Journal of Information Security and Applications 79 (2023) 103643

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Evolution of web tracking protection in Chrome

Ronghao Pan, Antonio Ruiz-Martínez [*]

Department of Information and Communications Engineering, Faculty of Computer Science, University of Murcia, Spain

ARTICLE INFO

Keywords:
Privacy
Tracking
Fingerprinting
Privacy assessment
Chrome
PrivacyScanner

ABSTRACT

In our society, protecting users' privacy is of utmost importance, especially when users access websites. Increased awareness of privacy concerns has led web browsers to implement new mechanisms to improve privacy while browsing the Internet. In each new version of web browsers, it is claimed that they provide better improvements to protect our privacy. However, there is no analysis of these improvements. To cope with this issue, in this paper, we present an analysis of the privacy of different versions of the Chrome web browser. This analysis is based on the PrivacyScanner tool, which we have improved with the detection of additional tracking techniques. Our findings reveal that tracking protection has seen modest enhancements (namely, between Chrome version 83 and 90, we observed a 7.55% reduction in trackers and 4.76% decrease in Google Analytics elements). Therefore, despite these improvements, there is still ample room for further enhancement.

1. Introduction

As Internet users, we visit a plethora of websites every day. Generally, access to these services is "free". However, this means that we ourselves become the product, since the service is free. For this reason, websites embed ads and track us to create profiles of us that allow them to serve ads based on our preferences, making conversion more likely.

In order to track us, different mechanisms can be used [1–6]: IP address, cookies, HTTP headers, WebRTC, web browser fingerprinting, etc. This tracking is used for multiple purposes including targeted marketing [7], behavioural analysis, and surveillance [8], which compromises user privacy. In fact, tracking on the Internet is widely considered one of the biggest threats to Internet privacy on the Internet [1,9]. We have seen that various measures have been taken to protect our privacy from both a legal and technical perspective.

From a legal point of view, we have seen that various regulations have appeared to create legal mechanisms to improve user privacy and transparency in relation to the collection of user information when browsing the Internet and the management of personal data. Examples of these regulations are the General Data Protection Regulation,[1] the ePrivacy Directive,[2] or the California Consumer Privacy Act[3] [3].

These regulations provide for changes in various areas of an organization: Procedures, Applications, and Infrastructures [10]. In countries where these regulations apply, users are warned that tracking will be performed. In many cases, they must consent to this tracking in order to access the website. In other cases, it is possible to refuse tracking, but since the user must configure it, the user ends up accepting it. Moreover, it is true that in many cases the user's wish is not properly implemented, and it is important to follow some guidelines to design usable privacy interfaces [11]. Finally, it should be noted that all major web browsers have a goal to block all third-party cookies and identifiers for advertisers [12] by the end of 2024.

On the other hand, from a technical point of view, we have seen how various tools have been built in to prevent web tracking [1,6,7,9]: Cookie erasers, private browsing mode in web browsers, adblockers, virtual private networks and anonymous communications networks as Tor. In addition, some browsers claim to have improved their privacy protection mechanisms and configured more privacy options by default (Chrome, Firefox, etc.). At the same time, some privacy-focused browsers have been launched (e.g. Tor browser, Brave, Iron, Waterfox, etc.). As long as no personal information is provided when browsing the Internet, combining these tools with the use of anonymous communication systems can improve our privacy [6,13], since encrypted connections are not sufficient to protect against traffic identification [14].

We have also seen that there are open source tools that allow us to analyse privacy issues [4,15] when we visit a website, and that is interesting for detecting potential issues before we visit a website [4]. As an example of these tools, we can refer to the following: Privacy

* Corresponding author.
E-mail address: arm@um.es (A. Ruiz-Martínez).

[1] https://gdpr.eu.

[2] https://edps.europa.eu/data-protection/our-work/subjects/eprivacy-directive.

[3] https://www.oag.ca.gov/privacy/ccpa.

https://doi.org/10.1016/j.jisa.2023.103643

Available online 8 November 2023

Score, OpenWPM, Chamaleon, and PrivacyScanner. These tools have been used to analyse the main websites and know the different types of tracking mechanisms they use [16,17], how adblockers behave [18–20], etc. So, these tools are very useful to identify the privacy risks when we access websites. We have also found they have been used to compare between (mobile) web browsers [8]. However, we missed that there is a study that analyses the evolution of web browsers and finds out if they provide better tracking and privacy protection as they evolve. This would allow us to determine if the privacy improvements in web browsers are real and effective.

To address this issue, i.e., to measure the evolution of privacy mechanisms in web browsers, in this paper, we present the analysis we performed for the most widely used web browser, Google Chrome [21,22], in different versions (83, 86, and 90). In our analysis, we investigated whether their privacy improvements through different versions have enabled better user browsing privacy protection while browsing some popular websites. For this analysis, we relied on a website privacy scanner called PrivacyScanner, which is used as a headless browser to Google Chrome. However, before the analysis, we added new features to PrivacyScanner to make the privacy analysis more comprehensive. In this paper, we will explain the features and the process followed.

Thus, the main contributions of our work are:

- A comparative analysis of current tools for measuring privacy risks associated to websites.
- Improvement of the PrivacyScanner tool with new features to make the privacy analysis of a website more comprehensive.
- An analysis of privacy tracking mechanisms used in the 50 most visited websites in Spain.
- An analysis of Chrome's evolution regarding the techniques introduced to protect user privacy has been conducted. The effectiveness of these techniques has been tested across different Chrome versions using PrivacyScanner.

The remainder of the paper is organized as follows. Section 2 presents background information related to tracking mechanisms and the main privacy mechanisms supported by web browsers, focusing on Chrome. Section 3 covers related work on the evolution of privacy mechanisms on Chrome and the different tools we can use to measure privacy risks on the web. Then, in Section 4, we present the PrivacyScanner tool and the various improvements we have made to it. Section 5 presents an assessment of the evolution of tracking protection in Chrome, and finally, Section 6 presents conclusions and future work.

2. Background

To track us on the web, there are several mechanisms that can be used [2–6]. At the same time, to protect our privacy, there are a variety of tools [6,7,16,23]. These include virtual private networks [1], anonymous communication systems [24,25], web browsers, and extensions and plug-ins for web browsers [7,16,26–29].

Since this article focuses on analysing the evolution of privacy protection mechanisms provided by the Google Chrome web browser, as it is the most widely used, in this section we first analyse the different mechanisms that are used to track us while browsing the web.

Next, we will see how web browsers work to improve security and privacy by improving web and JavaScript security or by incorporating security headers [30] such as the Same Origen Policy or the Content Security Policy [16,28].

Finally, since we focus on Google Chrome, we will then discuss its features in more detail.

2.1. Tracking on the web

In general, web tracking mechanisms can be divided into two main groups [2,8,12,20]: stateful web tracking and stateless web tracking (or fingerprinting).

In stateful web tracking, the tracker stores some information in our web browser that can be used to identify users across multiple websites [31]. Stateless web tracking, on the other hand, is based on capturing properties and characteristics of the browser or device and the user's configuration [8]. From this information, unique identifiers are derived to identify users.

In general, stateful web tracking is used to track users. Within this group, we can mention third-party trackers as the main mechanism both on the web and in mobile apps [32,33]. However, others could be mentioned [31], such as Local Shared Objects, HTML5 storage, or HTTP ETags. Recently, Demir et al. [33] noted that tracking performed by the first party can be utilized by a third party to circumvent standard tracking preventing techniques.

Although stateful web tracking is the most used mechanism, stateless web tracking is progressively being adopted. Indeed, Iqbal et al. [34] reported that more than 10% of the top-100K websites 10% of websites were using browser fingerprinting. At the same time, web browsers are becoming more fingerprintable because of their APIs [35].

