PixelConfig: Longitudinal Measurement and Reverse-Engineering of Meta Pixel Configurations

Paper #9, 13 pages body, 18 pages total

ABSTRACT

Tracking pixels are widely used to optimize online ad campaigns through personalization, re-targeting, and conversion tracking. While prior research has primarily focused on detecting the prevalence of tracking pixels, limited attention has been given to variations in their configurations across websites. A tracking pixel may be configured differently on different websites. In this paper, we propose a differential analysis framework: `PixelConfig`, to reverse-engineer tracking pixel configurations across websites. Using this framework, we investigate three types of Meta Pixel configurations: activity tracking (i.e., what a user is doing on a website), identity tracking (i.e., who a user is or who the device is associated with), and tracking restrictions (i.e., mechanisms to limit the sharing of potentially sensitive information).

Using data from the Internet Archive's Wayback Machine, we analyze and compare Meta Pixel configurations on 18K health-related websites with a control group of the top 10K websites from 2017 to 2024. We find that features for activity tracking, such as automatic events (where the Pixel automatically collects button clicks and page metadata), and identity tracking, such as first-party cookies (which are unaffected by third-party cookie blocking), reached adoption rates of up to 98.5% and 98.8%, respectively, largely driven by the Pixel's default settings. We also observe the Pixel to be used to track potentially sensitive information, such as user interactions related to booking medical appointments or clicking buttons associated with specific medical conditions (e.g., erectile dysfunction) on health-related websites. Tracking restriction features, such as the core setup, are typically configured by Meta on up to 37% of health websites and 15% of control websites. Overall, we find that, even when implemented, these tracking restriction features are not comprehensive, can be ineffective, and can be circumvented.

1 INTRODUCTION

Online platforms such as Google, Meta, and TikTok provide tracking pixels that advertisers can install on their websites. The information collected through these tracking pixels is used to run ads. For example, an advertiser may want to run ads targeting Facebook or Instagram users who took a specific action (e.g., added an item to cart) on their website. This is called retargeting [59]. Advertisers can also ask an online platform to target users who "look like" a particular set of users who visited their website. This is called lookalike targeting [67]. Advertisers can also run ad campaigns where the platform automatically configures targeting parameters to optimize specific outcomes (e.g., item purchase). This is called conversion optimization, which relies on tracking outcomes to auto-adjust targeting parameters [31, 60].

As our society's reliance on the Internet has grown, online advertising has become the dominant model for reaching customers. Online advertising in the US is set to exceed $300 billion in 2024 [29], with Google and Meta accounting nearly half of the online ad spend. Almost every business has an online presence today and most of them run ad campaigns to reach new customers and engage with the existing ones. This means that websites install tracking pixels provided by platforms to track their customers [42]. In fact, over 90% of websites include at least one tracking pixel today [17, 20].

Tracking pixels have advanced in their capabilities over the years. Tracking pixels used to be simple 1x1 image elements loaded from a tracking server. When a user visited a webpage where the pixel was installed, the user's IP address, cookies, and information about the page's URL was shared with the tracker. Modern tracking pixels (now often referred to as tags instead of pixels) increasingly rely on JavaScript to gather a richer set of information that is available via various web APIs. JavaScript-based tracking pixels can automatically or through additional configuration collect much richer set of information. Therefore, modern tracking pixels can be configured and customized in a variety of ways to use different tracking features. For example, modern tracking pixels can be configured to use first-party cookies [66, 92] or share information about button clicks on a page [33, 64].

As tracking has become more widespread, the research community has conducted large-scale studies to measure the prevalence of tracking pixels [9, 20, 42, 77]. For example, nearly a decade ago, Englehardt and Narayanan found third-party tracking pixels on more than 80% of top-million websites. Using data archived by Internet Archive's Wayback Machine, Lerner et al. [42] reported increasing prevalence, number, and variety of tracking pixels between 1996 and 2016. There is also ample research on tracking pixels that engage in a specific kind of tracking such as fingerprinting [8, 37, 38, 98] and session replay [1, 91, 102].

Prior research on online tracking measurement has primarily focused on detecting the presence of tracking pixels across the web. These studies have overlooked the fact that the same tracking pixel installed on two websites may be

configured differently. Since modern JavaScript-based tracking pixels offer a myriad of configurable features, simply detecting their presence (i.e., whether a pixel is present or absent on a page) is insufficient to understand the full scope of their tracking capabilities.

To address this gap, we aim to study different configurations of the same tracking pixel across the web. However, studying tracking pixel configurations at scale is technically challenging. First, public documentation of tracking pixels is often vague and incomplete, making it difficult for researchers to enumerate different possible configurations and develop methods to study them at scale. Second, studying tracking pixel configurations in the advertiser portal could be helpful in theory; however, that requires having access to advertiser accounts – which is infeasible. Third, dynamic network traffic analysis of tracking pixels through website crawling suffers from completeness issues. This is because exhaustive website crawling would require (a) running a deep crawler to identify relevant pages on a website, and (b) simulating robust user interactions such as typing in form fields and clicking on buttons to trigger various tracking features. Additionally, dynamic analysis restricts the scope to live installation of pixel on the website, limiting retrospective analysis of configuration changes over time. Fourth, static analysis of JavaScript source code to study tracking pixel configuration is challenging because the code is typically obfuscated and minified.

To address these challenges, we present `PixelConfig` – a reverse-engineering framework that combines static and dynamic approaches to perform a differential analysis of tracking pixel configurations deployed across the web and over time. This involves iteratively patching different tracking pixel configurations of a given website for ablation analysis of specific parts of the code pertaining to specific configurations. Next, we compare the network traffic of the original pixel configuration code and patched pixel configuration code replayed in the browser. We further create a developer account and install the tracking pixel on a test website to enumerate different possible configurations. These differences in the pixel's source code and network traffic allow us to pin-point the exact part of the pixel source code that reflects a particular pixel configuration.

In this paper, we apply `PixelConfig` to investigate Meta Pixel configurations across the web from 2017 to 2024. We select Meta Pixel because it is arguably the most sophisticated tracking pixel today and Meta is the second-largest digital advertising company in the world by revenue after Google [77]. Building on the approach introduced by Lerner et al. [42] and most recently used by Bahrami et al. [8], we rely on Meta Pixel's source code archived by Internet Archive's Wayback Machine. The Wayback Machine has archived Meta Pixel's source code that is installed across millions of websites over the years [7].

Using the data from the Wayback Machine, we conduct a longitudinal analysis of the configurations of Meta Pixel installed on 18K health websites and a control group of top-10K websites. Tracking pixels installed on health websites can collect potentially sensitive information when users search for specific medical conditions, schedule medical appointments, or access medical records. Health information is considered more sensitive [41, 50, 94] and laws/regulations (e.g., HIPAA [35]) provide heightened protections for health information. Using `PixelConfig`, we study the adoption and configuration of the following three categories of Meta Pixel features:

- **Activity Tracking:** Meta Pixel tracks user activity – what a user is doing on a website – through about 20 default and standard events such as `PageView`, `Purchase`, and `AddToCart` that can be automatically setup. Meta Pixel also allows advertisers to define custom events. Through a variety of standard and custom events that can be configured in a number of ways, Meta Pixel can track a wide range of information such as page URL, referrer URL, page title, page meta-data, button text, form fields, etc.
- **Identity Tracking:** Meta Pixel tracks user identifying information – who a user is or who the device is associated with – using PII (e.g., `email`, `phone`, `firstname`, `lastname`) harvested from the website, first- and third-party cookies containing account and device identifiers, as well as IP address and detailed device properties such as in user-agent.
- **Tracking Restrictions:** Meta Pixel has recently introduced tracking restrictions that can be configured to limit the information shared with Meta. These restrictions include recently introduced Core Setup [69] as well as blacklisted and sensitive keys that limit tracking by Meta Pixel.

We aim to address the following research questions:

(1) How has Meta Pixel configuration evolved over time? Over the years, Meta Pixel has offered various new features and capabilities. It is unknown how these features and capabilities have been adopted and configured. Understanding tracking pixel configuration over the time can shed light on how online platforms and advertisers have responded to new features introduced by Meta as well as impact of new privacy laws, regulations, and enforcement actions.

(2) How does Meta Pixel configuration compare across health-related and a control group of top websites? A tracking pixel installed on a generic website (e.g., a pet supply store) presents a different privacy risk as compared to the same pixel installed on a health website, due to its sensitive nature and heightened protections under laws/regulations. The collection and sharing of potentially sensitive health information by tracking pixels has invited scrutiny from regulators such as HHS and FTC [4, 24].

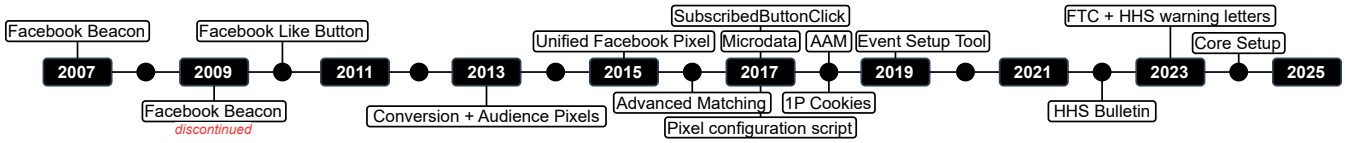


Figure 1: Timeline depicting key events in the evolution of Meta Pixel: features, configurations, regulatory actions.

Our findings show that the adoption of tracking features of Meta Pixel is driven by their default settings. For example, Meta Pixel's automatic event tracking feature that collects button clicks and page metadata information was adopted by up to 98.5% of the websites. As another example, Meta Pixel's first-party cookie feature that is unaffected by third-party cookie blocking was adopted by up to 98.8% of the websites. This is because both of these features were configured by default. Other features that were not turned on by default but still were widely adopted were driven by Meta's messaging and nudging. For example, Automatic Advanced Matching where Meta Pixel extracts user identifying information such as email addresses and phone numbers from form fields was adopted by nearly 50% of the websites due to Meta's messaging around its ability to improve effectiveness of ad campaigns given the adoption of anti-tracking features being introduced in web browsers [57, 73].

Meta Pixel has also introduced features for tracking restriction. These tracking restriction features include blacklisted and sensitive keys (to block tracking of certain URL parameters) in 2020 and 2021 respectively, and more recently Core Setup in 2023. Our findings show that these tracking features are typically enabled and configured by Meta. While these these features were adopted on more health websites as compared to control group of website, their adoption on health websites remained relatively low (36% and 37% for sensitive keys and Core Setup respectively). We also find that these restrictions, even when implemented, were not always effective (e.g., not all potentially sensitive URL parameters were blocked and not all Pixels on a health website were placed in Core Setup).

2 BACKGROUND & RELATED WORK

2.1 Introduction to Meta Pixel

Meta Pixel (initially known as Facebook Pixel) is a JavaScript based tracking pixel that allows Meta to track information about certain actions taken by a user visiting a website [56]. The information tracked by Meta Pixel on a website is used to optimize ad campaigns on Meta. For example, an advertiser may run ad campaigns to retarget users on Instagram who previously added an item to cart but did not ultimately complete the purchase on the website. Meta Pixel is currently used on millions of websites [78], accounting for roughly one-quarter of the web according to various measurements [9, 17, 75, 95].

