




# Towards a Privacy Web Scanner for End-Users

Myriam Massardier-Meca and Antonio Ruiz-Martínez<sup>(✉)</sup> 

Department of Information and Communications Engineering,  
Faculty of Computer Science, University of Murcia, Murcia, Spain  
{myriam.m.m,arm}@um.es

**Abstract.** Internet users progressively have realized that due to our online activities our privacy can be compromised and that much personal information can be gathered. To cope with this problem, both technological solutions and regulations have emerged which are steadily being improved. But, apart from these privacy-preserving tools, we need tools to show privacy risks and that end-users be aware the risks they might be exposed to when they access a website. Currently, there are some tools of this kind. However, they are not oriented to end-users (users with not a high/moderate knowledge on technical issues related to tracking). To address this issue, we have started the development of a Web scanner, named Privacy Web Scanner, that is, in charge of analyzing a website and provide in a simple and graphical way the privacy implications of accessing that site for end-users. In the paper, we present the main issues that should be considered in this kind of scanner, its design and the features of the current beta version.

**Keywords:** Privacy · Tracking · Web scanner

## 1 Introduction

Surfing on the Web is a common activity that every Internet user makes a lot of times along the day. In general, the content that is accessed is “for free”. In return this free access, the Web content provider or publisher makes revenues from showing adverts in the content that is provided or by gathering personal information that is used to offer us personalized services. This information later will be sold to other third parties.

So that the different adverts are customized and, therefore, it is more probable that they are clicked by the users, different entities try to create a user’s profile that associates to this user (identified or not) what his/her preferences are, browsing behaviour, and so on. Thus, advertisers can show customized adverts according to the user’s interests. This is what is called Online Behavioral Advertising (OBA) [3]. In these profiles, although the user, in general, is not identified, he/she can be tracked when he/she is surfing due to when the user access to a website he/she is also accessing to the same time to third-parties that are tracking the user into the different sites. These third parties can be web analytics

services, advertisers, social networks. This unidentified profile could be linked to a user if, in some transactions, the user reveals some personally identifiable information. Furthermore, nowadays, as we are “hyperconnected” to the Internet, we are widely traceable by third-parties [6].

The tracking of the users is made using different kinds of mechanisms [5, 11]. The most well known is the use of cookies. There are also other mechanisms such as tracking the IP address, fingerprinting techniques, and local storage [8, 9, 11].

To protect from this tracking, users have used different types of tools such as cookie erasers, ad blockers, privacy-aware browsers or anonymous communication networks such as Tor [2, 8, 9, 11].

So far many users were not aware that they were being tracked. However, some regulations, such as the General Data Protection Regulation (GDPR) that have made that a website has to inform users about the use of the cookies and third-party trackers, whether they are gathering user’s information, what they are going to do with that information and how the can query, modify and delete their information.

Recently, different web scanner services have appeared to show the different risks we are exposed to when we access a website. The aim of these web scanners is that a user obtains information on the different mechanisms that the website is using to track him/her by showing information on cookies used, the third parties the website is working with, and so on. Thus, users can be aware of the impact a website can have on their privacy. From our point of view, this kind of solution is quite interesting to promote users’ literacy in privacy. However, from our point of view, the main problem these tools have is that, in most of the cases, the information they are providing will only be understood by advanced users (with a technical background on privacy). However, the information shown can be overwhelming to non-experts users and they might not understand it.

As a response to this problem, in this paper, we present a web scanner, named PrivacyWebScanner (PWS), whose aim is showing privacy risks associated with a website in a simple way that can be understood by end-users. Thus, this paper presents the goals that should satisfy this kind of scanner, what kind of information should be shown, the design of the scanner we have made and a use case where we show the information that the scanner presents to an end-user.

The rest of this paper is organized as follows. In Sect. 2 we present related work. Section 3 presents the requirements and goals established for our privacy web scanner, and the architecture of the tool we consider it should have. In Sect. 4 we present some implementation issues. The results of the access to a website are shown in Sect. 5. After that, in Sect. 6, we present the limitations of PWS and, finally, in Sect. 7, we present conclusions and future work.

## 2 Related Work

This section introduces the different mechanisms that can be used to track a user. These mechanisms has to be taken into account in a Privacy Web Scanner to show the possible risks a user is exposed to when he/she is going to access a website. After that, we present the main web tools developed so far to analyze a website.