Due to some regulations such as the General Data Protection Regulation (GDPR) or the California Consumer Privacy Act (CCPA), many websites display cookie consent banners to users to inform them about the use of cookies on the website they are visiting and (in many cases) give them the option to accept all cookies, reject all or accept some [3]. However, the implementation of these directives is not fully satisfied by (web) applications or easy for users [7,32,36]. Ultimately, however, tracking is performed prior to this consent, and tracking is also switched to stateless web tracking mechanisms [3].

2.2. Privacy mechanisms of the main web browsers

Lerner et al. [37] studied third-party HTTP requests on the top 1 million websites, offering insights into tracking practices across the web. The study concluded that Google can track users across nearly 80% of these sites through its various third-party domains. Despite advancements in third-party tracking techniques in different browsers, web tracking has evolved beyond simple HTTP cookies to encompass more persistent methods, such as persistent data storage or the emergence of browser fingerprinting [31]. New fingerprinting techniques are continually being discovered and subsequently employed to track users on the web. For instance, Jiang et al. [38] proposed an enhanced version of the Electronic Frontier Foundation's (EFF) Browser Fingerprinting. This version demonstrated high performance and the capability to swiftly and accurately associate users with browser configuration changes across various time periods.

The problem of browser fingerprinting and the use of JavaScript to collect a large amount of data about the user's browser and device. The collected data can form a unique combination of information that represents the digital fingerprint of each user. There are currently two approaches to implementing a fingerprint detector [34]:

- **The static approach** looks for suspicious snippets in JavaScript source codes, i.e., it consists of searching all JavaScript files, both internal and external, to find suspicious elements or functions. A weakness of this technique is that it marks the fingerprint even if the suspicious fragments are not executed.
- **The dynamic approach** is a solution to the weakness of the static approach, as it inspects the function calls that are executed after visiting a website.

As a consequence, nearly all web browsers are implementing security and privacy standards to enhance and ensure varying degrees of user privacy, either by default or through customizable browser settings. E.g., recently web browsers are starting to block third-party cookies considering various degrees [15]. The protection a web browser can provide is important because, as highlighted in [14], more than 75% of tracking activities are made previous to the cookie consent banner or when user rejects all of them.

The privacy protection mechanisms in the most commonly used browsers are as follows:

- **Mozilla Firefox**: Mozilla Firefox uses a number of privacy protection mechanisms to strengthen users' online privacy. These include Enhanced Tracking Protection, which blocks third-party trackers and cookies; Total Cookie Protection, which isolates cookies in separate containers to enhance privacy, and Fingerprint Protection, which minimizes device information accessible to websites. In addition, DNS over HTTPS encrypts DNS traffic, while the browser's privacy protections dashboard provides insights and customization options.
- **Google Chrome**: Google has begun to gradually improve its users' privacy protection starting in 2019 through a proposal called Privacy SandBox. Privacy SandBox is Google's new initiative to develop a set of open standards to improve Internet privacy on the web. Currently, this initiative is still in development and looking for new ideas, but some measures have already been implemented. The main measures implemented in the latest versions of Chrome are the SameSite header, the Referrer-Policy header, and cache partitioning.
- **Apple Safari**: Apple Safari uses some privacy protection mechanisms to improve users' online privacy. These include Intelligent Tracking Prevention (ITP), which blocks cross-site tracking cookies and provides insight into tracking attempts via a privacy report. Fingerprinting prevention reduces the data accessible to websites for unique identification, while Enhanced ITP further limits cookie use and storage. Cross-site tracking and sandboxing prevention provides isolation and security between websites. In addition, Apple's App Tracking Transparency (ATT) obtains user consent for tracking, and enhanced privacy settings allow for customization.
- **Microsoft Edge**: Microsoft Edge includes several privacy features to enhance users' online privacy. These include built-in tracking prevention that blocks trackers and cookies, Microsoft Defender SmartScreen to protect against malicious websites, and InPrivate Browsing for session-based data wiping. The browser also offers enhanced password protection, safeguards against downloaded malware, and family safety settings. The Do Not Track setting allows users to specify that they do not want not to be tracked. The Tracking Prevention dashboard provides insights and customization options, while Secure DNS and the option to delete browser data further bolster privacy.

2.3. Chrome privacy protection techniques

As our work is focused on evaluating the evolution of web tracking protection in Chrome, in this section, we enter into detail in the three privacy protection techniques that are built into the latest versions of Chrome. These protections are:

- **SameSite header**: This header allows you to specify whether a cookie should be restricted to the first-party or the same-site context, that is, it prevents the browser from automatically sending a specific cookie when the request comes from an external domain. In addition, if an application is to be accessed in the cross-site context, it can do so only over an HTTPS connection, i.e. all browser requests must be made over HTTPS. Thanks to this mechanism, browsers reduce the risk of information leakage and improve protection against Cross Site Request Forgery (CSRF) attacks. There are three possible values for the SameSite attribute (None, Lax, and Strict), each of which behaves differently. Next, we explain their behaviour. (1) *None*: This setting enables cross-site access and allows cookies to be passed in the context of a third party. With this setting, the browser sends the cookie for both same-site and cross-site requests. (2) *Lax*: Instructs the browser to use the cookie for requests in the same site context. In the cross-site context, only secure HTTP methods such as the GET request can use the cookie. (3) *Strict*: Cookies with this setting are sent only for requests originating from the domain itself. The use of this header is progressively being adopted using Lax as default policy [39].
- **Referrer-Policy**: HTTP requests can optionally include the *Referer* header, which specifies the origin or URL of the web page from which the request was made. The *Referer-Policy* header specifies what data is provided in the Referer header and, for navigation and iframes, in the destination's document referer. So, the information that is sent in the Referer header when a request is made from your website is defined by the Referer-Policy header that we set. If no policy is set, the default browser setting is used. Until recently, the default policy for most browsers was *no-referrer-when-downgrade*. The no-referrer-when-downgrade policy sends a full URL along with requests from a TLS-protected environment configuration object to a potentially trusted URL and requests from non-TLS-protected clients to any origin. In the case of navigation and iframes, the data in the Referer header can also be accessed via JavaScript via Document Referrer. Therefore, third parties can use this header to create a browsing history of each user. In combination with the use of set cookies, this allows them to identify users and create a list of their previously visited websites. Currently, browsers are moving towards a privacy-enhancing default referrer policy. Chrome has taken the lead, changing the default policy (no-referrer-when-downgrade) to *strict-origin-when-cross-origin* starting with version 85. The strict-origin-when-cross-origin policy specifies that when a request has the same origin and the same secure destinations, a full URL is sent (HTTPS to HTTPS) and no headers are sent for less secure destinations (HTTPS to HTTP).
- **Partitioning the cache**: Since version 85, Chrome caches resources retrieved from the network, using the respective URLs of the resources as cache keys. This mechanism has worked well from a performance point of view for a long time. However, the time it takes for a website to respond to HTTP requests can show that the browser has accessed the same resource in the past, allowing for a number of security and privacy attacks, such as the following:
  - **Detecting whether a user has visited a specific website**: A third party can determine a user's browsing history by checking whether the cache contains a resource that could point to a website.
  - **Cross-site search attack**: An attacker can determine whether an arbitrary string is included in the user's search results by checking whether a "no search results" image used by a particular website is in the browser cache.
  - **Cross-site tracking**: The cache can be used to store cookie identifiers as a tracking mechanism between websites.

To address this issue, Chrome has changed the cache key to a network isolation key in addition to the resource URL. The network isolation key is composed of the URL of the top-level site and the current frame. In this way, the cache stores resources on a per-website and per-by-resource basis. This improvement results in a cache error if the user requests the same resource but from a different domain. This would affect the performance of some web services, especially those that serve large amounts of cacheable resources, such as popular fonts and scripts. According to the results published by Chrome, the overall rate of cache misses would increase by 3.6 percent and the total percentage of bytes loaded from the network would increase by about 4 percent [40].

Below, we will analyse in which version of Chrome these mechanisms are included and whether they improve the user's protection against tracking.

