Computing&AI Connect
ISSN: 3104-4719
2025, Article ID. 2025.0013, Cite as: https://doi.org/10.69709/CAIC.2025.159980

SCIFINITI PUBLISHING

Research Article

OPEN ACCESS

Comprehensive Classification of Web Tracking Systems: Technological Insights and Analysis

Theofanis Tasoulas[✉,1] **Alexandros Gazis**[✉,2,*] **Aggeliki Tsohou**[✉,1]

[1] Department of Informatics, Ionian University, Corfu, 49100, Greece
[2] Department of Electrical Engineering and Computer, Democritus University of Thrace, 67100 Xanthi, Greece

Article History
Received: January 30, 2025 Accepted: April 11, 2025 Published: May 06, 2025

Abstract
Web tracking (WT) systems are advanced technologies that are used to monitor and analyze online user behavior. Initially focused on HTML and static webpages, these systems have evolved with the proliferation of IoT, edge computing, and Big Data, encompassing a broad array of interconnected devices with APIs, interfaces, and computing nodes for interaction. WT systems are pivotal in technological innovation and business development, although trends like GDPR complicate data extraction and mandate transparency. Specifically, this study examines WT systems from a technological perspective, excluding organizational and privacy implications. A novel classification scheme based on technological architecture and principles is proposed, compared to two pre-existing frameworks. The scheme categorizes WT systems into six classes, emphasizing technological mechanisms such as HTTP protocols, APIs, and user identification techniques. Additionally, a survey of over 1,000 internet users, conducted via Google Forms, explores user awareness of WT systems. Findings indicate that knowledge of WT technologies is largely unrelated to demographic factors such as age or gender, but is strongly influenced by a user's background in computer science. Most users demonstrate only a basic understanding of WT tools, and this awareness does not correlate with heightened concerns about data misuse. As such, the research highlights gaps in user education about WT technologies and underscores the need for a deeper examination of their technical underpinnings. This study provides a foundation for further exploration of WT systems from multiple perspectives, contributing to advancements in classification, implementation, and user awareness.

Keywords:
web tracking systems; classification scheme; technological architecture; internet user awareness; HTTP protocol; API tracking; data privacy; behavioral analytics

1. Introduction

Web tracking (WT) systems are technologies that automatically record data and behaviors of internet users. Their utilization by service providers is growing rapidly. WT tools can be examined from various perspectives, including company profits, privacy protection, information security, and system architecture [1]. For companies, WT tools are invaluable for collecting data from internet users. By analyzing this data, companies apply personalization techniques and optimize marketing and advertising strategies [2].

However, WT technologies also pose a threat to individual privacy rights. Internet users may be aware of how companies process their data, especially when it is consciously shared (e.g., through online or registration forms). Research indicates that users often lack awareness of the extent of personally identifiable information they share or who can access it, even when shared consciously. Awareness is even more limited when data is collected via WT technologies without users' knowledge.

Another critical aspect is the use of security mechanisms, like cryptography, to protect against unauthorized data disclosure or modification through WT tools. Addi-

* **Corresponding Author:**
Alexandros Gazis, Department of Electrical Engineering and Computer, Democritus University of Thrace, 67100 Xanthi, Greece; agazis@ee.duth.gr

tionally, the architecture of tracking systems is crucial, as the knowledge they produce depends on how each system operates in the background [3,4].

Most WT research focuses on privacy concerns, highlighting the lack of understanding about the technical workings of these systems. There is also a lack of consistent classification schemes for WT systems. For example, a classification framework based on observable behaviors and properties has been introduced, such as a tracker's scope (within-site or cross-site tracking) by [5,6]. Similarly, a scheme based on how websites use WT tools has been developed by [6–8]. Such works are essential for a better understanding of WT systems. Currently, most internet users have limited familiarity with WT systems (beyond cookies) and possess only basic knowledge of these technologies, as shown by recent surveys [5,9–12].

WT systems are web software tools that collect data and store it in files or databases. This data is transmitted through internet communication protocols, such as HTTP, during interactions between a web client (e.g., a browser) and a web server. The process begins with a request from the web client. For example, when a web client requests to view a website (i.e., the front-end technologies consisting of static HTML and CSS or dynamic webpages of JavaScript, and HTML/CSS files), it sends information through the HTTP protocol [13].

If the client does not provide its IP address, the server cannot determine where to deliver the website.This information is added to the HTTP protocol in text form. After the client gathers the requested data, the server sends the website to the client and stores the data received from the client (e.g., IP address, operating system details). These data points can be saved on either the client's or the server's side.

Data can also be transferred via APIs (application programming interfaces), which are sets of subroutine definitions, communication protocols, and programming tools designed to enhance software functionality. APIs make tracking significantly easier and more efficient. There are various methods to track user data, which will be examined in the following sections [7,14].

2. Related Work

In this section, we briefly describe the necessary technological components that are connected with web tracking technology and web tracking systems.

2.1. Cookies

Website owners and third parties commonly use HTTP cookies to track users and monitor their behavior via the HTTP protocol. They are among the most well-known WT systems. A cookie is not a program but a text file containing key-value pairs (e.g., Visiting Time = 21:12, IP Address = 127.0.0.0). HTTP cookies are stored on the client's computer.

Whenever a user visits a website, the server retrieves the cookie to efficiently access the necessary information. If no cookie exists for that site, the server assumes the user is visiting for the first time. The server then generates a new identifier for the visitor in its database and sends a cookie. Cookies store user data, such as passwords, allowing users to avoid reentering them each time they log in. This benefits both websites and internet users [7,8,15–18].

Permanent cookies, also known as zombie cookies or ever-cookies, are HTTP cookies controlled by scripts. When deleted, they are recreated from backups stored in the HTML5 local storage or on the client's hard drive. This occurs when a user visits a webpage that secretly sends a script or executable program via the browser. The script checks a specific location on the client's disk to restore the deleted cookie from a backup or save the cookie's current state [19,20].

2.2. Local Shared Objects

Local shared objects are cookies that do not rely solely on the HTTP protocol for tracking but require software manipulation. For example, Flash cookies are managed by Adobe Flash Player, while Silverlight Secure Storages are handled by Microsoft Silverlight [21,22]. These development tools enhance web content and user experience by using their APIs and scripts to communicate with browsers, retrieve data from viewed pages, or access browser storage.

