, and a screenshot to capture the main rationales presented in the webpage's user interface (UI).

For our analysis, we focus on telemetry from Chrome, which is known to be representative of popular websites, as demonstrated in recent research [52]. Specifically, we acquired a Chrome telemetry dataset from December 2022, comprising 770K publicly accessible URLs. Each URL corresponds to a specific webpage where at least 50 users across all platforms encountered a permission prompt within the last 28 days, as summarized in Table 1. The URLs were sanitized to prevent exposure of any sensitive personal information.

Platform	🖥		📱		Total	
Perm.	**Pages**	**Sites**	**Pages**	**Sites**	**Pages**	**Sites**
Geoloc.	192,728	46,450	272,007	47,210	464,735	93,660
Notif.	263,835	30,523	424,264	35,054	688,099	65,577
Mic.	11,046	4,054	11,217	3,898	22,263	7,952
Camera	9,336	3,863	18,407	6,835	27,743	10,698
Total	476,945	77,086	725,895	86,572	**770,349**	118,371

Table 1: Permission prompts seen by unqiue desktop/mobile clients.

During each webpage visit, the crawler tracks calls to permission-restricted APIs and records the permissions requested. If a permission request appears, the crawler rejects it using the CDP and captures a second snapshot to identify any secondary rationales that might appear when permission is not granted (whether ignored, dismissed, or denied). To maximize the detection of prompts and capture associated rationales, the crawler also interacts with the webpage by clicking on elements likely to trigger permission-related actions. The full list of interaction heuristics is provided in Table 11 of Appendix A. With these heuristics, our crawler detects nearly twice as many permission prompts compared to a non-interactive approach (see Appendix A.1).

5.2 Rationale Identification

To identify and extract rationale sentences from the crawled webpages, we followed a multi-step process involving text extraction, dataset construction, and model training.

We used BeautifulSoup [51] to parse each webpage's HTML and extract raw text, focusing on rationale sentences while excluding headers, footers, and stylistic elements. The text was deduplicated to retain unique samples for further processing.

5.2.1 Ground-Truth Dataset Creation with LLM Filtering. Using the unique extracted texts, we then constructed a ground-truth dataset to train a rationale classifier. Because permission rationales represent a small fraction of the vast text data, random sampling was impractical due to a low probability of identifying rationale sentences at scale. Previous research [36, 37, 43, 44] addressed a similar challenge using keyword matching, but this approach restricts collected samples to those containing predefined keywords (e.g., "camera" or "webcam") and often results in models overfitting to these terms. To avoid this, we adopted a keyword-agnostic method, employing few-shot prompting with the Mistral-7B language model [3] to identify relevant text snippets around *permission-protected concepts* such as camera, microphone, notifications, and geolocation. This filtering substantially reduced irrelevant samples.

After filtering, we applied random sampling to the remaining data, manually labeling each sample to identify rationale sentences. This process was repeated until we had a sufficient number of positive examples. We then balanced the dataset by under-sampling non-rationale examples, creating an equal mix of positive and negative

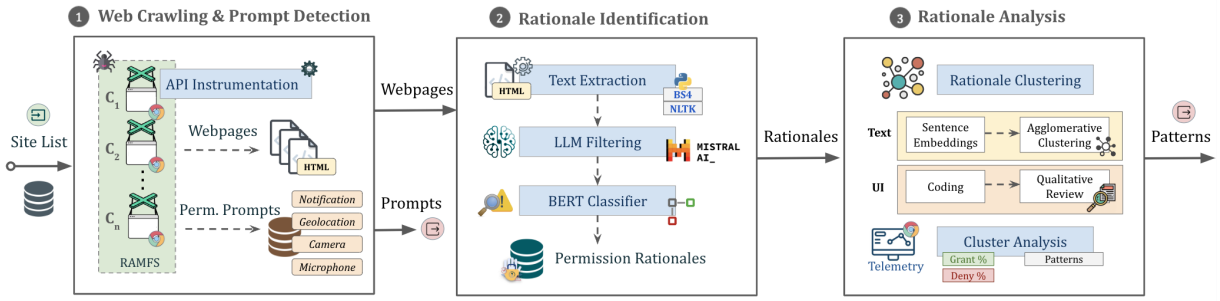


Figure 3: Overview of our methodology. We address RQ1 in *1) Web Crawling & Prompt Detection* and *2) Rationale Identification*. We answer RQ2 in *3) Rationale Analysis*, which includes two key phases: *Analysis of Rationale Text & UIs Patterns* and *Exploring the Effect of Rationales on User Decision-Making*.

samples for training. Details of the prompt used for the language model are provided in Appendix D.

5.2.2 Large-Scale Rationale Classification. Using the labeled dataset, we trained a BERT classifier [17], as it outperformed alternative models such as T5 in our context. The classifier was trained over 10 epochs with a learning rate of 0.0001, ensuring gradual and stable model updates. A batch size of 16 was chosen to balance computational efficiency with the need for sample diversity, and gradient accumulation steps were set to one, allowing parameter updates after each batch. A weight decay of 0.005 was also applied to improve generalization by discouraging large parameter values. These parameters were chosen based on best practices in BERT fine-tuning [1, 21, 50] and our initial experiments. Finally, we used the trained BERT classifier to categorize the unique text extracted from the webpages.

5.2.3 Manual Review and False Positive Analysis. To eliminate false positives, two researchers independently reviewed the samples classified as rationales, verifying their accuracy by examining the associated webpages or UIs. For cases of disagreement, a third researcher conducted an additional review, followed by a discussion among the three researchers to resolve conflicts. All reviewers were experts in web and usable security.

5.3 Analysis of Rationale Text & UIs Patterns

To answer RQ2, we analyzed the classification results and grouped the rationales into clusters to identify common patterns, considering both rationale text and UI components.

5.3.1 Rationale Text Patterns. For text analysis, we generated embeddings of rationale text snippets using the all-MiniLM-L6-v2 sentence transformer [18] and applied agglomerative clustering [53] to capture syntactic and semantic relationships. Through random sampling, we examined each cluster, compiling a comprehensive list of codes and attributes related to message sentiment and content until no new information emerged (saturation). We labeled the rationale samples based on these attributes, initially using one-shot prompting with GPT-4 to assign attributes, followed by a manual review for accuracy. This allowed us to group rationales with similar attributes. The results of this stage are presented in §8.1.2.

5.3.2 Rationale UI Patterns. In the UI analysis, we performed a qualitative review of different UI designs. This included both pages where a text rationale was detected by our ML pipeline and random samples from pages with observed prompts but no detected rationales. For each case, we analyzed screenshots captured before and after the denial of a permission prompt. If screenshots lacked the rationale (e.g., due to complex interactions missed by the crawler), we used a semi-automated approach to capture them by manually interacting with the page. After obtaining two screenshots for each rationale, two independent reviewers analyzed and coded the UI components, continuing this process until saturation. In total, we analyzed 7,413 webpages, resulting in 749 distinct UI rationales, which we will discuss in §8.2.

5.4 Exploring the Effect of Rationales on User Decision-Making

We conducted an exploratory analysis to determine to what extent the attributes of rationale texts and UIs can influence users' decisions to grant, deny, dismiss, or ignore browser permission prompts. Additionally, we sought to evaluate how users perceive their overall experience with permission requests, particularly in terms of annoyance and ease of use. To that end, we relied on two datasets that we gained access to from Google's Chrome browser:

- **Chrome Telemetry Data.** We analyzed Chrome desktop data on user interactions with permission prompts. This data is collected when users (1) enable the setting to "Make searches and browsing better" by sending URLs of visited pages to Google and (2) when at least 50 Chrome users visit the page and respond to a permission prompt. The data we used covers the 28 days leading up to August 8, 2024.
- **Chrome User Sentiment Data.** Chrome fielded experience sampling questionnaires to understand how users feel about web permissions on desktop platforms. Users were eligible to answer a questionnaire if they had enabled the "Help improve Chrome's features and performance" setting and had not seen another questionnaire in the last 180 days. We focused on responses from users who shared URLs with their answers, which allowed us to link their feedback to specific reasons and user interface designs. URLs are available when users opted into the "Make searches and browsing better" setting. Questionnaires were available to Chrome users with an English language setting between November 2, 2023, and January 15, 2024, and collected 118,949 complete responses. The questionnaire was originally fielded for a Google-internal

project and included four questions, two of which we were able to leverage to understand user sentiment on permission prompts. Respondents rated both their annoyance and ease of decision-making on a 5-point Likert scale (see Appendix C for exact wording and more details on the data collection).

We cross-referenced webpages with coded rationale texts and UIs with those containing telemetry and user sentiment data to obtain our sample described below.

5.4.1 Sample Description. The telemetry sample included data for 242 of the websites with coded UI rationales and for 2,687 websites with coded text rationales. The user sentiment sample consisted of 1,351 responses across 282 URLs (169 geolocation, 59 camera, 55 microphone) for coded text rationales and 443 responses across 97 URLs (72 geolocation permission, 17 camera, 8 microphone) for coded UI rationales. Given that these datasets reflect website usage, popular sites are more likely to appear, increasing the likelihood of detecting patterns common on more popular sites.

Additionally, we incorporated control samples from pages with permission prompts but no detected rationale for which we also had telemetry and user sentiment data available. For the UIs, we included all samples that were manually inspected but did not show a rationale for a total of 89 URLs. For the texts, we randomly selected 500 URLs from the set of manually verified text samples.

5.4.2 Statistical Tests. We used regression models to evaluate the impact of rationale attributes on user behavior and sentiment toward permission prompts. Analyses were conducted in R 4.4.1. For user behavior, we applied exploratory linear regression to websites with telemetry data, modeling each prompt action (allow, deny, dismiss, ignore) separately. For sentiment, we used logistic regression on top-2-box Likert-scale scores. Permission type was included as a factor, acknowledging varying grant rates [23]. These models were exploratory and not further optimized or validated.

The following sections will provide a detailed presentation of the results for each section discussed above.

6 Web Crawling & Prompt Detection Results

In February 2023, we used the 770K seed URLs from Chrome telemetry dataset as a starting point to initialize our crawling infrastructure, deploying 100 parallel browser instances. As a result, we successfully collected snapshots of 739K pages from an EU vantage point. To ensure comprehensive coverage, we attempted to recrawl each failed page up to three times, followed by a manual review. For 31,114 URLs across 4,834 domains, all three crawl attempts failed–mostly due to inactive URLs or pages timing out (taking over 30 seconds to load). The data collection process spanned approximately seven weeks.

Table 2 provides an overview of the captured permission prompts and the triggered calls to permission-gated APIs recorded by our interactive crawler. It also quantifies the proportion of permission prompts observed within the Chrome telemetry data. Overall, our crawler found 29K domains with at least one web permission API call, with a total of ~1.6M API calls across 161K webpages. We observed that the geolocation API is the most widely used, with almost 1.6M calls across 99K pages, and also the most widespread, being present on more than 22K sites, which is followed by push notifications present on 6.6K websites with 73K API calls.

	# Sites	# Pages	# Calls
Seed URLs	118,371	770,349	-
Collected successfully	113,537	739,235	-
Geolocation	22,036	99,241	1,608,729
Notification	6,657	69,567	73,231
Microphone	139	233	334
Camera	188	220	322
Total	29,020	**161,775**	1,682,616

Table 2: Summary of collected webpages and observed prompts.

	Processing		Validated Rationales			
Perm.	*LLM Filt.*	*Classif.*	*Confirm.*	*Instances*	*Pages*	*Sites*
Notif.	6,918	2,675	1,666	22,739	14,855	1,950
Geoloc.	127,552	2,305	1,063	2,136	1,680	894
Camera	14,082	1,005	495	848	543	364
Mic.	7,878	1,543	617	1,087	587	322
Total	155,093	7,254	**3,674**	**26,810**	17,333	3,237

Table 3: Processing steps for ~6M unique text snippets from in-the-wild webpages: (1) LLM filtering samples, (2) Large-scale BERT classification results, (3) Manually confirmed rationales, (4) Total count of rationale instances (real-world distribution), (5) unique webpages, and (6) unique domains.

Our crawler detected permission prompts on ~20% of seed pages from Chrome telemetry, despite all using popular permission-gated web APIs (per Chrome telemetry). Investigating 100 missed cases revealed that 73% stemmed from crawling challenges like complex user interactions and authentication, 11% were inactive pages, and 16% involved privacy-sanitized URLs causing discrepancies. See Appendix A.2 for further details. As we will show next, our approach can still detect rationales in many of these cases, particularly when the rationale text is present in the DOM, even if the permission prompt is not triggered by the crawler.

7 Rationale Identification Results

Starting with 739K pages across 113K websites from our seed list, we extracted ~20M English text samples. After deduplication, we retained 6M unique samples.

7.1 Ground-Truth Dataset Creation with LLM Filtering

To create a relevant, smaller-scale dataset for training our ML classifier, we used few-shot prompting with the Mistral-7B LLM, as described in §5.2. We evaluated the LLM-based filtering on a manually compiled dataset of 143 rationales, selected via random sampling (denoted as DS1). The filtering approach achieved a high recall rate of approximately 93%, with 133 true positives (TPs) and 10 false negatives (FNs). Our goal was to maximize TPs while filtering out potential false positives (FPs), which would be handled later by the classifier. From the filtered set, we extracted about 155K unique rationale candidate texts from the 6M unique texts in our dataset. Table 3 presents an overview of these results and their distribution across permission types.

Next, we applied an iterative sampling method to the 155K filtered samples. We randomly selected batches of 100 samples for manual annotation until we gathered sufficient rationales. After reviewing 2,100 samples, we identified 262 rationales. Combined with the 143 rationales from DS1, this yielded 405 positive samples. We then added 1,785 negative samples to create a labeled training dataset of 2,190 samples, referred to as DS2.

We split DS2 into training, validation, and test sets using an 80%-10%-10% ratio. To address class imbalance, we under-sampled the training set to equalize the number of positive and negative samples. Using this labeled dataset, we trained a BERT classifier, as described in §5.2. The classifier achieved an F1 score of 0.82 on the validation set and 0.83 on the test set, indicating robust performance.

7.2 Large-Scale Rationale Classification

We used the BERT classifier to automatically annotate labels to the 155K samples and rule out potential false positives. The classifier flagged 7,254 unique samples as positive. These unique rationales correspond to 40,996 rationale instances across 28,538 unique pages that themselves belong to 5,798 unique domains.