Meta Pixel builds on Meta's earlier tracking tools, which date back more than 15 years as depicted in Figure 1. Beacon was the first tracking tool introduced in 2007 that enabled Facebook to track user activities on non-Facebook websites [18, 87]. When a user visited a website where Beacon was installed, the user's activity on the website was (without explicit user interaction) shared on the user's Facebook News Feed. Facebook Beacon was discontinued in 2009 due to privacy concerns [12]. In 2010, Facebook introduced the Like button that could be installed on non-Facebook websites [85]. Both Beacon and the Like button automatically tracked user activity (e.g., URL of the page and a third-party cookie) but the Like Button required a user to explicitly click on the button for the user's website activity to be shared on the user's Facebook News Feed [83, 85].

In 2013, Facebook introduced two advertising-focused pixels that were ultimately merged into a unified Facebook/Meta Pixel [10, 22, 51, 93]. The Custom Audience Pixel enabled creation of an audience of users on the website that visited a particular URL [39]. The Conversion Tracking Pixel enabled tracking information about a specific actions users took on the website via five standard events (i.e., Checkouts, Registrations, Leads, Key Page Views, Adds to Cart) [45]. In 2015, Facebook unified the two pixels with the launch of the Facebook Pixel [10, 22]. The unified Facebook Pixel enabled tracking of nine standard events (e.g., `InitiateCheckout`, `Purchase`) and define custom events [10, 51]. Just like Beacon and the Like Button, Facebook Pixel automatically tracked user activity on non-Facebook websites but with more detailed information and the main purpose being to optimize ad campaigns on Facebook.

In an update in 2017, Facebook Pixel introduced two new events to automatically collect button clicks (`SubscribedButtonClick`) and page metadata (`Microdata`) on non-Facebook websites [55]. Unlike previous iterations of the Facebook Pixel, the collection of these automatically collected events did not require any manual configuration from website developers [64, 79]. Using automatic events, Facebook Pixel now could automatically infer that these events are associated with standard events such as `Purchase` or `AddToCart`. Beginning with this update, as discussed later in Section 2.3, Facebook Pixel's source code was split into two scripts: a generic *fbevents.js* script [54] and a *signals/config* script [52] that contains configuration information about a Meta Pixel.

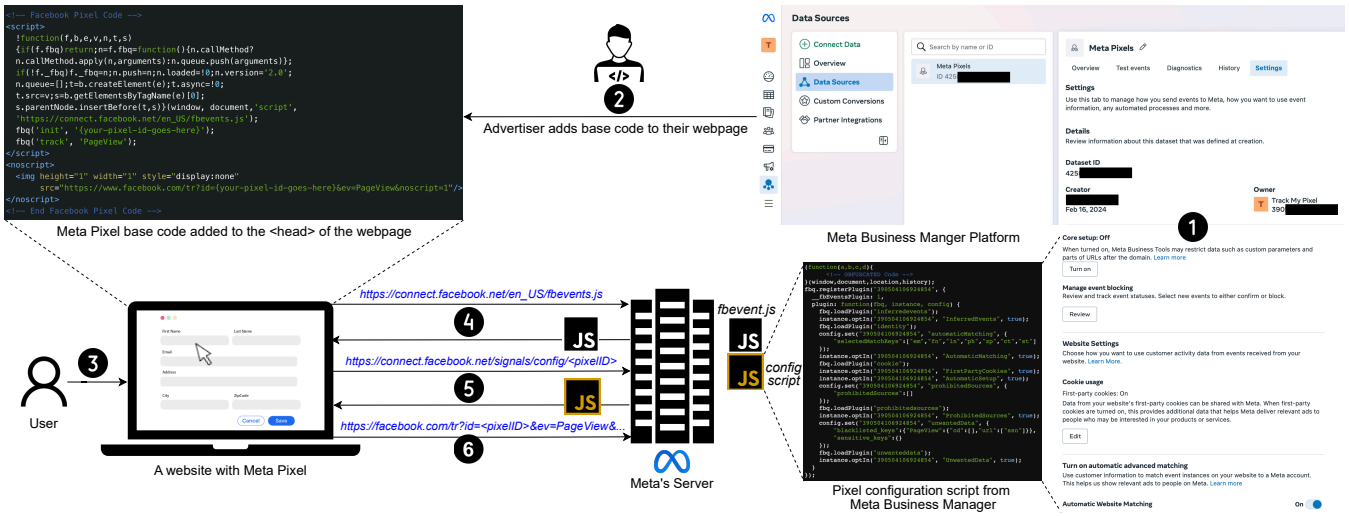


Figure 2: Steps depicting the process of loading a configured Meta Pixel.

2.2 How does Meta Pixel track users?

At its core, Meta Pixel tracks a user's activity on non-Facebook websites and matches it with the user's Facebook account.

• **Activity Tracking:** Meta Pixel by default supports the `PageView` event for each page laod. In addition, Meta Pixel supports two automatic events – `SubscribedButtonClick` and `Microdata` [79]. The `SubscribedButtonClick` event captures button click information such as button text. The `Microdata` event captures metadata of the webpage that is defined on the page using OpenGraph, Schema.org, or JSON-LD format. Beyond these, Meta Pixel supports 17 predefined standard events, representing common user actions indicative of conversions, such as product searches, views, and purchases. Each standard event may include parameters such as product name, product identifier, product category, product price, quantity purchased, etc. Meta Pixel also supports custom events, which allow defining and capturing any user action not captured by the standard events [61, 63, 68].

Each Meta Pixel event is sent in an HTTP GET/POST request to Meta's *https://www.facebook.com/tr/?id=[PixelID]* server, where PixelID is the Meta-assigned unique identifier for the pixel instance. The payload of each HTTP request include the page URL in the `dl` parameter and referrer page URL in the `rl` parameter [56]. Beyond standard URL parameters, event-specific and contextual details are included in `cd[<parameterKey>]` parameters, where `parameterKey` may be `buttonText`, `buttonFeatures`, `pageFeatures`, and `formFeatures` for the `SubscribedButtonClick` event and DataLayer such as OpenGraph, Schema.org, and JSON-LD for the `Microdata` event. Other standard and custom events may also include relevant information in the `cd[<parameterKey>]` parameters. Lastly, to facilitate user-matching, user data manually sent by advertisers is included in the `ud[<parameterKey>]`, or the `udff[<parameterKey>]` if captured automatically through form fields via Automatic Advanced Matching, where common `parameterKey` values include identifiers such as `em` (for email) as discussed below in identity tracking.

• **Identity Tracking:** Meta Pixel collects three categories of identifying information for each event and matches them to a user's Facebook account using a proprietary matching algorithm [73, 99].

First, Meta Pixel automatically collects third-party cookies set on the facebook.com domain with each event's HTTP request. These third-party cookies include the `c_user` cookie that stores the Facebook user ID in the plaintext, the `fr` cookie that contains encrypted Facebook user ID and a browser ID, and the `datr` cookie that contains a browser ID [46, 65]. Given some browsers restrict third-party cookies [11, 14], Meta Pixel also stores first-party cookies (i.e., set on the advertiser's domain), `_fbp` and `_fbc`, since 2018 [19]. The `_fbp` cookie enables *same-site* tracking while the `_fbc` cookie enables cross-site tracking by storing the `fbclid` click ID that is added as a URL parameter by Facebook during any Facebook to non-Facebook navigation [9]. The third-party cookies are shared with Facebook in the Cookie header while the first-party cookies are shared in parameters.

Second, in addition to cookies, Meta Pixel automatically collects IP address, user agent, and other device properties such as screen width and height. While account and device identifiers in cookies are deterministically matched to a Facebook user, IP address, user agent, and other device properties are probabilistically matched [90].

Finally, Meta Pixel also supports advanced matching since 2016, where it collects certain information a user enters in web forms such as email, first name, last name, phone number, gender, birth date, city, state, zip, and country information [74, 76, 91].

• **Tracking Restrictions:** Meta Pixel has recently introduced tracking restrictions to limit the sharing of certain types of information with Meta. For example, Meta Pixel introduced `UnwantedData` configuration, which filters certain URL (or payload) parameters. The filtering is based on what Meta calls `sensitive` and/or `blacklisted` keys. In 2023, Meta introduced Core Setup that imposed more stricter restrictions on websites that belong to sensitive categories [96]. In Core Setup [69], Meta Pixel is supposed to not share custom parameters and the URL information is to be limited at the domain level. While an advertiser can in theory enable Core Setup on its own, it is mostly turned on by Meta [80, 96].

2.3 How is Meta Pixel Installed and Configured on a Website?

Figure 2 illustrates how Meta Pixel is loaded on a website. When an advertiser creates a pixel, Meta assigns it a unique identifier called Pixel ID. Meta provides the base code that advertisers are recommended to include in the head section of their website. When a user visits the advertiser's website, the base code executes and *fbevents.js* [53] is downloaded. This script is common across all websites on which Meta Pixel is installed. The *fbevents.js* script subsequently loads a configuration script that is specific to the Pixel ID. This configuration script reflects the information about various features and settings defined for the Pixel ID. Once Meta Pixel is loaded, each event is collected via a HTTP GET/POST request at Meta's *https://www.facebook.com/tr/?id=[PixelID]* server.

Our analysis shows that a portion of the configuration script is minified and obfuscated. However, a portion towards the end of the configuration script, within the fbq.registerPlugin(...) function, includes structured information that is related to adoption and configuration of various features. It comprises of the following five sets of functions:

(1) fbq.registerPlugin(...) function encapsulates the entire configuration segment, including all other functions.

(2) fbq.loadPlugin(<moduleName>) function loads modules associated with specific features, such as for `UnwantedData` as listed below:

```
fbq.loadPlugin("unwanteddata");
```

(3) instance.optIn(<pixelID>, <configName>, <bool>) opts in to the respective configuration for the Pixel ID as shown in the example below:

```
instance.optIn("1234567891234567", "UnwantedData", true);
```

(4) config.set(<pixelID>, <configName>, <configJSON>) allows setting the configuration at a finer granularity as dictated by the input JSON as shown in the example below:

```
config.set("1234567891234567","unwantedData", {
    "blacklisted_keys": {
        "ViewContent": {
            "cd": ["em"],
            "url": ["lat", "lng"]
        },
    },
    "sensitive_keys": {
        "PageView": {
            "cd": ["d3857b12b4cea ..."], // SHA-256 hash
            "url": []
        }
    }
});
```

(5) fbq.set(<configurationName>, <pixelID>, <list>) lists specific configurations on how a Pixel ID handles data collection and processing.

```
fbq.set("estRules", "1234567891234567", [{
    "condition": {
        "type": 1,
        "conditions": [{
            ...
            "value": "Submit my portfolio"
        }]
    },
    "derived_event_name": "SubmitApplication",
    ...
    "rule_id": "3690133590227007"
}]);
```

This set of functions in the Pixel configuration script capture various features and their configurations. Therefore, we rely on the configuration script in developing our differential analysis based reverse-engineering framework in Section 3.

