PIIxel Leaks: Passive Identification of Personally Identifiable Information Leakage through Meta Pixel

Paschalis Bekos
Stony Brook University
Stony Brook, NY, USA
pbekos@cs.stonybrook.edu

Panagiotis Papadopoulos
FORTH
Heraklion, Greece
panpap@ics.forth.gr

Nicolas Kourtellis
Telefonica Research
Barcelona, Spain
nkourtellis@gmail.com

Michalis Polychronakis
Stony Brook University
Stony Brook, NY, USA
mikepo@cs.stonybrook.edu

Abstract

Web pixels are one of the predominant techniques for tracking conversions and user behavior on the Web. The integration of Meta Pixel (the most widely used tracking pixel) into a website enables Meta to collect sensitive information about the website's visitors and match it with their Facebook or Instagram profiles. In addition to detailed navigation history, Meta Pixel also collects personally identifiable information (PII) entered by visitors in online forms present on the website, such as emails and phone numbers.

In this paper, we present a scalable and comprehensive approach for measuring PII leakage through Meta Pixel by *passively* inspecting its core components, without the need to interact with the dynamic elements of a website. This is possible by statically identifying and analyzing the configuration profile of a Meta Pixel instance and extracting the information it is set up to collect. By developing a hybrid crawling approach (static and headless), we analyzed the top-1M most popular websites and found that *12.2%* of them leak at least one instance of PII to Meta. We also found that in addition to email addresses and phone numbers, Meta Pixel also tracks PII such as age, gender, and geographical information, which can be used to not only reveal the identity of a user, but also their demographic characteristics. Finally, we assess the ability of Meta Pixel to track the browsing journey of a user by recording the sequence of full URLs visited across sub-pages.

CCS Concepts

• **Security and privacy → Social network security and privacy**;
• **Networks → Network privacy and anonymity**.

Keywords

Online advertising, personally identifiable information, Meta Pixel, online tracking, web privacy

CCS '25, Taipei, Taiwan

ACM ISBN 979-8-4007-1525-9/2025/10
https://doi.org/10.1145/3719027.3765113

ACM Reference Format:
Paschalis Bekos, Panagiotis Papadopoulos, Nicolas Kourtellis, and Michalis Polychronakis. 2025. PIIxel Leaks: Passive Identification of Personally Identifiable Information Leakage through Meta Pixel. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3719027.3765113

1 Introduction

The tracking of online user activity by third parties is one of the most pressing issues that affect user privacy. As advertising remains one of the core pillars of revenue, most companies rely not only on the collection of behavioral data, but also of personally identifiable information (PII) for more accurate ad attribution and targeted advertising. Regulations such as GDPR [1], CCPA [2], HIPAA [3], COPPA [4], and VPPA [5], as well as browser initiatives such as Chrome's Privacy Sandbox [6] and Mozilla's referrer policy [7] are a few examples of efforts aiming to protect user privacy. However, advertising entities rely heavily on user tracking for accurate targeted advertising and adapt to these safeguards with new techniques.

One of the oldest and most prominent browsing tracking techniques is through web pixels (also known as web beacons [8]). Modern "pixels" are actually pieces of JavaScript code [9–11] that operate as event-driven callbacks triggered by certain events that report information about user activities to third parties. Their most prominent uses include (i) tracking pageviews (e.g., to determine how many times a particular web page was viewed); (ii) conversion tracking for capturing user actions (e.g., account subscription or product purchase) to measure the effectiveness of an ad campaign; (iii) measuring ad performance by tracking views and clicks; and (iv) gathering user information (e.g., IP address, device or browser characteristics, cookies, and other PII) that helps with attribution of an action to a user's social networking profile.

Meta introduced its own tracking pixel, namely Meta Pixel [12], in 2013 (known as Facebook Pixel at that time), with its second major version released in 2018. In 2019, Meta developed a feature for Meta Pixel called Automatic Advanced Matching (AAM) [13], which enables a website that uses Meta Pixel to automatically collect visitor data and match it with users on Meta's platforms (i.e., Facebook, Instagram). If a website contains input forms, AAM will collect various types of PII, such as email addresses, birthdays, names, home addresses, and phone numbers, and transmit them

(hashed) to Meta. The purpose is to track and profile visitors so that the rendered ads can be more targeted and effective. Meta may also use this information to improve its own services and ad-targeting capabilities, thus benefiting other advertisers that use its pixel and ad platform. A similar feature of Meta Pixel, called Automatic Events (AE), enables websites to automatically track user interactions, such as button clicks, searches, and menu selections. Consequently, when a visitor navigates to a website or app, AE collects data about these interactions and reports them to Meta.

Before the wide adoption of Meta Pixel, both third-party and first-party cookies (e.g., `_fbc` and `c_user`) were used to attribute a user's activity to their Facebook profile [14]. This, however, requires an intermediate interaction with the social media platform. The user either must be logged in on Facebook so that the `c_user` cookie can be attributed to their profile, or land on a website coming from Facebook with a click ID tag embedded in the URL (e.g., `fbclid`). At the same time, modern browsers have started to block third-party cookies by default, preventing this type of tracking.

In contrast, Meta Pixel can attribute a user's activity to a Facebook profile *without* requiring *any* intermediate interaction with the social networking platform. When some user-specific PII (e.g., an email address submitted through a form) is leaked through Meta Pixel, there is no need for any intermediate interaction with the platform for successful attribution. Consequently, Meta Pixel can reveal the Facebook identity of an online user without requiring any user identifiers generated by the platform.

It is apparent that although Meta Pixel was introduced to aid advertisers reach their audience, it has also raised significant privacy concerns. Indicatively, in March 2023, the Austrian Data Protection Authority decreed that the use of Facebook's tracking pixel directly violates the GDPR [15]. In June 2024, the Swedish Data Protection Agency issued a fine of €1.34M to a website for breaching GDPR in five different cases related to Meta Pixel [16]. An FTC report [17] highlights how pervasive tracking through Meta Pixel transmits to third parties a great amount of information about user interests. As another example, AMC+ settled a class action lawsuit regarding a violation of the Video Privacy Protection Act (VPPA) [18].

In this work, we present a new approach for investigating Meta Pixel's privacy implications, focusing on the leakage of PII. Our approach does not require any interaction with the visited web pages, and is thus highly scalable. By analyzing the *configuration file* that is always present as part of Meta Pixel's integration in web pages, we can identify the specific PII-related parameters it is configured to capture and transmit to Meta. To the best of our knowledge, we are the first to leverage this core component of Meta Pixel for studying its PII leakage implications. Using our methodology, we are able to scale our exploration across one million websites from the Tranco list [19]. An important advantage of our approach compared to previous studies [20–22] is that it *does not require any interaction* with dynamic elements (e.g., form submissions), which otherwise would negatively impact a website under study (e.g., by affecting ad budgets or network traffic).

In addition to PII leakage, we also uncover how Meta Pixel can track in detail the specific URLs a user visits on a website *despite* existing countermeasures. Although modern browsers employ HTTP referrer trimming to prevent such tracking, Meta Pixel reports both the visited URL and the URL from which the visit originated across different sub-page visits. More importantly, we show through several case studies how the combination of the above two types of leakage can lead to indirect leakage of *additional* sensitive contextual information about a user activity or intent, such as the type of health care a user is interested in, or the type of medication they are looking for.

The main contributions of our work are summarized as follows:

- We propose a new approach for studying the types of PII leaked through Meta Pixel by statically analyzing its configuration code present in a webpage, without the need to interact with form fields or submit any information.
- We developed a scalable *hybrid* (static and headless) web crawler that identifies instances of Meta Pixel.
- We crawled 1M websites and present an in-depth analysis of the types of PII leaked through Meta Pixel. We also measure the extent to which Meta Pixel performs detailed activity tracking even on sensitive websites (e.g., health-related) by capturing the sequence of URLs a user visits.
- We showcase how Meta Pixel can collect not only the navigation history and PII of a user, but also contextual information from a page's content. This is even more concerning in "sensitive" websites, potentially revealing insights into the user's offline needs and habits.

Our prototype implementation and complete data set are publicly available through https://github.com/pasxalisbekos/PIIxel_Leaks.

2 Background

2.1 Meta Pixel

To provide effective advertising campaigns, platforms like Meta must have the ability to measure and attribute engagement and conversions (actions). One of the most popular techniques to achieve this is using web pixels [9–11] (or beacons [8]). Meta Pixel is a piece of JavaScript code hooked up to Facebook Ads Manager that developers can include in their website to: (i) measure ad performance, (ii) optimize ad campaigns, (iii) build audiences for ad campaigns, (iv) track page views and conversions, and (v) get insights on how people use their website. Each Meta Pixel consists of a base code script that is added in a webpage, which loads two additional JavaScript components in the following order:

- `fbevents.js` library: Includes the components required for the core functionality of Meta Pixel.
- Configuration file: Contains options specific to the marketing campaign and website (e.g., parameters tracked or filtered), and is tied to a unique Meta Pixel ID.

Fouad et al. [23] investigated the presence of Pixels as user tracking mechanisms across the web, finding that 83% of Alexa's top 100K websites included at least one such tracker. Bekos et al. [24] studied the mechanics of Meta Pixel for tracking page events, and how their attribution to a Facebook profile could be achieved through first-party cookies and URL tags. Aside from first-party cookies, Meta Pixel can also relay PII to advertisers, offering better accuracy for user attribution. In this work, we propose a different approach for identifying PII leakage based on analyzing the core components of Meta Pixel, allowing us to identify the conversion-specific functionality tailored to the marketing campaign each website runs.

2.2 Advanced Matching and Customer Information Parameters

In July 2020, Meta Pixel introduced Customer Information Parameters [25], a set of user identifiers shared alongside event information. These parameters are defined as `user_data` objects, and include the user's email, first and last name, phone number, gender, date of birth, and geolocation information. According to Meta, these can help increase the quality of event attribution, which is important as only *matched* events can be used for ad attribution and delivery.

While this process can be configured manually [13], an easier approach is to use Meta's Events Manager [26]. Events Manager offers a direct way of defining which parameter Meta Pixel will track without requiring any modification to the website's or Meta Pixel's code. The owners of an advertising page on Facebook can simply enable "Advanced Matching" and toggle on the parameters they wish to track. These actions will be reflected in the configuration of the corresponding Meta Pixel where those parameters are defined.