7.3 Manual Review and False Positive Analysis

Three human analysts conducted a thorough manual review of all 7.2K discovered rationales to eliminate potential false positives following the methodology detailed in §5.2. When considering only the text, we observed a false positive rate of 19.6%, which is consistent with the figures observed in our test split during the training phase. When additionally considering the UI context, we observed a false positive rate of 49%, identifying 3,674 cases as true positives. The primary reason for this relatively high false positive rate is that many texts initially appeared to be valid rationales when evaluated in isolation. However, we observed that the context in which this text appears is crucial for accurate identification. For example, text snippets found on tutorial sites describing messages from other webpages, or user-generated content (e.g., comments) may initially seem like rationales but are not upon closer UI analysis. However, automatically extracting such contextual information from webpages remains highly challenging due to the dynamic nature of webpages and the complexity of HTML structures and semantics. This insight underscores the importance of both content and context in accurately identifying rationales on the web, which in this work, we tackled using a semi-automated approach.

7.4 Catalog of Rationales and Comparison with Permission Prompts

Through our extensive manual and automated analysis, we compiled a catalog of rationales containing 3,674 unique samples, totaling 26.8K instances across 17.3K unique webpages and 3,237 unique domains. Notification rationales were the most common, with over 22K instances, while camera rationales were the least common, with only 848 cases. A summary of our rationale catalog across various permission types can be found in Table 3 (columns 4-7).

Our crawler detected permission prompts on 161K webpages, but according to our ML-based detection, only 7.5K of those pages (4.6%) included a text-based rationale. This lower rate is expected, as many pages trigger prompts without including rationales, or the rationales may be embedded in non-text formats. On the other hand, our ML-based approach identified 9.8K pages with text rationales where no prompt was observed, suggesting that rationales can still be present even when the crawler is unable to trigger a prompt.

7.5 Rationales in Libraries and Prevalence

We observed that certain rationales in our catalog exhibited a notably high prevalence across the web. Intrigued by this pattern, we manually reviewed the most frequent rationales. We found that these cases are associated with geolocation and notification permissions, and implemented via 10 distinct third-party libraries.

7.5.1 Library Signatures. For each library, we extracted specific code signatures based on HTML elements (such as `id` and `name`) that these libraries incorporate into webpages. Our goal was to search these signatures within our broader dataset of webpage snapshots collected during web crawling, allowing us to uncover any instances that our machine learning-based detection pipeline might have overlooked. In total, we created 32 detection rules for libraries. The complete list of rules is in Table 14. As we show in Appendix E, our rules are robust against false positives.

7.5.2 Libraries and Prevalence. Table 4 presents the ten libraries we identified and their rationale messages. For each rationale, the table shows the number of webpages we found using our (i) machine learning-based approach and (ii) signature search approach, including the union and intersection of both methods. Our analysis reveals that signature searching significantly enhances our findings, uncovering over 67K additional instances of rationale messages for one of these libraries in the wild, compared to only 15K instances of these libraries detected using our initial dataset. We observed that the top three used libraries are OneSignal [46], iZooto [30] and Smart Push [29]. However, the majority of the newly discovered instances belong to OneSignal, with over 63K instances identified through signature searching alone. In addition, the majority of these 10 libraries are focused on push notifications, with only a few supporting or being designed for geolocation permissions. In total, we identify 82.8K webpages that use a rationale from a library, and 85,093 webpages that use either a custom or library rationale. Overall, this strategic approach not only refined our understanding of rationale prevalence but also strengthened the comprehensiveness of our rationale catalog.

The ML-based approach missed these rationales because the captured webpage snapshots did not include the rationale text, which required complex user interactions (e.g., clicks) to appear. Our interactive agent in §5.1 failed to simulate these interactions. However, the HTML code signatures of libraries were present, enabling the signature-based method to find them. The ML-based approach was able to find the text of these rationales on other page snapshots that used the same libraries but did not require user interaction to load their content. We refer interested readers to Appendix E, where we discuss the complementary nature of ML and signature-based rationale detection methods.

We note that when the library signatures appear in the DOM, we cannot guarantee that the rationale message will be always visible. Also, as we will discuss in §10.1, it is challenging to fully

Library	Perm.	Rationale	ML ∪ SS	ML ∩ SS	SS only	ML only
iZooto [30]	Notif.	Real time notifications have been turned off. Enable them to get important and timely updates.	5,274	5,074	0	200
	Notif.	Real time notificatios are turned off. You can enable it to receive timely updates.	5,274	5,074	0	200
OneSignal [46]	Notif.	We'd like to show you notifications for the latest news and updates.	64,194	473	63,716	5
	Notif.	Would you like to be aware of all the hottest news and events from $SITE?	55	1	55	0
PushEngage [48]	Notif.	Subscribe to notification	730	205	525	0
Smart Push [29]	Notif.	Give us a permission to receive push notification messages and we will keep you posted	1,174	652	522	0
	Notif.	Give us a permission to receive push notification messages and we will keep you posted	1,174	652	522	0
	Notif.	You can choose to turn off notifications later anytime using browser settings.	1,174	746	428	0
Moe-push [40]	Notif.	This website would like to send you awesome updates and offers! Notifications can be turned off anytime from browser settings. Don't Allow	491	55	436	0
PushOWL [49]	Notif.	Get Updated with Latest Offers and Products.	685	262	412	11
Perfecty [47]	Notif.	Do you want to receive notifications?	354	90	264	0
	Notif.	I want to receive notifications	354	287	67	0
Webpushr [67]	Notif.	You are unsubscribed to Push Notifications	397	192	201	4
	Notif.	You are subscribed to Push Notifications	397	202	191	4
	Notif.	Subscribe to receive push notifications on latest updates	400	207	186	7
	Notif.	You have blocked Push Notifications. Follow these instructions to enable Push Notifications.	397	208	185	4
Superstorefinder-wp [58]	Geoloc.	Location service is not enabled. Continue anyway | Share my location	79	49	30	0
Storerocket [28]	Notif.	Get notified of new locations.	56	41	14	1
	Geoloc.	Allow the geolocation on your browser and refresh the page.	75	71	3	1
	Geoloc.	Your browser blocked our request to get your location.	75	71	3	1
Total			**82,809**	14,612	**67,760**	438

Table 4: Rationale messages from libraries and their prevalence on the Web based on the number of webpages. The table shows the contribution of library code signature searching to identify additional webpages that use one of the rationale messages in our catalog, and compares it with results from our ML-based detection pipeline. The library detection rules are in Table 14. Legend: SS= signature search; ML= machine learning; $SITE= a placeholder for site name.

Cluster	Subcl.	Rationales	Pct.	Instances
C1: N	28	1,655	45%	22,727
C2: G	21	1,046	28.4%	2,116
C3: M	11	468	12.7%	877
C4: C	7	348	9.4%	641
C5: C_M	2	139	3.7%	197
C6: Other	6	18	0.49%	21
N_G	1	7	0.19%	7
C_M_G	1	4	0.11%	5
C_N_M_G	1	3	0.08%	4
M_G	1	2	0.05%	3
N_M	1	1	0.03%	1
C_G	1	1	0.03%	1
Total	70	3,674	100.00%	26,810

Table 5: Overview of text-based rationale clusters ordered by size. The top part shows prevalent clusters, while the bottom part highlights unique behaviors that are grouped in the *other* cluster. The left part shows the percentage of *unique* samples per cluster, whereas the right part (i.e., instances) shows their real-world distribution, i.e., non-unique count of rationales across webpages in our data set. Legend: Subcl = Subcluster. Pct = Percentage. N = Notification. G = Geolocation. C = Camera. M = Microphone.

disentangle library and custom rationales at scale, since webpages may use both or customize them.

8 Analysis of Rationale Text & UI Patterns

We analyzed the collected rationales to identify common patterns, following the methodology outlined in §5.3.

8.1 Rationale Text Patterns

Our automated clustering method leveraging all-MiniLM-L6-v2 sentence transformer organized the 3.6K rationales into 75 clusters based on both textual syntax and semantics. Out of these, we consolidated six clusters that included only few samples, into a larger one named *Other*, reducing the total to 70 clusters. To simplify, we applied hierarchical clustering and further merged clusters that rely on the same set of permissions together, resulting in six higher-level clusters. Table 5 provides a summary of the clusters by permission type, detailing both their unique count and their prevalence on the web. We refer interested readers to Appendix F, which provides examples of the 70 individual subclusters.

We found that the *notification* cluster is the largest, making up about 45% of the unique samples with over 22K instances observed in the wild. The *geolocation* cluster follows, representing over 28% of unique rationales in our catalog, but with significantly lower prevalence at around 2.1K instances.

8.1.1 Rationale Text Attributes. We undertook a qualitative analysis of the 70 clusters to extract their characteristics by manually examining random samples from each cluster until saturation, in line with the methodology detailed in §5.3. Other than permission type, we found that rationale texts can vary widely across four dimensions: (i) sentiment and tone, (ii) encouragement style including benefits and consequences, (iii) necessity of permission granting for proper functionality, (iv) and message content such as errors, instructions, and reassurance on data use. In the following, we discuss these attributes with real-world examples.

Message Tone. The tone of a rationale text indicates the emotional or attitudinal stance conveyed to the user. A *positive* tone (POS) employs language that contains excitement, such as the message *"This website would like to send you awesome updates and offers!"* In contrast, a *neutral* tone (NEUT) presents information in a factual manner, as seen in examples like *"This website requests access to your location."* Conversely, a *negative* tone (NEG) communicates caution or potential errors, as illustrated by statements like *"Sorry! We can't access your webcam and/or audio recorder."*

			p50 (Median)			
Category	**Attribute**	**Count**	**grant %**	**deny %**	**dismiss %**	**ignore %**
Message Tone	None	498 (18.6%)	15.1	7.8	31.0	35.3
	Neutral	2,026 (75.5%)	12.5	8.9	25.5	41.2
	Negative	102 (3.8%)	48.2	6.2	20.3	11.6
	Positive	49 (1.8%)	39.5	6.3	24.5	21.2
Encouragement	None	2,386 (89.2%)	14.5	8.4	26.5	37.8
	Motivation	284 (10.6%)	12.1	11.3	24.1	45.3
	Consequence	5 (0.2%)	60.7	3.3	32.1	1.6
Necessity: Required	True	66 (2.5%)	66.6	5.5	17.9	6.1
	False	2,609 (97.5%)	13.5	8.7	26.4	39.5
Necessity: Optional	True	23 (0.9%)	53.2	5.4	22.8	10.4
	False	2,652 (99.1%)	13.9	8.7	26.3	39.0
Message Content	Permission Request	1,361 (50.9%)	18.7	7.8	24.5	31.7
	Func. Explanation	311 (11.6%)	18.6	9.8	19.7	31.5
	Error	265 (9.9%)	43.1	6.3	20.3	14.4
	Instruction	207 (7.7%)	8.7	11.1	23.6	50.4
	Emphasize Control	85 (3.2%)	14.6	9.0	26.1	39.6
	Data Use Reassurance	44 (1.6%)	67.2	5.1	17.2	5.8
	Loading Device	14 (0.5%)	91.9	2.2	2.9	1.5

Table 6: Distribution of attributes and the corresponding action rates across webpages in our dataset for which we had telemetry available (overall $n = 2,675$). The *"None"* attributes represent the control groups. Attributes in the Message Content section are not mutually exclusive and we omitted the action rates. The count column shows the count of webpages with rationales that have the corresponding attribute, based on the methodology in §5.3, for which we had telemetry data available. The percentages in brackets show the proportion of webpages exhibiting this attribute.

Encouragement. Encouragement in rationale texts could vary. A *motivating* approach (MOTIV) suggests actions by highlighting benefits, exemplified by statements like *"Allowing notifications will keep you updated with the latest news"* or *"Granting camera access will improve your experience."* Instead of the benefits, the message may convey the *consequences* of permission denial (CONSEQ), e.g., *"WARNING: If you select BLOCK, you cannot have a video call because your camera and microphone cannot be used"* or *"Blocking camera access may limit features of this website"*.

Permission Necessity. Necessity in rationales determines the perceived importance of a permission request. *Required* actions (REQU) indicate essential permissions, e.g., *"Permission to access your contacts is required to sync data."* These cases often directly instruct or mandate user action, such as stating, *"You must grant microphone access to continue."* Conversely, *optional* actions (OPT) suggest enhancements or alternatives, such as *"You can enter your address manually or allow automatic filling for convenience"* and *"You may get a popup asking you to Allow or Block your location. The search function will work with either option."*

Message Content. Rationale messages include different and sometimes multiple types of content. One type provides *guidance* for troubleshooting and resolving problems, such as *"Troubleshoot permission issues by resetting your browser settings"*. These types of rationales may also provide more precise, possibly step-by-step instructions (INSTRUCT), such as *"To enable microphone access, go to Settings > Privacy > Microphone."* In comparison, *error* messages (ERROR) alert users about incorrect actions or issues, as seen in messages like, *"Error: Location access denied. Please grant permission to proceed."* Other *Emphasize control* (CONTROL), such as *"Notifications can be turned off anytime from browser settings."* In addition, the rationale can *reassure* users about data usage and privacy risks (REASSURE), e.g., *"Access to your camera is necessary, but no personal data is collected"* and *"We need your location to provide you with the best experience. Your location is safe with us."* Other rationales state the permission status, such as *"Accessing camera, please wait..."* and *"Waiting for camera to load"* (LOADING), or simply contain a direct *permission request* (PREQ) like *"Please allow access to your location."* Finally, the message may also include a functionality *explanation* (FUNC_EXPL) clarifying for what purpose the permission is needed, e.g., *"In order to find a store near you, allow location access or use the search feature."* Websites rely on these types of rationales based on their specific scenarios and user needs.

8.1.2 Analysis of Rationale Text Attributes. Table 6 shows the prevalence of each rationale attribute across the webpages in our dataset for which we had telemetry available (overall $n = 2,675$) and the corresponding permission prompt action rates. Our clustering algorithm of §5.3 identified 123 rationale groups across 17.3K URLs. However, only 32 groups had samples from more than 10 URLs, and among those, only 18 had sufficient telemetry data. We focused on these 18 groups to analyze how various factors, such as message tone and encouragement, influence users' permission decisions.

Common Text Patterns. Table 7 provides an overview of the 18 common rationale clusters and their sizes, illustrating how various online platforms communicate their need for web permissions. Each cluster is categorized by a distinct set of attributes from §8.1.1, capturing their tone, encouragement, necessity, and message content. The N0 cluster is the most prevalent, appearing on over 11,8K webpages. These neutral messages typically prompt users to allow notifications and have significant telemetry presence, with 882 URLs, showing that users frequently encounter this scenario online.