2.4 Related Work

Prevalence of tracking pixels. The research community has extensively studied the prevalence of tracking pixels [5, 9, 20, 38, 42, 43, 49, 84, 100, 101]. Libert [43] reported Google Analytics and Facebook Like button on 46% and 21% of the top-million websites, respectively. Englehardt and Narayanan [20] reported Google and Facebook's tracking

pixels on 67% and 24% of top-1M websites, respectively. Complementing this line of work, Lerner et al. [42] conducted a longitudinal study of tracking pixels using data from the Internet Archive's Wayback Machine. The authors reported a steady increase in the prevalence, variety, and capabilities of tracking pixels between 1996 and 2016. Ruohonen et al. [86] found 1x1 image pixels on 31% of top-500 websites. Fouad et al. [27] expanded the analysis beyond 1x1 image pixels and reported image pixels on 95% of the crawled websites The authors further classified image pixels based on their tracking behaviors. More recent studies [6, 9] have shifted focus on more closely analyzing a specific tracking pixel. The research by Bekos et al. [9] is the most relevant to our work. The authors reported that 23% of the top-10K websites use Facebook Pixel and reported various events tracked across different categories of websites, including health websites. In addition to the PageView event, the authors found that certain websites track standard events such as "completeregistration" and "add payment info". Our work studies a wider range of Facebook Pixel configurations. In summary, much of prior work has focused on detecting the presence of tracking pixels and not their configurations over time. Our work aims to fill this gap.

Tracking pixels on health websites. The research community has also investigated the use of tracking pixels on potentially sensitive websites such as those related to health. Intuitively, tracking pixels on health websites pose different privacy risk as compared to those on a generic website. Health information is considered more sensitive [41, 50, 94] and observes greater protection under various laws and regulations (e.g., HIPAA [35]). Libert [44] reported that 91% of the 80,142 health-related web pages include one or more third-party, with 78% for Google, 38% for comScore, and 31% for Facebook. Huo et al. [36] reported that 67 and 7 out of the 459 patient portals include Google Analytics and Facebook Pixel, respectively. Markup's *Pixel Hunt* project [48] investigated the use of Meta Pixel across 100 hospital websites in the United States [47]. The academic and journalistic research into the use of tracking pixels on health websites has prompted regulatory action in the US. In 2022, the Federal Trade Commission (FTC) [28] issued enforcement actions against digital health platforms like *GoodRx* and *BetterHelp* for sharing sensitive health information due to their use of tracking technologies such as Google, Facebook, Snapchat, Criteo, and Pinterest [23, 30]. In 2022, Department of Health and Human Services (HHS) [82] released a bulletin [81] on the use of tracking technologies on health websites [4]. Subsequently, in 2023, the FTC and HHS issued warning letters to 130 healthcare providers using tracking technologies on their websites [4, 24]. Our longitudinal analysis sheds light on the impact of these enforcement actions on the adoption and configuration of tracking pixels on health websites.

3 PIXELCONFIG

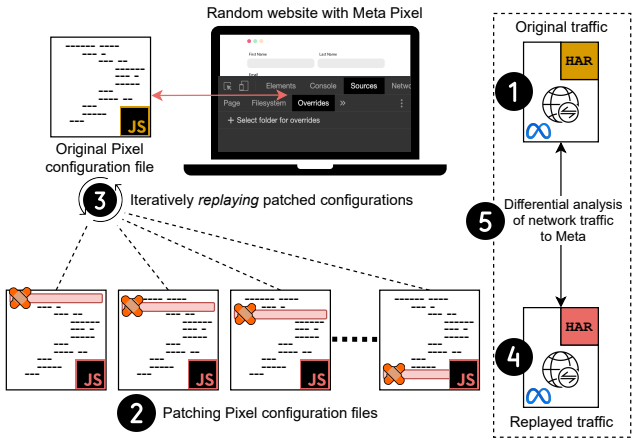
In this section we describe `PixelConfig`, a reverse-engineering framework to study the configurations of a Meta Pixel instance. `PixelConfig` involves a two-fold strategy. ***First***, we perform *code patching* of different configurations defined in a Meta Pixel instance by iteratively commenting out or removing specific lines related to a given configuration. We replay the patched configuration script on client-side using the override functionality in Chrome DevTools [32]. We inspect network traffic to Meta's servers (i.e., *facebook.com*) to understand the effect of patched line in the configuration script. ***Second***, we create a test website, sign-up for a Meta developer account, and iteratively create pixels to test the effect of different features and configuration settings available in Meta Business Manager. We do this because changing various feature settings result in changes in the client-side configuration script of the Meta Pixel instance. Thus, we perform differential analysis of network traffic to Meta's servers before and after replaying a patched configuration script as well as differential analysis of the configuration script before and after modifying various feature settings in Meta Business Manager. Below we explain the how we use this differential analysis to ascertain the three categories of Meta Pixel features: activity tracking, identity tracking, and tracking restrictions. Table 1 summarizes the mapping.

Table 1: Meta Pixel features: Summary of Reverse-engineered configuration mapping, timeline, defaults.

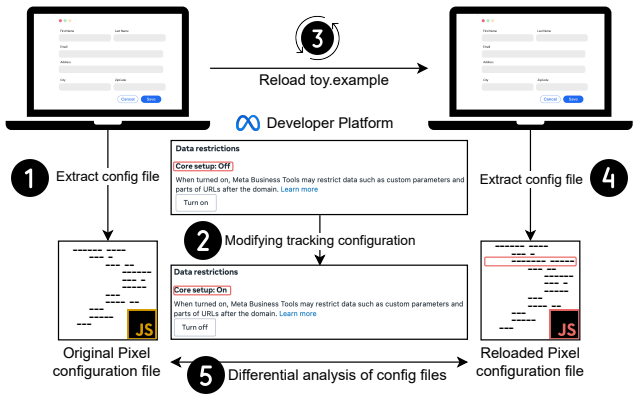
Tracking Category	Tracking Feature (Start Year)	Pixel Configuration (Since Year)	Default Status
Activity	Automatic Events (2017)	AutomaticSetup InferredEvents (2017)	Enabled
	Event Setup Tool (2019)	ESTRuleEngine (2023)	Disabled
Identity	First-Party Cookies (2018)	FirstPartyCookies (2018)	Enabled
	Auto. Adv. Matching (2018)	AutomaticMatching (2018)	Disabled
Restrictions	Prohibited Data Filter (Unknown)	UnwantedData (2020)	Disabled
	Core Setup (2024)	ProtectedDataMode (2023)	Disabled

3.1 Activity Tracking

Automatic Events. Meta Pixel implements the default `PageView` event along with the automatic `Microdata` and `Subscri-`



(a) Replaying patched Pixel configuration scripts on a random website with Meta Pixel, followed by differential analysis of network traffic to Meta before and after replays.



(b) Comparing changes in client-side Pixel configuration scripts before and after changing feature settings on Meta Business Manager platform.

Figure 3: `PixelConfig`: Our differential analysis based reverse-engineering framework comprising (a) & (b)

bedButtonClick events. While the PageView event is defined within the base Pixel code, the automatic Microdata and SubscribedButtonClick events are not. To examine their configuration, we patch the AutomaticSetup configuration by removing the call to instance.optIn(<PixelID>, "AutomaticSetup",true). Upon replaying the patched pixel, the Microdata event does not fire when the page is loaded. Similarly, when we patch both the AutomaticSetup and InferredEvents configurations from the Pixel configuration script by removing their respective instance.optIn calls and replay the patched script, the SubscribedButtonClick event no longer fires when a button is clicked. Both Microdata and SubscribedButtonClick events are enabled by default due to the presence of the AutomaticSetup and InferredEvents configurations in the Pixel configuration script. However, the AutomaticSetup configuration no longer appears in the script for newly created pixels, suggesting either its deprecation or consolidation into the InferredEvents configuration.

Event Setup Tool. In addition to automatic events, advertisers can configure standard and custom events that can be triggered by specific user actions such as button clicks. To investigate how such events are set up, we use the Event Setup Tool in Meta's Business Manager platform to configure a new event triggered by a button click. Comparing the configuration script before and after configuring the event, we observe that only the estRules object is updated to include a new object representing the newly configured event. Next, we investigate a website where a Submit Application event is configured to trigger upon a button click. We observe that when we remove the estRules object, represented by the fbq.set("estRules", ...) call in Listing 5 (Section 2.3), the Submit Application event no longer fires upon a button click. Therefore, we conclude that the derived_event_name key within the estRules object is responsible for defining the event type triggered by specified conditions. We also observe that the button's inner HTML matched the 'value' field within the condition object of the disabled estRules, which, in this instance, was "Submit my portfolio."

3.2 Identity Tracking

First Party Cookies. First-party cookies are used to track a user across different sessions and link interactions back to a specific device or session, especially when third-party cookies are blocked. By default, first-party cookies are enabled when a Meta Pixel is installed – transmitting _fbp and _fbc cookies as parameters with each event. The _fbp cookie is used for same-site tracking, while the _fbc cookie stores the fbclid click identifier, allowing Meta to link user interactions across different websites back to a Facebook click. To examine how the inclusion of first-party cookies is controlled, we configure a Meta Pixel instance on a test website with first-party cookies initially disabled. Upon enabling first-party cookies through the Events Manager, we observe an addition of the instance.optIn(<Pixel ID>, "FirstPartyCookies", true) call in the configuration script. Replaying the pixel with this configuration results in the inclusion of both _fbp and _fbc values in the event payload. Conversely, when we remove the instance.optIn(<Pixel ID>, "FirstPartyCookies", true) call and replay the pixel, the _fbp and _fbc values are no longer transmitted.

Automatic Advanced Matching (AAM). AAM enables advertisers to share hashed versions of user information such as email, phone number, first name, last name, gender, city, state, zip code, country, date of birth, and external ID. When AAM is enabled (it is not enabled by default), Meta Pixel automatically detects these input types when a user submits a form and shares the hashed versions of these

identifiers as udff[<parameterKey>] parameters in the payload of the SubscribedButtonClick event. For instance, the hashed email address is transmitted as udff[], the hashed phone number as udff[<ph>], and so on. To investigate how AAM operates, we first configure it on a test website. Upon enabling AAM in the Meta Events Manager, we observe inclusion of both – instance.opt- In(<Pixel ID>, "AutomaticMatching", true) and config.set (<PixelID>, "automaticMatching", {...}) calls in the Pixel configuration script. The config.set call also includes a selectedMatchKeys array that specifies which user information fields to track:

```
config.set("1286678629287552", "automaticMatching", {
    "selectedMatchKeys": [
        "em", "ph", "fn", "ln", "ge", "db",
        "ct", "st", "zp", "country", "external_id"
    ]
});
```

This array contains keys (e.g., em for email) corresponding to the user data fields that the advertiser chooses to track. Removing a key from the selectedMatchKeys array prevents transmission of the corresponding hashed user information. For instance, removing the ph key prevents the transmission of the hashed phone number. We further validate this behavior by disabling the instance.optIn(<Pixel ID>, "AutomaticMatching", true) call in the Pixel configuration script and replaying the pixel. Upon replay, no hashed user information is transmitted with the SubscribedButtonClick event, confirming that AAM is effectively disabled. This controlled testing aligns with our observations on other websites, where changes in the selectedMatchKeys array directly affected the data transmitted under AAM.