Meta Pixel collects `user_data` objects and includes them as additional query parameters in the body of the request that reports an event to Meta. While these sensitive parameters (e.g., email) are not shared as clear text but as SHA256 hashes, attribution of those hashes to real Facebook users is still possible. Figure 1 illustrates this process. Assume that an anonymous (towards Facebook) user visits a website that uses Meta Pixel. Once loaded, Meta Pixel requests (step 1) the core library (`fbevents.js`) and the configuration file of this Meta Pixel instance (step 2). In this example, the website has a Facebook advertising account (i.e., a Facebook page) that has configured the "email" parameter to be tracked.

After browsing for a while, the user decides to create an account on the website and fills in the registration form with their email. The submission of the form raises an event (e.g., `SubscribeButtonClick` or `CompleteRegistration`) that is captured by Meta Pixel (step 3). This event is then reported to Meta (step 4), including the hashed email as a value of the `udff[em]` parameter. If the user has a Facebook account tied to the same email, a simple comparison of the two hashes (step 5) can reveal the real Facebook identity of the previously anonymous user (step 6). In this way, Meta Pixel can reveal the Facebook identity of an unknown user without requiring any additional interaction with the platform (e.g., Facebook login).

2.3 Meta Pixel and Full URLs

While PII leakage is the main issue of concern, the leakage of navigation history to third parties also raises privacy implications. Regulations such as the Health Insurance Portability and Accountability Act (HIPAA) [3] and the Video Privacy Protection Act (VPPA) [5] restrict the ability of online services to share information that can reveal an individual's browsing habits or preferences. The fidelity of activity tracking is enhanced when conversion tracking from tools such as Meta Pixel also incorporates the *URLs* a user visits.

Apart from personal user identifiers, Meta Pixel also captures a user's navigation history on a website through two identifiers:

(1) `dl` (Direct Link): URL of the current page a user has landed on (extracted from `document.location.href`).
(2) `rl` (Redirect Link): URL of the previous page from which the visit originated (extracted from `document.referrer`).

This behavior of Meta Pixel can be disabled by configuring it using the "Core Set Up" [27], which is though disabled by default when installing Meta Pixel. The `dl` and `rl` parameters are populated from the corresponding DOM element values. Consequently, an alternative way to prevent Meta Pixel from sharing these values is to create website-specific scripts that prohibit reading those values (or clear them completely). Third-party scripts such as Google Tag Manager [28] (which can be used to install and manage Meta Pixel) can also override those query parameters.

Figure 2 illustrates how Meta Pixel can capture a user's visited pages during a browsing session on a particular website. In this example, a user visits a series of pages, starting with the home page (step 1), which triggers a `PageView` event. Next, the user visits a subpage for a specific product (step 2). This time, besides the current URL (`dl` parameter), Meta Pixel also reports the URL from which this visit originated (`rl`). Up to this point, Meta has collected the browsing journey of the anonymous user on the website, but has no way to attribute this activity to a specific Facebook user. In the next step, the user decides to create an account on the website (step 3). When the user submits the registration form, all tracked parameters will be sent to Meta, as shown in Figure 1. This allows Meta to associate the tracked navigation history on this website with a particular Facebook user. While the navigation history is collected from each individual website, when combining such information from multiple websites, Meta is in a position to identify part of a Facebook user's global browsing history.

3 Methodology

We begin by exploring how updates on Events Manager about Advanced Matching and customer information parameters are reflected in Meta Pixel's configuration file. We identify the association between configuration options and types of tracked PII, which we use to measure PII leakage across the web. To that end, we leverage the Tranco [19] list to explore a wide range of popular domains with respect to Meta Pixel utilization and its configurations.

3.1 Advanced Matching Configuration

The most widely used approach for deploying Meta Pixel is to set up Advanced Matching directly through Meta's Events Manager, removing the need for manual implementation. This requires no additional code and sets up Meta Pixel to search for recognizable form fields and other sources that contain user information.

To explore how the selected Advanced Matching capabilities are reflected in the configuration file, we used a personal Facebook account with advertising features and a custom website deployed using WordPress. Through our Events Manager, we configured a Meta Pixel and deployed it on our website, initially without setting any customer information parameters to be tracked. Having verified the installation, we then start enabling each parameter (or set of parameters in the case of first and last name) to be tracked. For each such change, we inspect the updated version of the configuration file to identify the presence of new parameters and map them to the corresponding Advanced Matching feature.

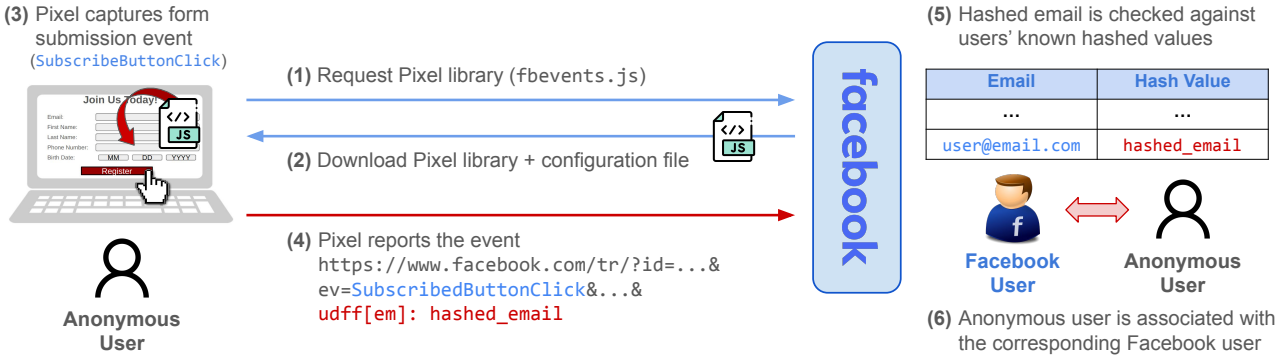

Figure 1: Attribution of an anonymous user event to an actual Facebook identity through the Meta Pixel.

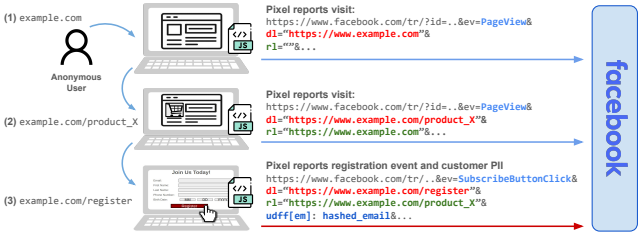

Figure 2: Meta Pixel tracking the pages a user visits during a browsing session on a website.

3.2 Hybrid Crawling

Having understood the core functionality and usage of Meta Pixel's configuration file, we aim to develop an approach for identifying its presence on any website. One of our first findings is that each Meta Pixel has a *unique* identifier (ID) that corresponds to its configuration file. Initially, we implemented a dynamic crawler that browses through websites and identifies the configuration file ID (PIXEL_ID) while inspecting any network activity related to Meta's servers on the website. We follow state-of-the-art approaches [24, 29–32], leveraging Puppeteer [33] to automate this browsing activity through Chrome. To handle cookie banners, we simulate a real user's choice of *"Accept All"* using the Consent-O-Matic extension. Papadogiannakis et al. [34], when also studying Tranco's 1M websites, showed that the consent option does not really affect the presence of third-party trackers such as Meta Pixel. With this dynamic ("headful") crawler, we explored the adoption of Meta Pixel across Tranco's top 10K websites.

While this dynamic crawling approach serves as a good starting point, we conclude that it is not a scalable solution for our needs due to i) its high resource consumption and time-consuming nature, and more importantly, ii) its inability to identify Meta Pixel instances that (for various reasons discussed below) are not immediately loaded on the page. To address these challenges, we developed a *hybrid* crawler that uses a combination of static and dynamic analysis, as illustrated in Figure 3. Initially, the crawler statically analyzes only the static components of the HTML code of a website through a wide range of heuristics that can identify the presence of Meta Pixel. If the static analysis is not fruitful, the crawler then performs a dynamic *but headless* crawl of the website, trying to identify any Meta Pixel instances present within any dynamic components.

3.2.1 Static Component. We developed a methodology that scans only the *static* HTML components of a website to identify the presence of Meta Pixel using four complementary strategies. Meta Pixel can be statically configured in a website through two implementation methods, which act as our baseline indicators for identifying its presence. The *base code* implementation is a `<script>` tag that provides the full functionality and tracking capabilities. The *lightweight* implementation is an `<image>` tag offering simplified integration but with limited capabilities.

Besides the above two integration methods, we also consider additional ways in which Meta Pixel can be loaded on a website through third-party scripts (that are statically present on the HTML). The most widely used third-party script for managing and loading Meta Pixel is the Google Tag Manager (GTM) [28]. GTM allows developers to easily update and manage measurement and other analytics code snippets (tags) on a website, without the need to modify the underlying code. For that reason, we adapt our crawler to also investigate the presence of Meta Pixel implementations inside GTM configurations.

Given the above possible integration methods, our static analysis method consists of four steps. First, we use a set of heuristics to identify either the base code or the lightweight implementation of Meta Pixel inside `<script>` and `<image>` HTML tags. Second, if this search yields no results, we proceed to scan all `<script>` tags for indicators related to Meta Pixel. Apart from the base code and lightweight implementations, these indicators also include invocations of pixel-specific functions (e.g., its initialization function call `fbq("init")`) or URLs (e.g., `facebook.com/tr?id=...`). Third, if still unsuccessful on retrieving a Meta Pixel ID, we perform a similar heuristic-based search on *all* script files loaded statically by the page. We perform this by retrieving each file through the `src` field of `<script>` tags and analyzing its content.

Finally, we scan for the presence of Google Tag Manager's (GTM) implementation and extract the corresponding Tag IDs. Similarly to Meta Pixel, GTM also uses a configuration file that specifies which tags, triggers, and variables are set up within the container. This configuration determines what data is collected, when it is collected, and how it is processed. We extract GTM's specific ID (`GTM-TAG_ID`)

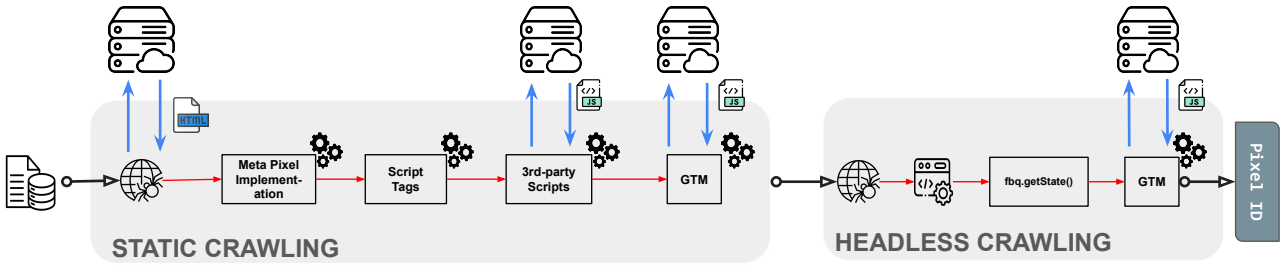

Figure 3: The hybrid crawler's underlying infrastructure using both static and dynamic (in headless mode) analysis to identify Meta Pixel's configurations.

and we use it to retrieve the JavaScript file corresponding to the GTM configuration. In this configuration file, we look for the presence of Meta Pixel using four different search strategies. These cover the base code and lightweight implementations, as well as pre-defined variables that may store a Meta Pixel ID. Those variables are used inside the invocations instead of using the direct usage of Meta Pixel ID, enabling easy updates of such parameters.