## 2.1 Tracking Mechanisms

A user can be tracked in a website by using a plethora of mechanisms. As commented by Estrada-Jiménez et al. [5], most of the mechanisms are based on the use of cookies and they could be classified on first-party tracking, third-party tracking, cookie matching, fingerprinting, flash cookies, canvas fingerprinting, and HTML5 local storage. Although there are other mechanisms such as Etags, or using iframes and social widgets [9].

In first-party tracking, the publisher tracks the user by means of cookies and information released by the user agent. Third-party tracking is produced when the user is accessing the publisher and within the content, there are links to other contents placed in other parties different from the publisher such as social networks, content providers, advertisers, demand-side platforms, supply-side platforms, etc. Thus, the user is accessing these third-parties and they gather information from the users and create the profile to offer later personalized advertising [10]. In general, this tracking is made using Web bugs, cookies and the Referer header of the HTTP request. However, as the cookies received from third-parties can be blocked by means of adblockers, other mechanisms have been developed such as evercookies, cookies matching, fingerprinting, and the use of HTML5 storage, canvas fingerprinting or iframes. In general, these mechanisms are based on the use of two main components: Javascript and data storage mechanisms on the client, which allow the storage of unique identifiers in multiple storage locations [9].

To protect from tracking, there are different privacy-enhancing technologies achieving different levels of privacy protection and that in many cases should be combined to achieve the best level of privacy protection. Some of these tools are anonymous communications tools, adblockers, cookies erasers, etc. More details can be found in different works [1, 2, 8, 9, 11]. We are not going enter into detail because the purpose of this paper is not to analyze protection mechanisms, we are interested in providing a view of the elements that we should analyze in a web page to determine if a user is exposed to some kind of privacy risk.

Taking into account the different tracking mechanisms previously mentioned, to know if we are being tracked, we should analyze whether a web page is using or including some of these elements: cookies, web bugs, Javascript, canvas, iframes, and HTML5 storage. In the case of Javascript, to be more specific, we should consider if Javascript is performing some kind of fingerprinting or storing some data on client storage.

## 2.2 Websites Showing Privacy Risks

Nowadays, due to privacy is becoming a more important issue in society by both end-users, regulators, data protection authorities and data protection non-governmental organizations, different (privacy) website scanning services are available. Next, we present some of the most well-known: urlscan.io<sup>1</sup>, webbkoll<sup>2</sup>,

<sup>1</sup> <https://urlscan.io>.

<sup>2</sup> <https://webbkoll.dataskydd.net/>.

WebCookies<sup>3</sup>, and PrivacyScore<sup>4</sup>. In all these scanners, the scanner access as if an end-user was browsing the web page and records some information. The information gathered depends on the scanner and we comment it next.

*urlscan.io* provides information about domains and IPs accessed, resources (Javascript, CSS, images, etc), HTTP requests, cookies, certificates, a screenshot of the site, indicators of compromise, and technologies used.

*webbkoll* allows a user to check the different data-protecting measures considering a web browser without plug-ins and with Do Not Track disabled. Namely, this tool offers information about the use of HTTPs, content security policy, strict transport security (HSTS), referrer policy, cookies, third-parties, HTTP headers protection, local storage, and information about the server location.

*WebCookies* is a scanner for Web application vulnerability and privacy. It mainly provides information about cookies (third-party, persistent, and session). Furthermore, it provides information about SSL/TLS security, security-related HTTP headers, HTML5 storage (local and session), advertising publisher identifiers, and resources (images, CSS, Javascript, web fonts, audio and video files, and iframes). One interesting information this scanner provides is a privacy impact score mainly based on the cookies' information gathered.

*PrivacyScore* is a web scanner that aims to show the security and privacy measures that websites are taking [7]. It shows information about third-parties (tracking or advertising companies), cookies, if Google Analytics is used, compliance with GDPR, whether web server and mail server are located in the same country, different tests about the encryption of Web traffic (certificates, HTTPS, HSTS, TLS, attacks as CRIME, BREACH, POODLE, etc.), the use of security-related headers, and encryption of mail traffic. Thus, they do not only cover tracking mechanisms but also software development errors can lead to user's privacy is put at risk [4]. Based on this information they provide an overall rating.