3.3 Tracking Restrictions

Unwanted Data. The UnwantedData configuration defines rules that limit the sharing of specific parameters with Meta. These rules are categorized into blacklisted_keys and sensitive_keys, and are applied at the event level (e.g., PageView, CompleteRegistration, etc.). Each event type can have rules for both custom data parameters (cd) and URL query parameters (url). The technical distinction between blacklisted_keys and sensitive_keys lies in how parameters are specified in the UnwantedData configuration. blacklisted_keys lists parameter names in plain text (e.g., ["lat", "lng"]). sensitive_keys consists of SHA-256 hashed parameter names (e.g., d3857b12b4cea... for a hashed parameter name as shown in Section 2.3). For instance, in the payload of a ViewContent event triggered on the URL *https://www.example.com?lat=40.00&lng=35.00*, the dl parameter is sanitized to *https://www.example.com?lat=_removed_&lng=_removed_* if lat and lng are either listed as blacklisted_keys or their SHA56 hashes are included in sensitive_keys. We confirm this by removing a parameter from the UnwantedData configuration for a specific event type (e.g., excluding lat from ViewContent's blacklisted_keys), which resulted in transmission of the parameter. Conversely, adding a parameter's SHA256 hash to sensitive_keys triggered its sanitization. For custom data parameters, including a parameter name in the cd array under blacklisted_keys or its hash under sensitive_keys suppresses its transmission entirely. To further investigate how these configurations are set, we conducted an experiment on our test website. We configure an event to share dob as a custom data parameter to Meta. Upon triggering the event, a notification in Meta Events Manager stated that the dob parameter was "Blocked by Meta" (Figure 16). Subsequent inspection of the downloaded Pixel configuration script revealed that the cd array within the blacklisted_keys had been automatically updated to include dob.

Core Setup. Core Setup is a configuration that enforces strict restrictions on the sharing of custom parameters and URL query parameters. When a website is placed under Core Setup, indicated by the instance.optIn(<Pixel ID>, "ProtectedDataMode," true) call), the cd parameters are omitted, and dl and rl parameters which typically include full URLs and referrer URLs, are truncated to only the domain. To validate this behavior, we configure a Meta Pixel on a test website and enabled Core Setup via the developer platform. Upon enabling Core Setup, we observe an addition of the instance.optIn(<Pixel ID>, "ProtectedDataMode," true) call in the configuration script. Upon removing the aforementioned instance.optIn call and replaying the pixel, we observe that cd is restored in the payload, and dl and rl revert to their original full URLs, including paths and query strings. This confirms that the ProtectedDataMode reflects Core Setup configuration. Our controlled testing on the test website aligns with our observations on other websites, where toggling Core Setup (via the instance.optIn(<Pixel ID>, "ProtectedDataMode," true) call), consistently modifies the payload structure in the specified manner.

4 CRAWLING METHODOLOGY

In this section, we first explain the selection of websites and longitudinal curation of their snapshots in Sections 4.1 and 4.2, respectively. Next, we explain the identification and extraction of Meta Pixel IDs from these snapshots, and the details about temporal crawling of Meta Pixel configuration scripts in Section 4.3.

4.1 Website Curation

We curate US-focused health websites from two sources – American Hospital Association (AHA) [3] and Centers for Medicare and Medicaid Services (CMS) [15]. We obtained

5,685 AHA member hospital websites from AHA DataQuery [2]. CMS provides a public dataset of 115,646 providers in the USA [26]. We identity 53,432 unique active providers. Unlike AHA data, CMS dataset does not contain website information so we rely on Google Search ("[*name*] [*city*] [*state*] official website") to identify the providers' websites. Overall, we identify 18,327 unique US-focused health websites – 3,272 websites from AHA and 15,055 websites from CMS. To compare health websites against a baseline, we use top-10K websites [40] as a control group. We limited the analysis to top-10K websites due to the crawling limitations of the Wayback Machine as explained below.

4.2 Crawling Website Snapshots

We rely on Internet Archive's Wayback Machine [7] for longitudinal analysis of tracking pixels on the set of health and control websites. The Wayback Machine archives websites and their resources (e.g., scripts, images). It has already archived more than 900 billion web pages since 1996.

We begin by crawling a website's snapshot on the Wayback Machine to detect and extract installed Meta Pixel IDs. We use the CDX Server API [13] to collect historical website snapshots available on the Wayback Machine. The CDX records provide metadata about available snapshots, including timestamps, which allow us to construct appropriate Wayback Machine URLs for each snapshot. The CDX Server API was queried in batches of up to 100,000 records per request for each website, retrieving website snapshot records from 2017 onwards. For the majority of websites, a single request was sufficient to retrieve all available records. By limiting the number of CDX API retries to five, snapshot timestamps obtained from the records were used to generate corresponding archive.org URLs of website snapshots. For example, URL for the Wayback's snapshot of *https://www.facebook.com/* dated Sept 1, 2024, is *https://web.archive.org/web/20240901000408/https://www.facebook.com/*.

Due to the limitations of Wayback Machine crawling, we capped fetching of website snapshots to twice per year (on January 1 and July 1) for all websites. In absence of a website snapshot on January 1 or July 1, we search for the closest snapshot. Each archived website snapshot was crawled using a Selenium [89] driven Chrome browser. Selenium's default page load strategy, ensures that each webpage is fully loaded during the crawl, allowing dynamic elements to be rendered in the HTML snapshot [88]. Moreover, given that website crawling on the Wayback Machine can face issues, we retry each archived website snapshot ten times.

4.3 Extracting Pixel ID and Configuration

To identify unique Pixel IDs on a website, we parse the HTML snapshots to locate the script tag that loads the configuration script and the initialization of the fbq() function. Once extracted, these Pixel IDs are used to identify and crawl the corresponding configuration scripts. Similar to how website snapshots are crawled, the CDX Server API is used to identify snapshots of Pixel configuration scripts by fetching the URL *https://connect.facebook.net/signals/config/<PixelID>* as a prefix. Unlike the website snapshots, which are crawled using Selenium, the configuration scripts are directly fetched using the Python's requests library to minimize overhead. To ensure accuracy, each configuration script is assigned to a website only for the years in which that Pixel ID was observed in the website snapshot.

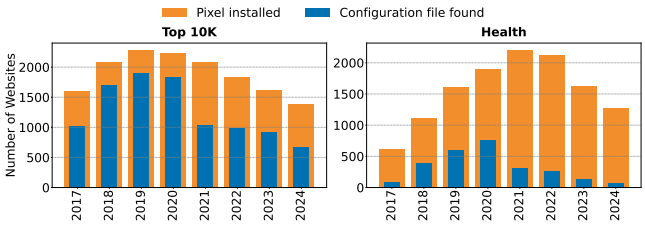


Figure 4: Number of control and health websites with Meta Pixel (orange) and number of websites with configuration script in that year (blue) from 2017-2024.

Figure 4 illustrates the temporal distribution of websites with Pixel installations and those for which configuration scripts were archived. Out of all the control and health websites, we find 3,486 and 3,375 websites, respectively, with at least one snapshot containing a Pixel installation across the years. The other websites in these groups showed no evidence of Pixel installation based on their configuration scripts. However, not all of these Pixel installations had corresponding configuration scripts archived by the Wayback Machine. Specifically, we find configuration scripts for 2,771 control and 1,174 health websites, indicating that while a Pixel was present, the associated configuration script was not always archived. This inconsistency in the Wayback Machine's archival coverage means that our longitudinal comparison across years may not comprise the same set of websites, as some configuration scripts are not consistently archived once every year. Nevertheless, the number of configuration scripts per year remain sufficient for trend analysis [42].

5 ANALYSIS OF META PIXEL CONFIGURATIONS

Using the `PixelConfig` framework described in Section 3, we map different Meta Pixel configurations to their corresponding tracking behaviors to assess their capabilities. Leveraging this framework, we conduct an archaeological analysis of Meta Pixel configurations on health and control websites from 2017 through 2024. In this section, we investigate differences in configuration patterns, privacy implications, and ownership responsibilities across activity tracking, identity tracking, and tracking restrictions.

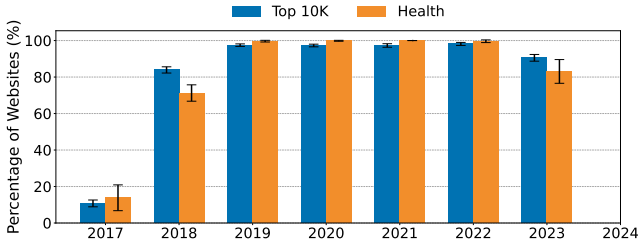


Figure 5: Comparing percentage of control and health websites using `AutomaticSetup` by year from 2017 to 2024. Error bars represent 95% confidence intervals.

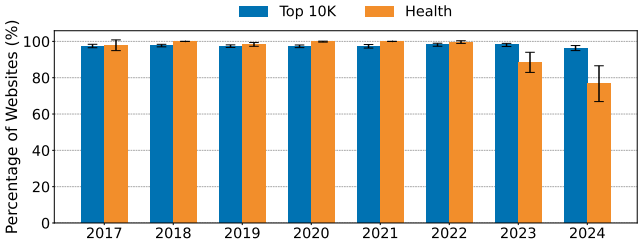


Figure 6: Comparing percentage of control and health websites using `InferredEvents` by year from 2017 to 2024. Error bars represent 95% confidence intervals.

5.1 Activity Tracking

Automatic Events. We saw that the `AutomaticSetup` configuration enables `Microdata` event, while the presence of either the `AutomaticSetup` or `InferredEvents` configurations enable the `SubscribedButtonClick`. Figure 5 shows that the adoption of `AutomaticSetup` configuration increased rapidly after its introduction in April 2017, becoming nearly ubiquitous by 2019. From 2019 to 2022, the configuration was present on over 97% of control websites and over 99% of health websites. However, in 2023, the adoption dropped to 89% for control and 82% for health websites. We surmise that this decline in 2023 is due to a platform-wide deprecation of `AutomaticSetup` by Meta, as we do not observe it in 2024 snapshots. This is likely because `InferredEvents` is sufficient to trigger the `SubscribedButtonClick` event containing page metadata information and Meta may instead be gathering microdata directly by crawling websites or through other mechanisms [70, 71].

Figure 6 shows the adoption trends for the `InferredEvents` configuration. Similar to `AutomaticSetup`, its prevalence remained consistently high through 2017-2022, ranging from 97.4% to 97.8% for control websites and 99.5% to 99.8% for health websites. In 2023, the adoption of `InferredEvents` declined among health websites (88.1%), and further decreased in 2024 (68.3%). Our analysis reveals that 94% of health websites and 77% of control websites that no longer use the `InferredEvents` configuration were placed in Core Setup starting in July 2023. This suggests that Meta may have begun automatically disabling the `SubscribedButtonClick` event under Core Setup to mitigate the collection of potentially sensitive information, particularly for health websites.

Event Setup Tool. We detected that the `estRules` object captures standard and custom event configurations set up through Meta's Event Setup Tool. Meta introduced its Event Setup Tool in April 2019 [21], but the `estRules` object does not appear in configuration scripts until January 2023. Since its introduction, we observe that 47.1% of health websites and 45.5% of control websites have configured events using the Event Setup Tool.