Initially we try to identify the base code or lightweight Meta Pixel implementation within GTM directly. We explore the body of the GTM implementation for any of the two (base code and lightweight) implementations. Next, we search for a variable named `vtp_pixelId` that is used for storing the pixel's ID value. This variable is used mostly when Meta Pixel components such as the ID or events names are declared through a global variable.

As a third option, we identified that one of the structural components of GTM is an array named "tags," which holds all the configured Pixels. We use a set of heuristics to identify Meta Pixel implementations inside this array. The tags array can be used to perform mapping between events and tags, i.e., declaring which tag or pixel should be activated based on a specific triggered event. Finally, if no Meta Pixel is found, we resort to a generic heuristic to identify IDs based on the format of the Meta Pixel ID (numerical value including 15–16 digits), and we flag any matches as *potential* IDs. These heuristics allowed us to identify Meta Pixel implementations inside GTM configuration files, and extract Meta Pixel IDs present either as a static strings or values of dynamic variables.

This static implementation efficiently reduces the population of websites requiring a dynamic investigation, speeding up significantly the overall crawling process across 1M websites. Our methodology depends solely on a GET request to retrieve the website's core HTML page, with minimal additional overhead for fetching and analyzing any supplementary JavaScript files (GTM or other scripts) included by the static HTML.

3.2.2 Dynamic Component. Our static analysis approach misses Meta Pixel instances that depend on the execution of dynamic components. Thus, we complement our static approach by further analyzing any websites where Meta Pixel was not statically identified through an additional headless crawl. We do not use headful crawling in this step, as it requires significantly more CPU and memory resources for the graphical interface of the browser. In contrast, headless crawling allows for significantly better parallelization. This translates to both faster processing times and more efficient resource utilization when dealing with our large-scale measurement requirements. Given that even headless crawling introduces more overhead than the static approach, our implementation also leverages parallelization (with respect to our resources).

When visiting a website, instead of inspecting the network traffic to identify the presence of Meta Pixel, we use one of its predefined methods, namely `getState()`. This method is defined in the core library `fbevents.js` and is universally accessible under any Meta Pixel implementation. The invocation returns an object containing information about the pixel, including its ID and event count.

For our headless crawling, we use Selenium and Chrome's Driver to automate browser interactions and exploration of the DOM structure and JavaScript execution states. When Meta Pixel's state inquiry method (`fbq.getState()`) yields no results, our implementation falls back to looking for Google Tag Manager implementations by extracting all unique GTM IDs from the document's HTML content. We proceed with retrieving the corresponding GTM configuration files of the website and then perform the same heuristic-based approach for identifying GTM-deployed Meta Pixel IDs used by the static analysis phase.

3.2.3 Verification of Meta Pixel IDs. To verify the identified Meta Pixel IDs, we perform a "sanitization" process. After removing any duplicate IDs, we *validate* the remaining IDs by visiting every potential Meta Pixel ID configuration URL of a given website using as parameters the configuration ID under consideration and the domain name. If no response is obtained, or if the response contains JS code indicating an invalid configuration ID, we remove this ID from our set. This validation step ensures that all considered Meta Pixel IDs in our evaluation correspond to valid configuration files.

4 Meta Pixel Adoption

In this section, we measure the adoption of Meta Pixel across the 1M most popular websites according to the Tranco list [19]. We first compare our hybrid approach with a fully dynamic (headful) based crawler with respect to the number of identified pixel instances and their validity, using a subset of websites. Our results show that our hybrid approach identifies a higher number of pixel instances due to its static analysis component. We then move forward with scaling our study to all 1M websites, and analyze the PII leakage occurring through the identified pixel instances.

4.1 Hybrid vs Dynamic: Top 10K

We compare our hybrid approach with a fully dynamic crawler using the same timeout values on both, for the subset of the top 10K websites. This experiment took place on a local machine with eight CPUs in a university environment.

The dynamic approach identifies 2,021 (20.21%) domains containing Meta Pixel by inspecting all network activity and capturing both the pixel's configuration file and any event-related requests. On the other hand, our hybrid crawler identifies the presence of Meta Pixel on a total of 2,572 (25.72%) domains. From those, 1,368 (53.19%) are identified by our *static* heuristic-based approach, and the remaining 1,204 (46.81%) by the headless-based approach. After validating and removing duplicate IDs, we are left with 2,496 (24.96%) websites containing Meta Pixel.

After analyzing the differences, we observe that our hybrid approach identifies 94.8% of the websites that the dynamic approach captures, missing only 107 (5.2%) domains. On the flip side, our hybrid approach yields an additional *595* domains that cannot be captured by the dynamic crawler. To understand this increase better, we manually analyzed each one of those extra websites, investigating the traffic and scripts responsible for loading Meta Pixel and its behavior with respect to any cookie banners or other interactions.

The majority of these extra websites (40.84%) load Meta Pixel upon accepting a cookie banner. While we do use Consent-O-Matic to automatically accept cookie banners during dynamic crawling, customized cookie banners can evade the identification process of the plugin. In contrast, static analysis is not affected by such issues.

The next most frequent issue occurring in 34.28% of the extra websites is that while Meta Pixel exists inside the GTM configuration file, the crawler's visit does not act as a trigger for GTM to load it. By randomly visiting a few sub-pages on those websites, we see that 22.5% of those insert the pixel only in the HTML code of sub-pages. A fraction of websites (6.05%) load Meta Pixel directly upon the first visit of the hybrid approach, indicating that those same websites likely blocked our dynamic crawler, or a network issue occurred during crawling. Interestingly, for 5.88% of the extra websites, Meta Pixel is loaded and activated after a second action upon the initial visit, such as closing a pop-up window. We also find Meta Pixel requests towards Meta been directly issued from GTM in 3.7% of the websites without loading the underlying Meta Pixel components (`fbevents.js`, configuration file) on the website.

We find websites where GTM (3.7%) or the third-party script (1.85%) that configures Meta Pixel is not loaded upon the initial visit due to various reasons (e.g., regional restrictions with respect to consent choices, login requirement, or website-specific issues). For those cases, we were able to inspect the network traffic and identify the components that configure Meta Pixel's implementation. On the other hand, for 1.18% of the websites, we could not browse the website at all to inspect any network activity (due to IP blocking or regional restrictions), or the corresponding requests to load Meta Pixel's components were blocked.

In addition, on 1.18% of the websites Meta Pixel exists in the HTML but does not issue any requests, and 0.67% of the websites require login to continue browsing. We also encounter cases where only the Meta Pixel core library was loaded (0.67%). Overall, our hybrid approach provides increased coverage, as it can handle cases in which GMT or Meta Pixel may be loaded under certain conditions not met during the visit of the dynamic crawler. These include cookie banner acceptance not properly being handled by Consent-O-Matic, having the pixel enabled only in sub-pages, or loading it only under certain conditions specified in the GTM configuration.

Table 1: Distribution of Meta Pixel instance types.

Identification Method	# of Websites	% of Instances
Static: GTM	69,033	36.87%
Static: Meta Pixel	62,260	33.25%
Headless: Meta Pixel	41,065	21.93%
Headless: GTM	12,346	6.59%
Static: Other Scripts	2,553	1.36%

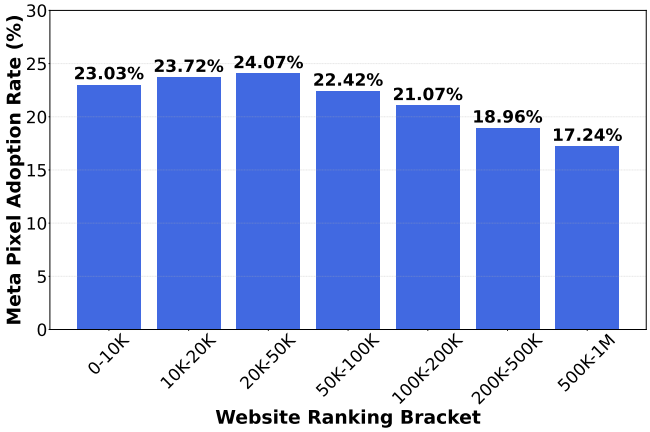

Figure 4: Adoption rates across website ranking buckets (from the most to the least popular).

4.2 Meta Pixel Across 1M Websites

We use the Tranco list generated on 16 December 2024[1] to collect the top 1M domains, and split the workload on batches of 100K websites across 10 Amazon EC2 instances (of type `t2.xlarge` with 16 GiB of memory and 4 virtual CPUs) hosted in the `us-east-1` region. Our measurement was conducted during the period between late February – mid March 2024.

Using our hybrid approach, we found Meta Pixel being used in 187,257 (18.7%) of the 1M websites. It is important to note that website reachability in our measurement is affected by the geographic location of our crawler and the fact that it is hosted on AWS (the IP ranges of which are public and potentially blocked in some cases).

As shown in Table 1, the majority of the pixel instances found by both the static and headless approaches correspond to "default" implementations, i.e., using Meta Pixel's code directly without the involvement of third-party scripts (55.18%). Supporting GTM is crucial, as it contributes to a significant portion of the instances found (36.87% for static and 6.59% for headless). A few websites integrate Meta Pixel through either custom or less popular third-party scripts (1.36%), many of which have similarities with GTM and similar frameworks.

[1]The specific list can be found at https://tranco-list.eu/list/J943Y/full.

When taking the popularity of the websites into account, our results align closely with previous studies. For instance, Chen et al. [35] identified 23.77% of the top-10K websites containing Meta Pixel through its first party cookies. Similarly, Bekos et al. [24] identified Meta Pixel in 23.08% of the Tranco's top 10K websites. As shown in Figure 4, our results closely align with these prior studies, as we found that 23.03% of the top-10K websites use Meta Pixel. To study the overall trend, we split our set into seven popularity buckets. We observe that the adoption rate increases up to the top-50K websites, after which a downward trend begins (with an average of 18.68% across the remainder of 100K–1M websites). Notably, Bekos et al. [24] report an adoption rate of 16.5% on those ranges while performing random sampling of 1K websites from each range, which is lower than the overall 18.68% we found when exploring the population as a whole.