After analyzing the different (privacy) web scanners mentioned, we consider that they reflect the importance of being aware of the different privacy risks we are exposed to when we access a website, how we can be tracked and by whom. Our analysis also reveals that not all the web scanners analyze all the information mentioned in the previous section and depending on the tool are more focused on some issues, e.g., some are more focused on cookies or in HTTP information. But there are elements such as iframes or canvas that are not analyzed and that can also be a privacy risk. Another important issue we have found is that, in many cases, the information is shown in a quite technical and difficult way to be understood by an end-user since it is not presented in a simple way that helps its understanding because of the amount technical and detailed information shown in the results.

---

<sup>3</sup> <https://webcookies.org/>.

<sup>4</sup> <https://privacyscore.org/>.

### 3 PrivacyWebScanner: A Web Scanner for Privacy Risks Identification

In this section we present the web scanner for privacy risk identification we have developed and named PrivacyWebScanner or PWS for short. Next, we define the goals and requirements that we establish for it, we describe its architecture and how it has been developed.

#### 3.1 Goals and Requirements for PWS

The main goal of PWS is to offer a web scanner that helps any end-user without any technical skills to understand and interpret the different privacy risks associated with the access to a website.

As requirements, we establish that PWS should allow a user to access a website where he/she can indicate a URL to be analyzed, then, the system should detect the elements previously mentioned in Sect. 2.1, next, the results should be presented in a simple way and using categories, finally, the system should work correctly both in desktop and mobile devices.

#### 3.2 PrivacyWebScanner Architecture

PWS is a web application that runs on a web server, so that the user can access it by just using a web browser. The proposed architecture follows a modular approach, which is depicted in Fig. 1. This architecture has been designed so that the following information can be gathered and shown as result of the analysis: web beacons, cookies, fingerprinting variables, resources (Javascript, iframes, canvas, and images), third-parties, and HTML web storage (more details are provided in Sect. 3.3). Next, we describe the different modules of PWS.

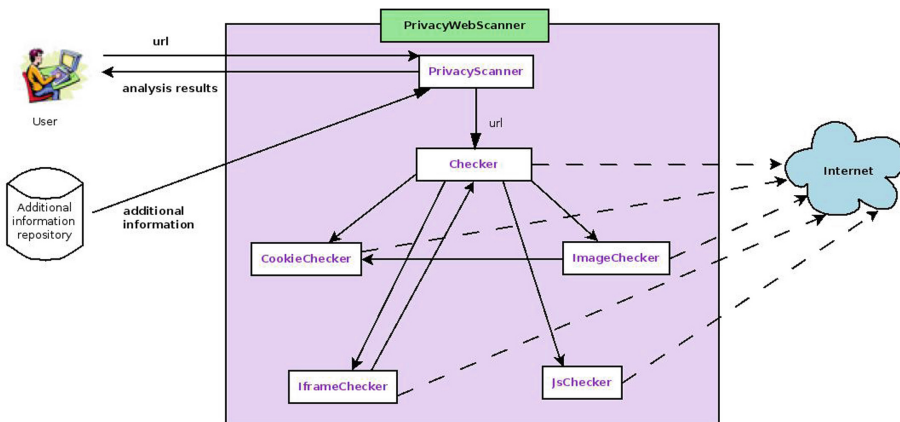


Fig. 1. PrivacyWebScanner system architecture.

As depicted in Fig. 1, *PrivacyScanner* module takes as input the URL of the target web page, loads the necessary information from the repository of additional information and passes the URL to the *Checker* module. Then, it will receive the final information on the analysis of the web page obtained from the other modules. Finally, it processes all the information to compose a graph with the results and shows it to the user along with other relevant information.

The *Checker* module is the PWS processing core. It checks if the target URL is valid and retrieves its HTML code. Next, it gets the DOM model and parses it to find web elements that can be malicious. For each element type, it invokes the module responsible for its analysis and stores all the information collected from the other modules. It also gets a snapshot of the target website.

The *CookieChecker* module is responsible for collecting all cookies. It gets a resource as input and returns a list of cookies including both HTTP cookies and those generated by script code. To obtain the latter, it uses a tool integrated into the system that executes JavaScript code<sup>5</sup>.

As Fig. 1 depicts, the *ImageChecker* module receives a list of images and cookies. For each image, this module checks whether it is a canvas or a web beacon. It also checks if the image belongs to a third party and invokes the *CookieChecker* module to obtain cookies. As a result, the module obtains a list of web beacons, a list of images containing canvas, a list of cookies generated when consulting the images, and a list of third parties.