Figure 7 illustrates the distribution of standard events configured through `estRules` across control and health websites in 2023-2024. The most commonly tracked event across both website groups is the `Lead` event, configured by approximately 25% of websites. This suggests that generating new leads is a primary focus for websites using the Event Setup tool. However, certain events are configured disproportionately on health websites, aligning with user interactions typical of healthcare platforms. For instance, health websites track the `Schedule` event to monitor patients booking appointments, the `Search` event to identify users seeking health-related information, and the `FindLocation` event to track searches for medical providers or facilities. Additionally, `Donate` events are configured on health websites to track interactions with fundraising initiatives, such as donations to medical centers or research institutions. Events such as `ViewContent` and `Contact` are also more prevalent on health websites. These events on health websites can result in collection of potentially sensitive health information by Meta.

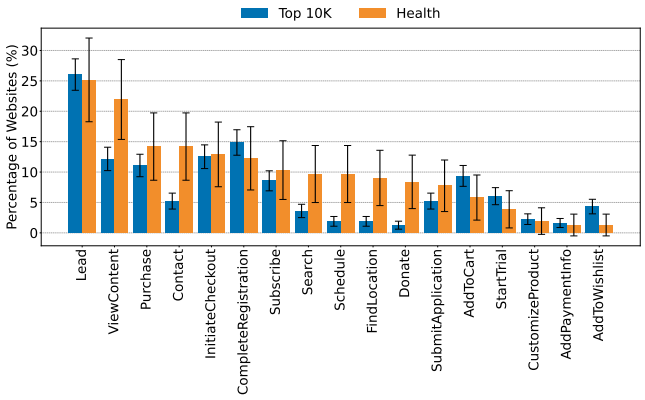


Figure 7: Percentage of websites configuring 17 standard events of Meta Pixel in 2023-2024 across control (N=1110) and health (N=155). These events are configured using Meta's Event Setup Tool and reflected in the `estRules` object.

Events configured through the Event Setup Tool are assigned a unique `rule_id`, which Meta can use to link specific user interactions to button text or values defined in the estRules configuration (see Listing 5 in Section 2.3). Since `rule_id` is collected for each event, Meta can map user interactions with specific buttons, revealing contextually sensitive content on the webpage captured via the button text. For instance, we observe *christushealth.org* (a healthcare provider) tracking a `Schedule` event when a user interacts with a button labeled "request lung nodule screening appointment." Similarly, *healthgrades.com* (a healthcare provider directory) tracks users clicking on buttons labeled as "hiv", "hpv and genital warts", "sexual health", "schizophrenia", and "autism" with a `ViewContent` event, and *docasap.com* (a healthcare appointment scheduling service provider) tracks `Lead` events when their users navigate to certain sections on the website using button clicks such as "birth control pills", "menstrual care products", and "erectile dysfunction." Another concerning example involves *equitashealth.com*, an LGBTQ+ focused healthcare provider, which tracks HIV appointment registrations using the `CompleteRegistration` event when a user clicks a button labeled "schedule your hiv sti testing appointment now". These examples illustrate that Meta collects potentially sensitive health information using events configured using the Event Setup tool.

5.2 Identity Tracking

First-Party Cookies. First-party cookies can be used for both same-site tracking via the `_fbp` cookie and cross-site tracking via the `_fbc` cookie, which stores the `fbclid` click identifier. On health websites, if a user performs actions that trigger event requests containing hashed identifiers (e.g., email, phone number) alongside the `_fbp` cookie, Meta can link previous user activities to their Facebook identity. In contrast, the `_fbc` cookie captures the `fbclid` click identifier when a user clicks on an external link on Facebook, also enabling Meta to link a user's activities on an external website to their Facebook identity.

Meta leverages dark patterns to encourage advertisers to enable first-party cookies in Meta Pixel. By default, first-party cookies are enabled when an advertiser creates a Meta Pixel, a practice that aligns with the *Bad Defaults* dark pattern, in which the less privacy-protective option is set as the default [34]. In contrast, disabling first-party cookies is notably more complex than toggling other features (such as Automatic Advanced Matching), requiring multiple steps that align with the *Privacy Maze* dark pattern [34]. Advertisers must first click 'Edit', then toggle off first-party cookies, and finally confirm the changes by saving (Figure 14). Besides, Meta nudges advertisers to keep first-party cookies enabled by promoting them as a mechanism to "help deliver relevant ads to people who may be interested in your products."

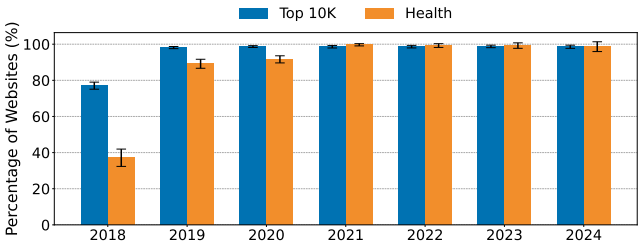


Figure 8: Percentage of control and health websites configuring `FirstPartyCookies` by year from 2018 to 2024. Error bars represent 95% confidence intervals.

Figure 8 shows that first-party cookie adoption surged between 2018 and 2019, soon after they were introduced in October 2018. From 2021 through 2024, pixels on almost all health and control websites configured first-party cookies. This shows that advertisers largely adhere to the default setting of first-party cookies, rarely disabling them.

Automatic Advanced Matching (AAM). AAM enables Meta to directly collect user attributes from websites. Meta hashes user attributes before matching it to the corresponding attributes of all Facebook users [74]. Therefore, hashing user attributes does not limit Meta's ability to link the attributes to Facebook users [25]. Moreover, hashing certain user attributes such as gender is meaningless since it has only two possible values ('M' or 'F') as per Meta's documentation.

We conduct a longitudinal analysis of 11 user attributes that Meta collects as defined in the `selectedMatchKeys` array in the Pixel configuration script. Figure 13 depicts the longitudinal trend of all 11 attributes. Since the trend is largely similar, for the sake of brevity, we discuss two of these attributes here. Figure 9 illustrates a steady increase in Meta's collection of user's email and phone attributes across control websites through 2024 and health websites up to 2021-2022. While AAM is not turned on by default, this increase can be attributed to Meta's design of the AAM setup process, which leverages dark patterns to encourage its configuration. During Pixel setup, Meta describes AAM as a means to "enhance remarketing" [74]. Moreover, enabling AAM through a single toggle automatically activates the collection of all user information attributes by default, aligning with the *Hiding Information* dark pattern as shown in Figure **??** [34]. To prevent specific identifiers from being collected, advertisers must expand the "Show customer information parameters" dropdown and manually opt-out for each user attribute. Meta introduced country, date of birth, and external ID as new identifiers in 2020, further expanding the range of user attributes that it collects through AAM.

The decline in AAM adoption among health websites beginning in 2023 can be linked to regulatory actions targeting tracking of potentially sensitive health information. In 2022, HHS published the bulletin about the use of tracking pixels

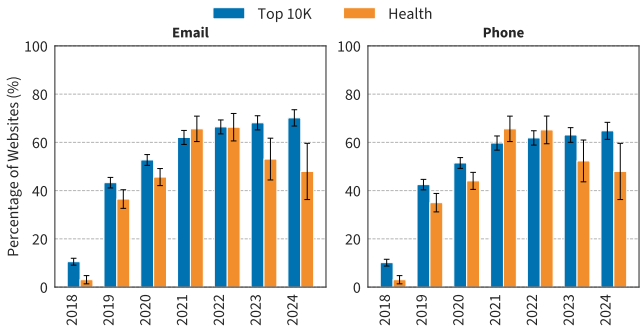


Figure 9: Percentage of control and health websites with Automatic Advanced Matching (AAM) configured for email and phone from 2018 to 2024. For trends related to all AAM keys, refer to Figure 13.

on health websites [81]. This was followed by warning letters issued by FTC and HHS in 2023 [24]. This and associated enforcement actions may have prompted health website advertisers to stop using AAM. As of 2024, approximately 30% of health websites continue to use AAM.

5.3 Tracking Restrictions

Unwanted Data. We detected that the `UnwantedData` configuration reflects filtering of URL and custom data parameters based on `blacklisted_keys` (specified in plain text) and `sensitive_keys` (specified as a SHA-256 hash). A key challenge in analyzing these filtering rules is the hashing of `sensitive_keys`. We attempt to reverse SHA-256 hashed `sensitive_keys` using a public database of pre-computed hashes available at CrackStation [16, 97]. This allowed us to successfully reverse 72.6% of the `sensitive_key` hashes. Overall, we find 4651 unique `blacklisted` and 954 unique decrypted `sensitive` keys.

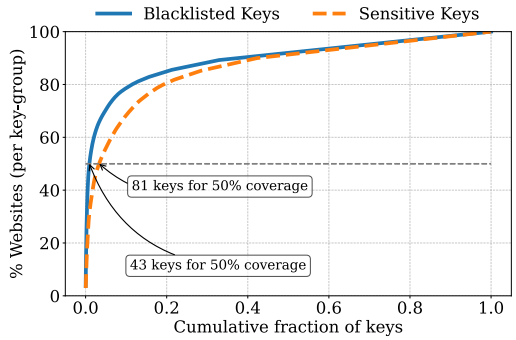


Figure 10: Cumulative distribution of websites containing a given fraction of keys in blacklisted or sensitive groups. A steep curve highlights that a small fraction of keys are present across a large proportion of websites, indicating concentrated key distribution.

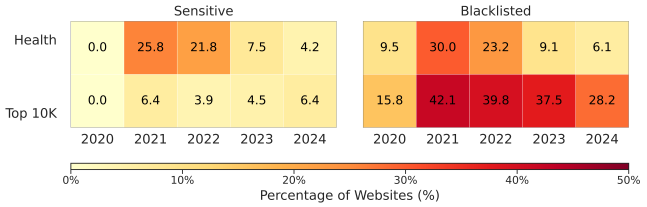


Figure 11: Percentage of control and health websites containing sensitive and blacklisted keys.

Our analysis revealed an overlap in `blacklisted_keys` and `sensitive_keys` across websites. As shown in Figure 10, half of the websites that contain `blacklisted` keys include at least one of the 43 common keys and half of the websites that contain `sensitive` keys include at least one of the 81 common keys. This overlap in commonly observed keys suggests that Meta is likely responsible for selecting `blacklisted` and `sensitive` keys. This is consistent with Meta's claim of detecting and filtering potentially prohibited information automatically [58].

We also identify 200 keys (e.g., `SearchTerm`, `locationName`, `q`) that are common across `blacklisted` and `sensitive` keys. However, despite similarities, there are clear differences between `blacklisted` and `sensitive` keys. Blacklisted keys (specified in plaintext) frequently contained substrings explicitly indicative of common PIIs, such as `name`, `address`, `password`, `em` (email), `dob` (date of birth), `phone`, `IP`, `lat` (latitude), and `long` (longitude). These PII-related substrings appear in 58.4% of the unique `blacklisted` keys but only 7.7% of the decrypted `sensitive` keys. In contrast, the hashed `sensitive` keys include parameters related to potentially sensitive health information. Examples include `doctor` (*towerhealth.org*), `specialty` (*balladhealth.org*), `height` (*meningerclinic.org*), `gender` (*doctor.webmd.com*), `hospital` (*jeffersonhealth.org*), `lgbtq` (*ucihealth.org*), `pregnant` (*investing.com*) and `physician` (*templehealth.org*).

Figure 11 shows that no `sensitive` keys were detected in Pixel configurations in 2020. From 2021 and 2022, health websites exhibited higher use of `sensitive` keys compared to control websites, with parameters such as `txtSearch`, `searchstr`, `childId`, `donor`, `gender`, `queryfilter`, etc. Over time, the gap between health and control websites in terms of use of `sensitive` keys narrows.