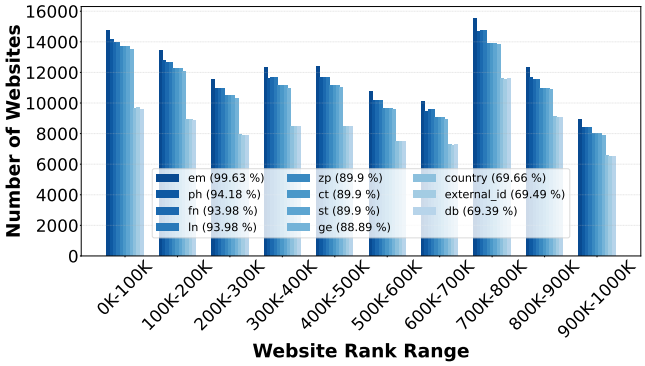

Figure 5: Parameter tracking rates across rank bins of 100K.

4.3 Key Takeaways

Our results show that the use of a hybrid crawler is beneficial for identifying Meta Pixel instances, as it can handle cases where the pixel is loaded only after specific actions that are challenging to fully automate. From the top-10K websites we are missing only about 1% compared to dynamic (headful) crawls, but we identify an additional 5.95% of websites that were missed by dynamic crawling (due to regional restrictions, cookie banner behavior, and the need for other interactions).

We identified Meta Pixel instances in 18.7% of the popular websites with adoption spikes in higher-ranked websites. The adoption rates follow a clear declining trend as website popularity decreases, starting at approximately 23% for the most popular sites (0-10K) and dropping to 17.24% for less popular sites (500K-1M). Our findings indicate that the adoption of Meta Pixel is correlated with website popularity, reflecting differences in marketing affiliations between high-traffic and low-traffic websites.

5 Customer Information Parameters

Having identified Meta Pixel instances and their corresponding PIXEL_IDs, we proceed to analyze the PII leakage through their configurations. We explore the presence of tracked PII in every configuration file corresponding to a PIXEL_ID. Using the unique IDs of each website, we can obtain and analyze the corresponding configuration files simply by visiting the URL https://connect.facebook.net/signals/config/PIXEL_ID. We identify the tracked PII by inspecting the following array: `"selectedMatchKeys": ["em", "fn", "ln", "ge", "ph", "ct", "st", "zp", "db", "country", "external_id"]`. We measure the frequency of each tracked parameter and their most common combinations across the analyzed websites.

5.1 Findings

Among the 187,257 websites that use Meta Pixel, we identify 122,577 (65.45%) configurations that track at least one PII-related parameter. To assess this leakage in relation to a website's popularity, we plot the observed tracking rate of each parameter per each 100K-rank bin within our 1M dataset, as shown in Figure 5.

These PII-related parameters can be used to enhance the attribution of a user's browsing activity (conversions) to their corresponding Facebook account. When creating a Facebook account, users must provide the following information: first and last name, email or mobile phone number, password, date of birth, and gender. To thwart the creation of fake accounts, additional verification is performed by requiring the user's *email* or *phone number*, making one of these two identifiers a prerequisite for account creation. As evident from Figure 5, those two parameters are the most commonly tracked, as they provide better accuracy for matching a conversion to a Facebook account. More specifically, the email parameter is tracked with a rate of 99.63%, and the phone number with a rate of 94.18%. In the rest of this section, we elaborate on the extent of this tracking across the web.

While the presence of such parameters in a website's configuration is a robust indication of PII tracking, relevancy attribution of ads is a multi-faceted process. Demographics can be used to cluster a group of users based on shared attributes, with the most common being the *age, gender,* and *region*. Gender and birthday (age) are also required fields when creating a Facebook account, but are not *unique* to the extent email accounts and phone numbers are. Although regional identifiers (e.g., country, state, city, and zip code) are not required, they can be collected through Facebook Location Services [36] if a user enables them. To identify the most commonly tracked groups of parameters, we measure the presence of each parameter combination tracked through the collected configurations.

As shown in Figure 6, the most common combination, used on average by 43.24% of all configurations, includes *all* parameters. More specifically, across websites that have configured Meta Pixel to track at least one PII the adoption of this combination is 65.8%. This can be attributed to websites wanting to achieve the most accurate attribution of ads to Facebook users, and thus aggressively track as much information about the user as possible. Our findings indicate that websites adopt an *"all or nothing"* philosophy when it comes to PII tracking, with Meta Pixel tracking either none (34.5%) or the vast majority of PII parameters (57.2%), i.e., either all or the following eight: email, phone number, first and last name, gender, city, state and ZIP code.

The 10 most frequent combinations of PII parameters (covering 96.3% of all configurations) heavily favor the collection of user information related to account creation (email, phone number, first and last name). As shown in Table 2, the email parameter is included in all combinations, followed by the phone number (80%), which is the

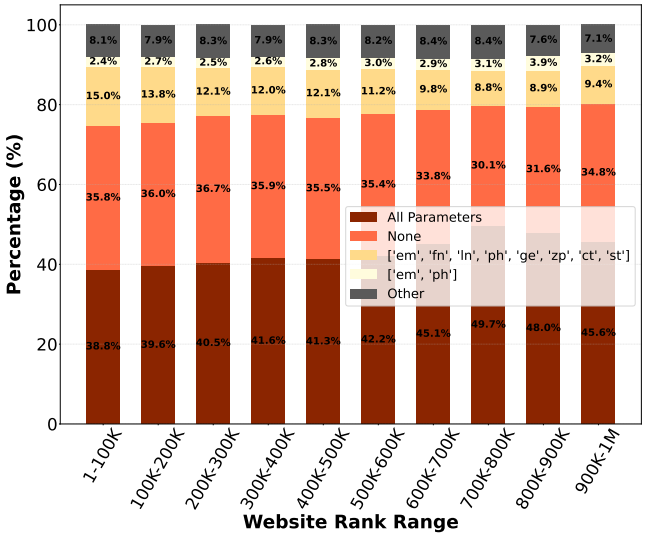

Figure 6: Top-5 most commonly tracked combinations of Meta Pixel parameters.

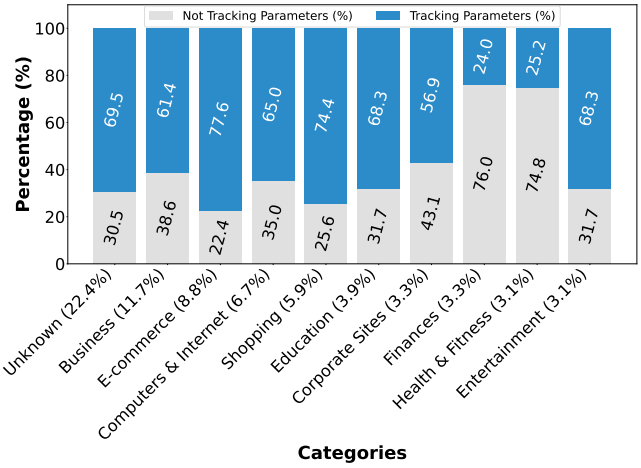

Figure 7: Tracking configurations across Meta Pixel implementations in the top-10 website categories.

Table 2: Frequency of tracked parameters across top-10 combinations.

PII	Parameter	% top combin.	Required
email	em	100%	Yes
Phone Number	ph	80%	Yes
First Name	fn	80%	Yes
Last Name	ln	80%	Yes
Gender	ge	70%	Yes
City	ct	70%	No
State	st	70%	No
ZIP code	zp	70%	No
Country	country	40%	No
Birthday	db	30%	Yes
external_id	external_id	40%	No

only other alternative way of verifying a Facebook account. Those two parameters are followed by the first and last name (80%). We also observe an aggressive tendency of websites to collect the user's gender (70%). On the other hand, country and date of birth (40% and 30%, respectively) are not favored for conversion attribution. However, specific regional identifiers such as state, city, and zip code (70%) are more widely tracked.

The `external_id` parameter is used less frequently (40%) across these combinations. As explained in the relevant documentation by Meta [37], this is a user identifier issued by the advertiser's system. As most websites rely on Facebook user identifiers to achieve better conversion to ad attribution, this identifier is not as frequently present as the rest.

5.2 Key Takeaways

From all inspected configurations, 65.45% track at least one PII-related parameter, with the email address (99.63%), the phone number (94.18%), and first and last name (93.98%) being the most widely tracked ones. As the email and phone number are among the requirements for creating a user account, Meta Pixel collection of these parameters can lead to highly accurate conversion attribution to Facebook profiles. Finally, we find that the predominant combination of tracked parameters includes *all* of them. This combination is used by 43.24% of the websites utilizing Meta Pixel.

6 Domain Categorization

To understand further the impact of PII leakage through Meta Pixel, we must consider each website's specific context. To that end, we categorize the domains in our dataset and investigate the leakage of each parameter across those categories. For the categorization process, we use the SafeDNS URL category check [38].

6.1 Meta Pixel Use Across Different Categories

We successfully categorize 77.6% of the websites that use Meta Pixel across 63 different categories. For the remaining 22.4% we receive no category type, and mark them as "Unknown". We find the majority of the categories having a ratio close to 2:1 with respect to tracking and non tracking PII configurations, as shown in Figure 7. Categories such as "E-commerce" and "Shopping" have a ratio of 3.5:1 and 3:1, respectively. We also see that "Financial" and "Health" oriented websites exhibit the least amount of PII leakage with a ratio of 1:3. Although these categories fall into the broader subcategory of "Sensitive" websites (with respect to their content), later on we will see that this is not a common trend.

When closely inspecting the websites that have configured Meta Pixel to track PII parameters, we see that the most widely tracked parameter is the *email* with a tracking rate of 99.6%, as shown in Figure 8. The phone number (96.3% adoption on average) is heavily favored in corporate and institutional websites ("Business," "Computers & Internet," "Education," "Corporate," and "Financial").

Figure 8: Parameter tracking rate by Meta Pixel across the top-10 website categories.