Cl.	Attributes	Rationale Example	Count	Telem.
N0	NEUT	Click Allow to receive notifications.	11,863	882
N4	NEUT, PREQ, FUNC_EXPL	click Allow to get notified about low-cost dental care	663	171
N33	NEUT, INSTRUCT	Step 2. Tap the toggle switch to turn the notification off and on.	470	147
N2	NEUT, CONTROL	The website $SITE would like to send you push notifications. Notifications can be turned off anytime from browser settings.	1,031	46
N20	POS, MOTIV, PREQ, CONTROL	$SITE would like to send awesome offers for your furry friend! Notifications can be turned off anytime from browser settings. Don't Allow	247	26
N14	NEUT, INSTRUCT, PREQ	Subscribe to receive push notifications on latest updates You have blocked Push Notifications. Follow these instructions to enable Push Notifications.	239	15
N35	NEUT, MOTIV, FUNC_EXPL	Apply to jobs anytime, anywhere and get notified instantly when your application is reviewed.	32	13
G1	NEUT, PREQ, FUNC_EXPL	Please allow location permission from your browser to view nearby leases	1,233	432
G10	NEG, ERROR	Opps! Unable to retrieve your location, please enable location access in your browser OK No Cancel	198	62
G29	POS, MOTIV, PREQ	$SITE requires access to location. To enjoy all that $SITE has to offer, turn on your GPS and give $SITE access to your location.	47	13
G34	NEUT, ERROR, REASSURE	Your Location access is blocked! Please provide location access to proceed further. Your location is safe with us.	22	12
C6	NEUT	You will be asked to enable camera access	248	64
C17	NEUT, PREQ	When prompted, click "Allow" and you'll see your camera	50	28
C9	NEUT, INSTRUCT	Use your Camera to start VideoChat. Allow access the Camera in your browser's Settings Your webcam is active on $SITE. To use the webcam here please close $SITE	36	10
M7	NEUT, PREQ	Enable microphone access on this site by clicking the big "Enable Microphone" button.	163	56
M5	NEUT	Use the audio devices on your computer to speak and listen	101	40
M27	NEUT, FUNC_EXPL	After pressing the call button, a window appears in the upper right corner asking you to allow access to the microphone, you should click enable to start the free call. If you accidentally clicked on the disallow button, try reloading the page.	70	29
M96	NEG, ERROR, FUNC_EXPL, REQU	No microphone found. Unable to continue.	13	13

Table 7: Summary of text rationale clusters having more than 10 samples based on their distinct number of URL-permission pairs for which Chrome telemetry was available. For each cluster, the table shows its attributes, unique count (i.e., number of URLs in the dataset belonging to that cluster), number of cluster URLs with telemetry data, and an example. Legend: Cl. = Cluster. $SITE = a placeholder for site name. N = Notification. G = Geolocation. C = Camera. M = Microphone.

Another key cluster, G1, appears on over 1.2K webpages and underscores the widespread use of geolocation-related rationales. Lastly, the N2 cluster, focused on push notifications, appears on nearly 1K URLs and reassures users that they can deactivate notifications at any time. However, the lower telemetry counts suggest these pages are less frequently visited.

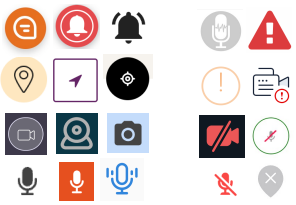
Message Tone. We observed that the majority of the rationale messages (75.5%) maintain a neutral tone across most clusters, such as N0, G1, C6 and M7 in Table 7, aiming for a straightforward and factual manner. Positive and negative tones are rarer, accounting for about 2-4% of our samples. Positive tones are employed mostly in notification and geolocation rationales such as N20 and G29, together with motivations that highlight the benefits of granting permissions, such as receiving offers or enhancing their experience. Negative tones are often used in geolocation and microphone rationales, such as G10 and M96.

Encouragement. Encouragement strategies that highlight the benefits of granting permissions, such as receiving timely updates, discounts, and price alerts via push notifications, or ensuring the best experience through location tracking, were common, appearing in 10.6% of samples, as seen in clusters like N20 and G29. In contrast, using consequences as a cautionary tactic is notably rarer, appearing in just 0.2% of our dataset, typically to warn users about potential limitations in functionality if permissions are not allowed.

Permission Necessity. We found that most rationales avoid explicit categorization, with only 2.5% labeling actions as required and 0.9% as optional. Instead, the required permissions are only clearly emphasized in critical contexts where the service cannot function without specific permissions, such as the mandatory geolocation permission for account creation (geo-restricted) in cluster G0 of Table 15 and M96 for microphone access in Table 7.

Message Content. Rationale messages vary in content, often combining multiple types. The most common are permission requests (50.9%), such as G1, N4 and M7, mirroring the primary purposes of most rationale texts. Other notable types include emphasizing that the permission can be deactivated anytime (3.2%), such as N2, and the provision of functionality explanations (11.6%), such as M27. These elements highlight the efforts of websites to inform users and reinforce control [7]. Rationales involving error messages (9.9%) and guiding instructions (7.7%) are less frequent but commonly used in scenarios where user action is needed to resolve issues, such as N33, C9, G34, and M96. Finally, reassurances on data use are rare (1.6%), like G34.

Summary of Insights. Overall, our systematic analysis reveals a spectrum of strategies in permission-related communications, reflecting varying levels of urgency and user autonomy. The distinctions across clusters underscore the tailored approaches websites take depending on the specific functionality and sensitivity of the data involved, highlighting the intricate balance between user experience and operational necessity. The distribution of the identified rationales highlights a preference for neutral messaging, with larger clusters reflecting more common and broadly applicable requests. These messages generally avoid explicitly stating the necessity of permissions, allowing users to infer their importance. In contrast, the smaller clusters, while less frequent, are mostly tailored to specific user interactions, such as permission-dependent functionalities (e.g., camera for identity verification), troubleshooting or reassurance, which may require more detailed or emotionally nuanced messaging.



(a) (b)

Figure 4: Icons in rationales. (a) Representing different permissions, arranged from top to bottom: notifications, geolocation, camera, and microphone. (b) Indicating denied permissions.

8.2 Rationale UI Patterns

We analyzed rationale UIs following the methodology described in §5.3, compiling a dataset of 749 rationale designs across 631 webpages. Our analysis revealed several patterns in how permission requirements are communicated, identifying eight distinct layout patterns. Each layout is characterized by a unique combination of attributes, which we will introduce first in the following section.

8.2.1 Rationale UI Attributes. A rationale, which may be linked to one or more permissions, has a distinct layout with several notable attributes. We identified the following attributes:

Position. Rationales can be *inline*, seamlessly integrated into the page, or *floating*, overlaying content or protected elements like a camera feed or map. Floating rationales can appear anywhere on a 2D plane: top, bottom, left, right, or center.

Elements. Rationales often include additional elements such as *buttons*, *icons*, *alternative options*, and *visual or textual instructions*. Regarding *buttons*, we identified three types based on their functionality. A positive button is designed to trigger the browser prompt when clicked, typically labeled with *"Allow"*, *"Subscribe"*, *"Use My Location"*, or *"Turn On"*. Conversely, acknowledge/dismiss buttons serve to acknowledge and dismiss the rationale and were labeled with *"OK"*, *"Got It"*, *"Cancel"*, *"Don't Allow"*, or *"Later"*. Websites represented these buttons also by an *"X"*, typically located in the upper right corner of the rationale. We also found instances where links were provided to help and troubleshooting pages, labeled with phrases such as *"How to Grant Access"*, *"Troubleshooting Tips"*, or *"The Help Center"*. We found that when multiple buttons are present, developers often use opinionated design by applying distinct styling–such as different colors, sizes, or visual cues–to the positive button. While this approach can subtly encourage users to grant permissions, it is important to note that not all patterns aimed at increasing grant rates are legitimate. In some cases, this kind of nudging can cross the line into dark patterns that manipulate users into granting permissions they might not otherwise agree to. We leave investigating the prevalence of dark patterns for future work.

Besides buttons, we observed that rationales may include *icons* referencing the permission-protected functionality, as illustrated in Figure 4a. Icons may also indicate permission status, such as a crossed-out or disabled icon for denied permissions, or an exclamation mark signaling missing permissions, as shown in Figure 4b. Then, instead of granting permission, rationales may offer *alternative options* to users. For example, users might manually search for a location instead of granting geolocation access (Figure 5a), join an online meeting without camera and microphone access (Figure 5b), or view all shops instead of only seeing the nearest ones.

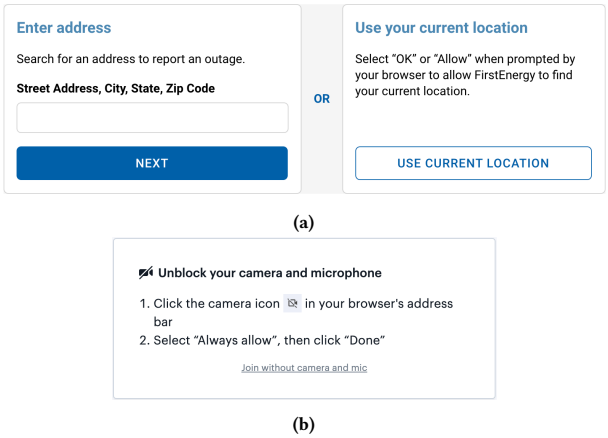


Figure 5: Buttons in rationales. (a) Users can grant permission to access their current location or manually search for an address. (b) Option to join a meeting without granting permissions, with an alternative button.

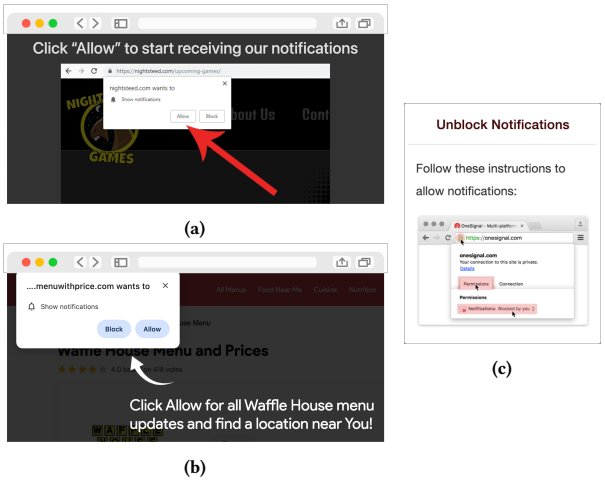
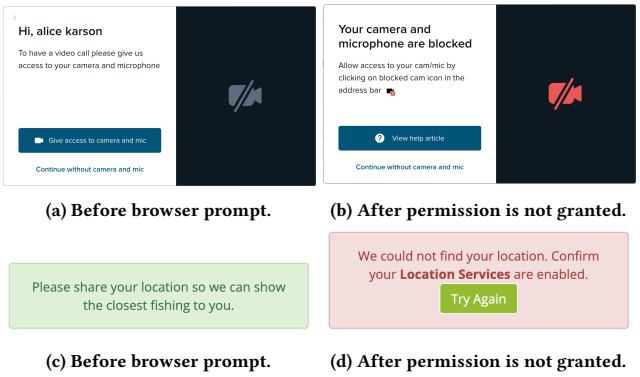


Figure 6: Textual and visual instructions. (a) As screenshot showing how to grant permission when prompted. (b) With an arrow directing attention to the browser prompt. (c) Showing how to re-enable denied permission.

Figure 7: Rationale displayed alongside a browser prompt with a loading icon.

Finally, rationales may include *visual or textual instructions*, using screenshots (Figures 6a, 6c) or text (Figure 6b).

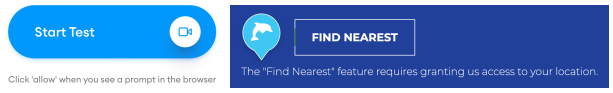
Timing. Rationales can be introduced at various points of the permission request cycle. *Before* requesting permission, a rationale can prepare the user for the request. *Alongside* a browser prompt, a rationale can guide the user's attention, often using a loading icon to indicate the webpage is waiting, such as Figure 7. *After*



(a) Before browser prompt. (b) After permission is not granted.

(c) Before browser prompt. (d) After permission is not granted.

Figure 8: Adaptive rationale with content updates after permission is not granted, shown as a dialog in (a) and (b), and as a banner in (c) and (d).



(a) (b)

Figure 9: Buttons implicitly requesting permission, with explanatory text.

dismissing, ingoring, or denying, a rationale can inform the user about the missing permission using indicators like bright colors, bold typography, and warning icons, as depicted in Figure 4b.

A webpage may present different rationales for the same permission depending on the timing. Some rationales remain static regardless of permission status, while others update dynamically. For example, an initial dialog may update with instructions on how to re-enable permission if denied, as shown in Figures 8a and 8b. Similarly, a banner rationale might change its content and color after permission is not granted (Figures 8c and 8d), and a fullscreen rationale can also differ before and after (see, e.g., Figure 15).

(a) (b)

Figure 10: Rationale as button with alternative option.

8.2.2 Analysis of Rationale UI Patterns. We outline the most common UI patterns, organized by layout and the frequently co-occurring attributes, providing a foundation for our exploratory analysis.

Text. Starting with the simplest pattern, rationales can be solely text integrated within static webpage content. These rationales remain unchanged regardless of permission status. For instance, a webpage might state: *To use live audio input, please allow access to your browser microphone when prompted or check your browser settings.*" Similarly, a help section might include steps like: *1) Plug in your headphones. 2) Allow browser access to your microphone. 3)…*".

Text-only rationales can also appear dynamically after permission is not granted, often highlighted with bold text, a distinct color (e.g., red), or an exclamation mark, such as: *"Could not get your current position! Location service must be enabled."* Furthermore, dynamic text rationales may appear while waiting for user interaction with a permission prompt, such as *Click allow for daily weather updates"* or *Trying to detect your location…"*.

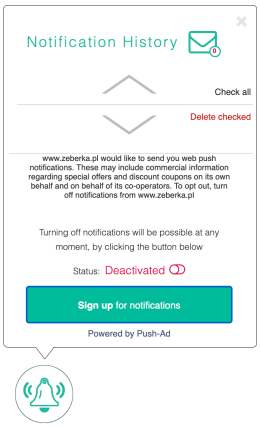


Figure 11: Clicking the button causes a rationale dialog to appear.