We next analyze the names of custom events observed in `blacklisted` or `sensitive` keys as reflected in the `UnwantedData` configuration (see Listing 4 in Section 2.3). These custom event names suggest tracking potentially sensitive health information such as "OCD" and "PTSD Quiz" (*rogersbh.org*), "Low Testosterone Form Submit" (*houstonmethodist.org*), "CardiovascularSurgery" (*mountsinai.org*), and specific treatments like "ImpressionGileadhiv" (*everydayhealth.com*).

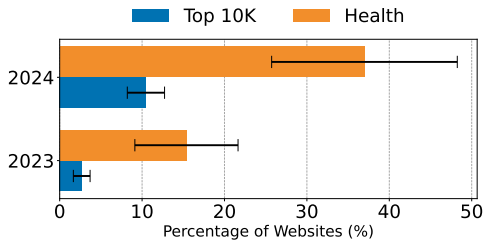


Figure 12: Percentage of control and health websites using `ProtectedDataMode` (i.e., Core Setup) over 2023 and 2024. Error bars represent 95% confidence intervals.

Core Setup. We detected that the `protectedDataMode` configuration reflects Core Setup. Recall that Core Setup restricts the data collection by Meta Pixel, particularly for health, finance, and consumer reports websites deemed sensitive by Meta [58]. While Core Setup was officially introduced in May 2024, we first observe the `protectedDataMode` configuration in Pixel configurations in July 2023. This suggests that Meta may have started deploying Core Setup on certain websites due to regulatory scrutiny even before its official announcements. This is consistent with advertisers reporting their pixels being placed under Core Setup [80].

Figure 12 shows a rise in Core Setup adoption from 2023 to 2024, with a notably steeper increase among health websites. In 2023, only 2.5% of control websites were under Core Setup, increasing to 8.6% in 2024. In contrast, Core Setup adoption rose sharply for health websites from 12.3% in 2023 to 44.4% in 2024. While the pixels on many health websites are in Core Setup, a majority have still not been placed in Core Setup. For example, Meta Pixel continues to collect search terms on *myamericannurse.com* in `s` parameter, *dshs.texas.gov* in `content` parameter, and *health.csuohio.edu* in `keys` parameter.

Once a pixel is placed under Core Setup, we observe that a JSON object listing the website's custom conversion and custom audience rules appears in its configuration. Note that advertisers can specify rules based on visited URLs or custom parameters to track conversions or create custom audiences for targeted advertising [62, 72]. Among websites in Core Setup, 74.4% and 93.2% have at least one custom conversion rule and one custom audience rule, respectively. For instance, the pixel on *healthline.com* tracks users visiting URLs containing "/health/boils-on-buttocks", "/health/vaginal-pimples", "urge-incontinence", and "infertility" using these custom rules. Similarly, the pixel on *psychcentral.com* tracks users visiting URLs containing "adhd" and "nicotine" through these custom rules. Note that the pixels on both sites were not placed in Core Setup until August 2024.

We also observe cases where websites deploy multiple pixels, but not all are placed in Core Setup. For instance, the *nationaljewish.org* hospital deployed four Meta pixels in late 2023, of which only one was placed in Core Setup. The pixel placed in the Core Setup was configured with custom conversion and audience rules to track specific health conditions by detecting presence of "…/lung-cancer-screening/thank-you", or "…/cardiac-rehabilitation-program" in the URL. Even when Core Setup is enabled and URL is stripped to the domain name, we observe pixels on some websites circumventing Core Setup restrictions by including hashed versions of the full URL. For instance, the pixel on *wexnermedical.osu.edu* is placed in Core Setup but the SHA-256 hash of the full URL is shared with Meta in the ud[dl] parameter. This effectively circumvents Core Setup's tracking restriction.

6 CONCLUSION

Meta Pixel has been available for more than a decade and is used on approximately a quarter of the websites today. Prior measurement studies were limited to studying the prevalence of tracking pixels, such as the Meta Pixel. In this paper, we present a deeper dive into the installation *and* configuration of Meta Pixel. Using `PixelConfig`, our framework to reverse-engineer Meta Pixel configurations, we conduct a longitudinal analysis of Meta Pixel configuration on health and a control set of websites from 2017 and 2024. Our work sheds light into the how Meta Pixel's activity tracking, identity tracking, and tracking restriction features have been adopted and configured. Overall, we find that websites stick to Meta Pixel's out-of-the-box configurations, driven in part by the defaults and dark patterns that nudge advertisers to not change them. For example, Meta Pixel was configured to automatically collect button click and page meta-data on up to 98.5% of websites and first-party cookies on up to 98.8% of the websites. We find evidence that Meta Pixel had been tracking potentially sensitive information from health websites such as user interactions related to booking medical appointments or clicking buttons associated with specific medical conditions (e.g., erectile dysfunction). While Meta later introduced controls for tracking restrictions such as Core Setup and Unwanted Data as it faced regulatory scrutiny due to the deployment of Meta Pixel on health websites, we find that the adoption of such controls—again driven by Meta rather than websites—is not comprehensive, can be ineffective, and can be circumvented.

Our work contributes to the measurement literature on tracking pixels by presenting a reverse-engineering framework and its application to analyze Meta Pixel configurations using the Wayback Machine. While this paper focuses on Meta Pixel, the underlying methodology of `PixelConfig` can be generalized to study the configurations of other tracking pixels. To that end and to foster future research on tracking pixels, we have open-sourced `PixelConfig` and released the data (list of health care provider websites and pixel source code) at *https://anonymous.4open.science/r/pixel-config*.

REFERENCES

[1] Gunes Acar, Steven Englehardt, and Arvind Narayanan. 2020. No Boundaries: Data Exfiltration by Third Parties Embedded on Web Pages. *Proceedings on Privacy Enhancing Technologies* 2020 (Oct. 2020), 220–238. https://doi.org/10.2478/popets-2020-0070

[2] AHA 2024. AHA DataQuery. https://www.ahadata.com/aha-dataquery

[3] AHA-Website 2025. AHA Website. https://www.aha.org/

[4] Steve Alder. 2023. OCR, FTC Publish Online Tracking Technology Warning Letters. https://www.hipaajournal.com/ocr-ftc-publish-online-tracking-technology-warning-letters/

[5] Adil Alsaid and David Martin. 2002. Detecting web bugs with bugnosis: Privacy advocacy through education. In *International Workshop on Privacy Enhancing Technologies*. Springer, 13–26.

[6] Nardjes Amieur, Walter Rudametkin, Oana Goga, et al. 2024. Client-side and Server-side Tracking on Meta: Effectiveness and Accuracy. In *24th Privacy Enhancing Technologies Symposium (PETS 2024)*, Vol. 2024. 431–445.

[7] Internet Archive. 2025. Wayback Machine. https://web.archive.org/

[8] Pouneh Nikkhah Bahrami, Umar Iqbal, and Zubair Shafiq. 2021. Fp-radar: Longitudinal measurement and early detection of browser fingerprinting. *Proceedings on Privacy Enhancing Technologies (2022)* (2021).

[9] Paschalis Bekos, Panagiotis Papadopoulos, Evangelos P. Markatos, and Nicolas Kourtellis. 2023. The Hitchhiker's Guide to Facebook Web Tracking with Invisible Pixels and Click IDs. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 2132–2143. https://doi.org/10.1145/3543507.3583311

[10] Antonio Calero. 2015. How to Use Facebook's Upgraded Website Custom Audience Pixel - Jon Loomer Digital. https://www.jonloomer.com/facebook-upgraded-pixel/

[11] Dave Camp. 2019. Firefox Now Available with Enhanced Tracking Protection by Default Plus Updates to Facebook Container, Firefox Monitor and Lockwise. https://blog.mozilla.org/en/products/firefox/firefox-now-available-with-enhanced-tracking-protection-by-default ETP was enabled by default starting with Firefox 69..

[12] CBC. 2009. Facebook Shuts Down Beacon - CBC. https://www.cbc.ca/news/science/facebook-shuts-down-beacon-marketing-tool-1.832698

[13] CDX 2025. Wayback CDX API. https://archive.org/developers/wayback-cdx-server.html

[14] CJ 2018. What You Need to Know about Apple Intelligent Tracking Prevention (ITP). https://junction.cj.com/article/what-you-need-to-know-about-apple-intelligent-tracking-prevention-itp

[15] CMS 2025. CMS.gov. https://www.cms.gov/

[16] Crackstation 2025. Crackstation. https://crackstation.net/

[17] Savino Dambra, Iskander Sanchez-Rola, Leyla Bilge, and Davide Balzarotti. 2022. When Sally Met Trackers: Web Tracking From the Users' Perspective. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 2189–2206. https://www.usenix.org/conference/usenixsecurity22/presentation/dambra

[18] Dave. 2024. Facebook Beacon & Privacy Settings for External Websites (ex: BustedTees.com). https://500hats.typepad.com/500blogs/2007/11/facebook-beacon.html

[19] Digiday. 2018. WTF are Facebook's First-Party Cookies for Pixel? https://digiday.com/marketing/wtf-what-are-facebooks-first-party-cookies-pixel Explains that Facebook Pixel's first-party cookie option went live on October 24, 2018..

[20] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1388–1401. https://doi.org/10.1145/2976749.2978313

[21] Event Setup Tool 2019. Social Media Examiner - How to use Facebook's Event Setup Tool. https://www.socialmediaexaminer.com/how-to-use-facebook-event-setup-tool/

[22] Facebook Developers. 2015. Facebook Marketing API Offsite Pixels v2.4. https://web.archive.org/web/20151112221204/https://developers.facebook.com/docs/marketing-api/offsite-pixels/v2.4

[23] Federal Trade Commission 2023. Better-Help complaint - Federal Trade Commission. https://www.ftc.gov/system/files/ftc_gov/pdf/2023169-betterhelp-complaint_.pdf

[24] Federal Trade Commission 2023. FTC HHS joint letter | heart-risks tracking technologies. https://www.ftc.gov/business-guidance/blog/2023/07/ftc-hhs-joint-letter-gets-heart-risks-tracking-technologies-pose-personal-health-information

[25] Federal Trade Commission 2024. Hashing does not make your data anonymous. https://www.ftc.gov/policy/advocacy-research/tech-at-ftc/2024/07/no-hashing-still-doesnt-make-your-data-anonymous

[26] Centers for Medicare & Medicaid Services. 2025. Provider of Services File - Hospital & Non-Hospital Facilities Data. https://data.cms.gov/provider-characteristics/hospitals-and-other-facilities/provider-of-services-file-hospital-non-hospital-facilities/data

[27] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. 2018. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. *arXiv preprint arXiv:1812.01514* (2018).