The collected data is stored in local shared objects, which may include small databases, JSON-formatted files, or complex storage structures beyond simple key-value texts. These objects improve the user experience by storing privacy preferences, restoring time points (e.g., where a user stopped playing a Flash video), and more.

Additionally, they can link a user's activities across different websites or browsers and synchronize their cookies. Cookie synchronization involves linking data from various cookies using fingerprinting techniques to create user interest profiles. Local shared objects have a larger storage capacity than regular cookies and are harder to delete due to their ability to be recreated by the associated software [5,8,9].

2025, Vol. 2, Article ID. 2025.0013
https://doi.org/10.69709/CAIC.2025.159980

2.3. HTML5 Tracking

HTML5 offers numerous APIs to enhance the web experience, such as server-sent events, XMLHttpRequest2, Web Messaging, and Geolocation. By using HTML5 APIs, websites or software can gather user data through the browser and store it either on the server or the client's storage. Additionally, HTML5 introduces new client-side storage mechanisms, including Web Storage, Local Storage, and Indexed DB. These options allow for more efficient data storage using JSON-formatted files, NoSQL techniques, and API calls. Consequently, HTML5 provides an effective means of tracking through API interactions [7,8,10,14,23].

2.4. Web Beacons

A Web beacon (also known as a web bug, tracking bug, pixel tag, or clear GIF) is a technique that involves displaying a very small (typically 1×1 pixel) image in an HTML file or web programming code (e.g., JavaScript). For the image to appear, the user's browser must send a request to the server hosting it. This request allows the server to identify which user wants to view the image.

Due to its small size, the Web beacon is usually invisible to users, though it can be detected by examining the source file. This enables tracking without users' knowledge. Additionally, the image's URL may include a script written in server-side programming languages like PHP or ASP.NET. When executed on the server, this script collects even more user data. Typically, all data gathered by a Web beacon is stored on the server [7,8,15,23].

2.5. Spyware

Spyware tools are software programs installed on a computer to monitor activity or cause harm. They can access third-party APIs or components from browsers or operating systems (OS), such as Layered Service Providers and IConnectionPoint, to retrieve user data. Once collected, this data is saved in hidden files, and a background process transmits the files to a specific server via HTTP.

Additionally, Spyware often collaborates with other software or techniques, such as Adware or keystroke loggers. Keystroke loggers operate at the kernel level, monitoring binary data sent by input devices (e.g., keyboards) through OS APIs, allowing them to capture sensitive information like passwords.

Most Spyware tools are illegal and lack authorization for download, installation, or access [24–26].

2.6. Email Tracking

Email tracking is implemented using a web beacon or email services. It can operate on various networks (public, local, etc.) to reveal how often a message is read or transmitted and by whom, along with their IP address. If a server collects data from a specific web beacon, it can determine which email was opened, where, and when. Modern email services use scripts (usually involving POST and GET methods) on their servers to track the communication channel, sender, and receiver. They may also use web analytics services. All this data is stored on servers [27–29].

2.7. Local-Based Services

Recent advances in wireless location tracking, such as cellular networks or RFID tags, offer many opportunities to implement or enhance Location-Based Services. Using APIs and maps enables real-time traffic analysis, as APIs (like Google Maps API) provide extensive geographic information. APIs repeatedly communicate with telecommunication systems managed by location providers and brokers, requiring a stable connection between the client, server, and provider. Location-Based Services often process queries based on current location (e.g., "Where is the nearest hotel?") by calculating distances between the current and target locations. These services also track a person's location (e.g., via GPS) to improve traffic flow or assist with navigation to specific destinations [30,31].

2.8. Fingerprinting

Fingerprinting is a technique used to create a unique string that identifies internet users. This string is stored in a cookie or browser storage, allowing websites to request it and identify visitors. Trackers use fingerprints to enhance tracking, build user behavior profiles, and provide personalized services. They can also be used for multi-factor authentication and improved security. Some fingerprints are generated via HTTP mechanisms like ETags or scripts, while others are more complex, based on collected data. For instance, canvas fingerprinting sometimes uses the Canvas API [32] to gather information like fonts, graphics card details, screen size, and other features, which are then hashed to produce the fingerprint. Similarly, various APIs enable methods like audio, battery-based, font fingerprinting, and more [33–36].

2.9. Taint Tracking

Taint tracking, or information flow tracking, monitors how applications access and manipulate personal data. Specifi-

cally, when an application initiates one or more processes in IPC (Inter-process Communication), a taint tag is added to the process based on specific dataflow rules [37]. This can be implemented using virtual or hardware components, such as a VM interpreter or shadow memory, to access low-level OS APIs. The taint tracking system labels sensitive data sources and temporarily tags them as they flow through an application. When tainted data is transmitted over a network, the system records the tags, the responsible application, and the destination. This provides real-time feedback to users about how their data is being used [38–40].

2.10. Web Privacy Measurement

Many researchers use Web Privacy Measurements to track tracking services on websites. These tools employ browsers and task managers to convert high-level commands (e.g., "visit a website") into specific subroutines, distribute commands to browser administrators (e.g., create command threads per browser), and interact with browser APIs to analyze data flow. They also use COM interfaces or IConnectionPoint for deeper data collection and tracker detection. Additionally, Web Privacy Measurements crawls files and scripts from websites or data collected by them, manipulating and storing this information in databases for analysis. By crawling websites, they can identify WT systems, such as detecting "src" tags in HTML files leading to web analytics services or tracking scripts (e.g., cookies, fingerprints). Advanced tools can even reveal how personal data is used. These complex systems rely on APIs, databases, browser interfaces, and scripts, and often integrate machine learning algorithms, datasets from privacy tools like Ghostery, and more [41–44].

2.11. Other Tracking Techniques

2.11.1. Session IDs in Hidden Fields

Before cookies, users were tracked through session IDs stored in hidden fields. Basic user data from the HTTP protocol (e.g., IP address) or an identification string from ETag could be passed to another web page via the URL (using the GET method) or a hidden web form field (using the POST method). While this method works without client-side programming languages (e.g., JavaScript), it is limited to a single browsing session [45,46].