Button. Buttons, such as dismiss or allow buttons on a dialog, can be part of a larger rationale layout. However, buttons can also serve as the rationale itself, implicitly indicating the need for permission to enable a feature. Examples include buttons labeled *"Find Immediate Care Near You"* or *"Start Video Chat."* These buttons may be accompanied by explanatory text, as shown in Figure 9. For instance, a button labeled *"Start Test"* might be accompanied by the text *"Click 'allow' when you see a prompt in the browser."*

Whenever a button implicitly requests permission by activating a function or explicitly with labels like *"Allow Permission"*, an alternative to granting permission can be provided. For example, instead of granting microphone access, the user might click a button to upload an audio file (Figure 10a). Similarly, instead of clicking on *"Places Nearby"* button, the user could also use the *"See All Places"* option (Figure 10b).

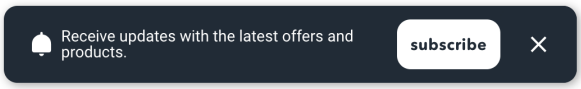
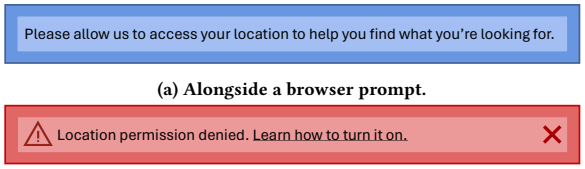


Figure 12: Clicking the button causes it to expand into a rationale banner.



(a) Alongside a browser prompt.

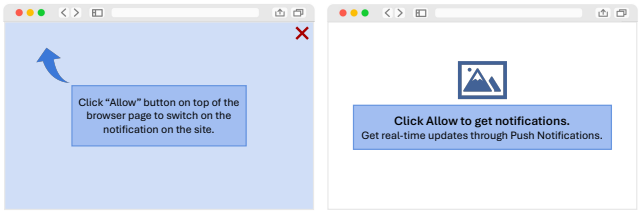
(b) After permission is not granted.

Figure 13: Banners.

Particularly for notification permissions, clicking a button can trigger the display of a rationale message. We observed several

variations in presenting this rationale. For instance, clicking a button with a notification-related icon, such as a bell, may display a rationale dialog either in the center of the screen or as a floating element above the button, as depicted in Figure 11. Alternatively, the button may expand horizontally to form a banner that contains the rationale message, as in Figure 12.

Banner Rationales in the form of a banner appear dynamically when awaiting user interaction with a permission prompt (see Figure 13a) or after permission is not granted (Figure 13b). In the latter case, they are usually displayed on a prominently colored banner, often red, featuring exclamation mark icons, distinct typography, and variably colored text.

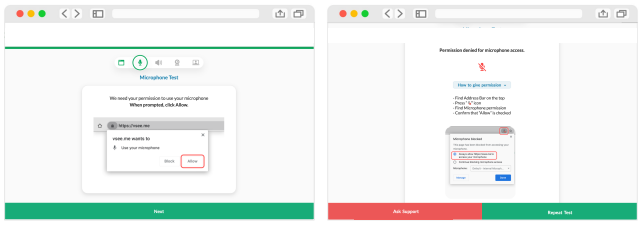


(a) Overlay. (b) Fullscreen.

Figure 14: Rationales alongside a browser prompt.

We found that rationales may include a link to a help or troubleshooting page to assist users in re-enabling permissions. Banners are often displayed inline with the webpage content but can also float above it, spanning the full width of the screen, mostly at the top as a header and occasionally at the bottom as a footer.

Overlay. Rationales in the form of overlays appear on top of the main content of a webpage, often dimming or obscuring the background to draw the user's focus to the rationale. These overlays frequently guide the user to the browser prompt with an arrow, as depicted in Figure 14a. While most overlays can be dismissed with an *"X"* button, we also observed instances where the overlay remains persistent until the user interacts with the browser prompt or re-enables a previously blocked permission. For the 'quieter' notification permission [24], a rationale overlay can point to the address bar, directing the user on how to enable notifications.



(a) Before browser prompt. (b) After permission is not granted.

Figure 15: Fullscreen rationales.

Fullscreen. When a fullscreen rationale is displayed alongside a browser prompt, it operates like an overlay rationale but with a solid background. This rationale prompts the user to click *"Allow"*. Similar to overlays, the user must take action to proceed since the current only contains the rationale, as illustrated in Figure 14b.

When the fullscreen rationale is shown before a browser prompt, it is typically part of a multi-step process to access a specific functionality. Users first grant permission, after which they can use the permission-protected feature. For instance, the fullscreen rationale in Figure 15a prepares the user for an upcoming browser prompt that appears when they click the button labeled *"Next"*. Other button labels may include *"Enable Location"* or *"Get Started"*.

If permission is not granted, the rationale prompts the user to grant permission and try again. Similar to rationales shown post-non-granting, it may include a *"Try Again"* button or a link to a help page. Instructions, often in the form of steps or screenshots, guide users on how to grant the necessary permission (Figure 15b).

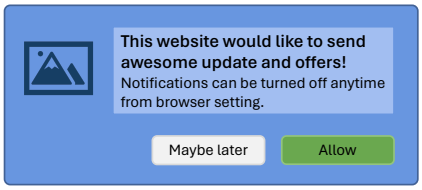
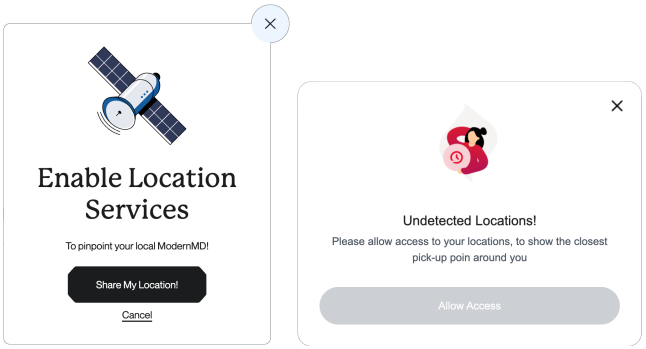


Figure 16: Example of a notification permission dialog displayed at the top center of the screen before browser prompt.

Dialog. Dialog rationales are typically centered on the screen, either in the middle or, particularly for notification permissions, at the top. Dialogs overlay the webpage content, which can be darkened to provide emphasis. Figure 16 shows a common notification permission dialog.



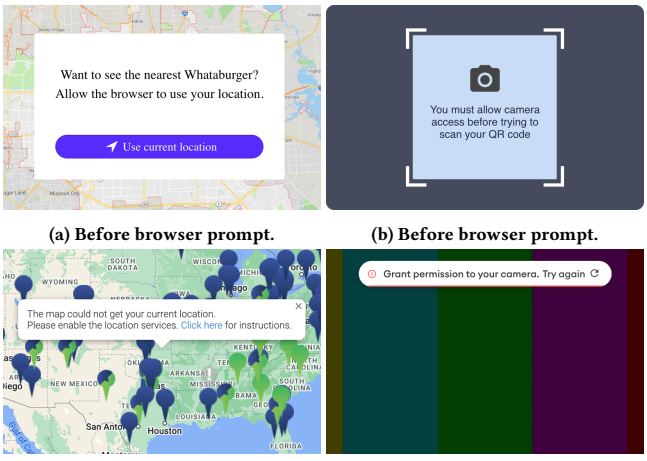
(a) Before browser prompt. (b) After permission is not granted.

Figure 17: Dialogs.

Dialogs presented before the browser prompt generally have an explicit button for granting permission, such as *"Allow Location"*, *"Detect My Location"*, or *"Give Access to Camera and Mic"*. In addition, they may include a dismiss button, such as *"OK"*, *"Got It,"* or an *"X"* button. Figure 17a shows a rationale dialog before the browser prompt. We observed that dialogs are also used to inform users when permission is required but has been denied. In these cases, dialogs may feature an icon indicating that permission is missing, as shown in Figure 17b.

	Geolocation		Notification		Camera		Microphone		Cam & Mic	
	Before	*After*	*Before*	*After*	*Before*	*After*	*Before*	*After*	*Before*	*After*
Count	**183**	**204**	**119**	**30**	**51**	**37**	**33**	**20**	**35**	**37**
Percent	47%	53%	80%	20%	58%	42%	62%	38%	49%	51%
Text	8.8%	13.7%	7.4%	–	35.2%	11.4%	30.2%	5.7%	12.5%	<5.0%
Button	18.9%	<5.0%	<5.0%	–	8.0%	<5.0%	17%	7.5%	9.7%	–
Banner	<5.0%	12.7%	<5.0%	<5.0%	<5.0%	5.7%	<5.0%	9.4%	–	6.9%
Dialog	10.6%	14.2%	42.9%	8.7%	<5.0%	10.2%	<5.0%	<5.0%	9.7%	18.1%
Fullscreen	<5.0%	<5.0%	21.5%	<5.0%	8.0%	9.1%	11.3%	11.3%	9.7%	13.9%
Overlay	<5.0%	<5.0%	<5.0%	6.7%	<5.0%	<5.0%	–	–	5.6%	<5.0%
On Protected	<5.0%	<5.0%	–	–	<5.0%	<5.0%	–	<5.0%	<5.0%	6.9%
Side of Map	–	6.2%	–	–	–	–	–	–	–	–
Alternative	36.4%		–		<5.0%		<5.0%		<5.0%	

Table 8: An overview of rationale layouts per permission type shows the distribution of rationale UI patterns before or alongside a browser prompt and after permission has not been granted.



(a) Before browser prompt. **(b) Before browser prompt.**

(c) After permission is not granted. **(d) After permission is not granted.**

Figure 18: Rationales on permission-protected content. (a) and (c) show permission on map. (b) and (d) show permission on camera feed.

On Permission-Protected Content. In previous patterns, we found that rationales can appear as floating elements on webpages, such as banners or dialogs. Additionally, they can overlay or replace permission-protected content, especially for geolocation and camera permissions, covering maps or camera/video feeds.

When displayed before or alongside a browser prompt, these rationales ask the user to grant the necessary permission, as illustrated in Figures 18a and 18b. Conversely, Figures 18c and 18d show a rationale over protected content when permission is not granted.

Side of Map. In the context of geolocation permission, a rationale is displayed on the sidebar of a map. Typically positioned on the right-hand side, this sidebar is utilized to present a list of nearby places, a feature that becomes inaccessible when the permission is denied. In such cases, the rationale message informs the user that *"current location could not be determined."* However, users are often provided with alternatives. They may be prompted to explore all locations by clicking a button labeled *"Show All Locations,"* or they can manually search for a specific location using the search bar.

Common UI Patterns. Table 8 shows the distribution of rationale layouts both before or alongside a browser prompt and following not granting permission. In total, we clustered 387 rationales for geolocation, 149 for notification, 88 for camera, 53 for microphone, and 72 for both camera and microphone permissions across the eight distinct layout patterns described above. We found that geolocation rationale UIs often include an alternative option with common layouts such as buttons, dialogs, text, and side-of-map rationales. Notification rationales typically appeared before or alongside a browser prompt, primarily as dialogs or fullscreens without alternatives. Camera and camera & microphone rationales were often presented as text, followed by dialogs and fullscreens. For Microphone permissions, text and buttons were the most frequent layouts.

Summary of Insights. Our analysis of rationale UIs reveals eight common layout patterns: *text*, *buttons*, *banners*, *overlays*, *fullscreens*, *dialogs*, *on-permission-protected content*, and *side-of-map*. Each pattern serves distinct purposes, such as presenting static or dynamic messages (text), encouraging action (buttons), or offering alternatives (e.g., for geolocation or camera permissions). Patterns like banners, dialogs, and overlays are often used dynamically to emphasize missing permissions, while fullscreens guide user interactions through focused layouts. Different permissions, such as geolocation, notification, and camera, influence the choice and frequency of these patterns.

9 Exploring the Effect of Rationales on User Decision-Making

This section presents an overview of our exploratory analysis of how text attributes and UI patterns in rationales influence user actions on permission prompts. As detailed in §5.4, we use the action rates from Chrome telemetry and user sentiment data from Chrome experience sampling.

	Grant Rate *Estimate (SE)*	**Deny Rate** *Estimate (SE)*	**Dismiss Rate** *Estimate (SE)*	**Ignore Rate** *Estimate (SE)*
Intercept	0.25 (0.01)***	0.10 (0.00)***	0.36 (0.01)***	0.28 (0.01)***
Permission				
Camera	0.37 (0.01)***	-0.06 (0.01)***	-0.16 (0.01)***	-0.15 (0.02)***
Microphone	0.43 (0.01)***	-0.07 (0.00)***	-0.20 (0.01)***	-0.16 (0.01)***
Notification	-0.24 (0.01)***	0.00 (0.00)	-0.06 (0.01)***	0.30 (0.01)***
Message Tone				
Negative	0.05 (0.02)*	0.00 (0.01)	-0.05 (0.02)**	0.00 (0.02)
Neutral	0.08 (0.01)***	0.02 (0.00)**	-0.05 (0.01)***	-0.04 (0.01)***
Positive	0.18 (0.03)***	-0.02 (0.01)	-0.06 (0.02)*	-0.11 (0.03)**
Encouragement				
Motivation	-0.03 (0.01)	0.02 (0.01)***	0.00 (0.01)	0.01 (0.01)
Consequences	-0.20 (0.08)**	0.01 (0.03)	0.12 (0.06)*	0.07 (0.08)
Permission Necessity				
Required	0.08 (0.02)***	-0.01 (0.01)	-0.03 (0.02)	-0.05 (0.02)*
Optional	-0.28 (0.04)	0.01 (0.01)	-0.02 (0.03)	0.03 (0.04)
Message Content				
Error	0.03 (0.01)	0.00 (0.01)	-0.01 (0.01)	-0.01 (0.02)
Instruction	0.00 (0.01)	0.01 (0.00)***	-0.01 (0.01)	0.00 (0.01)
Func. Explanation	0.04 (0.01)**	0.02 (0.00)***	-0.01 (0.01)	-0.04 (0.01)***
Loading Device	0.11 (0.05)*	0.01 (0.02)	-0.06 (0.03)	-0.07 (0.05)
Data Use Reassurance	0.11 (0.03)***	-0.02 (0.01)*	-0.06 (0.02)	-0.03 (0.03)
Permission Request	0.30 (0.01)***	-0.02 (0.00)***	0.00 (0.01)	-0.01 (0.01)
Emphasize Control	0.06 (0.02)**	0.01 (0.01)	0.00 (0.02)	-0.08 (0.02)***

Table 9: Results of four exploratory regression models using the four user action rates in permission prompts as dependent variables. Independent variables include the permission type as well as the identified rationale text attributes. Reference categories are geolocation for permission type, as well as "none" (i.e., no rationale text being present at all) for tone and encouragement. The remaining factors are binary, i.e. encode whether or not the given content type was present or not. Legend: SE = Standardized Error. * p <.05, ** p <.001, *** p <.0001.