3. Related work

In this section, we analyse related work from two points of view. First, we review the literature on the evolution of privacy protection in Chrome. Second, we review and compare the different tools we can use to automate the analysis of privacy on websites, because in order to measure Chrome's evolution in accessing popular websites, we need a tool that facilitates this process.

3.1. Evolution of privacy protection in Chrome

Since the introduction of Google Chrome in 2008, its market share has increased and it has become the leading web browser [21,22].

As a consequence, it has been analysed from various security and privacy perspectives: Attacks and protection against attacks [41,42], forensic aspects [43,44], support for HTTP security headers [30,39], analysis of its security compared to other web browsers [45,46], improvement of its security, privacy, and usability [47–49], fingerprinting [50–53], with private browsing mode being the most analysed feature [43,49,54–56].

However, as far as we know, there is no research that focuses on analysing Chrome's evolution from a user privacy perspective. Therefore, to the best of our knowledge, this is the first work that uses a website privacy analysis tool or a privacy scanner to analyse the protection that a number of different versions of Chrome provide for various tracking mechanisms.

3.2. Tools for automated web privacy analysis

Currently, there are a number of publicly available, research-oriented tools for website privacy analysis that have been used in other studies or projects [17,57].

Most of these tools that we have analysed behave as general website privacy scanners that allows an assessment of the privacy risk based on different tracking mechanisms of a given website, and on the other hand, there are browser extensions that report specific data related to privacy risks.

Among these tools we have analysed we can point out FourthParty,[4] FPDetective[5] [58], OpenWPM[6] [17], PrivacyScore,[7] PrivacyScanner,[8] WebbKoll,[9] Urlscan.io,[10] Blacklight,[11] and Chameleon.[12]

We can point out that except of FourthParty, FPDetective, and Chameleon which are focused on fingerprinting, the rest of the tools are web privacy scanners.

From these tools, it is relevant to mention OpenWPM, which is a web privacy measurement framework that has been used in over 75 studies.[13]

With the aim of determining which tool could be the most useful to analyse the evolution of Chrome, we have made a comparison between these privacy tools as for the different tracking mechanisms that they can detect. This comparison is shown in Table 1.

The *HTTP Headers* column of the table indicates the ability to collect both HTTP request and response header data, while the *Cookies* and *Third-PartyCookies* columns include the ability to detect both persistent and session cookies.

Finally, the *Active Fingerprinting* column refers to the ability to detect active fingerprinting techniques, and in this case, the detection of Canvas, Information Objects, WebGL, and WebRTC was considered. In addition to the techniques that each tool can detect, a new column has been added to indicate whether the tool is an add-on or not. Browser extensions or add-ons are additional programs that complement the browser and add functionality that the browser itself does not have.

As we can see in Table 1, the tracking mechanism that most tools detect is the HTTP headers. Next, many consider cookies (third-party and first-party, respectively) and fingerprinting. The least frequently detected mechanisms are Web storage, Google Analytics, and Facebook Pixel.

From Table 1, it can be seen that OpenWPM is the most comprehensive privacy measurement tool and is constantly being developed. The Selenium web driver is used by OpenWPM, a web automation tool built on Mozilla Firefox. Compatibility with Chrome was one of the original goals of the authors, but was never realized due to lack of resources and support for Chrome. Also, they have used Firefox-specific APIs in recent versions, which makes Chrome support even more difficult. For this reason, we do not consider it useful for our research.

One of the programs that offers a more thorough analysis is BlackLight, whose function is quite similar to PrivacyScore's, as both of them use the Chromium browser in headless mode to detect the presence of various tracking technologies. However, BlackLight is not able to enumerate first-party cookies. In this case, BlackLight was not suitable for our research, as it was not possible to add the missing features since it is not open source.

Chameleon is a Chrome privacy extension that focuses only on detecting and protecting fingerprinting techniques. As such, it was not useful for our research, as a more comprehensive tool was needed to facilitate the development and analysis process.

Since PrivacyScanner is built on Chrome, unlike OpenWPM, it provides more accurate analysis of websites, as Chrome is the most popular browser with 74.21% usage, according to data from NetMarketShare [22] or 64.76% according to StatCounter [21]. In addition, PrivacyScanner is the new scanning backend component of a future version of PrivacyScore, so it will be able to detect certain tracking techniques that PrivacyScore cannot, such as the detection of Canvas-based techniques.

The main weakness of PrivacyScanner is its inability to detect various tracking methods, including the existence of Facebook Pixel, WebGL, WebRTC, local and session storage, and informational JavaScript objects. In light of this, the idea to enhance PrivacyScanner was conceived. As one of the most comprehensive and Chrome-based tools, it could allow us to more thorough and precise assessment of user privacy concerns when browsing a particular website.

4. PrivacyScanner and new enhancements

For our work, we decided to improve PrivacyScanner. In this section, first, we present the design of the PrivacyScanner architecture and, then we describe the new features we have added to PrivacyScanner to provide a more comprehensive analysis of website privacy risks. Based on the added detection features, it is then compared to the previous version of PrivacyScanner to point out these new features. Finally, based on this new version of PrivacyScanner, we measured privacy on the 50 most visited websites[14] in Spain.

[4] https://github.com/fourthparty/fourthparty.

[5] https://github.com/fpdetective/fpdetective.

[6] https://github.com/mozilla/OpenWPM.

[7] https://privacyscore.org/.

[8] https://github.com/PrivacyScore/privacyscanner.

[9] https://webbkoll.dataskydd.net.

[10] https://urlscan.io/.

[11] https://themarkup.org/blacklight.

[12] https://github.com/ghostwords/chameleon.

[13] https://github.com/openwpm/OpenWPM.

[14] https://www.elsate.com/viewtopic.php?t=2686.

Table 1
Tracking detection mechanisms supported by the different web privacy scanners.

Tools	HTTP headers	Cookies	Third-party cookies	Web storage	Active fingerprinting	Google analytics	Facebook pixel	Add-on	Open source
FourthParty	■	□	□	□	■	□	□	□	■
FPDetective	■	□	□	■	■	□	□	□	■
OpenWPM	■	■	■	■	■	□	□	□	■
PrivacyScore	■	■	■	■	□	■	□	□	■
PrivacyScanner	■	■	■	□	◧[a]	■	□	□	■
WebbKoll	■	■	■	□	□	□	□	□	□
Urlscan.io	■	■	■	□	□	□	□	□	□
Blacklight	■	□	■	□	◧[a]	■	■	□	□
Chameleon	□	□	□	□	■	□	□	■	■

Notation:
■ Tracking detection mechanism supported.
□ Tracking detection mechanism not supported.
◧ Tracking detection mechanism partially supported.
Notes:
[a] Only Canvas.

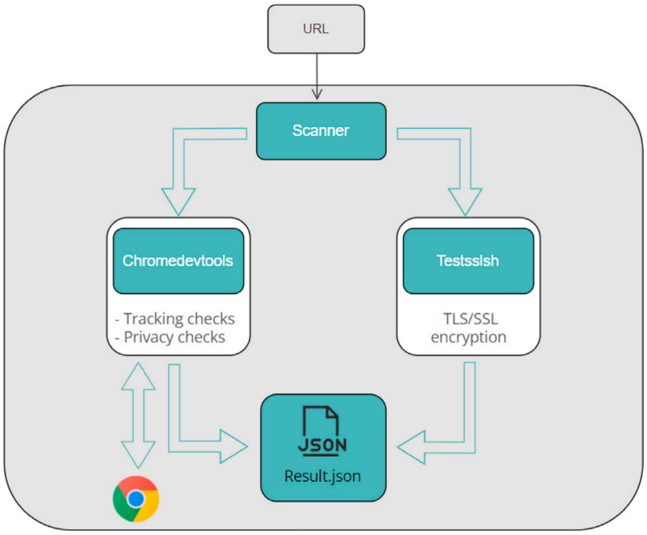

Fig. 1. High-level overview of PrivacyScanner architecture.