2.11.2. Web-Form Authentication

Some websites restrict resources to signed-up users, requiring login via web-form authentication (e.g., username and password). This method makes user identification accurate and straightforward, allowing WT systems to record all user activity on the website, whether stored on the client or server. It is also independent of the user's web browser, operating system, computer, or location [45,47].

2.11.3. Cookies Synchronization

Cookies synchronization, or cookies syncing, is a process used by websites' server to transmit identifiers or other elements of the users' cookies. In this way, cross-tracking is achieved, which is an effective way of sharing cookies' data and forming a profile for each user. Consequently, the user's preferences and behaviors are linked from various tracking websites [9,16,45–48].

2.11.4. Web Analytics Services

If a website wants to analyze its traffic, it can add a tag to its HTML files, which leads to web analytics services such as Google Analytics. Then, the scripts available in the web analytics services will run on their server, to collect and analyze data from visitors in real-time. Users' data are saved in both the client's and service's storage. As long as those services run on different websites, the service can define cookies in the user's browser, which will contain a unique identifier. As a result, users will contain the same cookie identifiers in lots of different websites' cookies, making cookie synchronization a very effective and yet very simple process. Web analytics services depend on big databases and servers for the great manipulation of users' data in real time. Thus, data are saved in both the client's and the website server's storage, mostly on servers [9,16,49].

3. Materials and Methods

3.1. Materials

The proposed classification of tracking systems is not very common in the literature, even though it can provide a better understanding of those systems. A classification scheme allows the separation of certain objects into categories, according to specified criteria. The current paper investigates WT systems based on how they work and their technological architecture, and for that reason, the proposed classification is developed based upon this. HTTP and APIs are the main methods of tracking, whether the data is stored on the client or on the server. Moreover, fingerprints are typically generated by APIs or scripts, but it would be better to distinguish them from other WT systems. That's because fingerprints don't collect data, but it's an exclusive category of identifying internet users to

2025, Vol. 2, Article ID. 2025.0013
https://doi.org/10.69709/CAIC.2025.159980

help other tracking mechanisms. And there are more complicated tracking systems as well, combining all the previous tracking methods. Hence, the current scheme proposed 6 classes, which are represented in Table 1. Lastly, to mention, this scheme is not based on any other previous research effort.

Table 1: Classification scheme based on technological architecture.

Class A HTTP Tracking	Class B API Tracking	Class C User identification	Class D Complex tracking
A1 (Client storage)	**B1 (Self-owned API)**		
Web Form Authentication	Flash cookies and local shared objects	Canvas fingerprinting	Analysis services
Session IDs in hidden fields	Persistent cookies, zombie cookies, evercookies	Synchronization of cookies	Web Privacy Measurements
HTTP Cookies	Local-Based Services	Etags	
	HTML5		
A2 (Server storage)	**B2 (Third party API)**		
HTML	Taint tracking		
Web Beacons	Spyware		
Email Tracking			

3.1.1. Class A---HTTP Tracking

This class includes technologies that rely on the HTTP protocol to collect data, often using HREFs, URLs, or HTTP GET requests. Code must be executed to store the data in files or databases, which can run either on the server or client. The data storage location isn't necessarily tied to where the code is executed. For example, code may run on the server, while data is stored on the client. Users can interact with these technologies by analyzing executable code and HTML files. Most tracking technologies depend on HTTP, as it facilitates basic communication between the server and client. However, this class excludes more complex, combinational methods. Therefore, Class A includes the following technologies:

(a) Web Form Authentication
(b) Session IDs in hidden fields
(c) HTTP Cookies
(d) HTML
(e) Web Beacons
(f) Email Tracking

Class A can be further divided into A1 and A2. A1 consists of technologies that store data on the client's computer, while A2 refers to technologies that store data directly on their servers. Thus, (a), (b), (c) belong to A1, while (d), (e), (f) are assigned to A2.

3.1.2. Class B---API Tracking

Technologies in Class B collect and store data in storage structures (databases or files) using APIs or other interfaces that mediate communication between objects and their environment. These interfaces often rely on low-level operating system APIs (such as Layered Service Providers), application APIs (e.g., HTML5, Adobe Flash), and browser interfaces (e.g., IConnectionPoint, COM). Users interact with these systems through web platforms, applications, or web pages. Implementing these tracking systems requires significant coding and a deep understanding of how APIs function. In conclusion, Class B includes the following technologies:

(a) Flash cookies and local shared objects
(b) Persistent cookies, zombie cookies, evercookies
(c) Local-Based Services
(d) HTML5
(e) Taint tracking
(f) Spyware

Class B can be divided into B1 and B2. B1 includes tracking systems that use their software APIs, while B2 comprises WT systems that rely on third-party APIs from programs, operating systems, or browser interfaces. Thus, (a), (b), (c), and (d) belong to B1, while (e), (f) belong to B2.

3.1.3. Class C---User Identification

Class C includes WT techniques that identify specific internet users. These systems collect data to create fingerprints, which uniquely identify users. A script or API generates a fingerprint based on collected data, storing it in cookies, local shared objects, or other storage. This process can be executed via simple scripts or APIs. The unique identifiers distinguish users as visitors to web pages or web browsers. Users may encounter these identifiers in cookies or local shared objects, often encrypted and difficult to recognize. Therefore, Class C includes the following systems:

(a) Canvas fingerprinting
(b) Synchronization of cookies
(c) E-tags

3.1.4. Class D---Complex tracking

Other tracking technologies combine HTTP, APIs, and web technologies or programming applications for more complex tracking. These systems often rely on large servers and databases to function effectively, as they may operate across multiple websites simultaneously, requiring significant resources. Additionally, complex WT systems often have research or commercial contexts that remain undisclosed, making them less known to both internet users and researchers. Lastly, Class D includes the following systems:

(a) Analysis services
(b) Web Privacy Measurements

3.1.5. Clarification on Categorization Criteria

It is noted that the classification of each tracking technology into a specific category was based on its core mechanism, i.e., its operational properties and the generic technological layering it relies on. For instance, session IDs in hidden fields are placed under Class A (HTTP Tracking) because they operate through the strictly HTTP protocol and do not require any scripting or external software entities. In contrast, persistent cookies do not operate purely through HTTP protocols, as, for example, zombie cookies or evercookies are included in Class B (API Tracking) as they depend on browser-based APIs or even client-side scripts (e.g., Flash or HTML4) to recreate or manipulate the stored data to extend their functionality beyond the strict boundaries of HTTP behavior. As such, this distinction reflects whether a system may function using straightforward protocol-level interactions or it may employ a more complex interface and programming logic for tracking.