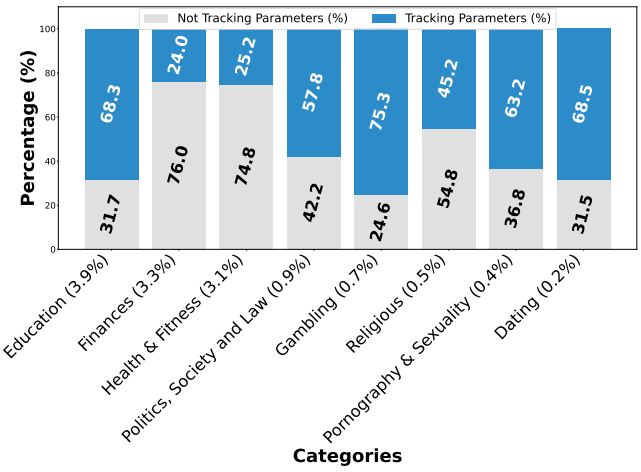

Figure 9: Tracking configurations across Meta Pixel instances in sensitive website categories.

On the other hand, consumer retail websites ("E-commerce" and "Shopping") tend to favor more the first and last name parameters (96.2% adoption on average).

6.2 Sensitive Categories

We identify 10 "sensitive" categories with respect to the content of the websites, using the same sensitive categorization process followed by prior works [39–41]. Apart from the six baseline sensitive categories ("Health," "Legal and Political," "Financial," "Sexuality," and "Religious"), we introduce three more categories from our findings. Inspired by an article [42] that revealed how Meta Pixel can identify and leak the academic interests of students, we categorize "Educational" websites as sensitive. Furthermore, we consider "Gambling" and "Alcohol & Tobacco" as sensitive due to both the nature of content and the user information they handle.

We identify a total of *30K* sensitive websites (16%) across these categories that use Meta Pixel. One would assume that in such websites, PII leakage would be significantly lower than the general web, but our findings, illustrated in Figure 9, indicate the opposite. Alarmingly, we find a tendency of "Gambling" websites to track PII with a ratio 3:1 (i.e., three out of four websites have configured Meta Pixel to track at least one PII parameter). In "Educational" and "Sexuality" websites this leakage is still prevalent with an approximate ratio of 2:1. Heavily moderated websites under strict regulations (GDPR, HIPAA), such as "Financial" and "Health," have an opposite trend, with a leakage ratio of 1:3. Finally, "Legal & Political" and "Religious" websites perform a moderate rate of PII tracking (4:3 and 1:1.2, respectively).

We analyze further the leaked PII for the websites of each category that track at least one PII parameter. As shown in Figure 10, the *email* is the most widely tracked parameter across all sensitive categories, with an average adoption rate of 99.5%, in contrast to the rest of the parameters. This is followed by *phone number* with an average rate of 95.5%, and *first and last name* with a rate of 93.6%.

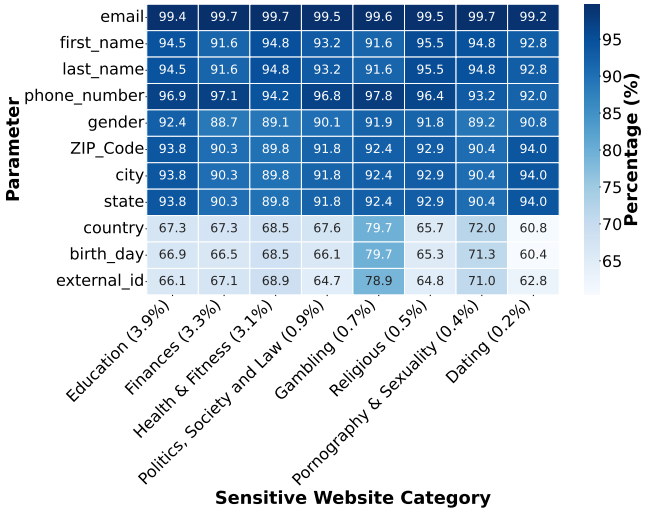

Figure 10: Parameter tracking rate by Meta Pixel across sensitive website categories.

The most aggressive form of tracking takes place on "Gambling" websites, in which even the least tracked parameters (country, date of birth, and `external_id`) are tracked in much higher rates. It is also important to note that across all categories, "Gambling" websites are those that tend to track phone numbers more frequently (97.8%). This is particularly concerning, as behavioral profiling on such websites can lead to exploitative targeted ads, pushing vulnerable users towards more risky behavior [43, 44]. A similar trend is observed in "Sexuality" oriented websites, in which these parameters are also tracked the most, with a slightly more aggressive rate on the first and last name than phone numbers. While the population of those websites is low, the nature of their content coupled with aggressive PII tracking raises significant privacy concerns.

6.3 Key Takeaways

We find that the most widely tracked parameters are the *email* and *phone number*, with rates higher than 90%, followed by *first* and *last name*, as these parameters offer higher accuracy for matching an anonymous user to their actual Facebook account. While the predominant 10 categories mostly include non-sensitive websites, we have also uncovered the use of Meta Pixel across eight sensitive categories. Among them, "Financial" and "Health" oriented websites tend to track PII parameters less frequently, but this is not the case for the remaining six sensitive categories. We find that in those sensitive categories, sensitive parameters are tracked with an approximate ratio of 2:1, while alarmingly, this ratio increases to 3:1 for "Gambling" websites.

7 Navigation History Tracking

In this section, we explore how the tracked PII can be coupled with additional information related to a user's behavior and habits for targeted advertising. In particular, we investigate the ability of Meta Pixel to track a user across different sub-page visits on a given website. Through the `dl` and `rl` parameters, Meta Pixel can track chronologically the precise URLs a user visits.

7.1 Full URL Leakage through Meta Pixel

Ad personalization of a user is a multi-faceted process that is enhanced when combining PII information with contextual and behavioral data. González et al. [45] showed that 73% of European Union Facebook users are labeled with potential sensitive interests for advertising purposes. Multiple personalization strategies [46–48] mention that browsing habits and behavioral information can be inferred by a user's navigation history on a website. Weinshel et al. [49] simulated how third parties can infer the interests of a user based on their browsing history.

One of the motivations behind the HTTP Referrer Policy [50] was to limit the browsing information that third parties could collect via the HTTP Referer [sic] field. Modern browsers such as Chrome [51] and Firefox [7] use a default policy of "strict-origin-when-cross-origin," that restricts the information transmitted to third parties through the referrer field. These policies, however, can be bypassed through Meta Pixel's URL tracking capabilities. We investigated the magnitude of URL leakage through Meta Pixel by focusing on seven website categories in which the collection of navigation history can reveal sensitive contextual information, as detailed in Table 3. We quantify URL tracking in this population by randomly sampling 2,000 websites from each category.

We instrumented Puppeteer to perform visits to all these websites, simulating a user that browses through several sub-pages of the same website. After landing on each website's homepage, and for any subsequent sub-page, we monitor the state of the page through two indicators: 1) `networkidle2`, which signifies that there have been no more than two network connections in the past 500 ms, and 2) `domcontentloaded`, which signifies that the `DOMContentLoaded` event is fired. When both are true, we assume that the navigation is complete. We use an overall timeout of two minutes, terminating the navigation after this threshold is passed. During this experiment, we inspect the network activity related to Meta Pixel requests and record all websites that have *at least one* sub-page reporting its full URL to Meta. If a website has deployed Meta Pixel through the Core Setup or uses a custom URL filtering mechanism, the `dl` parameter always includes only the root domain, in which case fine-grained tracking is not possible.

When the initial navigation on the homepage completes, the crawler extracts all links towards internal sub-pages from the HTML body of the page (we exclude any links towards third-party domains and non-HTML resources, such as PDF documents), and then randomly chooses one of them and navigates to it. This results in an internal navigation on the website, thus Meta Pixel will be able to capture both the current and previous URL and report them to Meta. After landing on a sub-page, we repeat the same process while avoiding the sub-pages that have already been visited (to prevent any endless loops between the same sub-pages).

The crawler keeps a navigation log with the originating and landing URLs of each visited sub-page. We do not keep track of websites that do not send the full URL to Meta, or websites that employ bot detection mechanisms (e.g., CAPTCHA) that block our crawler. In some rare cases, the crawler may experience timeouts or visit websites that do not contain any links to sub-pages.

Table 3: Information that can be inferred from URL navigation history per website category.

Category	# Websites	Inferred Information
News	6,651	Political leanings; geographic interests; topic preferences; consumption patterns; temporal habits
Politics and Law	2,177	Political ideology; civic engagement; legal concerns; advocacy interests; electoral participation
E-commerce	20,666	Brand loyalty; price sensitivity; purchase frequency; income approximation; shopping patterns
Shopping	13,931	Consumer preferences; spending habits; lifestyle choices; seasonal interests; fashion orientation;
Health	7393	Medical conditions; treatment preferences; medication usage; preventative behaviors; insurance status
Education	13,455	Academic interests; skill development needs; career aspirations; educational level; professional development focus
Real Estate	2,647	Economic status; life stage; location preferences; relocation timing; family needs; investment intent

Table 4: Full navigation history leakage per website category.

Category	# Websites	Navigation History Leakage %
E-commerce	1317	65.85
Shopping	1202	60.10
Education	1183	59.15
Real Estate	1153	57.65
News	1095	54.75
Politics & Law	1094	54.70
Health	226	11.30

7.2 Depth of Tracking

Our findings, displayed in Table 4, indicate that the most aggressive navigation history tracking is performed by "E-commerce" (65.85%) and "Shopping" (60.10%) websites. This is expected, as such websites would be more heavily invested in ad personalization (e.g., due to seasonal sales), and thus would be more involved in understanding in depth the habits of consumers. However, we also find that more than half of the rest of the categories also employ browsing activity tracking, with the most concerning case being "Education" websites (59.15%). This is closely followed by "Real Estate" websites (57.65%), while websites categorized as "News" or "Politics & Law" have a moderate URL leakage (54.7%). For "Health" oriented websites, we find that this kind of navigation tracking is less prominent, but still takes place in 11.3% of them.

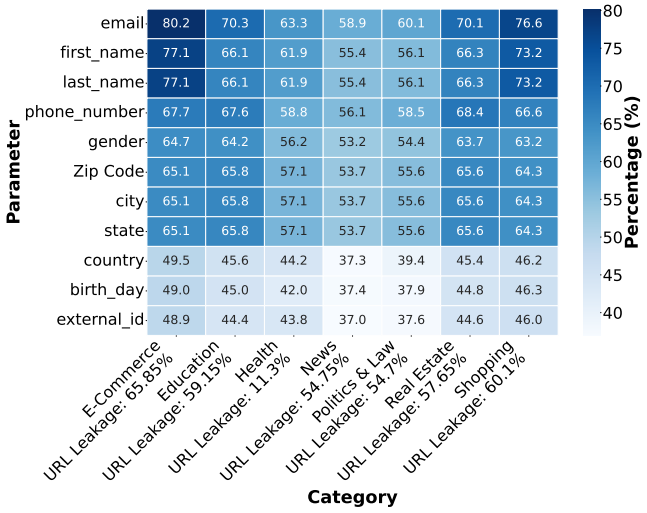

Figure 11: Parameter tracking rate by Meta Pixel across websites leaking users' navigation history.