4.1. PrivacyScanner architecture

PrivacyScanner is the new scanning component of the upcoming PrivacyScore application. PrivacyScore is an automated website scanner that allows us to analyse websites for privacy and security issues. It allows us to scan individual websites or a list of related websites to see how they compare.

Since PrivacyScanner uses the same architecture as PrivacyScore, but adds some enhancements that PrivacyScore does not have, such as Canvas detection, we can consider it an extension of PrivacyScore. It is an application developed in Python that uses Chrome's headless browser as an interaction browser, so it does not have a graphical interface. Unlike PrivacyScore, the PrivacyScanner tool cannot analyse a group of URLs, so we have to enter the URLs manually.

The application consists mainly of 3 components, as can be seen in Fig. 1:

- **Scanner**: The Scanner is a user-oriented module that pre-processes instrumentation data and converts high-level commands into automated actions in both *Chromedevtools* and *Testsslsh*.
- **Chromedevtools**: The Chromedevtools module is the core of the PrivacyScanner processing engine, as it is responsible for investigating privacy and web tracking risks using various analysers. A key advantage of PrivacyScanner is that it uses a Python package called `pychrome` to configure and debug the browser via Chrome DevTools. As a result, it supports a wide range of configurations and makes it easy to add new scripts to Chrome. This module, shown in Fig. 2, consists of the following main elements:
  - **ChromeScan**: This is a class responsible for initializing the browser in a temporary directory and with IP 127.0.0.1:9222. It also creates the `PageScanner` class to scan the page entered by the user.
  - **PageScanner**: The main task of this class is to perform HTTP requests in the browser. It is responsible for initializing the set of `Extractors` instances to extract and parse the website resources.
  - **Page**: This is a class created by `PageScanner` that is responsible for storing HTTP requests and responses so that `Extractors` can use this data during the analysis process.
  - **Extractors/Analysers**: These are a set of classes derived from the `Extractor` class that is responsible for analysing and testing various tracking techniques such as cookie detection and Canvas-based techniques.
  - **Result**: All parsers write the parsing results directly to a file in JSON format called *result.json*.
- **Testsslsh**: Testsslsh is a module that uses the Testssl.sh tool (a free command line and open source tool) to test TLS/SSL-enabled services for ciphers, protocols, and some supported cryptographic vulnerabilities.

PrivacyScanner uses a dynamic approach to detect Canvas-based tracking techniques. This detection is done by adapting the JavaScript engine of the headless Chrome browser and applying the Chrome DevTools protocol to instrument it. To detect the presence of Canvas, the Chrome Debugger and inserting breakpoints in specific API calls were used to extract data from the browser's function calls. Thus, we can replace or redefine a function of a particular API to capture its properties, analyse them, and check if some of them match those of the detection metrics. As shown in Listing 2, PrivacyScanner has redefined both the return function and the getter/setter function of the properties of a particular JavaScript object to include a breakpoint through the log function. This allows the debugger to record the properties of the captured function, such as its name and arguments. PrivacyScaner for detecting Canvas-based fingerprinting techniques is based on the following approaches:

- Scripts that extract the image using `toDataURL`, i.e., when the scripts use the `toDataURL` property of the `HTMLCanvasElement` API.
- The website is assumed to use Canvas-based font techniques if it uses the following APIs belonging to `CanvasRenderingContext2D`: `FillText`, `StrokeText` and `TargetImageData`.

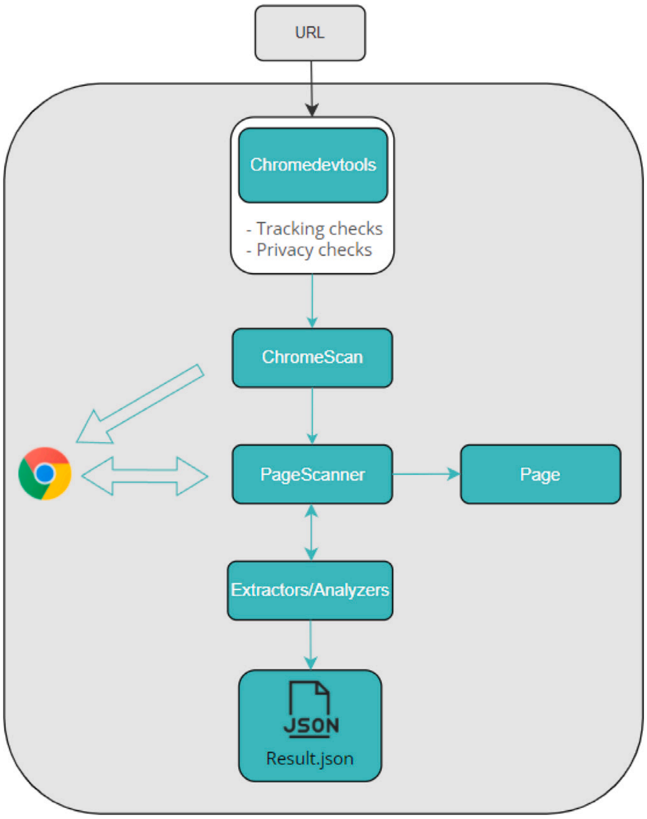

Fig. 2. Chromedevtools execution diagram.

```
function() {
    function log(type, message) {
        var setBreakpointOnThisLine;
    }
    window.alert = function() {};
    window.confirm = function() {
        return true;
    };
    window.prompt = function() {
        return true;
    };
   __extra_scripts__
}
```

Listing 1: The script to configure the debugger

Listing 1 shows the script that is previously inserted into the browser to configure the debugger and change the browser’s logging function.

```
function instrumentProperty(obj, prop, name,
     log_type) {
    ...
    Object.defineProperty(obj, prop, {
        get: function() {
            let value = origGetter.apply(this,
                 arguments);
            log(log_type, {
                ’type’: ’property’,
                ’name’: name,
                ’value’: value,
                ’access’: ’get’
            });
            return value;
        },
        set: function() {
            log(log_type, {
                ’type’: ’property’,
                ’name’: name,
                ’value’: arguments[0],
                ’access’: ’set’
            });
            return origSetter.apply(this,
                 arguments);
        }});}
```

Listing 2: Script that modifies the behaviour of function calls

4.2. PrivacyScanner enhancements

Once we have presented the original version of PrivacyScanner, this section describes the new extensions implemented in PrivacyScanner to detect additional tracking techniques not previously supported by PrivacyScanner, such as fingerprinting, WebRTC, and WebGL-based techniques. This new version of PrivacyScanner is available on GitHub[15] as open source.

4.2.1. WebGL fingerprinting

One of the techniques that PrivacyScanner cannot detect is WebGL fingerprinting, which is a technique that allows identifying the user’s device based on small changes in the rendering of an image . Therefore, one of our enhancements to the tool is to add the detection of this method.

The way the presence of WebGL is detected is very similar to that of Canvas, so we can reuse the methods: `instrumentFunction`, `instrumentProperty`, and `instrumentObject` to redefine the functions of the `WebGLRenderingContext` API and therefore collect the properties of these functions.

In detecting the WebGL-based technique, we used the criteria defined by [59], which assume that the website uses the WebGL technique if the website requests information from three properties of the API `WebGLRenderingContext`: `drawArrays`, `getSupportedExtensions`, and `getExtension`.

4.2.2. WebRTC fingerprinting

Another added enhancement is the detection of WebRTC fingerprints. Typically, a STUN server is used to track people using the WebRTC approach. Users can use a STUN server to discover their public IP address and NAT type through a specific local port. For this reason, browsers can use the WebRTC API to send requests to STUN servers, which then respond with the users’ local and public IP addresses.