3.2. Methods

Based on the four classes analyzed above, Table 1 showcases a comparison of these different classes. In this section, we compare our work with two previous studies. In [5], the authors introduced a classification framework, classifying web trackers based on tracking behaviors and properties. Specifically, the scheme identifies 5 behaviors and 7 properties, primarily observable from the client's side. A web tracker is classified as a behavior if it exhibits at least one of the specified properties, as detailed in Tables 2 and 3 [5].

Table 2: Classification scheme based on observable behaviors, tracking categories as derived from [5].

Category	Name	Profile Scope	Description	Example	Visit Directly?
A	Analytics	Within-Site	Acts as a third-party analytics engine for individual websites.	Google Analytics	No
B	Vanilla	Cross-Site	Employs third-party storage to track users across multiple websites.	Doubleclick	No
C	Forced	Cross-Site	Strictly requires users to visit the tracker directly (e.g., through popups or redirects).	InsightExpress	Yes (forced)
D	Referred	Cross-Site	Relies on another tracker (B, C, or E) to establish unique identifiers.	Invite Media	No
E	Personal	Cross-Site	Accessed directly by users in other contexts.	Facebook	Yes

2025, Vol. 2, Article ID. 2025.0013
https://doi.org/10.69709/CAIC.2025.159980

Table 3: Classification scheme based on observable behaviors, tracking behavior as a mechanism as derived from [5].

Property	Behavior
Tracker creates a state owned by the site itself (first-party state).	A
Requests made to the tracker expose site-owned data.	A
The third-party request to the tracker contains state from the tracker.	B, C, E
Tracker establishes its state from a third-party position; users do not directly engage with the tracker.	B
The tracker forces users to visit it directly.	C
Relies on interactions with another tracker (A, B, C, or E) to transmit data (not originating from the site itself).	D
Users voluntarily access the tracker's site.	E

Table 4: A classification of tracking methodologies, grouping techniques into classes and categories based on functionality and mechanisms [7].

Class	Category	Description
Explicit Cross-Domain Tracking	Basic Tracking	Standard tracking functionality, including the integration of third-party trackers.
	Third Party Included by a Tracker	Third-party domains are directly included by the tracker for data collection.
Implicit Cross-Domain Tracking	Basic Tracking Initiated by a Tracker	Tracking is initiated indirectly through other mechanisms.
	Third Parties That Include Trackers	Third parties that themselves host or incorporate tracking scripts.
Cookie Syncing	First to Third Party Cookie Syncing	Syncing cookies from first-party domains to third-party trackers.
	Third to Third-Party Cookie Syncing	Syncing cookies between multiple third-party trackers.
	Third-Party Cookie Forwarding	Forwarding cookies between third parties for enhanced tracking.
Analytics	Analytics	Analytical tools providing site-specific or cross-domain insights.

Another classification scheme was proposed in Imane Fouad's research to identify tracking domains and their interconnections. The authors in [7] focused on analyzing WT webpages using web beacons (pixel tags) by detecting these beacons. The arrows in [7], representing potential relationships between categories in a stateful crawl, suggest that categories within each class likely interact sequentially, as shown in Table 4.

By crawling 829,349 webpages and detecting web beacons, a tracking classification framework was developed, categorizing web trackers based on their behavior. This scheme consists of 4 classes and 7 categories, each explaining how the WT domain operates, its impact on user privacy, and statistical results from the research dataset. A web tracker using web beacons is classified based on its tracking behavior, as outlined in [7].

The Franziska Roesner classification scheme (Tables 2 and 3) focuses on web tracker behaviors and properties, primarily high-level criteria observable from the client (user's browser). In contrast, our scheme (Table 1) relies on lower-level criteria, emphasizing programming implementation and technological background, mostly observable from the server's side. Additionally, Imane Fouad's scheme (Table 4) covers more complex, non-basic WT systems based on the perspective of website usage. Our classification (Table 1), however, includes simpler mechanisms like HTTP and session IDs in hidden fields, classifying software rather than the website's tracking methods. In conclusion, the differences between these classification schemes stem from differing research perspectives, providing a more comprehensive understanding of WT mechanisms [5,7].

The next section will discuss internet users' awareness and their perspective on WT systems. The classification of WT systems in this research (Table 1) has limited relevance to users' perspectives, as it focuses on observable behaviors from the browser rather than what occurs on the server side. For example, users may notice cookie notifications but are less aware of fingerprints. Consequently, the most recognized WT tools are cookies (96.9% familiarity) and GPS (68.9%), while other tools are 30% less well-known (Appendix, Questions 7, 18). In conclusion, previous research and classification efforts are more aligned with users' perspective, which relies on observable behaviors from the client side. In contrast, this paper's classification, focused on server-side criteria, is less related to users' awareness.

4. Results and Discussion

The results are twofold. Firstly, we specify the questionnaire structure, followed by an explanation of the study's metrics and statistics.

It is noted that, to gain a detailed understanding of the user's knowledge and perceptions to analyze and explain what web tracking systems mean to them, the questionnaire was carefully designed with both structure and accessibility in mind. Specifically, it consisted of clear, concise questions formulated to avoid technical jargon or other misconceptions to ensure that participants, even without a strict technical or computer science-related background, could respond in a meaningful manner. As such, the survey included demographic items followed by targeted questions assessing both factual knowledge (identification of cookies/tracking methods) and subjective awareness (e.g., perceived danger or familiarity with terms). Multiple-choice questions were balanced with open-ended options, allowing for both quantitative future analysis and richer user insights. Our aim was not to merely explore what users know, but also to try to understand how they interpret and react to WT technologies.