Finally, we also find that navigation history tracking is closely coupled with Meta Pixel configurations that track at least one PII-related parameter, as shown in Figure 11. More specifically, we find that across all websites that track navigation history, the *email* is tracked with an average rate of 68.5%, and the *phone number* with an average rate of 63.4%. We further complement those empirical findings with a chi-squared test of independence to examine the relationship between PII leakage and navigation tracking. We find a highly significant association (x^2 = 225.97, p < 0.001, Cramer's V = 0.130) between them, confirming our empirical findings. About 58% of PII-collecting websites track a user's full navigation history compared to 44.6% of non-PII-collecting websites.

These findings reveal that websites tend to use the default Meta Pixel configuration, i.e., allowing Meta Pixel to collect both the full navigation journey and PII through forms. Thus, Meta can potentially link browsing activity and contextual information derived from this activity to Facebook users.

7.3 Key Takeaways

We find that more than half of the websites that use Meta Pixel leak the full navigation history of their visitors to Meta. This tracking can aid in revealing not only consumer habits, but also sensitive contextual information (e.g., medical conditions, academic interests) depending on the category of the visited website and the context of each visited sub-page. We investigate the implications of this contextual tracking further in the following section.

8 Case Study: Combining PII and URL Leakage With Page Context

As discussed so far, Meta Pixel requests may directly leak PII parameters and navigation history events. When taking into account the *context* of a page, however, this information can be used to infer additional details about a user's online and offline activities.

In September 2024, a report by the Federal Trade Commission [17] mentioned how tracking technologies such as Meta Pixel collect a concerning amount of information about user interests. Several class action suits [18, 52–54] revealed how the collection of both PII and contextual information violates privacy regulations such as VPPA and HIPAA. These cases identified how additional contextual information regarding a user's habits (e.g., video titles) or pharmaceutical needs (e.g., subscription re-fills, HIV tests, Plan B pills) was collected through Meta Pixel.

Query parameters contained in Meta Pixel requests (reporting an event), such as `content_name` and `pageFeatures`, can reveal additional contextual information about a user's activity. When such information is combined with PII parameters, it can be directly linked to the user's Facebook profile. In this section, through several case studies, we investigate how the combination of page context and tracked PII parameters can reveal additional sensitive information about a user.

8.1 Methodology

Meta for developers [13] defines three interesting attributes that enhance tracking: `pageFeatures`, `content_name`, and `buttonText`. The documentation for `pageFeatures` states that developers can configure Meta Pixel to avoid collecting button clicks or page metadata. The documentation for `content_name` [55] states that it can be used to retrieve the name of the page or a product, but its use is optional. Finally, `buttonText` is captured every time a `SubscribeButtonClick` event is fired, capturing contextual information regarding a button (such as the button's title) [56].

Given this information, we set to explore by conducting some manual experiments what other sensitive information about a user's preferences and actions on a website can be inferred through the data collected by Meta Pixel. From the categorized domains, we manually select a subset of them that could reveal sensitive information about a user's real life needs and activities. We manually visit each website, browse and interact with different elements (products, forms, or other components of the website), and inspect the network activity using Chrome's DevTools.

To fill in any forms present in a page, we use an email account created exclusively for this study, and we try to submit incomplete information to avoid creating any fake accounts (e.g., subscriptions) or unwanted traffic towards the website (e.g., through contact

forms). Senol et al. [57] showed that the `SubscribedButtonClick` event can be triggered to report tracked PII *without* the user having to actually click the form submission button. This allows us to identify Meta Pixel requests leaking form-related PII without the need to actually submit a completed form.

We focused our exploration on websites from the "Health & Fitness" category, as well as some University websites, after we confirmed that they are configured to track at least one PII-related parameter. By analyzing the content of each Meta Pixel request after an interaction with a page, we can infer a "story" about the user. Then, when performing any interaction with a form leaking PII to Meta through Meta Pixel, we showcase how this story can be potentially associated to the user's Facebook account.

8.2 User Profile Build Up

In one of the most concerning cases we encountered, a website that offers end-of-life services has Meta Pixel configured for both PII and activity tracking. When a user selects a specific facility, the location of this facility is shared in the `st` parameter of the request, reported by the `SubscribeButtonClick` event. This is accompanied by details regarding the facility and the services it offers, through the `pageFeatures` and `buttonText` parameters, respectively. When a user submits the contact (or "Stay Connected") form, their email is also propagated to Meta through a Meta Pixel request. This means sensitive contextual information regarding the type of service (end of life) and the location of the facilities the user is interested in, can be linked to their Facebook profile.

Another interesting case involves a childcare network that offers families the ability to locate daycare or nannies near their location. When a user interested in the service explores a daycare station by clicking on an element inside that sub-page ("See Credentials"), the full name of the daycare station and its state and ZIP code are captured by Meta Pixel and reported to Meta through the `pageFeatures` query parameter. When the user chooses to contact the daycare (through an online form), they are required to log in to their account on the platform. Through the login form, Meta Pixel captures and reports the email address of the user to Meta. Those actions can be linked to a Facebook user revealing insights regarding: (i) their family status, i.e., having kids, (ii) the location where they are searching for a daycare, and (iii) the full name of the daycare.

In total, we identified 10 different types of websites and two universities in which Meta Pixel can collect detailed information about a user's interests and needs, as reported in Table 5. For instance, when looking into university websites, we find that they have a strong tendency towards tracking full browsing activity (through the `rl`/`dl` parameters). Moreover, we encounter cases where the interest of a user regarding an area of study is also reported through the `buttonText` parameter. When submitting a contact form (e.g., "Contact Admissions" or "Request Information"), we find the email of the user also being reported to Meta. This means that insights regarding (i) academic status (field of study, educational level), (ii) research interests, and (iii) university locations can be linked to Facebook user accounts.

By enabling Meta Pixel to report a user's complete browsing journey (*Full URLs*), it becomes possible to infer the user's specific domains of interest and online behavior. Moreover by tracking and reporting PII (*PII leakage*), Meta Pixel facilitates the ability to match this navigation history to a Facebook account. Recorded actions (e.g., button clicks) provide a deeper understanding of the context of the elements that the user is interested in during their browsing. Finally, all these attributes provide the ability to infer a "story" regarding the offline needs and characteristics of the user as displayed by the column *User-tailored Needs*.

8.3 Key Takeaways

Meta offers the ability to limit the types of data collected through Meta Pixel, but not all website administrators consider any privacy implications when configuring their Meta Pixel instances. Our case study shows that Meta Pixel can collect a user's activity at a concerning level of detail, which, when coupled with tracked PII parameters, can be linked to their Facebook account. While this information aids targeted ad attribution, its context can reveal detailed insights of a user's offline habits or needs.

9 Related Work

Online user activity tracking as well as ad conversion and attribution methods have been extensively studied over the past decade. User profiling and conversion attribution is an active research field, with many studies exploring the capabilities of elaborate profile building by advertising third parties, as well the monetary effects of those processes [58–64]. Activity attribution has been also widely studied through third party cookies [65–72] or other more sophisticated mechanisms, such as fingerprinting, CNAME cloacking, and cookie synchronization [34, 73–81]. On one of the most recent tracking techniques, uncovered by Pantelina et al. [82], leverages browser synchronization for cross-device tracking.

More recently, advertisers started relying on first-party cookies to perform persistent tracking of users while browsing the web, leading to a large body of studies [24, 35, 41, 83, 84] exploring the evolution of entities, methods, and roles of this new tracking ecosystem. Shaoor et al. [85] also explored the inter-connectivity of those cookies among different advertising entities. Among all works on web tracking, Meta (formerly known as Facebook), was identified as an active and dominant actor across third party trackers existing on the online advertising ecosystem.

Personal Identifiable Information leakage has also been widely studied [20–22, 86, 87], focusing not only on cleartext PII leakage, but also leakage of encrypted or hashed values. More specifically, Senol et al. [57] performed a large-scale study on Tranco's top 100K websites, revealing that email leakage on web forms can occur even without the need for a user to actually submit the form. Another large-scale study by Kubicek et al. [88] automated the detection and submission of registration forms on Tranco's 1M websites, inferring PII leakage by inspecting the emails received on the evaluated accounts. Moreover, Venkatadri et al. [89] showed how PII collected by Meta Pixel can be used by adversaries using Meta's Ad Manager to directly de-anonymize users. Compared to our work, these studies focused on a specific type of web forms or category of websites, and relied on the actual submission of PII values (automatically or manually) while inspecting the generated network traffic.

Table 5: Detailed activity tracking and potential activity attribution for 10 sensitive websites.

Type of website	Full URLs	PII leakage (observed)	Additional Components	User-tailored Needs
End-of-life services	Yes	email, state	`pageFeatures` and `buttonText`: Type of service and location of facility	Type of service for end-of-life and location of the facility the user is interested in
Healthcare Retailer	Yes	phone	`buttonText`: Detailed title of medication	A user searching for Betahistine medication (24mg tablets)
Optical Retailer	Yes	email, first/last name	`buttonText`: Driver's license eye examination appointment	City of clinic, type of appointment
Dermatology clinic	Yes	email, gender	`buttonText`: Skin treatment type	Skin condition that the user looks for consultation sessions
Veterinary clinic	Yes	phone	`buttonText`: Location and Doctor's specialty and name, `buttonFeatures`: time of appointment	The location of the clinic, name and doctor information and time of appointment
Senior Care products	Yes	email, phone, first/last name, city, state, ZIP code	`content_name` and `pageFeatures`: Type and specific name of the product	The user is interested in senior care products and the specific conditions each product is responsible for
Childcare network	Yes	email	`pageFeatures`: Name and location of the daycare center	Name and detailed location (city, state, ZIP code) of daycare center or nanny the user is looking for
Healthcare products	Yes	email	`pageFeatures`: The name of the product	The issues that the user potentially is facing (e.g., sexual health), their gender and the product of interest
Pet nutrition	Yes	email, phone, first/last name	`content_name` and `pageFeatures`: Name and type of product	What type of pet the user has based on the products search/bought
Fitness store	Yes	email, phone, first name	`pageFeatures`: Supplement product name	User supplementary product preferences
University	Yes	email	`buttonText`: area of study, `pageFeatures`: academic level of interest	User's academic interests in this university and academic level
University	Yes	email, first/last name	`buttonText`: area of study, `pageFeatures`: location of the university	User's academic interests in this university

In addition to commonly referenced identifiers, some works also tackle the issue of URL leakage. More specifically, Sampsa et al. [90] identified URLs as part of sensitive information shared with third parties. Ouwerkerk et al. [91] showcase how despite the implementation of strict referrer policies by default in modern browsers, third parties can still retrieve detailed activity information by collecting the URLs and referrer links on 3K e-commerce websites. Finally, Matic et al. [39] designed a classifier that can identify sensitive URLs even in websites that are not categorized as sensitive, based on the nature of information contained within the URL itself.