For the detection of WebRTC-based techniques, we relied on the criteria defined by [59], who considers that WebRTC compromises user privacy if the website uses the following APIs, which belong to `RTCPeerConnection`: `createDataChannel`, `createOffer`, and `onicecandidate`.

4.2.3. Informative objects

The basis of fingerprinting is to get an accurate user identification, which consists of collecting information about the user’s browser and environment to identify the user. Websites mainly use the `Navigator` and `Screen` objects to collect information about the user’s environment, since they contain identifying attributes [59]. Some of these attributes are: Browser type and version, operating system, language, time zone, active plugins, font, CPU class, etc.

In order to keep a record of all the function calls to informational objects, we customized Chrome’s JavaScript engine via the Chrome DevTool protocol. This allows us to redefine the getter function of these

[15] https://github.com/ronghaopan/privacyscanner-master/tree/master.

objects, one unit is added to the corresponding counter each time the browser intercepts a getter request.

Once you have received the access number of each property, an analysis is performed to check if the website really uses this technique. In this case, the rules defined by [59] have been used as a basis. They consider tracking when websites query more than 80% of the properties of information objects. Then, we have enhanced PrivacyScanner so that can detect the query of the 21 different properties of information objects (the most relevant), 19 for the navigator and 2 for the screen.

4.2.4. Web storage

Web Storage is a storage system that allows data to be stored in key/value pairs. It is more powerful than cookies because it can store not only string structures but also more complex and diverse objects, from simple identifiers to personal documents. Another difference is that cookies are designed to be read by the server, which means that only the identifier is stored on the server, while all the information is kept in the browser. In contrast, web storage can only be read by the browser and is not limited to small volumes of data. However, they are not automatically sent to the server, but are retrieved directly via the JavaScript code.

Another interesting feature of the HTML5 web storage is that it provides a set of methods via its API to determine which stored data has been changed. So, if the session storage is used in two iframes, they can be informed about the changes in the other iframe. The same happens with local storage when multiple windows or tabs are open, as long as they point to the same domain and use the same protocol.

A third party can use a unique identifier stored in its local storage area to track a user across multiple sessions and build a profile of the user's interests. If the website also registers the identity of users (e.g., an e-commerce website that requires authentication credentials), the third-party provider can obtain a unique user identifier.

The HTML5 Web Storage API provides 2 objects for creating, querying, modifying, and deleting web storage for a specific domain: `window.localStorage` and `window.sessionStorage`. As shown in Table 1, PrivacyScanner does not have the functionality to detect the presence of web storage. Therefore, one of the enhancement added to the tool is a script that supports the detection of `window.localStorage` and `window.sessionStorage`.

Therefore, to detect the presence of web storage, we added a JavaScript script to the browser via the `Page.addScriptToEvaluate` function of Chrome DevTools to query the status of these two objects.

4.2.5. Web analytics tools

Web analytics tools use cookies and JavaScript to gather important statistics about a website, such as where visitors came from, how long they stayed on the website, how they found the website, and what online activities they performed on the website. These tools are often used for product analytics, social media analytics, and marketing analytics. Although there are laws that require the user's consent to their use and restrict certain uses of these tools, there is still a possibility that the user's privacy is not respected [3], as it is not known what data is collected and for what purpose.

In the following, we explain how the detection of the two most widely used web analytics tools has been implemented. The improvements and changes introduced in PrivacyScanner with respect to web analytics tools are the following:

- **Google analytics**: To detect the presence of Google Analytics, the PrivacyScanner's source code was modified to only look for network requests from the website under analysis that lead to a URL that starts with "stats.g.doubleclick" and also contains the prefix "UA" for the Google account identifier.

Table 2
Table comparing the old and new PrivacyScanner versions.

Tool	PrivacyScanner	PrivacyScanner (our version)
HTTP headers	■	■
Cookies	■	■
Third-party cookies	■	■
Web storage	□	■
Canvas	■	■
WebGL	□	■
WebRTC	□	■
Informative objects	□	■
Google analytics	■	■
Facebook pixel	□	■

- **Facebook pixel**: To detect the presence of the Facebook Pixel [60], we look for network requests from the website that go to Facebook, and look for data in the URL's query parameters that match the schema described in the Facebook Pixel documentation. We look for two different types of data: "standard events" and "custom events". Standard events are actions with default names that you know are supported by Facebook's Pixel advertising products. Custom events are any actions that are not standard events.

4.3. Assessment of PrivacyScanner enhancements in a real scenario

In this section, we present a comparison between the original version and our enhanced version of PrivacyScanner. To do so, we compare in Table 2 the different features supported by each version. It can be seen that the improvement is in the detection of web storage, new active fingerprinting techniques such as WebGL and WebRTC, the detection of informative objects in Javascripts, and the detection of Facebook Pixel tracking.

After implementing the previously mentioned enhancements (see Section 4.2), we decided to use PrivacyScanner in a real scenario. Namely, we decided to investigate the 50 most visited websites in Spain to check if these websites use any web tracking techniques or not.

Table 3 shows the results of the analysis of the 50 most visited websites in Spain. It can be seen that the websites in the categories news and media, e-commerce, and entertainment are the most visited, and they also show the highest use of tracking techniques to track their users. The news category has also been identified in other studies as the category with the most trackers in the desktop environment [8,31,61].

As for trackers, we find that Google and Facebook are the most commonly used third parties, as 50% of the analysed e-commerce websites use Google Analytics and 25% use Facebook Pixel. In [8], we can find that, for Chrome, similar results are obtained. In [33], regarding only first party cookies, they show similar results, where for Google is 54% and Facebook 14%. In [31], Google Analytics also is the most used.

In terms of JavaScript-based techniques, Canvas and WebGL-based techniques have been the most commonly used, as they are used to preserve the different hardware and software configurations of the system. These techniques are most commonly used on e-commerce and adult websites. WebRTC is most commonly used on entertainment, banking, and adult websites because it provides the ability to establish real-time communication that allows trackers to determine the IP of devices and know the location of users. Zafar and Das [8] estimated that fingerprinting activity was between 10% and 15% of websites. These results are consistent with our measurements in some categories such as news and media or social media, as can be seen in Table 3. However, in 5 categories, these fingerprinting techniques are not used, such as in the business or in culture categories.

When comparing our results with a large measurement (1-million-site) made by Englehardt and Narayanan [17], most of the results

Table 3
Summary of the results of the tracking analysis of the 50 most visited websites in Spain.

Category	No. of pages	Trackers	Local storage	Session storage	Informative objects	Canvas	WebGL	WebRTC	Facebook pixel	Google analytics
E-commerce	4	100%	100%	50%	25%	25%	25%	0%	25%	50%
Adults	5	20%	80%	0%	0%	20%	0%	20%	0%	40%
Bank	1	0%	0%	100%	0%	100%	100%	100%	0%	0%
Search engines	2	0%	0%	0%	0%	0%	0%	0%	0%	0%
Culture	1	0%	100%	0%	0%	0%	0%	0%	0%	0%
Entertainment	5	80%	60%	20%	0%	0%	20%	20%	0%	40%
News & Media	18	88.9%	17%	50%	44.44%	11.11%	16.66%	0%	38.88%	55.55%
Social media	7	14.28%	28.57%	28.57%	0%	0%	14.28%	0%	14.28%	28.57%
Services	4	50%	50%	50%	25%	0%	0%	0%	25%	50%
Travel	2	50%	100%	100%	50%	0%	0%	0%	0%	0%
Business	1	100%	100%	0%	0%	0%	0%	0%	0%	0%

match, even if they are not the same websites. For example, in terms of Google Analytics and Facebook usage, Englehardt's analysis found that the use of Google or third parties affiliated with Google is greater than Facebook. Furthermore, regarding the level of tracking on different categories of websites, news and e-commerce websites were found to contain the most trackers, which also matches with the results obtained in other works mentioned above.