4.1. Questionnaire Structure

In this research, an online questionnaire survey was conducted using Google Forms, with 1,032 participants from Greece. Different types of surveys offer varying advantages and disadvantages depending on the research goal. For our study, we chose an online questionnaire via Google Forms to reach as many internet users as possible and ensure ease of data collection. This approach was simple to create and share online, guaranteeing anonymity and no time pressure for participants (allowing them to review questions as needed). The questionnaire aimed to explore how much internet users know about WT systems and the factors influencing their knowledge. Additionally, we examined whether a lack of awareness about WT systems affects users' privacy concerns.

To ensure the questions were understandable for all participants, regardless of their technical knowledge, we avoided jargon and technical terms, opting for small, closed questions (up to 21 words). Most questions included open-ended answers, such as "other, describe in a few words." The data was analyzed using distributional descriptions, statistically comparing groups (e.g., correct vs. incorrect answers) and identifying possible associations between variables. The questionnaire began with demographic questions, including gender, age, daily internet usage, and computer science background (Appendix, Questions 1–4) to identify factors influencing awareness of WT systems. It then included four multiple-choice questions on WT systems, each with three incorrect answers and one correct option, such as: "Do you know what cookies are? a) Text files, b) Software, c) Virus, d) I don't know" (Appendix, Question 8). Some multiple-choice questions were more subjective or debatable, such as: "Which of the following tracking systems do you know? a) Local-based service b) Flash Cookies c) …" (Appendix A, Question 18) [50–52].

Lastly, regarding the survey methodology used, it is noted that participants were reached online using anonymous sharing, i.e., sharing of Google Forms; thus, we aimed for wide accessibility and voluntary participation. Specifically, we paid extra attention to ensure that the questionnaire was intentionally simple to avoid bias from technical terminology, and closed-ended questions were used as a means to facilitate statistical comparison. However, it must be noted that the sampling method may introduce limitations via this mode of operation, as it may lead to self-selection bias, and thus, the sample may not fully represent the broader spectrum of the population. As such, while descriptive statistics and basic comparisons were used as a means to analyze responses, in future analysis of a bigger sample, it should also help to include regression models to quantify the impact of different variables on user awareness and concern.

4.2. Structure Analysis

The first objective was to uncover how much internet users know about tracking systems. The average correct answers across four multiple-choice questions were 37.63%, although only 26.4% claimed to understand how WT systems work (Appendix, Question 6). People appear aware of the data websites collect and the reasons behind it, yet only 25.2% are familiar with the WT systems (33.6% in-

2025, Vol. 2, Article ID. 2025.0013
https://doi.org/10.69709/CAIC.2025.159980

cluding cookies) that websites employ (Appendix, Questions 16–19). In conclusion, internet users have some knowledge about tracking systems, but at a very basic level.

The second goal of the questionnaire was to identify the most significant factor affecting internet users' awareness (gender, age, daily internet usage, and relation to computer science). To achieve this, queries in SQL were utilized within an Oracle Apex database. The sum of correct answers was calculated for each factor. For example, if 51% of men answered correctly to Question 8, and 54% answered correctly to Question 9, the sum for men would be 51% + 54% + ... The goal was to assess the deviation between lower and higher correct answers for each factor. For the "Age" factor, 13–18-year-olds had the lowest sum of correct answers (446%), while 19–24-year-olds had the highest sum (540%). This indicates that younger people know the least, whereas 19–24-year-olds know the most about WT systems. The deviation is the difference between these percentages, 540% − 446% = 94%, indicating that age affects awareness by 0.94. Similarly, gender affects awareness by 0.78, daily internet usage by 0.7, and relation to computer science by 1.42. These numbers provide a comparative measure of how each factor influences correct answers. Thus, the "relation to computer science" emerged as the most important factor affecting internet users' awareness of WT systems.

The third and final objective was to examine whether internet users' lack of awareness correlates with increased concerns about WT tools. To accomplish this, SQL queries were used to identify users who answered with the term "virus" (i.e., "cookies are viruses" from Appendix, Question 8). These users (203 in total) are assumed to be concerned about WT systems. The average of correct answers for those concerned is 0.39 (let A1). By repeating this process for users not concerned about WT systems, the average of correct answers is 0.37 (let A2). If A1 significantly differed from A2, it would suggest that concerned users have less knowledge about WT systems than non-concerned users, implying that awareness reduces concerns about WT systems. However, since A1 and A2 do not differ significantly, it is concluded that awareness about WT systems is independent of concerns regarding WT tools. While initial results showed that awareness does not significantly correlate with concern about data misuse (as the average correct answers between concerned and unconcerned users were 0.39 and 0.37, respectively), further statistical analysis could help validate this finding. Future work could involve the use of inferential statistical methods such as hypothesis testing or regression models to confirm the absence or presence of statistically significant differences between user groups. This would provide a more robust understanding of whether concern is indeed independent of awareness or influenced by other hidden variables.

5. Conclusions

This paper provides a deeper understanding of the technological background of existing WT systems, which is largely underexplored in the literature. Other studies often examine aspects such as privacy impact, but they rarely analyze how these systems function. The current research focuses on the technological foundation of WT systems and introduces a new classification system based on their technical workings. We then compare our classification with prior systems. Other classification schemes can be developed depending on the perspective from which WT systems are studied (e.g., privacy impact, security, algorithmic efficiency), leading to more comprehensive knowledge of these systems. Furthermore, there may be other WT systems not covered by this study, especially those with research or commercial contexts. A deeper investigation into these systems will foster the development of more efficient WT technologies across various domains.

This paper also examines internet users' knowledge of WT systems through an online survey, identifying the key factors influencing their awareness. We found that "relation to computer science" is the most significant factor affecting knowledge, surpassing other variables like gender, age, and daily internet usage. Our findings reveal that internet users' understanding of WT systems remains at a very basic level. Additionally, the study shows that a user's lack of awareness doesn't necessarily translate into heightened concerns about WT systems. However, the paper does not address how users can acquire a better understanding of these technologies. As WT systems continue to evolve, finding effective ways to increase awareness, especially among those unfamiliar with computer science, remains essential.