Meta Pixel's ability to share PII with Meta raises a significant privacy concern when this type of leakage happens on websites belonging to sensitive categories. The Markup [92] revealed that Meta Pixel was actively leaking sensitive patient information from US based hospitals. Moreover, there is a large body of studies [29, 93–98] that study the presence of third-party trackers on websites belonging to specific sensitive categories (e.g., schools, religious websites, hospitals, health portals). Huo et al. [99] uncovered that 14% of the studied health portals leaked sensitive personal identifiers, such as a user's email or phone number, to third parties. Nayanamana et al. [100] studied the prevalence of such trackers on religious websites and applications, identifying leakage of sensitive information such as names, addresses, and emails.

10 Conclusion

Meta Pixel, while used by websites to improve the effectiveness of the rendered ads, also enables Meta to collect user information and PII (such as email addresses and phone numbers), along with users' detailed browsing activity and interactions when browsing a website. Such PII can be used for conversion attribution of anonymous users to their Facebook profile without requiring any intermediate interaction with the platform.

Contributions. We propose a highly scalable approach to measure PII leakage from Meta Pixel by passively analyzing its configuration file. Our hybrid approach allows us to identify the use of Meta Pixel in 18.7% of Tranco's 1M popular websites. The PII captured by Meta Pixel includes a user's email address, phone number, first and last name, gender, age, and geographic characteristics. By studying the structural components of Meta Pixel, we observe PII leakage in 65.6% of these websites. Our work complements and improves upon previous studies on PII leakage by introducing a new inference method. By relying on passively inspecting the configuration of Meta Pixel, we do not need to actually submit any information through forms and then inspect the generated traffic to identify leakage. Consequently, our methodology is not constrained to a particular category of forms (e.g., login) or specific types of websites.

We also study the ability of Meta Pixel to report the full browsing journey of a user, even in websites that are categorized as sensitive. Through two query parameters (`dl`/`rl`), Meta Pixel is able to track the order of the URLs that a user browsed in a given website, collecting detailed navigation history. When coupling this information with PII, Meta can potentially infer the habits and needs of Facebook users. This is especially problematic when coupled with components that further reveal the context behind each action of a user on a website, potentially exposing further their offline needs and habits.

Future Work. Meta Pixel is a prominent component within a broader ecosystem of tracking technologies, each employing distinct methods and configurations for conversion tracking. Future research could investigate whether other similar platforms (e.g., TikTok, Twitter, and Snapchat Pixels, Google Analytics) exhibit comparable configuration patterns and PII leakage behaviors. Furthermore, our experiments on URL leakage and the accompanying case study highlight that privacy risks are amplified when PII leakage is combined with contextual information about users. These findings underscore the need for a comprehensive approach to assess the extent of such tracking, moving beyond isolated case investigations, to establish a systematic framework for evaluating its pervasive impact across the web.

Availability

Our prototype implementation and complete data set are publicly available through https://github.com/pasxalisbekos/PIIxel_Leaks.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. We also thank Evangelos Markatos for his useful insights and comments on earlier versions of this work.

References

[1] General data protection regulation (GDPR). https://gdpr-info.eu/.
[2] California consumer privacy act (CCPA). https://cppa.ca.gov/regulations/.
[3] Health insurance portability and accountability act (HIPAA). https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html.
[4] Children's online privacy protection rule (COPPA). https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa.
[5] Video privacy protection act (VPPA). https://www.congress.gov/bill/100th-congress/senate-bill/2361.
[6] Google Inc. Privacy sandbox. https://developers.google.com/privacy-sandbox/cookies, 2024.
[7] Mozilla. Referrer-policy header. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy.
[8] David Martin, Hailin Wu, and Adil Alsaid. Hidden surveillance by web sites: Web bugs in contemporary use. *Communications of the ACM*, pages 258–264, 2003.
[9] Malcolm Higgins. What is a tracking pixel, and how does it work? https://nordvpn.com/blog/what-is-a-tracking-pixel/, 2023.
[10] Federal Trade Commission. Lurking beneath the surface: Hidden impacts of pixel tracking. https://www.ftc.gov/policy/advocacy-research/tech-at-ftc/2023/03/lurking-beneath-surface-hidden-impacts-pixel-tracking, 2023.
[11] The ABCs of tracking pixels: What they are and how they work. https://improvado.io/blog/what-is-tracking-pixel.
[12] Meta Pixel. https://www.facebook.com/business/tools/meta-pixel.
[13] Advanced matching. https://developers.facebook.com/docs/meta-pixel/advanced/advanced-matching.
[14] Georgios Kontaxis, Michalis Polychronakis, Angelos D. Keromytis, and Evangelos P. Markatos. Privacy-preserving social plugins. In *Proceedings of the 21st USENIX Security Symposium*, pages 631–646, 2012.
[15] Austrian DSB: Meta tracking tools illegal. https://noyb.eu/en/austrian-dsb-meta-tracking-tools-illegal, 2023.
[16] Swedish Data Protection Agency. Beslut efter tillsyn enligt dataskyddsförordningen mot Avanza Bank AB. https://www.imy.se/globalassets/dokument/beslut/2024/beslut-tillsyn-avanza.pdf, 2024.
[17] Federal Trade Commission. FTC staff report finds large social media and video streaming companies have engaged in vast surveillance of users with lax privacy controls and inadequate safeguards for kids and teens. https://www.ftc.gov/news-events/news/press-releases/2024/09/ftc-staff-report-finds-large-social-media-video-streaming-companies-have-engaged-vast-surveillance, 2024.
[18] AMC to pay $8m for allegedly violating 1988 law with use of Meta Pixel. https://arstechnica.com/tech-policy/2024/02/amc-to-pay-8m-for-allegedly-violating-1988-law-with-use-of-meta-pixel/, 2024.
[19] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.
[20] Oleksii Starov, Phillipa Gill, and Nick Nikiforakis. Are you sure you want to contact us? Quantifying the leakage of PII via website contact forms. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 20–33, 2016.
[21] Ha Dao and Kensuke Fukuda. Alternative to third-party cookies: Investigating persistent PII leakage-based web tracking. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*, pages 223–229, 2021.
[22] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. I never signed up for this! Privacy implications of email tracking. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 109–126, 2018.
[23] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 499–518, 2018.
[24] Paschalis Bekos, Panagiotis Papadopoulos, Evangelos P. Markatos, and Nicolas Kourtellis. The hitchhiker's guide to Facebook web tracking with invisible pixels and click IDs. In *Proceedings of the ACM Web Conference*, pages 2132–2143, 2023.
[25] Meta. Customer information parameters. https://developers.facebook.com/docs/marketing-api/conversions-api/parameters/customer-information-parameters/, 2025.
[26] Meta. Set up automatic advanced matching in Meta events manager. https://www.facebook.com/business/help/1993001664341800?id=1205376682832142, 2025.
[27] Meta. About core setup. https://www.facebook.com/business/help/124742407297678, 2025.
[28] Google Inc. Google tag management. https://tagmanager.google.com/.
[29] Zahra Moti, Asuman Senol, Hamid Bostani, Frederik Zuiderveen Borgesius, Veelasha Moonsamy, Arunesh Mathur, and Gunes Acar. Targeted and troublesome: Tracking and advertising on children's websites. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 1517–1535, 2024.
[30] Audrey Randall, Peter Snyder, Alisha Ukani, Alex C. Snoeren, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. Measuring UID smuggling in the wild. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*, pages 230–243, 2022.
[31] Salim Chouaki, Oana Goga, Hamed Haddadi, and Peter Snyder. Understanding the privacy risks of popular search engine advertising systems. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 370–382, 2023.
[32] Sebastian Zimmeck, Daniel Goldelman, Owen Kaplan, Logan Brown, Justin Casler, Judeley Jean-Charles, Joe Champeau, and Hamza Harkous. Website data transparency in the browser. In *Proceedings of the 24th Privacy Enhancing Technologies Symposium (PETS)*, 2024.
[33] Puppeteer. https://pptr.dev/, 2018.
[34] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In *Proceedings of the Web Conference*, pages 2130–2141, 2021.
[35] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. Cookie swap party: Abusing first-party cookies for web tracking. In *Proceedings of the Web Conference*, pages 2117–2129, 2021.
[36] Facebook location services. https://www.facebook.com/help/337244676357509, 2025.
[37] External ID. https://developers.facebook.com/docs/marketing-api/conversions-api/parameters/external-id/, 2024.
[38] SafeDNS category check. https://www.safedns.com/check-website.
[39] Srdjan Matic, Costas Iordanou, Georgios Smaragdakis, and Nikolaos Laoutaris. Identifying sensitive URLs at web-scale. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 619–633, 2020.
[40] Iskander Sanchez-Rola, Davide Balzarotti, and Igor Santos. BakingTimer: privacy analysis of server-side request processing time. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC)*, pages 478–488, 2019.
[41] Savino Dambra, Iskander Sanchez-Rola, Leyla Bilge, and Davide Balzarotti. When Sally met trackers: Web tracking from the users' perspective. In *Proceedings of the 31st USENIX Security Symposium*, pages 2189–2206, 2022.
[42] The Markup. Facebook watches teens online as they prep for college. https://themarkup.org/pixel-hunt/2023/11/22/facebook-watches-teens-online-as-they-prep-for-college, 2023.
[43] Shanti Das. We didn't click 'consent' on any gambling website. So how did Facebook know where we'd been? https://www.theguardian.com/technology/2025/feb/08/we-didnt-click-consent-on-any-gambling-website-so-how-did-facebook-know-where-wed-been, 2025.
[44] Adam Satariano. What a gambling app knows about you. https://www.nytimes.com/2021/03/24/technology/gambling-apps-tracking-sky-bet.html.
[45] José González Cabañas, Ángel Cuevas, and Rubén Cuevas. Unveiling and quantifying Facebook exploitation of sensitive personal data for advertising purposes. In *Proceedings of the 27th USENIX Security Symposium*, pages 479–495, 2018.
[46] Relja Denic. Data-driven personalization explained [with examples]. https://www.plainlyvideos.com/blog/data-driven-personalization.
[47] Website personalization: Definition, strategies and tips. https://www.indeed.com/career-advice/career-development/website-personalization.
[48] SuperOffice. 11 personalization strategies for marketing, sales and customer support teams. https://www.superoffice.com/blog/personalization/.
[49] Ben Weinshel, Miranda Wei, Mainack Mondal, Euirim Choi, Shawn Shan, Claire Dolin, Michelle L. Mazurek, and Blase Ur. Oh, the places you've been! User reactions to longitudinal transparency about third-party web tracking and inferencing. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 149–166, 2019.
[50] Ivan Dolnák. Implementation of referrer policy in order to control HTTP referer header privacy. In *Proceedings of the 15th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2017.
[51] A new default Referrer-Policy for Chrome - strict-origin-when-cross-origin. https://developer.chrome.com/blog/referrer-policy-new-chrome-default.
[52] Fan v. NBA Properties, Inc. et al. - 3:23-cv-05069. https://www.classaction.org/media/fan-v-nba-properties-inc-et-al.pdf, 2023.
[53] Lee v. Plex, Inc., 5:24-cv-02386, (N.D. Cal.). https://www.courtlistener.com/docket/68459584/lee-v-plex-inc/, 2024.
[54] Castillo v. Costco Wholesale Corp., No. 2:23-cv-01548-JHC. https://www.classaction.org/media/castillo-et-al-v-costco-wholesale-corporation.pdf, 2023.