As a conclusion, we can highlight that our enhancements provides a more comprehensive view of tracking mechanisms used in websites than the previous version of PrivacyScanner. Furthermore, these enhancements are useful because, as we saw in the analysis of the top 50 websites in Spain, there are websites that use web storage, WebGL, WebRTC, informative objects, and Facebook pixel to track users.

5. Assessment of the evolution of user privacy protection in Chrome

In this section, we assess the evolution of Chrome by means of three versions of this web browser: versions 83, 86, and 90. This assessment is approached from two perspectives. Firstly, we analyse the different techniques that have been incorporated in the versions under review. Secondly, we analyse the evolution of the privacy protection it offers by accessing various websites and observing the behaviour of each version.

Regarding the techniques supported in the Chrome evolution, Table 4 shows the significant privacy-preserving and privacy-offensive techniques used by Chrome in the versions tested in this work, as discussed in Section 2.2. In version 83, Chrome began to address privacy issues related to cookies, such as Cross-Site Request Forgery (CSRF), cross-site leaks, and certain Cross-Origin Resource Sharing (CORS) exploits, by incorporating the SameSite header mechanism. Starting from version 85,[16] Chrome changed its default policy from *no-referrer-when-downgrade* to *strict-origin-when-cross-origin* (known as the Referrer-Policy mechanism). However, in version 85, there was an issue where cached resources were fetched from the network, using their respective URLs as cache keys. Therefore, starting with version 86,[17] Chrome introduced HTTP cache partitioning to address this problem. No fingerprint protection mechanism was implemented in the three versions tested. Since the SameSite attribute and the new Referrer-Policy are already created in Chrome versions 86, none of these enhancements require user configuration in Chrome version 90. However, it is necessary to configure the cache mechanism via the control panel to enable this CSRF attack prevention mechanism.

Next, we proceed with an assessment of Chrome's protection level, based on the results obtained from accessing the 100 most visited websites according to Alexa ranking. We accessed these websites using our enhanced version of PrivacyScanner with different versions of Chrome. Namely, we have evaluated Chrome versions 83, 86, and 90. For each website, PrivacyScanner assessed the different privacy risks.

Table 4
Table comparing different versions of Chrome regarding protection techniques.

Techniques	Version 83	Version 86	Version 90
SameSite header	■	■	■
Referrer-Policy	□	■	■
Partitioning the cache	□	■	■
Fingerprinting	□	□	□

Fig. 3 shows the results of our analysis. We can see that the number of trackers has decreased in versions 86 and 90 in comparison to version 83, because the *SameSite* and *Referrer-Policy* headers have been integrated into these versions. For this reason, we can see a reduction of four pages in the *Tracker* column of version 90. Moreover, one of the pages that Chrome version 90 managed to block uses Google Analytics, resulting in a reduction of one page in the Google Analytics section. However, regarding fingerprinting techniques such as the informative object, Canvas, WebGL, and WebRTC, the browser did not have introduced protective mechanisms in these versions. Therefore, the same results were obtained in both version 83 and version 90.

Finally, based on the analysis performed, we can conclude that the latest versions of Chrome have reduced the risk of information leaks through the use of the SameSite attribute and the new Referrer-Policy, but still allow trackers to collect user or device characteristics through JavaScript-based tracking techniques and the use of external entities such as Google Analytics and Facebook Pixel.

6. Conclusions and future work

The importance of protecting the user's privacy when surfing the Web is a research topic that is becoming increasingly important. As a result of this research, we have more and more different types of privacy protection tools at our disposal. A fundamental element for the protection of privacy during our navigation is the mechanisms offered by the web browser. However, to our knowledge, no study has been conducted that has analysed the evolution of the mechanisms offered by web browsers.

In this research, we decided to investigate the evolution of Chrome to protect user privacy, as it is the most widely used. For the evaluation, we relied on the PrivacyScanner tool, to which we added a number of improvements to perform a more comprehensive analysis.

The results of our assessment show that between Chrome version 83 and 90 tracking protection has a 7.55% reduction in trackers and a 4.76% reduction in Google Analytics elements. Therefore, we can conclude that Chrome's improvements do not fully protect user privacy, as it is still possible for trackers to monitor users via their fingerprints, i.e., it still does not provide protection against fingerprinting techniques. One of the ways to protect against these techniques is to limit the amount of information about users that is published on websites so that it is not sufficient to identify and track users, and dynamic fingerprinting, which is the modification of certain user data so that the tracker cannot create a unique profile of the user.

[16] https://developer.chrome.com/blog/referrer-policy-new-chrome-default/.

[17] https://developer.chrome.com/blog/http-cache-partitioning/.

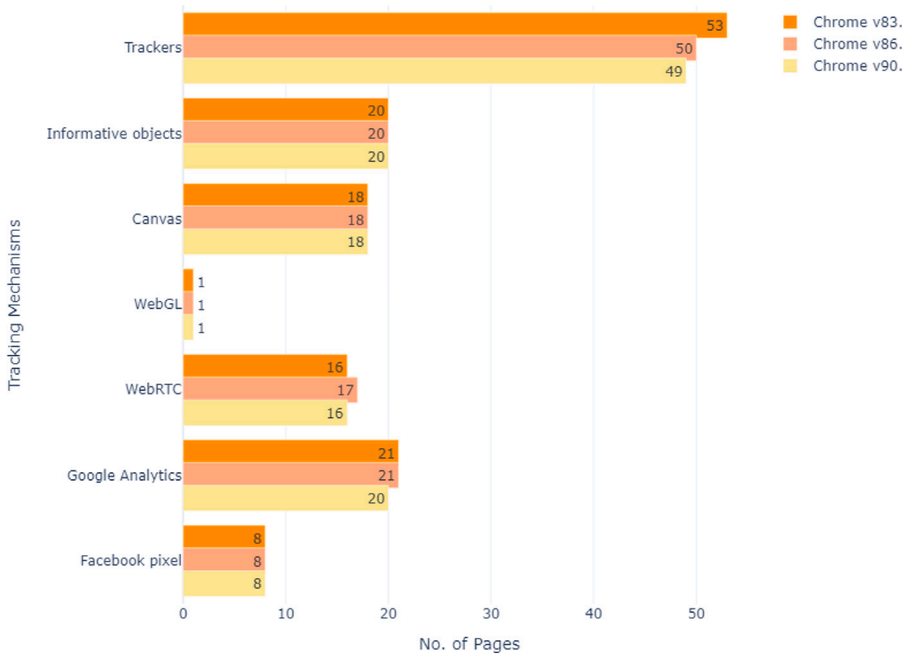

Fig. 3. Chrome evaluation results.

As future work, we consider that PrivacyScanner can be improved with the following features: the implementation that indicates the level of privacy risk when the user accesses the website, the possibility for the user to choose which types of techniques to analyse instead of performing a full analysis, and the addition of support for different web browsers other than Chrome.

CRediT authorship contribution statement

Ronghao Pan: Conception and design of study, Writing – original draft, Development, Data collection, Analysis. **Antonio Ruiz-Martínez:** Conception and design of study, Writing – original draft.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Antonio Ruiz-Martinez reports financial support was provided by European Commission.

Antonio Ruiz-Martinez reports was provided by Spanish Ministry of Science, Innovation and Universities.

Data availability

Data will be made available on request.

Acknowledgements

We would like to thank the reviewers for taking the time and effort to review the manuscript. We sincerely appreciate all the valuable comments and suggestions, which have significantly helped us improve the quality of the manuscript.

This work has been funded by the European Commission's H2020 Programme under the Grants Agreement Numbers 830929 (CyberSec4Europe project) and H2020-SU-DS-2019/883335 - PALANTIR (Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises), and the European Commission (FEDER/ERDF); the Spanish Ministry of Science, Innovation and Universities, under the ONOFRE 3 project (Grant No. PID2020-112675RB-C44).