Although this study focuses on the technological classification of web tracking systems and excludes the organizational and privacy implications by design, it is important to note that many of the tracking methods analyzed may pose significant security and privacy risks. Specifically, we must acknowledge that techniques such as persistent cookies, fingerprinting, spyware, and cookie synchronization can be used to track users across websites without explicit consent, thus potentially providing a gateway for unauthorized profiling or data exploitation. As such, while these concerns were not the central aim of this research, future studies should expand on the proposed classification scheme by integrating risk assessment or eval-

uating the ethical implications of the technological categories.

Lastly, it is also acknowledged that several WT systems used in this research, or in commercial environments, may not be fully covered in this study. Analytically, although the classification is based on widely used and technically established tracking methods, emerging technologies such as AI-enhanced fingerprinting techniques or blockchain-based tracking mitigation strategies, and specifically tracking mechanisms embedded in IoT ecosystems, represent promising areas for future analysis. As such, including such systems could provide a broader classification that would strengthen the taxonomy and categorization, but also make the study more adaptable to current and evolving digital environments. Additionally, understanding the technical characteristics of such systems can also support regulatory efforts, such as improving GDPR compliance mechanisms. This is especially important, as it helps scientists establish clearer standards for what qualifies as tracking and ensures that emerging technologies are covered by data protection frameworks.

List of Abbreviations

API	Application Programming Interface
COM	Component Object Model
CSS	Cascading Style Sheets
DB	Database
GDPR	General Data Protection Regulation
GET	HTTP GET method
GPS	Global Positioning System
HTML	HyperText Markup Language
HTML5	HyperText Markup Language version 5
HTTP	HyperText Transfer Protocol
ID	Identifier
IoT	Internet of Things
IPC	Inter-process Communication
JSON	JavaScript Object Notation
NoSQL	Not Only SQL
OS	Operating System
POST	HTTP POST method
SQL	Structured Query Language
VM	Virtual Machine
WT	Web Tracking

Author Contributions

Validation, Writing---review, Data curation, Conceptualization, Project administration, Supervision: A.T. Methodology, Formal analysis, Investigation, Writing, Citation ---original draft preparation: T.T. Validation, Writing---review and editing, Resources, Citation editing, Template Formatting: A.G. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials

Data supporting the results of this study are available upon request from the corresponding author.

Ethics Committee Approval and Consent to Participate

The study doesn't require Ethics committee approval. We confirm that informed consent was obtained from all participants.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Funding

No external funding was received for this research.

Acknowledgments

The authors confirm that they used DeepL to translate some Greek words from their original draft and Grammarly to improve the English level. No AI was used to generate new text, images, or any other content that does not adhere to the author guidelines.

Appendix A

Survey Questions

In this section, we showcase the Survey's Questions:

(1) What gender are you?

- □ Man
- □ Woman

(2) What is your age?

- □ 0 to 12
- □ 13 to 18
- □ 19 to 24
- □ 25 to 38
- □ 39 to 55
- □ 56 and above

(3) How much time do you spend online?

- □ I rarely use the internet
- □ I use the internet sometimes
- □ 0 to 2 hours a day
- □ 2 to 5 hours a day
- □ More than 5 hours a day

(4) What is your relation with Computer Science?

- □ I have not studied Computer Science
- □ I have not studied Computer Science, but my profession is related to it (e.g., I do some programming)

2025, Vol. 2, Article ID. 2025.0013
https://doi.org/10.69709/CAIC.2025.159980

□ I currently study Computer Science at University or college

□ I have a degree (bachelor, master degree, etc.) in Computer Science

(5) Do you know that the websites you visit online record data about you? (e.g., When did you visit a web site and from which device (smartphone, computer, etc.)?

□ Yes
□ No

(6) Do you know how this is done?

□ Yes, I know
□ Yes, but just a bit
□ No, I don't know

(7) Have you ever heard of cookies?

□ Yes
□ No

(8) What do you think cookies are?

□ They are text files
□ They are software
□ They are viruses
□ I don't know
□ Other (please describe in a few words)

(9) Do you think cookies are dangerous?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(10) Do you think a website can collect your information if you just view a picture online?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(11) Do you think a website can collect information about you through HTML (without cookies)?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(12) Do you think two or more sites can share cookies?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(13) Do you think it's easy for a website to distinguish two or more people? (e.g., to distinguish George from Maria)

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(14) Do you think there are ways to prevent a site from tracking you?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(15) Do you think Ad blocker or incognito mode makes it difficult for a site to track you?

□ Yes
□ Probably
□ No, I don't think so
□ I don't know

(16) Do you think it's possible to track you when they send you an email? If so, how? (select one or more answers)

□ Yes. They use web software (e.g., in JavaScript).
□ Yes. They use viruses.
□ Yes. They use images.
□ Probably
□ No, I don't think so
□ I don't know

(17) What information do you think a site can collect from you? (select one or more answers)

□ Date and time I visited a site
□ Browsing history (previous visits to other sites)
□ Location (e.g., city, home address)
□ Browser (e.g., Google Chrome, Mozilla Firefox)
□ Operating system (e.g., Windows, IOS, Android)
□ IP address
□ Data from other cookies on my computer
□ Passwords
□ Credit card numbers
□ Files from my computer (e.g., photos, videos, music)
□ I don't know

(18) Do you know any of the following tracking systems? (select one or more answers)

□ GPS or other Local Based Service
□ Flash cookies
□ Spyware
□ Email tracking

□ Web beacon (or Web bug, tracking bug, pixel tag)
□ HTML5 Tracking
□ Taint Tracking
□ I don't know any of them

(19) Finally, why do you think websites collect your personal data? (select one or more answers)

□ Maybe they want to hurt me
□ For security reasons and to know that I am not doing anything illegal on the internet
□ To improve their services
□ To make money, mainly through advertising
□ To sell my information to third parties and make money
□ I don't know
□ Other (please describe in a few words)

References

[1] N. Samarasinghe and M. Mannan, "Towards a global perspective on web tracking," *Comput. & Secur.*, vol. 87, Nov. 2019, Art. no. 101569. [CrossRef]