9.1 Analysis of Rationale Text Attributes

Table 6 provides an overview of the action rates across the various features of rationale texts we identified and Table 9 describes the regression models we fitted to explore the effects for each of the attributes we identified.

9.1.1 The Effect of Text Attributes on Permission Prompt Actions. Overall, our analysis suggests user behavior on permission prompts is influenced by how rationale messages are composed, primarily on grant rates. We detail the effects in the following paragraphs.

Insight #1: Message Tone. First and foremost, having any rationale message, even with a neutral or negative tone, positively influences the grant rate and is associated with lower dismiss and deny rates. Neutral and in particular positive tones in rationale texts are associated with higher grant rates (+8% and 18%, respectively).

Insight #2: Encouragement. The use of consequences as an encouragement strategy significantly decreases the grant rate (-20%). This attribute is also associated with a 12% higher dismiss rate. Highlighting benefits and motivations seems to have a small negative impact, with slightly lower grant rates (-3%) and slightly higher deny rates (+2%).

Insight #3: Permission Necessity. When functionality is stated as required in the rationale text, the grant rate is somewhat higher (+8%) and prompts get ignored slightly less frequently (-5%). Mentioning optional and alternative functionalities does not appear to have significant effects on user actions.

Insight #4: Message Content. Including most of the additional content into rationale messages leads to higher grant rates (+3% to +30%). In particular, the more neutral and very prevalent request for permission content type is identified as having the highest positive impact in our sample. Reassurance about use of the collected data and a notice about a delay because of the device also show substantial increases in grant rates (+11%), yet our dataset only included a smaller number of examples for these types of messages.

9.1.2 The Effect of Text Attributes on User Sentiment. We also ran logistic regression models on the user sentiment dataset as described in §5.4. These models did not yield any significant effects for any of the rationale text attributes on either permission prompts feeling annoying or being easy to make a decision on.

9.2 Analysis of Rationale UI Patterns

For the following analysis, we categorized the identified rationale patterns into two groups: those displayed before or alongside a browser prompt and those shown after permission is not granted (i.e., dismissed, ignored, or denied). This is a central distinction as some users may never see the rationales shown after the interaction with the prompt. It should also be noted that some rationales are

	Grant *Estimate (SE)*	**Deny** *Estimate (SE)*	**Dismiss** *Estimate (SE)*	**Ignore** *Estimate (SE)*	**Not Annoy.** *Odds R. (SE)*	**Is Easy** *Odds R. (SE)*
Intercept	0.32 (0.02)***	0.11 (0.01)***	0.32 (0.01)***	0.24 (0.02)***	6.89 (1.40)***	3.35 (1.31)***
Permission						
Camera	0.27 (0.04)***	-0.06 (0.01)***	-0.14 (0.03)***	-0.07 (0.04)	5.05 (2.39)	0.66 (1.65)
Microphone	0.26 (0.05)***	-0.06 (0.01)***	-0.16 (0.03)***	-0.04 (0.05)	2.20 (2.03)	0.97 (1.73)
Notification	-0.30 (0.03)***	0.00 (0.01)	-0.01 (0.02)	0.31 (0.03)***	–	–
Before						
Banner	0.08 (0.12)	0.02 (0.03)	0.04 (0.07)	-0.14 (0.12)	–	–
Button	0.06 (0.05)	-0.01 (0.01)	-0.06 (0.03)*	0.00 (0.05)	2.86 (3.06)	1.12 (1.82)
Dialog	0.18 (0.04)***	-0.03 (0.01)*	-0.07 (0.02)**	-0.08 (0.04)*	0.36 (1.58)*	0.32 (1.46)**
Fullscreen	0.33 (0.08)***	-0.08 (0.02)**	-0.20 (0.05)***	-0.05 (0.08)	0.03 (6.23)	–
On Protected	0.20 (0.10)	-0.04 (0.03)	-0.15 (0.06)*	-0.01 (0.10)	0.15 (3.19)	–
Overlay	0.41 (0.08)***	-0.06 (0.02)*	-0.19 (0.05)***	-0.16 (0.08)	0.17 (2.92)	1.04 (3.29)
Text	0.19 (0.04)***	-0.01 (0.01)	-0.08 (0.02)**	-0.11 (0.04)**	0.21 (1.90)*	0.68 (1.57)
After						
Banner	0.15 (0.04)***	-0.03 (0.01)*	-0.02 (0.03)	-0.10 (0.04)*	0.53 (1.84)	1.06 (1.48)
Button	0.23 (0.11)*	-0.04 (0.03)	0.01 (0.07)	-0.21 (0.11)	0.05 (4.90)	0.58 (2.36)
Dialog	0.15 (0.04)***	-0.02 (0.01)*	-0.05 (0.02)*	-0.07 (0.04)	0.23 (1.67)**	0.45 (1.55)
Fullscreen	-0.06 (0.09)	0.00 (0.03)	0.08 (0.06)	-0.03 (0.09)	29.08 (7.32)	–
On Protected	0.15 (0.07)*	-0.03 (0.02)	-0.01 (0.04)	-0.12 (0.07)	1.13 (2.16)	0.57 (1.77)
Overlay	0.08 (0.09)	-0.03 (0.03)	-0.04 (0.06)	-0.01 (0.09)	0.18 (5.16)	0.61 (4.90)
Side of Map	0.15 (0.07)*	-0.03 (0.02)	-0.05 (0.04)	-0.07 (0.07)	0.20 (2.69)	0.63 (2.12)
Text	0.09 (0.04)*	-0.04 (0.01)**	-0.03 (0.03)	-0.03 (0.04)	0.68 (3.13)	0.33 (2.14)
Additional						
Prompt	-0.09 (0.07)	0.02 (0.02)	0.00 (0.04)	0.06 (0.07)	1.79 (3.00)	1.40 (2.46)
Alternative	-0.09 (0.04)*	0.02 (0.01)	0.02 (0.02)	0.05 (0.04)	2.89 (2.01)	0.71 (1.67)

Table 10: Results of six exploratory regression models using the four user action rates in permission prompts, user annoyance, and ease of decision-making as dependent variables. Independent variables include the permission type as well as the identified UI rationale clusters. Reference categories are geolocation for permission type, as well as "none" (i.e., no rationale being present at all) for before browser prompt and after permission denial. The additional factors are binary, i.e. encode whether or not the given element was present or not. Not Annoy. = Not Annoying. Legend: SE = Standardized Error. Odds R. = Odds Ratio. Before = Before browser prompt. After = After permission denial. * p <.05, ** p <.001, *** p <.0001.

consistently displayed throughout the permission request cycle, remaining static. This particularly applies to text embedded within the main content of the webpage (referred to as *Before: Text* in Table 10) or to buttons that trigger a permission-protected function (referred to as *Before: Button*).

We also introduced two additional variables: *Alternative*, indicating whether the rationale includes an alternative to granting permission, and *Prompt*, indicating whether the rationale is displayed at the same time as the browser prompt. We then conducted a logistic regression on the patterns, as detailed in Table 10, using the methodology outlined in §5.4.

9.2.1 The Effect of UI Patterns on Permission Prompt Actions. Similarly to the rationale texts, our exploratory analysis of rationale UIs showed a consistent pattern: the presence of rationales generally increases grant rates while reducing deny, ignore, and dismiss rates. The primary distinction between different rationale patterns lies in the magnitude of their effects.

Insight #1: Timing. Rationales presented before or alongside a browser prompt had a stronger impact than those shown after permission was not granted. In this context, overlays resulted in the highest increase in grants (+41%), followed by fullscreens (+33%), text (+19%), and dialogs (+18%). The largest reductions in deny rates were observed with fullscreen prompts (-8%), followed by overlays (-6%), before or alongside a browser prompt. Similar effects were noted for dismiss rates, with fullscreen (-20%), overlays (-19%), and rationales on permission-protected content (-15%) significantly decreasing the likelihood of dismissing a permission request. Regarding ignore rates, significant reductions were seen with text (-11%) and dialogs (-8%) presented before or alongside prompts, and banners shown after permission was not granted (-10%).

Insight #2: Actionable Buttons. For websites that offer rationales after permission was not granted, the button layout resulted in the highest increase in grant rates (+23%) by offering users an actionable option to grant permission after experiencing the site without the requested permission.

Insight #3: Alternative. When a rationale offered users an alternative option to granting permission, the likelihood of users granting the requested permission decreased by 9%.

9.2.2 The Effect of UI Patterns on User Sentiment. In analyzing the user experience data, we found that only dialogs and text significantly impacted user perception. Users were more likely to report an increase in annoyance when a rationale was presented as a

dialog before a browser prompt and after. Text before a prompt was also associated with increased annoyance. Additionally, dialogs before a prompt made it less likely that respondents rated the decision-making process as somewhat or very easy.

10 Summary and Discussion

We discuss threats to the validity, summarize our main findings, and outline their broader implications.

10.1 Threats to Validity

We relied on web crawling to collect snapshots of webpages and their associated permission rationales. However, crawling is a challenging task [13, 55] and we may have missed pages containing permission rationales, such as rationales behind user authentication and those presented exclusively to specific geographical regions or specific web clients like mobile browsers. Furthermore, we focused on rationales based on English text. Consequently, our findings likely represent a lower-bound estimate of rationale prevalence on the web and may have missed mobile-specific, geo-specific, and patterns that depend on more complex user interactions.

In addition, to assess the effects of rationales on user decisions, we relied on Chrome telemetry and user sentiment data. However, telemetry data may not always correspond to the specific rationales we identified in webpages. Also, these were collected almost 1.5 years after our data collection, posing risks that some webpages may have changed in the meantime. Future research could build on our work and address these challenges by integrating more advanced crawling techniques and by conducting controlled and longitudinal studies to capture evolving web content over time.

Furthermore, future work could investigate whether libraries diverge significantly from the observed patterns and effects, and assess how their rationale text and design influences permission decisions. Our dataset does not allow us to fully isolate library and custom rationales at scale, as webpages may use both or customize library rationales, which complicates analyzing their distinct effects.

Then, we only assess how rationales observed on websites in the wild impact user behavior and sentiment. Therefore, we have a limited number of website samples and user sentiment responses for each of the various dimensions we identified. The websites in our sample also span a wide variety of use cases, given the differing nature of the most common permission-gated web APIs. It is therefore likely that properties of these use cases as well as other aspects such as brand reputation influenced the efficacy of the rationales we found. A controlled experiment across the dimensions we identified is necessary to more rigorously establish which types of rationales are truly effective. Such an experiment should also include a more thorough evaluation of user sentiment towards such rationales, given that our sentiment dataset was limited to only two of many plausible measures.

Finally, a validity threat may stem from users who never saw Chrome's permission prompt due to lack of interaction with the webpage rationale, leaving them unaccounted for in Chrome telemetry and experience sampling responses. This exclusion may bias the reported action rates and sentiment proportions, further highlighting the need for controlled experiments.

10.2 Open Science

To support future research, we have made our catalogs of rationale text and UI publicly accessible [15].

10.3 Web Permission Rationales

Detection Technique and Rationale Catalog. We present the first approach to systematically detect and study web permission rationales at scale, leveraging interactive web crawling, advanced semantic capabilities of LLMs, and BERT classification models. We instantiated our system against 779K webpages and created a comprehensive catalog of 3.6K unique, manually-vetted rationale text samples and 749 UIs. Furthermore, we found that the most common rationale messages are associated with 10 specific libraries, for which we developed 32 code signatures and HTML detection rules.

Status-Quo and Prevalence. Our automated crawling observed over 1.6M permission API calls on the surface of web applications, accounting for over 162K webpages and 29.1K sites, with the majority belonging to geolocation and notification permission prompts. In total, our ML-based rationale detection pipeline, combined with mining of library signatures, identified over 85K webpages that present either a custom or library rationale.

10.4 The Effect of Rationales on User Decision-Making

We present and discuss the key insights from our study on how rationale text and UI elements influence user decisions regarding web permission prompts.

Insight #1: Timing is Everything–Early Rationales Drive Grants. Our results show that the timing of permission rationales is critical in influencing user decisions, which is in line with prior findings for rationales in mobile apps [16]. Rationales presented before or alongside a browser prompt significantly increase the likelihood of users granting permissions. Overlays displayed at this stage resulted in the highest boost in grant rates (+41%), followed by fullscreen rationales (+33%). These early interventions effectively set the stage for a positive user response, underscoring the importance of timing in permission request strategies. At the same time, those most effective rationales take up a large part of the screen, so can feel very heavy handed and might not be suitable for all types of permission use cases, especially when a capability is not a central part of the user journey.

Insight #2: Second Chances–Post-Prompt Buttons Can Be Helpful. Interestingly, we found that offering users actionable options after not granting a permission can substantially increase the likelihood of grants overall. Buttons presented in such situations were associated with 23% higher grant rates, giving users a second chance to reconsider their decision after experiencing the site without the requested permission. This finding highlights the value of providing users with a clear path to revisiting their initial choices.

Insight #3: Consequence-Based Messaging Less Effective. Our findings indicate that consequence-based rationales–those that emphasize what users stand to lose if they do not grant permission–are less effective and can even backfire. These strategies were associated with a 20% decrease in grant rates and a 12% increase in dismissals

in our dataset. This suggests that emphasizing negative outcomes may undermine user trust and lead to resistance rather than compliance. Encouragement strategies should, therefore, focus on positive reinforcement rather than fear-based tactics.

Insight #4: Balancing User Annoyance and Effectiveness. While many rationales effectively increased grant rates, we also observe potentially unintended consequences on user experience. For example, text and dialog rationales presented before or alongside prompts were generally effective in increasing grant rates (+19% and +18%, respectively) and reducing dismiss rates. However, the dialogs included in our sample were also associated with higher levels of user annoyance. Similarly, text presented before a prompt made it more likely that user rated the decision-making process as more challenging. These findings suggest that while rationales are crucial for guiding user behavior, the challenge lies in balancing effectiveness with user satisfaction, ensuring that rationales are both persuasive and user-friendly.

Insight #5: Clarity on Essential Functions Drives Compliance. Our study also suggests that clearly stating the necessity of a functionality in the rationale text significantly improves grant rates (+8%) and reduces the likelihood of users ignoring the prompt (-5%). Users respond positively when they understand that granting a permission is essential for the core functionality of the site, as already posited by prior work [23]. This highlights the importance of clear, direct communication in rationale messages, particularly when the permission is crucial for the website's operation.

Insight #6: Message Tone Matters. Including a rationale message, even with a neutral or negative tone, improves grant rates and reduces dismiss and deny rates. Neutral tones increase grants by 8%, while positive tones increase them by 18%.