[55] Meta Pixel Reference. https://developers.facebook.com/docs/meta-pixel/reference/, 2025.

[56] Automatic Facebook pixel events. https://www.pixelyoursite.com/major-facebook-pixel-update-automatic-facebook-pixel-events.

[57] Asuman Senol, Gunes Acar, Mathias Humbert, and Frederik Zuiderveen Borgesius. Leaky forms: A study of email and password exfiltration before form submission. In *Proceedings of the 31st USENIX Security Symposium*, pages 1813–1830, 2022.

[58] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. Tracing information flows between ad exchanges using retargeted ads. In *Proceedings of the 25th USENIX Security Symposium*, pages 481–496, 2016.

[59] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez Rodriguez, and Nikolaos Laoutaris. If you are not paying for it, you are the product: How much do advertisers pay to reach you? In *Proceedings of the Internet Measurement Conference (IMC)*, pages 142–156, 2017.

[60] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. The cost of digital advertisement: Comparing user and advertiser views. In *Proceedings of the World Wide Web Conference*, pages 1479–1489, 2018.

[61] David Rebollo-Monedero Silvia Puglisi and Jordi Forné. On web user tracking of browsing patterns for personalised advertising. *International Journal of Parallel, Emergent and Distributed Systems*, pages 502–521, 2017.

[62] Hiroaki Kikuchi and Ayaka Aoyama. Targeted ads analysis: What are the most targeted personas? In *Proceedings of the IEEE International Conference on Big Data (BigData)*, pages 5512–5518, 2023.

[63] Juan Miguel Carrascosa, Jakub Mikians, Ruben Cuevas, Vijay Erramilli, and Nikolaos Laoutaris. I always feel like somebody's watching me: Measuring online behavioural advertising. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2015.

[64] Pushkal Agarwal, Sagar Joglekar, Panagiotis Papadopoulos, Nishanth Sastry, and Nicolas Kourtellis. Stop tracking me bro! Differential tracking of user demographics on hyper-partisan websites. In *Proceedings of The Web Conference*, pages 1479–1490, 2020.

[65] Richard Gomer, Eduarda Mendes Rodrigues, Natasa Milic-Frayling, and M.C. Schraefel. Network analysis of third party tracking: User exposure to tracking cookies through search. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 549–556, 2013.

[66] Abdelberi Chaabane, Mohamed Ali Kaafar, and Roksana Boreli. Big friend is watching you: analyzing online social networks tracking capabilities. In *Proceedings of the ACM Workshop on Online Social Networks (WOSN)*, pages 7–12, 2012.

[67] Xuehui Hu and Nishanth Sastry. Characterising third party cookie usage in the EU after GDPR. In *Proceedings of the 10th ACM Conference on Web Science*, pages 137–141, 2019.

[68] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M. Pujol. Tracking the trackers. In *Proceedings of the 25th International Conference on the World Wide Web*, pages 121–132, 2016.

[69] Arnold Roosendaal. *We Are All Connected to Facebook … by Facebook!*, pages 3–19. Springer, 2012.

[70] Arnold Roosendaal. Facebook tracks and traces everyone: Like this! Legal Studies Research Paper 03/2011, Tilburg Law School, November 2010.

[71] Ali Rasaii, Devashish Gosain, and Oliver Gasser. Thou shalt not reject: Analyzing accept-or-pay cookie banners on the web. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, pages 154–161, 2023.

[72] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. An empirical study of web cookies. In *Proceedings of the 25th International Conference on the World Wide Web*, pages 891–901, 2016.

[73] Xu Lin, Frederico Araujo, Teryl Taylor, Jiyong Jang, and Jason Polakis. Fashion faux pas: Implicit stylistic fingerprints for bypassing browsers' anti-fingerprinting defenses. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 987–1004, 2023.

[74] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 541–555, 2013.

[75] Imane Fouad, Cristiana Teixeira Santos, Arnaud Legout, and Nataliia Bielova. My cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 79–98, 2022.

[76] Soumaya Boussaha, Lukas Hock, Miguel A Bermejo-Agueda, Rubén Cuevas, Angel Rumin, David Klein, Martin Johns, Luca Compagna, Daniele Antonioli, and Thomas Barber. FP-tracer: Fine-grained browser fingerprinting detection via taint-tracking and entropy-based thresholds. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 540–560, 2024.

[77] Peter Eckersley. How unique is your web browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies (PETS)*, 2010.

[78] Imane Fouad, Cristiana Santos, and Pierre Laperdrix. The devil is in the details: Detection, measurement and lawfulness of server-side tracking on the web. *Proceedings on Privacy Enhancing Technologies*, pages 450–465, 2024.

[79] Ha Dao, Johan Mazel, and Kensuke Fukuda. CNAME cloaking-based tracking on the web: Characterization, detection, and protection. *IEEE Transactions on Network and Service Management*, pages 3873–3888, 2021.

[80] Pierre Laperdrix, Oleksii Starov, Quan Chen, Alexandros Kapravelos, and Nick Nikiforakis. Fingerprinting in style: Detecting browser extensions via injected style sheets. In *Proceedings of the 30th USENIX Security Symposium*, pages 2507–2524, 2021.

[81] Ismael Castell-Uroz and Pere Barlet-Ros. A first look into utiq: Next-generation cookies at the ISP level. In *Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 315–320, 2024.

[82] Pantelina Ioannou and Elias Athanasopoulos. Been here already? Detecting synchronized browsers in the wild. In *Proceedings of the 8th IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 913–927, 2023.

[83] Shaoor Munir, Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. CookieGraph: Understanding and detecting first-party tracking cookies. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 3490–3504, 2023.

[84] Nurullah Demir, Daniel Theis, Tobias Urban, and Norbert Pohlmann. Towards understanding first-party cookie tracking in the field. In *Proceedings of GI SECURITY*, pages 19–34, 2022.

[85] Nurullah Demir, Daniel Theis, Tobias Urban, Norbert Pohlmann, and AG Networks. Towards understanding CNAME based first-party cookie tracking in the field. https://api.semanticscholar.org/CorpusID:246485578.

[86] Justin Brookman, Phoebe Rouge, Aaron Alva, and Christina Yeung. Cross-device tracking: Measurement and disclosures. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 133–148, 2017.

[87] Gunes Acar, Steven Englehardt, and Arvind Narayanan. No boundaries: Data exfiltration by third parties embedded on web pages. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, pages 220–238, 2020.

[88] Karel Kubicek, Jakob Merane, Ahmed Bouhoula, and David Basin. Automating website registration for studying GDPR compliance. In *Proceedings of the ACM Web Conference*, pages 1295–1306, 2024.

[89] Giridhari Venkatadri, Athanasios Andreou, Yabing Liu, Alan Mislove, Krishna P. Gummadi, Patrick Loiseau, and Oana Goga. Privacy risks with Facebook's PII-based targeting: Auditing a data broker's advertising interface. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 89–107, 2018.

[90] Sampsa Rauti. Lessons learned from studying third-party data leaks in web services. In *Proceedings of the 8th International Conference on Information Systems Engineering*, pages 125–129, 2024.

[91] Thomas van Ouwerkerk. Evading the policy: A measurement on referrer policy circumvention in 3k e-commerce websites. Master's thesis, University of Amsterdam, 2022.

[92] Facebook is receiving sensitive medical information from hospital websites. https://themarkup.org/pixel-hunt/2022/06/16/facebook-is-receiving-sensitive-medical-information-from-hospital-websites.

[93] Xiufen Yu, Nayanamana Samarasinghe, Mohammad Mannan, and Amr Youssef. Got sick and tracked: Privacy analysis of hospital websites. In *Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 278–286, 2022.

[94] Jake Chanenson, Brandon Sloane, Navaneeth Rajan, Amy Morril, Jason Chee, Danny Yuxing Huang, and Marshini Chetty. Uncovering privacy and security challenges in K-12 schools. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2023.

[95] Sampsa Rauti, Robin Carlsson, Sini Mickelsson, Tuomas Mäkilä, Timi Heino, Elina Pirjatanniemi, and Ville Leppänen. Analyzing third-party data leaks on online pharmacy websites. *Health Technology*, pages 375–392, 2024.

[96] Sampsa Rauti, Esko Vuorinen, Robin Carlsson, and Panu Puhtila. Data leaks to third-party services on medical websites. In *Proceedings of the 16th International Conference on Security of Information and Networks (SIN)*, 2023.

[97] Alexander R. Zheutlin, Joshua D. Niforatos, and Jeremy B. Sussman. Data-tracking among digital pharmacies. *Annals of Pharmacotherapy*, pages 958–962, 2022.

[98] Alan B. Friedman, Raina M. Merchant, Atheendar Maley, Khaled Farhat, Kristin Smith, Julia Felkins, Robert E. Gonzales, Lauren Bauer, and Matthew S. McCoy. Widespread third-party tracking on hospital websites poses privacy risks for patients and legal liability for hospitals. *Health Affairs (Millwood)*, pages 508–515, 2023.

[99] Mingjia Huo, Maxwell Bland, and Kirill Levchenko. All eyes on me: Inside third party trackers' exfiltration of PHI from healthcare providers' online systems. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, pages 197–211, 2022.

[100] Nayanamana Samarasinghe, Pranay Kapoor, Mohammad Mannan, and Amr Youssef. No salvation from trackers: Privacy analysis of religious websites and mobile apps. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS International Workshops*, pages 151–166, 2023.