The *IframeChecker* module receives a list of iframe resources and the collected cookies. For each iframe, it invokes the *Checker* module to perform a complete scan of the resource. As a result, it returns a list of the third-parties detected and all the elements referenced in the iframes.

The *JsChecker* module is responsible for script code processing. It takes both the URL and the DOM document of a script as input. Then, it parses the resource in order to find fingerprinting variables or calls to other scripts. Finally, this module provides as output a list of variables that could be used to perform fingerprinting and another list of suspicious scripts.

The *Additional information repository* has also been defined with additional information about the most common trackers and cookies that are usually used by web pages. More in detail, we have distinguished three types of trackers:

- Advertising: third-parties that collect information about the user to create a profile and thus offer personalized advertising.
- Web analytics: third-parties that collect information of the user mainly to generate statistics about certain website parameters.
- Social: third-parties associated with social networks. For example, the well-known *Like* button is also used to track users.

Therefore, when the target website makes use of a known third party, our tool will indicate what type of tracker is and it will show information about such a third party. To elaborate on the list of *known* third-parties, we have considered the works of Zimmeck et al. [13] and Starov and Nikiforakis [12].

<sup>5</sup> ChromeDriver, <http://chromedriver.chromium.org/>.

It is also interesting to provide users with information about cookies. To prepare the list of cookies, we rely on the Google Analytics cookies<sup>6</sup> and the Cookiepedia<sup>7</sup> as primary sources. The rest of the list has been created during the evaluation of our tool.

### 3.3 PWS Output

From the information gathered in the analysis, PWS shows the following items:

- A diagram with the third-parties detected in the analysis of the website, detailing, for each of them its type, the number of web beacons, cookies, iframes, fingerprinting variables, JavaScript files and web storage.
- A summary of the number of third-parties and suspect elements detected, with a link to detailed information.
- A screenshot of the website so that the end-user can easily see if the website contains many advertising elements.
- A list of the third-parties detected on the target website split by category. If the tool has information in the repository about the third parties, this information will also be displayed.
- A list of the images used on the website, indicating their size and highlighting those that are web beacons or canvas.
- A list of session and persistence cookies with their properties. We include relevant parameters of a cookie such as the domain, value, expiration date, if it is safe or has the “http” field. Furthermore, additional information about the cookie is also displayed if it has been cataloged as known.
- A list of HTML5 web storage where we distinguish between local and session storage.
- Finally, a list of scripts used on the website.

## 4 Implementation Issues

PWS has been developed with J2EE and two JSF Frameworks: JSF's Primefaces<sup>8</sup> and BootsFaces<sup>9</sup>. These frameworks facilitate the development of a responsive design. Furthermore, the open-source library Jsoup<sup>10</sup> has been used to manage HTTP requests. For more detailed processing of the web page, we have used the Selenium<sup>11</sup> WebDriver API with the browser driver implementation for Chrome, i.e., ChromeDriver<sup>12</sup> in headless mode, and HtmlUnitDriver<sup>13</sup>.

Hereafter, we describe the implementation for each of the system modules.

<sup>6</sup> <https://policies.google.com/technologies/types>.

<sup>7</sup> <https://cookiepedia.co.uk/>.

<sup>8</sup> <https://www.primefaces.org/>.

<sup>9</sup> <https://www.bootsfaces.net/>.

<sup>10</sup> <https://jsoup.org/>.

<sup>11</sup> <https://www.seleniumhq.org/projects/webdriver>.

<sup>12</sup> <https://github.com/SeleniumHQ/selenium/wiki/ChromeDriver>,  
<http://chromedriver.chromium.org/>.

<sup>13</sup> <https://github.com/SeleniumHQ/htmlunit-driver>.

## 4.1 PrivacyScanner

This module is responsible for initializing PWS. It verifies that the user entered a valid URL. Then, it invokes the *Checker* module and, it processes its output to show the result of the risk analysis.

PrivacyWebScanner uses the additional information repository of trackers and cookies. It has been implemented as JSON files that are processed by the open-source library Gson<sup>14</sup>.

This module is also responsible for the user interface and builds a graph in which the central node is the target website and the rest of the nodes represent the third-parties used by the website. Each node shows the information on the suspicious elements analyzed by also indicating whether the third party is a well-known tracker.