Insight #7: Effective Encouragement. Encouragement strategies play a critical role, with the use of consequences decreasing grant rates by 20% and raising dismiss rates by 12%. Conversely, highlighting benefits and motivations has a slight negative impact, lowering grant rates by 3% and increasing deny rates by 2%.

Insight #8: Functionality Requirements. Stating that functionality is required in the rationale results in an 8% higher grant rate and a 5% reduction in ignored prompts. Mentioning optional functionalities in text, however, does not significantly affect user actions. Yet, when there is an alternative interaction available, we found that grant rates are reduced by 9% in our UI-based analysis. This highlights that some users prefer alternative options when offered.

Insight #9: Message Content Impact. Providing additional context in rationale messages increases grant rates by 3% to 30%. Neutral permission requests have the strongest effect, while reassurances on data use and and notifications about device delays increase grant rates by 11%, though they were less common in our dataset.

10.5 Differences in Permission Requests Between Desktop Web and Android

In this study, we focused on permission experiences on the web when using desktop platforms, while prior work primarily addressed mobile app permissions. When comparing permission requests between the desktop web and mobile apps, web prompts offer users more interaction options. Users can grant or deny permissions, as well as ignore or dismiss requests. In contrast, Android or iOS permission prompts are inherently blocking, meaning the app's execution is paused until the user responds. Dismissing a prompt is loosely comparable to using the back button on Android while it is impossible on iOS.

Both platforms exhibit similar rationale layouts, but Android's blocking prompts introduce distinct differences. On the desktop web, rationales can be displayed alongside permission requests due to the larger screen sizes, whereas on Android, they are shown either before or after a prompt. Additionally, limited screen space on Android made it less common to include supplementary content next to a rationale. For example, previous studies [14] did not encounter rationales placed beside maps or inline text and banners, which typically appeared after permission was not granted.

Interestingly, when comparing our findings with previous work [14], the phrasing and content of rationales across the two platforms were largely consistent. Most rationales aimed to encourage users to grant permissions, with 50.9% doing so on the desktop web compared to 67.0% on Android. A smaller proportion provided guidance (7.7% on the desktop web vs. 24.0% on Android). On both platforms, rationales emphasized the benefits of granting permissions to motivate users, with 10.6% of desktop web rationales and 18.0% of Android rationales highlighting this aspect. Less common themes, such as privacy assurances (1.6% vs. 2.0%) and alternative options (0.9% vs. 2.0%), followed similar trends. The main differences were in terminology, reflecting platform-specific contexts. For example, web rationales referred to *"websites"* and *"browser settings,"* while Android rationales mentioned *"apps"* and *"app settings."* A web rationale might state, *"To start your webcam, you need to allow our website to use it,"* whereas an Android rationale might say, *"We need access to your camera for the app to function properly."*

Finally, some capabilities, like geolocation, vary in usefulness depending on the attributes of devices typically used on the respective platforms. This influences how developers approach permission requests and design their features. For example, websites accessed mainly on desktop devices might provide alternative ways to locate the nearest store, as desktops often lack GPS sensors and provide lower-quality location data. Desktops also tend to stay in one place, making frequent location updates less relevant. In contrast, mobile devices like smartphones almost always have GPS sensors, offering high-quality location data and supporting features that rely on frequent updates, such as navigation.

10.6 Rationales and Dark Patterns

Dark patterns refer to design strategies in user interfaces that manipulate or deceive users into making decisions that may not align with their best interests [10]. These patterns exploit cognitive biases, making it harder for users to choose desirable options while subtly promoting unfavorable ones. We observed a tendency for such patterns in permission rationales, where the design may not fully qualify as a dark pattern but appears to nudge users toward a specific choice.

In our investigation of rationales, we noticed that websites often emphasize the "Allow" button, a practice also observed in Android rationales [14]. Additionally, visual cues such as arrows pointing to

the "Allow" button in browser prompts, as shown in Figure 14a, can serve as nudges. Attention icons may also draw focus to rationales, creating a sense of urgency. A particularly interesting case we found involved a rationale with a countdown timer, which caused the rationale to disappear after a set time. While it is not definitively a dark pattern, the countdown could influence users to make quick, potentially uninformed decisions. These observations highlight the importance of understanding these patterns to evaluate the ethical implications of such practices and to design user interfaces that support informed decision-making.

10.7 Decoupling Rationale Detection from Permission Prompts

Our crawler triggers about 20% of permission prompts in the Chrome telemetry dataset—a 100% improvement over non-interactive agents via our interaction heuristics (Appendix A.1). This rate should not be mistaken for successfully identified rationales, the study's primary focus. Many applications present prompts without any rationale. Furthermore, since the ML pipeline relies on rationale text, it can detect rationales whenever the text is present in the DOM of webpages, regardless of whether the crawler triggers the prompt or not. In contrast, when rationales only appear in the DOM after triggering a prompt and the crawler cannot simulate the required user interactions, our approach will miss them. Quantitatively speaking, in 1000 random pages from telemetry, 113 had rationales following manual analysis, of which only 19 required user interaction for authentication. Accordingly, the crawler can detect rationales for over 80% of the pages presenting a rationale. Our results only provide a lower bound on the prevalence of rationales on the web. We refer interested readers to Appendix A.2 for more details.

10.8 ML-based Rationale Detection and Future Work

Our study provides a systematic ML-based framework for detecting and analyzing permission rationales, offering a strong foundation for similar large-scale studies. For example, future work can use our tool and dataset of permission rationales to design and create more powerful rationale classifiers. While our methodology is reusable, we acknowledge areas for further development and encourage future research to build upon our findings.

Future research should explore the integration of complementary methods, such as signature-based detection of third-party libraries, to capture rationales not easily identified by text-based ML approaches. Expanding the method to automatically detect non-textual rationale UIs (e.g., image processing) and incorporating multimodal learning could further enhance detection capabilities.

To advance towards a fully automated process, researchers should also focus on integrating LLM-based crawlers able to complete tasks, such as simulating complex user actions [56], which could improve rationale coverage. While our pipeline effectively detects DOM-present rationales, manual analysis was necessary for validating the context of extracted sentences (e.g., distinguishing between tutorial text and true rationales). Future research should investigate automated methods, such as more powerful LLMs, to discern the context of potential rationales, enhancing the accuracy of the rationale detection pipeline.

10.9 Concluding Remarks

In this study, we adopted a predominantly quantitative approach to identify and analyze web permission rationales in the wild. Our findings indicate that web rationales do influence user behavior, though a complete list of possible effects is still unknown. To gain a more complete understanding, additional qualitative studies and controlled experiments are needed. For example, investigating the role of third-party libraries and their rationales is crucial for a thorough assessment of rationales' influence on user decisions. Similarly, more rigorously establishing the effects of rationale presentation based on the patterns we identified needs additional, controlled studies. In the meantime, our findings already provide actionable insights for researchers and practitioners into the diverse rationales present in the web ecosystem and their attributes.

References

[1] Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review* 54 (2021), 5789–5829.

[2] Matthew Finifter Devdatta Akhawe Adrienne Porter Felt, Serge Egelman and Berkeley David Wagner, University of California. 2012. How to Ask for Permission. In *Proc. 7th USENIX Workshop on Hot Topics in Security (HotSec'12)*.

[3] Mistral AI. 2024. *Mistral 7B Model.* Retrieved November 21, 2024 from https://mistral.ai/news/announcing-mistral-7b.

[4] Hazim Almuhimedi, Florian Schaub, Norman M. Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. 2015. Your Location has been Shared 5, 398 Times!: A Field Study on Mobile App Privacy Nudging. In *Conference on Human Factors in Computing Systems (CHI'15)*.

[5] Apple. 2024. *App Store review policy – privacy.* Retrieved November 21, 2024 from https://developer.apple.com/app-store/review/guidelines/#privacy.

[6] Pieter Arntz. 2024. *Browser push notifications: a feature asking to be abused.* Retrieved November 21, 2024 from https://blog.malwarebytes.com/security-world/technology/2019/01/browser-push-notifications-feature-asking-abused.

[7] Igor Bilogrevic, Balazs Engedy, Judson L Porter III, Nina Taft, Kamila Hasanbega, Andrew Paseltiner, Hwi Kyoung Lee, Edward Jung, Meggyn Watkins, PJ McLachlan, et al. 2021. "Shhh... be quiet!" Reducing the Unwanted Interruptions of Notification Permission Prompts on Chrome. In *Proc. 30th USENIX Security Symposium (SEC'21)*.

[8] Kerstin Bongard-Blanchy, Jean-Louis Sterckx, Arianna Rossi, Verena Distler, Salvador Rivas, and Vincent Koenig. 2022. An (Un)Necessary Evil - Users' (Un)Certainty about Smartphone App Permissions and Implications for Privacy Engineering. In *Proc. 7th IEEE European Symposium on Security and Privacy (EuroS&P'22)*.

[9] Bram Bonné, Sai Teja Peddinti, Igor Bilogrevic, and Nina Taft. 2017. Exploring decision making with Android's runtime permission dialogs using in-context surveys. In *Proc. 13th Symposium on Usable Privacy and Security (SOUPS'17)*.

[10] Harry Brignull. 2024. *Deceptive Patterns.* Retrieved November 21, 2024 from https://www.deceptive.design.

[11] Weicheng Cao, Chunqiu Xia, Sai Teja Peddinti, David Lie, Nina Taft, and Lisa M. Austin. 2021. A Large Scale Study of User Behavior, Expectations and Engagement with Android Permissions. In *Proc. 30th USENIX Security Symposium (SEC'21)*.

[12] Ian Clelland. 2024. Permissions Policy. *W3C Working Draft* (2024). Retrieved November 21, 2024 from https://www.w3.org/TR/permissions-policy.

[13] Adam Doupé, Ludovico Cavedon, Christopher Kruegel, and Giovanni Vigna. 2012. Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner. In *Proc. 21st USENIX Security Symposium (SEC'12)*.

[14] Yusra Elbitar, Alexander Hart, and Sven Bugiel. 2025. The Power of Words: A Comprehensive Analysis of Rationales and Their Effects on Users' Permission Decisions. In *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*.

[15] Yusra Elbitar, Soheil Khodayari, Marian Harbach, Gianluca De Stefano, Balazs Csaba Engedy, Giancarlo Pellegrino, and Sven Bugiel. 2024. *Catalogs of rationale text and UI.* Retrieved November 21, 2024 from https://osf.io/6cqds/?view_only=dddc898e5d4f4a93b80c9a19898c0cb5.

[16] Yusra Elbitar, Michael Schilling, Trung Tin Nguyen, Michael Backes, and Sven Bugiel. 2021. Explanation Beats Context: The Effect of Timing & Rationales on Users' Runtime Permission Decisions. In *30th USENIX Security Symposium (USENIX Security 21)*.

[17] Hugging Face. 2024. *BERT Base Model.* Retrieved November 21, 2024 from https://huggingface.co/google-bert/bert-base-uncased.

[18] Hugging Face. 2024. *Sentence Transformer all-MiniLM-L6-v2*. Retrieved November 21, 2024 from https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[19] Adrienne Porter Felt, Serge Egelman, and David Wagner. 2012. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proc. Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'12)*. 33–44.

[20] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proc. 8th Symposium on Usable Privacy and Security (SOUPS'12)*. 1–14.

[21] Eduardo C Garrido-Merchan, Roberto Gozalo-Brizuela, and Santiago Gonzalez-Carvajal. 2023. Comparing BERT against traditional machine learning models in text classification. *Journal of Computational and Cognitive Engineering* 2 (2023), 352–356.

[22] Google. 2024. *Google Play console help - Declaring permissions for your app.* Retrieved November 21, 2024 from https://support.google.com/googleplay/android-developer/answer/9214102.

[23] Marian Harbach. 2024. Websites Need Your Permission Too – User Sentiment and Decision-Making on Web Permission Prompts in Desktop Chrome. In *Conference on Human Factors in Computing Systems (CHI'24)*.

[24] Marian Harbach, Igor Bilogrevic, Enrico Bacis, Serena Chen, Ravjit Uppal, Andy Paicu, Elias Klim, Meggyn Watkins, and Balazs Engedy. 2024. Don't Interrupt Me - A Large-Scale Study of On-Device Permission Prompt Quieting in Chrome. In *31st Annual Network and Distributed System Security Symposium (NDSS'24)*.

[25] Marian Harbach, Markus Hettig, Susanne Weber, and Matthew Smith. 2014. Using personal examples to improve risk communication for security & privacy decisions. In *Conference on Human Factors in Computing Systems (CHI'24)*.

[26] Marian Harbach and Thomas Steiner. 2024. *Web permissions best practices.* Retrieved November 21, 2024 from https://web.dev/articles/permissions-best-practices.

[27] Mohammadreza Hazhirpasand, Mohammad Ghafari, and Oscar Nierstrasz. 2020. Tricking Johnny into Granting Web Permissions. In *Proc. 24th Evaluation and Assessment in Software Engineering (EASE'20)*.

[28] WP Hive. 2024. *Storerocket Library.* Retrieved November 21, 2024 from https://wphive.com/plugins/storerocket-store-locator.

[29] Ana Hoffman. 2024. *Smart Push Library.* Retrieved November 21, 2024 from https://www.smartpush.ai/blog/push-notifications.

[30] iZooto. 2024. *iZooto Library.* Retrieved November 21, 2024 from https://help.izooto.com/docs/enable-notifications-from-inside-newshub.

[31] Patrick Gage Kelley, Sunny Consolvo, Lorrie Faith Cranor, Jaeyeon Jung, Norman Sadeh, and David Wetherall. 2012. A Conundrum of Permissions: Installing Applications on an Android Smartphone. In *Proc. 16th International Conference on Financial Cryptography and Data Security (FC'12)*.

[32] Jialiu Lin, Bin Liu, Norman M. Sadeh, and Jason I. Hong. 2014. Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings. In *Proc. 10th Symposium on Usable Privacy and Security (SOUPS'14)*.

[33] Jialiu Lin, Norman M. Sadeh, Shahriyar Amini, Janne Lindqvist, Jason I. Hong, and Joy Zhang. 2012. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *ACM Conference on Ubiquitous Computing, (Ubicomp'12)*.

[34] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun (Aerin) Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti. 2016. Follow My Recommendations: A Personalized Privacy Assistant for Mobile App Permissions. In *Proc. 12th Symposium on Usable Privacy and Security (SOUPS'16)*.

[35] Bin Liu, Jialiu Lin, and Norman M. Sadeh. 2014. Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help?. In *Proc. 23rd International World Wide Web Conference (WWW'14)*.