All authors approved the final version of manuscript to be published

Funding

This research has been funded by the European Commission's H2020 Programme under the Grants Agreement Numbers 830929 (CyberSec4Europe project) and H2020-SU-DS-2019/883335 - PALANTIR (Practical Autonomous Cyberhealth for resilient SMEs & Microenterprises), and the European Commission (FEDER/ERDF); the Spanish Ministry of Science, Innovation and Universities, under the ONOFRE 3 project (Grant No. PID2020-112675RB-C44).

References

[1] Perdices D, López de Vergara JE, González I, de Pedro L. Web browsing privacy in the deep learning era: Beyond VPNs and encryption. Comput Netw 2023;220:109471. http://dx.doi.org/10.1016/j.comnet.2022.109471.

[2] Fouad I, Santos C, Legout A, Bielova N. My cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting. Proc Priv Enhanc Technol 2022. http://dx.doi.org/10.56553/popets-2022-0063.

[3] Papadogiannakis E, Papadopoulos P, Kourtellis N, Markatos EP. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In: Proceedings of the web conference 2021. 2021, p. 2130–41.

[4] Massardier-Meca M, Ruiz-Martínez A. Towards a privacy web scanner for end-users. In: Thampi SM, Martinez Perez G, Ko R, Rawat DB, editors. Security in computing and communications. Communications in computer and information science, Springer; 2020, p. 174–85. http://dx.doi.org/10.1007/978-981-15-4825-3_14.

[5] Estrada-Jiménez J, Parra-Arnau J, Rodríguez-Hoyos A, Forné J. Online advertising: Analysis of privacy threats and protection approaches. Comput Commun 2017;100:32–51.

[6] Ruiz-Martínez A. A survey on solutions and main free tools for privacy enhancing web communications. J Netw Comput Appl 2012;35(5):1473–92. http://dx.doi.org/10.1016/j.jnca.2012.02.011.

[7] Mehrnezhad M, Coopamootoo K, Toreini E. How can and would people protect from online tracking? Proc Priv Enhanc Technol 2022;1:105–25.

[8] Zafar A, Das A. Comparative privacy analysis of mobile browsers. In: Proceedings of the thirteenth ACM conference on data and application security and privacy. CODASPY '23, New York, NY, USA: Association for Computing Machinery; 2023, p. 3–14. http://dx.doi.org/10.1145/3577923.3583638.

[9] Maryam Abdulaziz Saad Bubukayr MF. Web tracking domain and possible privacy defending tools: A literature review. J Cyber Secur 2022;4(2):79–94. http://dx.doi.org/10.32604/jcs.2022.029020.

[10] Pereira F, Crocker P, Leithardt VR. PADRES: Tool for PrivAcy, data regulation and security. SoftwareX 2022;17:100895. http://dx.doi.org/10.1016/j.softx.2021.100895.

[11] Alharbi JA, Albesher AS, Wahsheh HA. An empirical analysis of E-governments' cookie interfaces in 50 countries. Sustainability 2023;15(2). http://dx.doi.org/10.3390/su15021231, URL https://www.mdpi.com/2071-1050/15/2/1231.

[12] El Hana N, Mercanti-Guérin M, Sabri O. Cookiepocalypse: What are the most effective strategies for advertisers to reshape the future of display advertising? Technol Forecast Soc Change 2023;188:122297. http://dx.doi.org/10.1016/j.techfore.2022.122297.

[13] Meek S, Holguin I, Das S. Can johnny really be anonymous? Evaluation of user data privacy within tor. In: Proceedings of the 6th workshop on technology and consumer protection (ConPro '22) co-located with the 43th IEEE symposium on security and privacy (IEEE S&P). San Francisco, California; 2022.

[14] Papadogiannaki E, Ioannidis S. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. ACM Comput Surv 2021;54(6). http://dx.doi.org/10.1145/3457904.

[15] Rasaii A, Singh S, Gosain D, Gasser O. Exploring the cookieverse: A multi-perspective analysis of web cookies. In: Brunstrom A, Flores M, Fiore M, editors. Passive and active measurement. Lecture notes in computer science, Cham: Springer Nature Switzerland; 2023, p. 623–51. http://dx.doi.org/10.1007/978-3-031-28486-1_26.

[16] Hiremath PN, Armentrout J, Vu S, Nguyen TN, Minh QT, Phung PH. MyWebGuard: Toward a user-oriented tool for security and privacy protection on the web. In: Dang TK, Küng J, Takizawa M, Bui SH, editors. Future data and security engineering. Cham: Springer International Publishing; 2019, p. 506–25.

[17] Englehardt S, Narayanan A. Online tracking: A 1-million-site measurement and analysis. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. CCS '16, New York, NY, USA: Association for Computing Machinery; 2016, p. 1388–401. http://dx.doi.org/10.1145/2976749.2978313.

[18] Castell-Uroz I, Sanz-García R, Solé-Pareta J, Barlet-Ros P. Demystifying content-blockers: Measuring their impact on performance and quality of experience. IEEE Trans Netw Serv Manag 2022.

[19] Bertmar S, Gerhardsen J, Ekblad A, Höglund A, Mineur J, Öknegård Enavall I, Le M-H, Carlsson N. Who's most targeted and does my new adblocker really help: A profile-based evaluation of personalized advertising. In: Proceedings of the 20th workshop on workshop on privacy in the electronic society. WPES '21, New York, NY, USA: Association for Computing Machinery; 2021, p. 245–9. http://dx.doi.org/10.1145/3463676.3485617.

[20] Merzdovnik G, Huber M, Buhov D, Nikiforakis N, Neuner S, Schmiedecker M, Weippl E. Block me if you can: A large-scale study of tracker-blocking tools. In: 2017 IEEE European symposium on security and privacy (EuroS P). 2017, p. 319–33. http://dx.doi.org/10.1109/EuroSP.2017.26.

[21] Statcounter. Browser market share worldwide. 2023, URL https://gs.statcounter.com/browser-market-share.

[22] NetMarketShare. Browser market share. 2023, URL https://www.netmarketshare.com/browser-market-share.aspx.

[23] Winkler S, Zeadally S. An analysis of tools for online anonymity. Int J Pervasive Comput Commun 2015;11(4):436–53. http://dx.doi.org/10.1108/IJPCC-08-2015-0030.

[24] Huete Trujillo DL, Ruiz-Martínez A. Tor hidden services: A systematic literature review. J Cybersecur Priv 2021;1(3):496–518. http://dx.doi.org/10.3390/jcp1030025.

[25] Alidoost Nia M, Ruiz-Martínez A. Systematic literature review on the state of the art and future research work in anonymous communications systems. Comput Electr Eng 2018;69:497–520. http://dx.doi.org/10.1016/j.compeleceng.2017.11.027.

[26] Bubukayr M, Frikha M. Effective techniques for protecting the privacy of web users. Appl Sci 2023;13(55):3191. http://dx.doi.org/10.3390/app13053191.

[27] Kumar K, Gupta MK, Jaglan V. Privacy protection in personalized web search using software applications–tools and plug-ins. In: Proceedings of the international conference on innovative computing & communication (ICICC) 2021. 2021, http://dx.doi.org/10.2139/ssrn.3884638.

[28] Phung PH, Pham H-D, Armentrout J, Hiremath PN, Tran-Minh Q. A user-oriented approach and tool for security and privacy protection on the web. SN Comput Sci 2020;1(4):1–16.

[29] Mazel J, Garnier R, Fukuda K. A comparison of web privacy protection techniques. Comput Commun 2019;144:162–74. http://dx.doi.org/10.1016/j.comcom.2019.04.005.

[30] Karopoulos G, Geneiatakis D, Kambourakis G. Neither good nor bad: A large-scale empirical analysis of HTTP security response headers. In: Fischer-Hübner S, Lambrinoudakis C, Kotsis G, Tjoa AM, Khalil I, editors. Trust, privacy and security in digital business. Cham: Springer International Publishing; 2021, p. 83–95.