[28] ftcc 2025. https://www.ftc.gov/

[29] Jeremy Goldman. 2024. US digital ad spend to exceed $300 billion in 2024. https://www.emarketer.com/content/us-digital-ad-spend-exceed--300-billion-2024

[30] GoodRx Complaint 2023. Complaint for permanent injunction, civil penalties and other relief. https://www.ftc.gov/system/files/ftc_gov/pdf/goodrx_complaint_for_permanent_injunction_civil_penalties_and_other_relief.pdf

[31] Google. 2025. About Google Performance Max Campaigns. https://support.google.com/google-ads/answer/10724817?hl=en

[32] Google. 2025. Chrome Override Functionality. https://developer.chrome.com/docs/devtools/overrides

[33] Google. 2025. Google Clicks Documentation. https://support.google.com/google-ads/answer/6331304?hl=en

[34] Colin M. Gray, Cristiana Teixeira Santos, Nataliia Bielova, and Thomas Mildner. 2024. An Ontology of Dark Patterns Knowledge: Foundations, Definitions, and a Pathway for Shared Knowledge-Building. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 1–22. https://doi.org/10.1145/3613904.3642436

[35] HIPAA 1996. HIPAA act report. https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996

[36] Mingjia Huo, Maxwell Bland, and Kirill Levchenko. 2022. 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*. 197–211.

[37] Maxime Huyghe, Clément Quinton, and Walter Rudametkin. 2025. FP-Rainbow: Fingerprint-Based Browser Configuration Identification. In *WWW'25 - ACM International World Wide Web Conference*. ACM, Sydney, Australia, 1–11. https://doi.org/10.1145/3696410.3714699

[38] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *2021 IEEE Symposium on Security and Privacy (SP)*. 1143–1161. https://doi.org/10.1109/SP40001.2021.00017

[39] Harry Kierbow. 2014. Pinpoint Advertising with Facebook Website Custom Audiences | GoSmallBiz.com. https://gosmallbiz.com/facebook-website-custom-audiences/

[40] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings 2019 Network and Distributed System Security Symposium (NDSS 2019)*. Internet Society. https://doi.org/10.14722/ndss.2019.23386

[41] Pedro Giovanni Leon, Blase Ur, Yang Wang, Manya Sleeper, Rebecca Balebako, Richard Shay, Lujo Bauer, Mihai Christodorescu, and Lorrie Faith Cranor. 2013. What matters to users? Factors that affect users' willingness to share information with online advertisers. In *Proceedings of the ninth symposium on usable privacy and security*. 1–12.

[42] Ada Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 2016. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th USENIX Security Symposium (USENIX Security 16)*.

[43] Timothy Libert. 2015. Exposing the hidden web: An analysis of third-party HTTP requests on 1 million websites. *arXiv preprint arXiv:1511.00619* (2015).

[44] Timothy Libert. 2015. Privacy implications of health information seeking on the web. *Commun. ACM* 58, 3 (2015), 68–77.

[45] Jon Loomer. 2025. Facebook Ads Conversion Tracking: How to create an offsite Pixel. https://www.jonloomer.com/facebook-ads-conversion-tracking-offsite-pixel/

[46] Facebook Ireland Ltd. 2011. Report of audit. https://www.pdpjournals.com/docs/87980.pdf

[47] Markup. 2022. 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

[48] Markup 2022. Markup's pixel hunt series. https://themarkup.org/series/pixel-hunt

[49] David Martin, Hailin Wu, and Adil Alsaid. 2003. Hidden surveillance by Web sites: Web bugs in contemporary use. *Commun. ACM* 46, 12 (2003), 258–264.

[50] William Melicher, Mahmood Sharif, Joshua Tan, Lujo Bauer, Mihai Christodorescu, and Pedro Giovanni Leon. 2016. (Do Not) track me sometimes: Users' contextual preferences for Web tracking. *Proceedings on Privacy Enhancing Technologies* (2016).

[51] Meta 2015. Meta 2015 Pixel Update. https://developers.facebook.com/ads/blog/post/2015/06/10/upgrades-to-conversion-tracking/

[52] Meta. 2017. Configuration file - Wayback Archive. https://web.archive.org/web/2017*/https://connect.facebook.net/signals/config*

[53] Meta. 2017. fbevents.js file. https://connect.facebook.net/en_US/fbevents.js

[54] Meta. 2017. fbevents.js file - Wayback Archive. https://web.archive.org/web/20170322231523/https://connect.facebook.net/en_US/fbevents.js

[55] Meta. 2017. Meta Pixel 2017 - Wayback Archive. https://web.archive.org/web/20170729045537/https://www.facebook.com/business/help/1292598407460746

[56] Meta. 2019. About Facebook Pixel | Facebook Ads Help Center. https://www.facebook.com/business/help/742478679120153?id=1205376682832142

[57] Meta. 2022. Optimize: Automatic Advanced Matching - Wayback Archive. https://web.archive.org/web/20220210085303/https://www.facebook.com/business/m/signalshealth/optimize/automatic-advanced-matching

[58] Meta. 2025. https://www.facebook.com/business/help/361948878201809?id=188852726110565

[59] Meta. 2025. *Facebook Business - Retargeting*. https://www.facebook.com/business/goals/retargeting

[60] Meta. 2025. Meta Advantage Plus. https://www.facebook.com/business/help/733979527611858

[61] Meta. 2025. Meta Business - Automatic Events. https://www.facebook.com/business/help/1292598407460746

[62] Meta. 2025. Meta Business - Custom Conversion. https://www.facebook.com/business/help/780705975381000?id=1205376682832142

[63] Meta. 2025. Meta Business - Event Setup Tool. https://www.facebook.com/business/help/777099232674791

[64] Meta. 2025. Meta Clicks Documentation. https://developers.facebook.com/docs/meta-pixel/advanced/

[65] Meta. 2025. Meta Cookies Policy. https://www.facebook.com/privacy/policies/cookies?subpage=subpage-1.3

[66] Meta. 2025. Meta First Party Cookies Documentation. https://www.facebook.com/business/help/471978536642445?id=1205376682832142

[67] Meta. 2025. Meta Lookalike Audience. https://www.facebook.com/business/help/164749007013531?id=401668390442328

[68] Meta. 2025. Meta Pixel Advanced - developer docs. https://developers.facebook.com/docs/meta-pixel/advanced/

[69] Meta. 2025. Meta'Business - Core Setup. https://www.facebook.com/business/help/124742407297678

[70] Meta. 2025. Microdata - developer docs. https://developers.facebook.com/docs/marketing-api/catalog/guides/microdata-tags#learn-more

[71] Meta. 2025. Web Crawlers - developer docs. https://developers.facebook.com/docs/sharing/webmasters/web-crawlers/

[72] Meta. 2025. Website Custom Audiences - developer docs. https://developers.facebook.com/docs/marketing-api/audiences/guides/website-custom-audiences/

[73] Meta Platforms, Inc. 2025. About Automatic Advanced Matching for Web. https://www.facebook.com/business/help/611774685654668

[74] Meta Platforms, Inc. 2025. Advanced Matching for Facebook Pixel. https://developers.facebook.com/docs/meta-pixel/advanced/advanced-matching/

[75] Shaoor Munir, Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. 2023. CookieGraph: Understanding and Detecting First-Party Tracking Cookies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 3490–3504. https://doi.org/10.1145/3576915.3616586

[76] MuteSix. 2016. New Advanced Matching Features Introduced To The Facebook Pixel. https://mutesix.com/blog/new-advanced-matching-features-introduced-to-the-facebook-pixel/

[77] Arvind Narayanan and Dillon Reisman. 2017. The Princeton web transparency and accountability project. *Transparent data mining for big and small data* (2017), 45–67.

[78] pixel on millions of websites 2025. Trends - Facebook Pixel. https://trends.builtwith.com/websitelist/Facebook-Pixel

[79] PixelYourSite. 2017. Pixel Your Site. https://www.pixelyoursite.com/major-facebook-pixel-update-automatic-facebook-pixel-events

[80] Reddit 2024. Pixel is in Core Setup - Comment. https://www.reddit.com/r/FacebookAds/comments/1btiwe4/pixel_is_in_core_setup/

[81] Office for Civil Rights. 2024. Use of online tracking technologies by HIPAA covered entities and business associates. https://www.hhs.gov/hipaa/for-professionals/privacy/guidance/hipaa-online-tracking/index.html
[82] Office for Civil Rights. 2025. OCR Home | HHS.gov. https://www.hhs.gov/ocr/index.html
[83] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation* (San Jose, CA) *(NSDI'12)*. USENIX Association, USA, 12.
[84] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and defending against Third-Party tracking on the web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. 155–168.
[85] Arnold Roosendaal. 2010. Facebook Tracks and Traces Everyone: Like This! *SSRN Electronic Journal* (2010). https://doi.org/10.2139/ssrn.1717563
[86] Jukka Ruohonen and Ville Leppänen. 2018. Invisible pixels are dead, long live invisible pixels!. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. 28–32.
[87] Betsy Schiffman. 2007. Facebook CEO Apologizes, Lets Users Turn Off Beacon. https://www.wired.com/2007/12/facebook-ceo-apologizes-lets-users-turn-off-beacon/
[88] Selenium. 2024. Browser Options. https://www.selenium.dev/documentation/webdriver/drivers/options/
[89] Selenium. 2025. WebDriver. https://www.selenium.dev/documentation/webdriver/
[90] Abhishek Sen, Anissa Connor, Brooks Dobbs, Chris Watts, Eli Heath, Emma Raz, NumberEight, Giovanni Gardelli, Jamie Zoufal, Jay Rakhe, Keith Kilpatrick, Melissa Ng, Shabneez Khan, IAB Technology Laboratory, Shailley Singh, and Miguel Morales. 2023. Identity Solutions Guidance. , 2–38 pages. https://iabtechlab.com/wp-content/uploads/2024/05/Identity-Solutions-Guidance-FINAL.pdf
[91] Asuman Senol, Gunes Acar, Mathias Humbert, and Frederik Zuiderveen Borgesius. 2022. Leaky Forms: A Study of Email and Password Exfiltration Before Form Submission. In *31st USENIX Security Symposium (USENIX Security 22)*. 1813–1830. https://www.usenix.org/conference/usenixsecurity22/presentation/senol
[92] Tiktok. 2025. Tiktok Pixel - Cookies Documentation. https://ads.tiktok.com/help/article/using-cookies-with-tiktok-pixel?lang=en
[93] Conversion Tracking. 2013. Conversion Measurement Rolls out. https://martech.org/track-on-conversion-measurement-rolls-out-for-all-facebook-advertisers/
[94] Blase Ur, Pedro Giovanni Leon, Lorrie Faith Cranor, Richard Shay, and Yang Wang. 2012. Smart, useful, scary, creepy: perceptions of online behavioral advertising. In *proceedings of the eighth symposium on usable privacy and security*. 1–15.
[95] Tobias Urban, Yash Vekaria, Zubair Shafiq, Chris Böttger, and Barry Pollard. 2024. Third Parties. https://doi.org/10.5281/zenodo.14193384
[96] Bram Van Der Hallen. 2024. Forced to core setup? - Post. Linkedin. https://www.linkedin.com/posts/bramvanderhallen_facebookads-digitalmarketing-digitaladvertising-activity-7159168207549792256-F-PH/
[97] Marie Vasek, Joseph Bonneau, Ryan Castellucci, Cameron Keith, and Tyler Moore. 2017. The Bitcoin Brain Drain: Examining the Use and Abuse of Bitcoin Brain Wallets. In *Financial Cryptography and Data Security*, Jens Grossklags and Bart Preneel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 609–618.
[98] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. 2018. FP-STALKER: Tracking Browser Fingerprint Evolutions. In *2018 IEEE Symposium on Security and Privacy (SP)*. 728–741. https://doi.org/10.1109/SP.2018.00008
[99] Goutham Veerabathini. 2024. Advanced Matching In Facebook For Web. https://www.customerlabs.com/blog/advanced-matching-in-facebook-for-web
[100] Yash Vekaria, Vibhor Agarwal, Pushkal Agarwal, Sangeeta Mahapatra, Sakthi Balan Muthiah, Nishanth Sastry, and Nicolas Kourtellis. 2021. Differential tracking across topical webpages of indian news media. In *Proceedings of the 13th ACM Web Science Conference 2021*. 299–308.
[101] Natalija Vlajic, Marmara El Masri, Gianluigi M Riva, Marguerite Barry, and Derek Doran. 2018. Online tracking of kids and teens by means of invisible images: COPPA vs. GDPR. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*. 96–103.
[102] Xiufen Yu, Nayanamana Samarasinghe, Mohammad Mannan, and Amr Youssef. 2022. Got Sick and Tracked: Privacy Analysis of Hospital Websites. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE Computer Society, 278–286. https://doi.org/10.1109/EuroSPW55150.2022.00034