[36] Xueqing Liu, Yue Leng, Wei Yang, Wenyu Wang, Chengxiang Zhai, and Tao Xie. 2018. A Large-Scale Empirical Study on Android Runtime-Permission Rationale Messages. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'18)*.

[37] Xueqing Liu, Yue Leng, Wei Yang, Chengxiang Zhai, and Tao Xie. 2018. Mining Android App Descriptions for Permission Requirements Recommendation. In *26th IEEE International Requirements Engineering Conference (RE'18)*.

[38] Kristopher Micinski, Daniel Votipka, Rock Stevens, Nikolaos Kofinas, Michelle L Mazurek, and Jeffrey S Foster. 2017. User Interactions and Permission Use on Android. In *Conference on Human Factors in Computing Systems (CHI'17)*.

[39] Microsoft. 2020. *Reducing distractions with quiet notification requests.* Retrieved November 21, 2024 from https://blogs.windows.com/msedgedev/2020/07/23/reducing-distractions-quiet-notification-requests.

[40] MoEngage. 2024. *Moe-push Library.* Retrieved November 21, 2024 from https://moengage.github.io/android-api-reference/moe-push-firebase/index.html.

[41] Mozilla. 2024. *Geolocation API.* Retrieved November 21, 2024 from https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API.

[42] Mozilla. 2024. *Restricting notification permission prompts in Firefox.* Retrieved November 21, 2024 from https://blog.mozilla.org/futurereleases/2019/11/04/restricting-notification-permission-prompts-in-firefox.

[43] Debjyoti Mukherjee, Alireza Ahmadi, Maryam Vahdat Pour, and Joel Reardon. 2020. An Empirical Study on User Reviews Targeting Mobile Apps' Security & Privacy. *arXiv:2010.06371* (2020). Retrieved from https://arxiv.org/abs/2010.06371.

[44] Duc Cuong Nguyen, Erik Derr, Michael Backes, and Sven Bugiel. 2019. Short Text, Large Effect: Measuring the Impact of User Reviews on Android App Security & Privacy. In *Proc. 30th IEEE Symposium on Security and Privacy (SP'19)*.

[45] Katarzyna Olejnik, Italo Dacosta, Joana Soares Machado, Kévin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. 2017. SmarPer: Context-Aware and Automatic Runtime-Permissions for Mobile Devices. In *Proc. 28th IEEE Symposium on Security and Privacy (SP'17)*.

[46] OneSignal. 2024. *OneSignal Library.* Retrieved November 21, 2024 from https://documentation.onesignal.com/docs/slide-prompt.

[47] Perfecty. 2024. *Perfecty Library.* Retrieved November 21, 2024 from https://perfecty.org.

[48] PushEngage. 2024. *Pushengage Library.* Retrieved November 21, 2024 from https://pushengage.com.

[49] pushowl. 2024. *PushOWL Library.* Retrieved November 21, 2024 from https://pushowl.com.

[50] Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, and Abdulwahab Ali Almazroi. 2022. A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. *Journal of healthcare engineering* 2022 (2022), 3498123.

[51] Leonard Richardson. 2015. *Beautiful Soup Library.* Retrieved November 21, 2024 from https://beautiful-soup-4.readthedocs.io.

[52] Kimberly Ruth, Deepak Kumar, Brandon Wang, Luke Valenta, and Zakir Durumeric. 2022. Toppling top lists: evaluating the accuracy of popular website lists. In *Proc. ACM Internet Measurement Conference (IMC'22)*.

[53] scikit learn. 2024. *Sklearn Agglomerative Clustering.* Retrieved November 21, 2024 from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.

[54] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. 2021. Can Systems Explain Permissions Better? Understanding Users' Misperceptions under Smartphone Runtime Permission Model. In *Proc. 30th USENIX Security Symposium, (SEC'21)*.

[55] Aleksei Stafeev and Giancarlo Pellegrino. 2024. SoK: State of the Krawlers-Evaluating the Effectiveness of Crawling Algorithms for Web Security Measurements. In *Proc. 33rd USENIX Security Symposium (SEC'24)*.

[56] Aleksei Stafeev, Tim Recktenwald, Gianluca De Stefano, Soheil Khodayari, and Giancarlo Pellegrino. 2025. YURASCANNER: Leveraging LLMs for Task-driven Web App Scanning. In *32nd Annual Network and Distributed System Security Symposium, (NDSS'25)*.

[57] Karthika Subramani, Xingzi Yuan, Omid Setayeshfar, Phani Vadrevu, Kyu Hyung Lee, and Roberto Perdisci. 2020. When Push Comes to Ads: Measuring the Rise of (Malicious) Push Advertising. In *Proc. ACM Internet Measurement Conference (IMC'20)*.

[58] SuperStoreFinder. 2024. *Superstorefinder-wp Library.* Retrieved November 21, 2024 from https://superstorefinder.net/superstorefinderwp.

[59] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. 2023. Stuck in the Permissions With You: Developer & End-User Perspectives on App Permissions & Their Privacy Ramifications. In *Conference on Human Factors in Computing Systems (CHI'23)*.

[60] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David A. Wagner. 2014. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Conference on Human Factors in Computing Systems (CHI'14)*.

[61] Christopher Thompson, Maritza Johnson, Serge Egelman, David Wagner, and Jennifer King. 2013. When it's better to ask forgiveness than get permission: attribution mechanisms for smartphone resources. In *Proc. 9th Symposium on Usable Privacy and Security (SOUPS'13)*.

[62] Lynn Tsai, Primal Wijesekera, Joel Reardon, Irwin Reyes, Serge Egelman, David A. Wagner, Nathan Good, and Jung-Wei Chen. 2017. Turtle Guard: Helping Android Users Apply Contextual Privacy Preferences. In *Proc. 13th Symposium on Usable Privacy and Security (SOUPS'17)*.

[63] Daniel Votipka, Seth M Rabin, Kristopher Micinski, Thomas Gilray, Michelle L Mazurek, and Jeffrey S Foster. 2018. User Comfort with Android Background Resource Accesses in Different Contexts. In *Proc. 14th Symposium on Usable Privacy and Security (SOUPS'18)*.

[64] W3C. 2024. *Media Capture and Streams.* Retrieved November 21, 2024 from https://www.w3.org/TR/mediacapture-streams.

[65] W3C. 2024. *Web Push API.* Retrieved November 21, 2024 from https://www.w3.org/TR/push-api.

[66] Katie Watson, Mike Just, and Tessa Berg. 2023. A comic-based approach to permission request communication. *Comput. Secur.* 124 (2023), 102942.

[67] webpush3. 2024. *Webpushr Library.* Retrieved November 21, 2024 from https://www.webpushr.com.

[68] Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu, and Michalis Faloutsos. 2012. Permission evolution in the Android ecosystem. In *Proc. 28th Annual Computer Security Applications Conference, (ACSAC'12)*.

[69] WHATWG. 2024. *DOM Living Standard.* Retrieved November 21, 2024 from https://dom.spec.whatwg.org.
[70] WHATWG. 2024. *Web Notifications API.* Retrieved November 21, 2024 from https://notifications.spec.whatwg.org.
[71] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. 2015. Android Permissions Remystified: A Field Study on Contextual Integrity. In *Proc. 24th USENIX Security Symposium (SEC'15).*
[72] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David A. Wagner, and Konstantin Beznosov. 2017. The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences. In *Proc. 28th IEEE Symposium on Security and Privacy (SP'17).*
[73] Bo Zhang and Heng Xu. 2016. Privacy Nudges for Mobile Applications: Effects on the Creepiness Emotion and Privacy Attitudes. In *Proc. 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW'16).*

A Crawler

A.1 Contributon of DOM Interactions

Our crawler, described in §5.1, interacts with webpages by clicking on elements likely to trigger permission-related functionalities. This interaction is guided by a manually curated list of heuristics, which we created by reviewing the source code of 500 random sites with permission prompts. Particularly, we checked if the occurrence of prompts on these sites relied on user interaction and identified the relevant DOM selectors (e.g., node ID attribute, class name, etc). Then, we grouped these selectors based on their similarity and created a set of heuristics to guide the crawler on which elements to click. Table 11 presents the complete list of heuristics.

We conducted additional experiments to quantify the contribution of our interactive crawler in triggering permission prompts and APIs. To do that, we randomly selected 100 URLs from the 770K seed URLs that use any permission concept, and an additional 100 URLs from each of the four permission concepts to ensure that there are sufficient samples from each permission type, resulting in 500 URLs. We then compared the number of observed API calls or prompts with and without crawler page interactions. To increase the confidence in results, we repeated the random sampling and our experiment two times, testing a total of 1K webpages, and take the aggregated result.

In total, we observed that incorporating page interaction heuristics more than doubled the likelihood of encountering browser prompts (and thereby rationales) at runtime, increasing observed calls to permission-gated APIs from 5.1% to 10.4% of webpages. Table 12 summarizes our experimental results.

A.2 Understanding Prompt Detection Challenges

Our automated crawler observed permission prompts on only ~20% of the pages from the seed list (all URLs on the seed list use at least one of the most popular permission-gated web API according to Chrome telemetry). To understand the underlying causes, we randomly selected 100 webpages where the crawler missed permission prompts and manually investigated the reasons.

We found that 73% of the cases were due to common crawling challenges: reaching deep application states (28%), handling DOM interactions (20%), authentication barriers (14%), and bot detection mechanisms (10%). In 11% of the cases, the target webpages were no longer active. Additionally, 16% of the URLs from the telemetry dataset were sanitized for privacy, leading to discrepancies between the pages our crawler visited and the actual pages where permissions were observed. Table 13 summarizes our findings.

Perm.	DOM Selector

Notif.	`[id=onesignal-slidedown-allow-button]`
	`button[class*="cleverpush-confirm-btn-allow"]`
	`button[class*="dn-slide-accept-btn"]`
	`button[class*="js-pushowl-yes-button"]`
	`//button[contains(.,"notification")]`
	`[data-test-id="push-subscription-cta-accept"]`
	`div[id="btn-allow"]`
	`'div[class*="btn-notification"]`
	`//div[text()="Zulassen"]`
	`//div[text()="allow"]`
	`a[class*="allow"]`
	`[class*="allow"]`
	`[id*="allow"]`
	`[id=push-popup-yes]`
	`[class*="approve"]`
	`[class*="btn-notification"]`

Geo.	`[data-qa-id="use-my-location-btn"]`
	`[class*="location-btn"]`
	`getElementsByTagName("m-locate-me")`
	`[class*=js-location-button]`
	`[class*=location]`
	`[id*=location]`

Cam.	`[id*=video]`
	`[class*=allow-camera]`
	`[class*=enable-camera]`
	`[class*=use-my-camera]`
	`[class*=use-camera]`
	`[class*=camera]`
	`//p[contains(.,"Use my camera")]`
	`//button[contains(.,"Get started now")]`

Mic.	`[id*=microphone]`
	`[class*=microphone]`
	`[class*=soundcheck]`
	`[class*=tuneron]`
	`[class*=input__voice-search]`
	`[title*=speech-to-text]`
	`[class*=speech-to-text]`
	`[class*=voice]`
	`[class*=btn-record]`
	`[class*=music-box__buttons__button]`

Table 11: The complete list of node selector heuristics the crawler uses for page clicks to trigger permission prompts. Legend: Perm. = Permission. Notif. = Notification. Geo. = Geolocation. Cam. = Camera. Mic. = Microphone.

B Role of Prompts in Rationale Identification

We conducted two experiments to explore the presence and location of rationales alongside web permission prompts. In the first, we manually analyzed 100 randomly selected pages from the dataset where the crawler detected prompts. Among these, only 10 contained a discernible rationale, whether in text, UI elements, or both, accounting for 10% of the cases, of which almost half (i.e., 4.6%)

Crawler	Exper.	Pages	Calls	Obs.	C	M	G	N
Baseline	Run #1	S_1: 500	23	4.6%	3	1	2	17
	Run #2	S_2: 500	28	5.6%	0	0	11	17
	Total	1,000	51	**5.1%**	3	1	13	34
Interactive	Run #1	S_1: 500	54	10.8%	0	3	21	30
	Run #2	S_2: 500	50	10%	0	3	19	28
	Total	1,000	104	**10.4%**	0	6	40	58

Table 12: Contribution of crawler DOM interactions in triggering permission API calls or prompts based on heuristics in Table 11. The left part shows the percentage of pages with captured API calls, whereas the right part shows the absolute number of pages with observed calls for individual permissions. Legend: S_i represents the random subset i of the dataset. Exper. = Experiment. Obs. = Observed. C = Camera. M = Microphone. G = Geolocation. N = Notification.

#	Reason	Count
1	Complex Application State	28
2	DOM Interaction Required	20
3	URL Sanitized	16
4	Authentication Required	14
5	Geoblocked Access	12
6	Page Inactive	11
7	Captcha/Bot Prevention	10

Table 13: Distribution of reasons automated crawling missed permission prompts on a sample of 100 sites.

were purely based on English text. Extrapolating this finding to the whole dataset of 162K pages with observed prompts suggests a noteworthy scarcity of rationales provided alongside permission prompts, amounting to approximately 16.2K pages in total, of which 7.4K are expected to contain rationales based on English text.

In the second experiment, we analyzed 1K random pages identified as having prompts. We created accounts and logged in to assess rationales after login, although in rare cases, this was not feasible due to account requirements. Our analysis revealed rationales on 113 sites, with 19 of these found after login. This indicates that approximately 17% of the rationales are post-authentication. We observed that the total number of rationales discovered (113 out of 1K) is close to the 10% found in the first experiment, suggesting consistency across both experiments.

C Experience Sampling Questionnaire

Figure 19 depicts how the experience sampling questionnaire appeared in Chrome. Please also see [23] for additional details on this method.

Question text variables:

- *$capability = {"geolocation", "camera", "microphone"}*

Questions:

Q0. A website just asked for access to your $capability. Help us improve how websites ask for access by taking this 1-minute survey!
Q1. [not shared with us]
Q2. How annoying did you find having to make a decision on $capability access for this website?
- Not at all annoying
- Slightly annoying
- Somewhat annoying
- Very annoying
- Extremely annoying

Q3. How easy or difficult did you find making a decision on $capability access for this website?
- Very difficult
- Somewhat difficult
- Neither difficult nor easy
- Somewhat easy
- Very easy

Q4. [not shared with us]
Q5. Thank you for helping to improve Chrome!

D LLM Filtering Prompt

We used the following few-shot prompt to identify text snippets related to permission concepts. The few-shot examples are from real websites.