[31] Yang Z, Yue C. A comparative measurement study of web tracking on mobile and desktop environments. Proc Priv Enhanc Technol 2020;2020(2):22–44.

[32] Paci F, Pizzoli J, Zannone N. A comprehensive study on third-party user tracking in mobile applications. In: Proceedings of the 18th international conference on availability, reliability and security. ARES '23, New York, NY, USA: Association for Computing Machinery; 2023, p. 1–8. http://dx.doi.org/10.1145/3600160.3605079.

[33] Demir N, Theis D, Urban T, Pohlmann N. Towards understanding first-party cookie tracking in the field, Vol. P-323. 2022, p. 19–34. http://dx.doi.org/10.18420/sicherheit2022_01.

[34] Iqbal U, Englehardt S, Shafiq Z. Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. In: 2021 IEEE symposium on security and privacy (SP). 2021, p. 1143–61. http://dx.doi.org/10.1109/SP40001.2021.00017.

[35] Akhavani SA, Jueckstock J, Su J, Kapravelos A, Kirda E, Lu L. Browserprint: an analysis of the impact of browser features on fingerprintability and web privacy. In: Liu JK, Katsikas S, Meng W, Susilo W, Intan R, editors. Information security. Cham: Springer International Publishing; 2021, p. 161–76.

[36] Mehrnezhad M, Coopamootoo K, Toreini E. How can and would people protect from online tracking? Proc Priv Enhanc Technol 2022;2022(1):105–25.

[37] Lerner A, Simpson AK, Kohno T, Roesner F. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In: USENIX security symposium. 2016.

[38] Jiang W, Wang X, Song X, Liu Q, Liu X. Tracking your browser with high-performance browser fingerprint recognition model. China Commun 2020;17(3):168–75. http://dx.doi.org/10.23919/JCC.2020.03.014.

[39] Khodayari S, Pellegrino G. The state of the SameSite: Studying the usage, effectiveness, and adequacy of SameSite cookies. In: 2022 IEEE symposium on security and privacy (SP). 2022, p. 1590–607. http://dx.doi.org/10.1109/SP46214.2022.9833637.

[40] Eiji Kitamura. Gaining security and privacy by partitioning the cache. 2020, URL https://developer.chrome.com/blog.

[41] Lin X, Ilia P, Polakis J. Fill in the blanks: Empirical analysis of the privacy threats of browser form autofill. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. CCS '20, New York, NY, USA: Association for Computing Machinery; 2020, p. 507–19. http://dx.doi.org/10.1145/3372297.3417271, URL https://dl.acm.org/doi/10.1145/3372297.3417271.

[42] Vila P, Köpf B. Loophole: Timing attacks on shared event loops in chrome. 2017, p. 849–64, URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/vila.

[43] Gabet RM, Seigfried-Spellar KC, Rogers MK. A comparative forensic analysis of privacy enhanced web browsers and private browsing modes of common web browsers. Int J Electron Secur Digit Forensics 2018;10(4):356–71. http://dx.doi.org/10.1504/IJESDF.2018.095126.

[44] Nelson R, Shukla A, Smith C. Web browser forensics in google chrome, mozilla firefox, and the tor browser bundle. In: Digital forensic education. 2020, URL https://api.semanticscholar.org/CorpusID:199586332.

[45] Zafar A, Das A. Comparative privacy analysis of mobile browsers. In: Proceedings of the thirteenth ACM conference on data and application security and privacy. CODASPY '23, New York, NY, USA: Association for Computing Machinery; 2023, p. 3–14. http://dx.doi.org/10.1145/3577923.3583638, URL https://dl.acm.org/doi/10.1145/3577923.3583638.

[46] Madhusudhan R, Surashe SV. Privacy and security comparison of web browsers: A review. In: International conference on advanced information networking and applications. Springer; 2022, p. 459–70.

[47] Kariryaa A, Savino G-L, Stellmacher C, Schöning J. Understanding users' knowledge about the privacy and security of browser extensions. 2021, p. 99–118, URL https://www.usenix.org/conference/soups2021/presentation/kariryaa.

[48] Bhadana K, Panda SP. Free services - a threat to privacy: Ensuring a safe online presence using chrome browser extension. In: 2021 3rd international conference on advances in computing, communication control and networking (ICAC3N). 2021, p. 1605–8. http://dx.doi.org/10.1109/ICAC3N53548.2021.9725431, URL https://ieeexplore.ieee.org/document/9725431.

[49] Abu-Salma R, Livshits B. Evaluating the end-user experience of private browsing mode. In: Proceedings of the 2020 CHI conference on human factors in computing systems. CHI '20, New York, NY, USA: Association for Computing Machinery; 2020, p. 1–12. http://dx.doi.org/10.1145/3313831.3376440, URL https://dl.acm.org/doi/10.1145/3313831.3376440.

[50] Akhavani SA, Jueckstock J, Su J, Kapravelos A, Kirda E, Lu L. Browserprint: An analysis of the impact of browser features on fingerprintability and web privacy. In: Information security: 24th international conference, ISC 2021, virtual event, november 10–12, 2021, proceedings 24. Springer; 2021, p. 161–76.

[51] Karami S, Ilia P, Solomos K, Polakis J. Carnus: Exploring the privacy threats of browser extension fingerprinting. In: Proceedings of the 27th network and distributed system security symposium (NDSS). 2020.

[52] Fiore U, Castiglione A, Santis AD, Palmieri F. Countering browser fingerprinting techniques: Constructing a fake profile with google chrome. In: Proceedings of the 2014 17th international conference on network-based information systems. NBIS '14, USA: IEEE Computer Society; 2014, p. 355–60. http://dx.doi.org/10.1109/NBiS.2014.102.

[53] Eckersley P. How unique is your web browser? In: Privacy enhancing technologies: 10th international symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings 10. Springer; 2010, p. 1–18.

[54] Fernández-Fuentes X, Pena TF, Cabaleiro JC. Digital forensic analysis methodology for private browsing: Firefox and chrome on linux as a case study. Comput Secur 2022;115:102626.

[55] Hughes K, Papadopoulos P, Pitropakis N, Smales A, Ahmad J, Buchanan WJ. Browsers' private mode: Is it what we were promised? Computers 2021;10(1212):165. http://dx.doi.org/10.3390/computers10120165.

[56] Tsalis N, Mylonas A, Nisioti A, Gritzalis D, Katos V. Exploring the protection of private browsing in desktop browsers. Comput Secur 2017;67:181–97. http://dx.doi.org/10.1016/j.cose.2017.03.006.

[57] Eubank C, Melara MS, Perez-Botero D, Narayanan A. Shining the floodlights on mobile web tracking — A privacy survey. 2013.

[58] Acar G, Juarez M, Nikiforakis N, Diaz C, Gürses S, Piessens F, Preneel B. Fpdetective: dusting the web for fingerprinters. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. CCS '13, New York, NY, USA: Association for Computing Machinery; 2013, p. 1129–40. http://dx.doi.org/10.1145/2508859.2516674, URL https://dl.acm.org/doi/10.1145/2508859.2516674.

[59] Ashouri M. A large-scale analysis of browser fingerprinting via chrome instrumentation. In: ICIMP 2019, the fourteenth international conference on internet monitoring and protection. 2019.

[60] Dimova Y, Franken G, Le Pochat V, Joosen W, Desmet L. Tracking the evolution of cookie-based tracking on facebook. In: Proceedings of the 21st workshop on privacy in the electronic society. WPES '22, New York, NY, USA: Association for Computing Machinery; 2022, p. 181–96. http://dx.doi.org/10.1145/3559613.3563200.

[61] Sørensen J, Kosta S. Before and after GDPR: The changes in third party presence at public and private European websites. In: The world wide web conference. WWW '19, New York, NY, USA: Association for Computing Machinery; 2019, p. 1590–600. http://dx.doi.org/10.1145/3308558.3313524.