[2] P. P. Kusumojati and E. Mediawati, "Web-based asset management information systems in higher education," *Int. J. Bus. Law Educ.*, vol. 5, no. 1, pp. 398–411, Jan. 2024. [CrossRef]

[3] T. Bujlow, V. Carela-Español, J. Sole-Pareta, and P. Barlet-Ros, "A survey on web tracking: Mechanisms, implications, and defenses," *Proc. IEEE*, vol. 105, no. 8, pp. 1476–1510, Mar. 2017. [CrossRef]

[4] C. Pilton, S. Faily, and J. Henriksen-Bulmer, "Evaluating privacy-determining user privacy expectations on the web," *Comput. & Secur.*, vol. 105, Jun. 2021, Art. no. 102241. [CrossRef]

[5] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proc. 9th Usenix Symp. Netw. Syst. Des. Implement. (NSDI 12)*, San Jose, CA, USA, 2012, pp. 155–168. [Online]. Available: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner

[6] C. Utz, S. Amft, M. Degeling, T. Holz, S. Fahl, and F. Schaub, "Privacy rarely considered: Exploring considerations in the adoption of third-party services by websites," *arXiv*, Mar. 21, 2022. [CrossRef]

[7] I. Fouad, N. Bielova, A. Legout, and N. Sarafijanovic-Djukic, "Missed by filter lists: Detecting unknown third-party trackers with invisible pixels," *arXiv*, Dec. 4, 2018. [CrossRef]

[8] I. Fouad, C. Santos, and P. Laperdrix, "The devil is in the details: Detection, measurement and lawfulness of server-side tracking on the web," in *Proc. 24th Priv. Enhancing Technol. Symp. (PETS 2024)*, Bristol, UK, 2024, Vol. 2024, No. 4. [Online]. Available: https://hal.science/hal-04617727/

[9] N. Singh, Y. Do, Y. Yu, I. Fouad, J. Kim, and H. Kim, "Crumbled cookies: Exploring E-commerce websites' cookie policies with data protection regulations," *ACM Trans. Web*, vol. 18, no. 1, Jan. 2024, Art. no. 11. [CrossRef]

[10] O. Pitkänen and V. K. Tuunainen, "Disclosing personal data socially—An empirical study on Facebook users' privacy awareness," *J. Inf. Priv. Secur.*, vol. 8, no. 1, pp. 3–29, Jan. 2012. [CrossRef]

[11] A. Soumelidou and A. Tsohou, "Towards the creation of a profile of the information privacy aware user through a systematic literature review of information privacy awareness," *Telemat. Inform.*, vol. 61, Aug. 2021, Art. no.101592. [CrossRef]

[12] R. L. Wakefield and L. T. Wakefield, "How do job seekers respond to cybervetting? An exploration of threats, fear, and access control," *ACM Sigmis Database: Database Adv. Inf. Syst.*, vol. 55, no. 1, pp. 136–155, Feb. 2024. [CrossRef]

[13] J. Byrne, C. Heavey, and P. J. Byrne, "A review of Web-based simulation and supporting tools," *Simul. Model. Pract. Theory*, vol. 18, no. 3, pp. 253–276, Mar. 2010. [CrossRef]

[14] I. Castell Uroz, "A novel approach to web tracking detection and removal with minimal functionality loss," *UPCommons*, 2023. [CrossRef]

[15] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan, "An empirical study of web cookies," in *Proc. 25th Int. Conf. World Wide Web*, Montreal, QC, Canada, 2016, pp. 891–901. [CrossRef]

[16] A. Rasaii, S. Singh, D. Gosain, and O. Gasser, "Exploring the cookieverse: A multi-perspective analysis of web cookies," in *Proc. Int. Conf. Passiv. Act. Netw. Meas. Online Conf.*, Cham, Switzerland: Springer Nature, 2023, pp. 623–651. [CrossRef]

[17] M. Kretschmer, J. Pennekamp, and K. Wehrle, "Cookie banners and privacy policies: Measuring the impact of the GDPR on the web," *ACM Trans. Web (TWEB)*, vol. 15, no. 4, pp. 1–42, Jul. 2021. [CrossRef]

[18] A. Ioannou, I. Tussyadiah, and G. Miller, "That's private! Understanding travelers' privacy concerns and online data disclosure," *J. Travel Res.*, vol. 60, no. 7, pp. 1510–1526, Sep. 2021. [CrossRef]

[19] T. Ermakova, B. Fabian, B. Bender, and K. Klimek, "Web tracking—A literature review on the state of research," in *Proc. 51st Hawaii Int. Conf. System Sciences (HICSS)*, Maui, HI, USA, 2018. [Online]. Available: https://aisel.aisnet.org/hicss-51/os/information_security/5/

[20] S. Belloro and A. Mylonas, "I know what you did last summer: New persistent tracking mechanisms in the wild," *IEEE Access*, vol. 6, pp. 52779–52792, Sep. 2018. [CrossRef]

[21] A. N. Kia, F. Murphy, B. Sheehan, and D. Shannon, "A cyber risk prediction model using common vulnerabilities and exposures," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no.121599. [CrossRef]

[22] N. Debnath and A. K. Jain, "A comprehensive survey on mobile browser security issues, challenges and so-

lutions," *Inf. Secur. J. A Glob. Perspect.*, pp. 1–20, May 2024. [CrossRef]

[23] R. Kirchner, S. Koch, N. Kamangar, D. Klein, and M. Johns, "A black-box privacy analysis of messaging service providers' chat message processing," *Proc. Priv. Enhancing Technol.*, vol. 3, pp. 1–8, 2024. [View Online]

[24] T. F. Stafford and A. Urbaczewski, "Spyware: The ghost in the machine," *Commun. Assoc. Inf. Syst.*, vol. 14, no. 1, Sep. 2004, Art. no. 49. [CrossRef]

[25] M. Naser, H. Albazar, and H. Abdel-Jaber, "Mobile spyware identification and categorization: A systematic review," *Informatica*, vol. 47, no. 8, Sep. 2023. [CrossRef]