You are an assistant trying to help users manage their browsers. You are given a sentence and you have to decide if it is a rationale or not. The definition of rationale is the following: a rationale is a sentence from a website that asks (directly or indirectly) a user to allow access to one of the following devices: webcam, push notifications, microphone, user's location. If you decide that the sentence is a rationale, you have to write the name of the relevant device.

Sentence: Get Breaking News Alerts. We'll send you latest news updates through the day. You can manage them any time from your browser settings.
Answer: notifications

Sentence: However, you are not logged in. Log in or Sign up to receive price alerts.
Answer: No

Sentence: error we did not manage to get access to your location
Answer: location

Sentence: this is a very good model, it can record audio and video
Answer: No

Sentence: In order to reliably test your equipment, this page requires your browser's permission to detect your webcam and microphone.
Answer: webcam, microphone

Sentence: Look at streamers on cam!
Answer: No

Sentence: You need to connect a microphone.
Answer: microphone

Sentence: You can find us at the following address: 1234 Main St, Anytown, CA, 12345
Answer: No

Sentence: [TEXT_PLACEHOLDER]
Answer: ?

E Library Detection Rules

Table 14 provides a summary of the library detection rules derived from the permission rationales in our rationale catalog. We used these rules to mine similar patterns and identify additional uses of these libraries within our broader dataset, which consists of

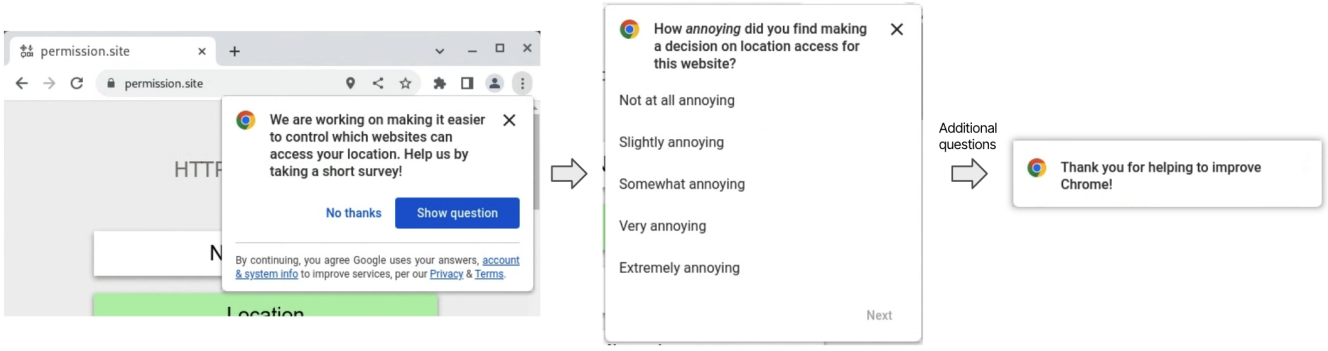


Figure 19: Screenshot of questionnaire invitation and subsequent screens. [23]

snapshots of hundreds of thousands of webpages collected during our web crawling.

False Positives of Library Detection Rules. We designed our library detection rules to be strict, minimizing the chance of false positives even at the expense of potential false negatives. To assess the false positive rate, we randomly selected three instances hit by each of the 32 signatures, and manually vet if the hit was a false positive. The results confirmed that our detection rules are robust, with all reviewed cases being true positives. This finding was expected, as the majority of the rules test for the presence of specific and relatively long string identifiers within HTML tag attributes, significantly reducing the likelihood of collisions.

Machine Learning vs. Library Signatures. We observed that signature-based rationale detection method, which operates directly on HTML code, excels at capturing instances where rationale text may not be immediately visible or requires user interaction to load, though compiling and maintaining a comprehensive list of signatures can be challenging. Conversely, the ML-based approach targets text within the rendered DOM, detecting both library and custom rationales, including libraries not found via the signature matching approach due to missing signatures in webpages, but may miss cases where text is split or not fully loaded, underscoring the complementary nature of these techniques.

F Rationale Clustering and Examples

We used agglomerative clustering in §5.3 to group together similar rationale embeddings generated via all-MiniLM-L6-v2 [18] sentence transformer. We chose the agglomerative clustering algorithm due to its capability to merge data points based on proximity measures, thereby facilitating the identification of semantic relations and keyword occurrences within the rationale texts. Our implementation uses the clustering model from the sklearn library [53], setting the affinity parameter to Euclidean distance and using a distance threshold of 3.5 to ensure precise and meaningful clusters. Table 15 presents examples of rationale texts from each of the 70 clusters identified in §8.1 following the above clustering methodology.

Library	Rule
iZooto	doc.find_all(attrs='class':'iz-news-hub-noti-blockd-txt') doc.find(attrs='id':'enable-turned-off-notis-cta')
OneSignal	doc.find_all(attrs='class':'modal-dialog') doc.find_all(attrs='class':'modal-notify') doc.find_all(attrs='class':'modal-body-message') doc.find(attrs='id':'onesignal-slidedown-container')
PushEngage	doc.find(attrs='id':'pe-widget-bell-launcher-message')
Smart Push	doc.find(attrs='id':'smart_push_smio_msg') doc.find(attrs='id':'smart_push_smio_note') doc.find(attrs='id':'smart_push_smio_not_allow') doc.find(attrs='id':'smart_push_smio_allow') doc.find(attrs='id':'smart_push_smio_footer') doc.find(attrs='id':'smart_push_arrow_bottom') doc.find(attrs='id':'smart_push_smio_agreement_contents') doc.find(attrs='id':'smart_push_smio_agreement_option') doc.find(attrs='id':'smart_push_gdpr_icon_message')" doc.find(attrs='id':'smart_push_smio_note') doc.find(attrs='id':'smart_push_smio_not_allow') doc.find(attrs='id':'smart_push_smio_allow')
Moe-push	doc.find(attrs='id':'moe-push-div')
PushOWL	doc.find(attrs='id':'pushowl-simple-toast-content') doc.find_all(attrs='class':'pushowl-simple-toast')
Perfecty	doc.find(attrs='id':'perfecty-push-settings-subscribed') doc.find(attrs='id':'perfecty-push-dialog-container')
Webpushr	doc.find_all(attrs='class':'webpushr-bell-theme-dark') doc.find_all(attrs='class':'webpushr-toggle-bell-popup')
Superstore-finder-wp	doc.find(attrs='id':'storeLocator__mapStatus__inner') doc.find(attrs='id':'storeLocator__mapStatus__closer')
Storerocket	doc.find_all(attrs='class':'storerocket-lead') doc.find_all(attrs='class':'storerocket-message-list') doc.find_all(attrs='class':'storerocket-initial-message-content') doc.find_all(attrs='class':'storerocket-error')
Total Rules	32

Table 14: Summary of library detection rules that we extracted from web permission rationales. The rules are based on the Beautiful Soup HTML parser [51].

Table 15: Examples of rationales from each of the 70 sentence transformer subclusters. IDs in the table represent cluster names and are composed of the first letter of the permission name followed by a group identifier. For example, G0 stands for Geolocation0 and CM0 for camera_microphone0 subcluster.

ID	Rationale	Domain
G0	It is mandatory to allow location of your browser to open an account through Video KYC	onlinesb.pnbindia.in
G1	To order online, please use the store locator below. To save your location for future online orders, press the "Set My Location" button on the location you are ordering from.	picklemans.com
G2	Requesting location access...	creedboutique.com
G3	Step 2: Click on "Location" in the options presented and then choose "Share live location".	imyfone.com
G4	Click Allow to easily find a bank and be in the know for all bank information!	allusbanks.com
G5	Click map to set your location	navigateme.lincoln.ac.uk
G6	We're searching for local stores. Your browser may ask for permission to use your location. Click "Allow" to sort the search results by distance.	theroomplace.com
G7	Please Allow GPS So That App Features May Be Enabled. Please Enable Location Service for Browser, and Clear Browser History Before Retry	app.masa.plus
G8	Geolocation Information. We may request access or permission	test2fly.carekore.app
G9	Your location is not permitted	rctiplus.com
G10	Click Allow for all Jet's Pizza menu updates and find a location near You!	menuwithprice.com
G11	Enter your address or zip code in the search bar below, adjust your search radius in the dropdown on the right, and click search. You may also click the arrow to geolocate and search from your current location.	rotech.com
G12	Click Now to find available Free Dental Clinic in your area.	livefit101.com
G13	Use your current location or enter search criteria in the form. Then choose a search radius and select the Search button to find dealers in your area.	windsorwindows.com
G14	Please turn on your location setting for your browser to see your nearest store. Alternatively, you can search by entering your city/postcode above or simply browse the map below.	charlestyrwhitt.com
G15	Allow the browser to use your location. Use current location	grubhub.com
G16	Click Accept and an initial pop-up will appear on your screen. Click "Continue" to go to your device's native Permission For Tracking pop-up.	playtikaprod.service-now.com
G17	Your location could not be determined. Click here to use your current location or enter your zip code in form above.	centier.com
G18	Allow us to access your location. We need your location to provide you with the best experience. Your location is safe with us. Allow Location	ajio.com
G19	Please enable your browser to allow this site to use your location	deltadentalnc.com
G20	Click Allow to get more free information about Public Housing Waiting List!	uslowcosthousing.com
M0	Once the number is entered, simply click on the "Call" button on the bottom of the dialpad. You will be prompted to allow PopTox to access your mic. Click on "Allow" for us to connect your call. Make sure to not "Deny" mic permission.	poptox.com
M1	To identify your range we will need to use your microphone.	singingcarrots.com
M2	Voice To Text Converter Click on the microphone icon and begin speaking for as long as you like.	unicodeconverter.info
M3	Your camera access is blocked. We can't continue without video. To connect with sign language support, allow access to your camera and microphone. Allow Access	signtime.apple
M4	If you are prompted, click to Allow access to the microphone.	htsdl.com
M5	Click here to test your mic.	xujenna.com
M6	To record audio messages, you must allow access to the microphone. I have authorized access, try again.	donationalerts.com

ID	Rationale	Domain
M7	You'll get a pop up from your browser asking to allow to use the microphone. Click to allow, so the violin tuner can pick up the note you're playing and tell you if it's in tune.	violinlounge.com
M8	This is a simple online microphone test so you can check whether your microphone works correctly. It's great before you start a Zoom call or any other video or audio-only call that requires a working microphone to be connected to your desktop or laptop computer. To begin the mic test, simply click the 'Start Test' button above.	test-microphone.com
M9	The microphone is not connected	micworker.com
M10	You will be asked to provide access to your microphone. App does not send any audio stream data to the servers.	bpmtech.no
N0	Get notified when you move less. The reminders function will ensure you are always on track with your health goals.	reliancedigital.in
N1	You are advised to subscribe with sarvgyan to receive all latest updates & notification for these & other exams.	sarvgyan.com
N2	Don't forget to subscribe to receive notifications of our new free recipes.	patterns.xn–amgurum-sfb.com
N3	Allow notification permission and refresh this page.	alerts.tbsnews.net
N4	Sign up to receive updates	bata.com.pk
N5	Get a notification when price drops below Rs.699.00 PKR.	jobsearch.childrens.com
N6	Get notified about opportunities that may interest you.	hannity.com
N7	Don't miss out on important news! Click 'Allow' for informative articles and updates.	unifi.com.my
N8	Looking to boost your credit? Allow updates to receive personalized alerts	creditcardsearching. thed-imepress.com
N9	Click Allow to stay updated with all DMV practice tests!	dmv-test-pro.com
N10	Allow your browser to receive notifications	rainbowloom.de
N11	Sign Up for Alerts. Receive alerts from Berkeley County	berkeleycountysc.gov
N12	Join our notification feed if you want to get the latest Movies, TV Series, Exciting updated Content, and Many More!!!	sunplex.net
N13	Get notified about ride updates & discounts For example: "Your Lyft driver is here!" or "Get $5 in credit" Notifications are blocked. Please follow these instructions to allow this site to show notifications.	ride.lyft.com
N14	Allow alternet.org to send web push notifications to your desktop.	alternet.org
N15	Click Allow for all latest coupons and discounts for Vistaprint!	coupon.hoursguide.com
N16	With your subscription, you'll get email alerts and push notifications to keep you up to speed on the action.	tradersmith.in
N17	Subscribe to our push notifications. No Thanks Allow	in.tubecorporate.com
N18	You have blocked receiving notifications from https://www.lostiempos.com. Please change the browser site settings in order to receive notification	www-lostiempos-com.gravitec.net
N19	So don't wait. Fill out our form to request the loan you've been searching for with Quick Loans. Stay updated on your loan! Click 'Allow' to ensure you receive important updates	quickloans.cash
N20	marionetka.com Would like to send you notifications: Allow, Discard	marionetka.com
N21	Stay up-to-date with SET News	careers360.com
N22	To receive notification from SmartThings Find, you must turn on the notification under settings.	samsung.com
N23	www.zeberka.pl would like to send you web push notifications. These may include commercial information regarding special offers and discount coupons on its own behalf and on behalf of its co-operators. To opt out, turn off notifications from www.zeberka.pl Turning off notifications will be possible at any moment, by clicking the button below.	relaxandwax.com
N24	No locations found near you, but we'd love to change that. Get notified when we add a location nearby Notify Me Reset Search Oops! Something went wrong. This page didn't load Google Maps correctly. See the JavaScript console for technical details.	cashify.in
N25	Disable notifications for WhatsApp to go Invisible On WhatsApp	samsung.com
N26	Don't miss out on best offers! Allow us to send you awesome updates and offers! Don't Allow	sarkariyojnaa.com
N27	CIO wants to show you notifications	cio.com
C0	Click Allow for all latest tricky DMV road sign tests!	flirt4free.com
C1	Under Camera, select "Allow" or "Ask".	readypay.co
C2	Hit the SCAN NOW button to launch the in-browser scanner. You may be prompted for camera access.	coomeet.me
C3	Activate your camera and start chatting. Video chat applications are a fun means to meet all different sorts of people from all over the globe.	echat.live
C4	Turn on the camera permission in your browser to continue further	qrcodescanneronline.com
C5	Use your Camera to start VideoChat	veed.io
C6	After allowed camera permission, just focus device camera to the WiFi QR Code and this tool will scan WiFi QR Code immediately.	megavirt.com
CM0	Give us access to this device, if You have to make free video calls.	globfone.com
CM1	You will need to allow access to your camera and microphone for the video consultation. You can use any computer with a webcam and microphone enabled or a smartphone with a camera.	essential.doxy.me
CMG0	Give access to your webcam, mic, and location if required. Click "Allow" where necessary.	omeglealternative.com