A ETHICS

This study complies with ethical guidelines for research involving data collection and usage. Our research methodology is based on the analysis of publicly available data obtained from the Internet Archive's Wayback Machine and controlled experiments conducted on a researcher-managed test website. We observe Internet Archive's access limits to avoid overwhelming their infrastructure. The primary data sources we analyze include archived Meta Pixel configuration scripts and archived website snapshots, all of which are public records. Our research did not involve the collection, interception, or processing of any actual user's Personally Identifiable Information (PII). We strictly analyzed publicly accessible code and configuration data as deployed by websites. Thus we strongly believe that our work does not pose any ethical concerns.

B ERROR BARS

For a given feature and in a given year, we analyze only those websites for which we find at least one configuration script in Wayback. Considering this subset, we compute the proportion p of websites that exhibit the feature in that year. We then construct a 95% confidence interval using a t-test based margin of error:

$$\text{ME} = t_{0.975,\,n-1}\sqrt{\frac{p\,(1-p)}{n}} \times 100$$

Error bars in the plots are thus placed at $p \times 100 \pm \text{ME}$.

Note that similar to Lerner et al. [42], we do not claim statistical significant comparisons but rather perform trend analysis of pixel configurations longitudinally on the web.

C OTHER PIXEL CONFIGURATIONS

Figure 13 shows the longitudinal trend of all 11 Automatic Advanced Matching (AAM) keys corresponding to different user attributes.

Figures 14, 15, and 16 showcase UI controls in Meta Business Manager platforms for first-party cookies, AAM, and parameter blocking by Meta, respectively.

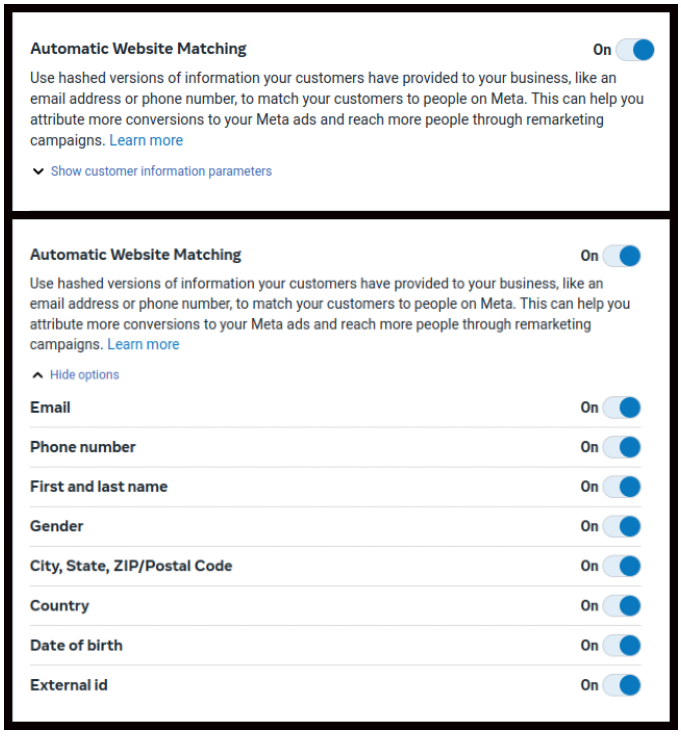


Figure 15: Activating the AAM toggle while the customer information parameters are concealed automatically enables all associated parameters.

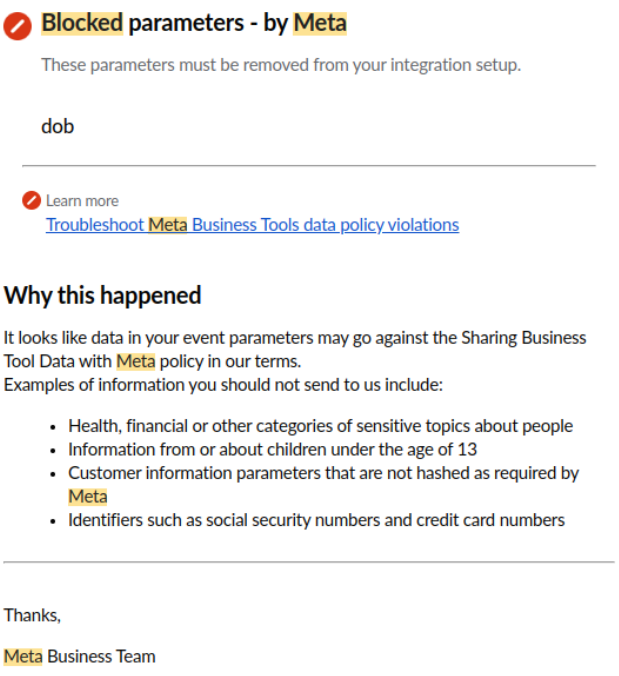


Figure 16: Email notification from Meta indicating that the 'dob' parameter—configured on our controlled website to share users' date-of-birth data—has been blocked.

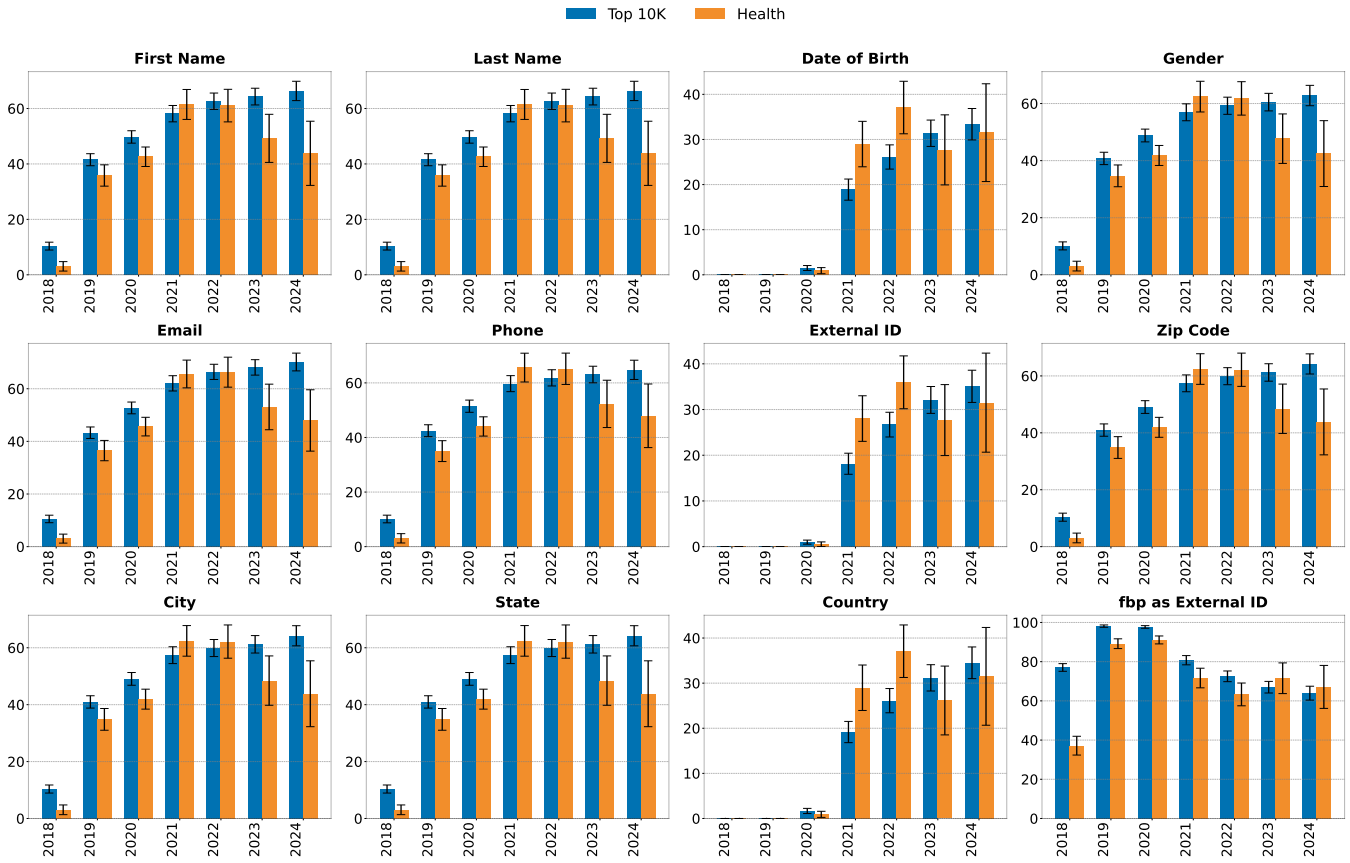


Figure 13: Percentage of control and health websites with specific match keys configured for Automatic Advanced Matching (AAM) from 2018 to 2024. The plots show longitudinal trends for First Name, Last Name, Date of Birth, Gender, Email, Phone, External ID, Zip Code, City, State, Country, and instances where the fbp cookie is used as an external_id (cases where first-party cookies are enabled but external_id through AAM is not). Error bars represent 95% confidence intervals.

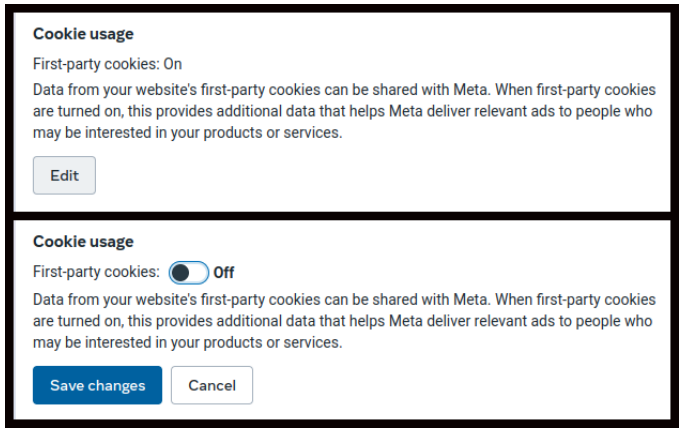


Figure 14: The complex process for enabling first-party cookies requires advertisers to first click 'Edit', then toggle the setting, and finally press 'Save Changes'.