[26] Å. Wengelin and V. Johansson, "Investigating writing processes with keystroke logging," in *Digital Writing Technologies in Higher Education: Theory, Research, and Practice*, Cham, Switzerland: Springer, 2023, pp. 405–420. [CrossRef]

[27] S. Englehardt, J. Han, and A. Narayanan, "I never signed up for this! Privacy implications of email tracking," *Proc. Priv. Enhancing Technol.*, 2018. [CrossRef]

[28] B. Fabian, B. Bender, B. Hesseldieck, J. Haupt, and S. Lessmann, "Enterprise-grade protection against e-mail tracking," *Inf. Syst.*, vol. 97, Mar. 2021, Art. no. 101702. [CrossRef]

[29] J. Zheng and Y. Sun, "Emailtracker: An intelligent analytical system to assist email event tracking using artificial intelligence and big data," *Proc. CS & IT—CSCP*, pp. 271–276, 2022. [CrossRef]

[30] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 1, pp. 1–36, Jan. 2021. [CrossRef]

[31] D. Dhinakaran, M. R. Khanna, S. P. Panimalar, S. P. Kumar, and K. Sudharson, "Secure android location tracking application with privacy enhanced technique," in *Proc. 2022 5th Int. Conf. Comput. Intell. Commun. Technol. (CCICT)*, Sonepat, India, 2022, pp. 223–229. [CrossRef]

[32] P. Lubbers, B. Albers, and F. Salim, "Using the HTML5 canvas API. Pro HTML5 programming: Powerful APIs for richer internet application development," in *Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development*, Indianapolis, IN, USA: Apress, 2010, pp. 25–63. [CrossRef]

[33] N. Nikiforakis, W. Joosen, and B. Livshits, "Privaricator: Deceiving fingerprinters with little white lies," in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, 2015, pp. 820–830. [CrossRef]

[34] M. D. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle, "Flash cookies and privacy II: Now with HTML5 and ETag respawning," *SSRN*, Jul. 29, 2011, Art no. 1898390. [CrossRef]

[35] A. Fernandez-de-Retana and I. Santos-Grueiro, "Keep your identity small: Privacy-preserving client-side fingerprinting," *arXiv*, Sep. 14, 2023. [CrossRef]

[36] T. Saito and R. Koshiba, "Examination and comparison of countermeasures against web tracking technologies," in *Innov. Mob. Internet Serv. Ubiquitous Comput. Proc. 13th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. (IMIS-2019)*, Sydney, Australia, 2020, pp. 477–489. Cham, Switzerland: Springer International Publishing. [CrossRef]

[37] Z. Ahmad, S. Casarin, and S. Calzavara, "An empirical analysis of web storage and its applications to web tracking," *ACM Trans. Web*, vol. 18, no. 1, pp. 1–28, Oct. 2023. [CrossRef]

[38] W. Enck et al., "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. Comput. Syst. (TOCS),* vol. 32, no. 2, pp. 1–29, Jun. 2014. [CrossRef]

[39] S. Arzt et al., "Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps," *ACM Sigplan Not.,* vol. 49, no. 6, pp. 259–269, Jun. 2014. [CrossRef]

[40] R. Kanyal and S. R. Sarangi, "PanoptiChrome: A modern in-browser taint analysis framework," in *Proc. ACM Web Conf. 2024*, Singapore, 2024, pp. 1914–1922. [CrossRef]

[41] H. Metwalley, S. Traverso, and M. Mellia, "Unsupervised detection of web trackers," in *Proc. 2015 IEEE Glob. Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, 2015, pp. 1–6. [CrossRef]

[42] S. Englehardt, C. Eubank, P. Zimmerman, D. Reisman, and A. Narayanan, "OpenWPM: An automated platform for web privacy measurement," in *Proc. ACM CCS*, 2016. [Online]. Available: https://github.com/openwpm/OpenWPM?tab=readme-ov-file#citation

[43] D. Bui, B. Tang, and K. G. Shin, "Do opt-outs really opt me out?," in *Proc. 2022 ACM SIGSAC Conf. Comput. Commun. Secur.*, Los Angeles, CA, USA, 2022, pp. 425–439. [CrossRef]

[44] W. Rieder, P. Raschke, and T. Cory, "Beyond the request: Harnessing HTTP response headers for cross-browser web tracker classification in an imbalanced setting," *arXiv*, Feb. 2, 2024. [CrossRef]

[45] D. Martin, H. Wu, and A. Alsaid, "Hidden surveillance by websites: Web bugs in contemporary use," *Commun. ACM,* vol. 46, no. 12, pp. 258–264, Dec. 2003. [CrossRef]

[46] A. Fortier and J. Burkell, "Hidden online surveillance: What librarians should know to protect their own privacy and that of their patrons," *Inf. Technol. Libr.,* vol. 34, no. 3, pp. 59–72, Sep. 2015. [CrossRef]

[47] A. Senol, A. Ukani, D. Cutler, and I. Bilogrevic, "The double-edged sword: Identifying authentication pages and their fingerprinting behavior," in *Proc. ACM Web Conf. 2024*, Singapore, 2024, pp. 1690–1701. [CrossRef]

[48] P. Papadopoulos, N. Kourtellis, and E. Markatos, "Cookie synchronization: Everything you always

wanted to know but were afraid to ask," in *Proc. World Wide Web Conf.*, San Francisco, CA, USA, 2019, pp. 1432–1442. [CrossRef]

[49] J. Smith, "Review of cookie synchronization detection methods," *arXiv*, Oct. 28, 2022. [CrossRef]

[50] R. Andonie and I. Dzitac, "How to write a good paper in computer science and how will it be measured by ISI Web of Knowledge," *Int. J. Comput. Commun. & Control,* vol. 5, no. 4, pp. 432–446. [View Online]

[51] A. Williams, "How to… write and analyse a questionnaire," *J. Orthod.*, vol. 30, no. 3, pp. 245–252, Sep. 2003. [View Online]

[52] H. Taherdoost, "Designing a questionnaire for a research paper: A comprehensive guide to design and develop an effective questionnaire," *Asian J. Manag. Sci.*, vol. 11, no. 1, pp. 8–16, Mar. 20, 2022. [CrossRef]