## 4.2 Checker

This module manages the HTTP connection with the target website and processes the obtained HTML code. Next, it selects on that page the resources corresponding to the web elements that are suspicious. For this goal, it calls each module by passing the elements found and the cookies. This module also obtains the HTML5 Web Storage by invoking the *Chrome Driver API* and, finally, it takes a snapshot of the website.

## 4.3 CookieChecker

The main goal of this module is to retrieve the cookies. On the one hand, it searches the cookies that appear in the header (e.g. *Cookie*, *Set-Cookie*, *Cookie2*, and *Set-Cookie2* fields) of the HTTP requests to the resources of the analyzed web page. On the other hand, it retrieves cookies that are generated by script code by using the *HtmlUnitDriver*.

## 4.4 ImageChecker

This module processes the images obtained from the target website. For each image, it checks if it is a canvas or a web beacon by calculating its size. Later, the *CookieChecker* module is called by passing the cookies of the target website, if the image belongs to the domain, or third-party cookies, if there were otherwise.

## 4.5 IframeChecker

This module receives a list of iframes and cookies. For each iframe, it invokes the *Checker* module to perform a complete analysis. Depending on the domain of the iframe, the module invokes with the cookies of the domain or with those of the third-party and updates the *Referer* header of the HTTP request header.

<sup>14</sup> Google Gson, <https://github.com/google/gson>.



## 4.6 JsChecker

This module receives script elements and analyzes them to detect if they use variables that can be used to perform fingerprinting. For this goal, the script is parsed. In the case of external scripts, we use the *HtmlUnitDriver* driver to retrieve the script code and analyze the variables suspicious to be used to perform fingerprinting. Besides, this module detects the use of three social networks<sup>15,16,17</sup> by using regular-expression patterns.

## 5 Use Cases

We have tested PWS by accessing to a tourism website, namely, the Official Guide to New York City<sup>18</sup>. The main results of the PWS analysis are shown in Fig. 2.

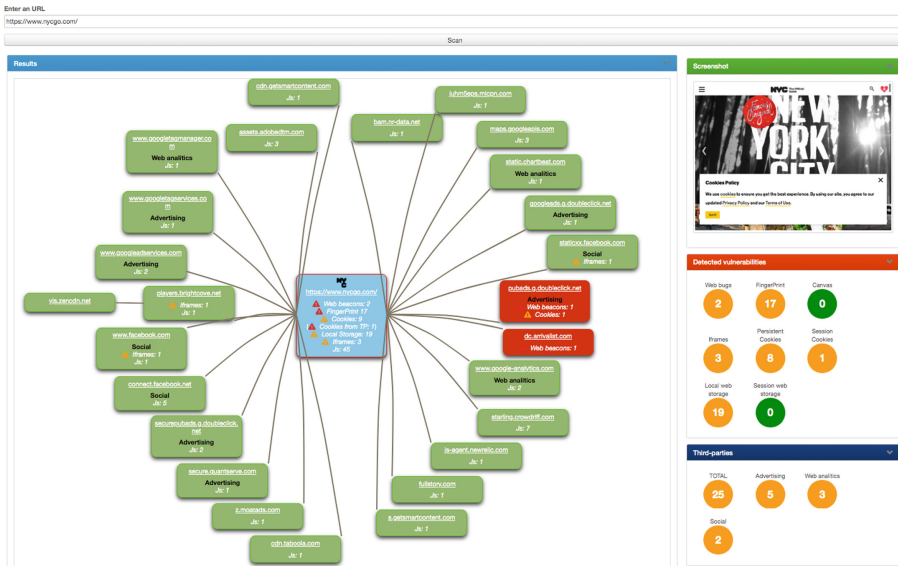


Fig. 2. Analysis of the website <https://www.nycgo.com>

Based on the analysis performed, we observe that the website delegates to a large number of third-parties. Among them, we find usual third-parties for advertising as Google, Doubleclick, and Chartbeat, social networks third-parties

<sup>15</sup> <https://developers.facebook.com/docs/javascript/quickstart/>.

<sup>16</sup> <https://developer.twitter.com/en/docs/twitter-for-websites/javascript-api/guides/set-up-twitter-for-websites.html>.

<sup>17</sup> <https://developers.pinterest.com/docs/sdks/js/>.

<sup>18</sup> <https://www.nycgo.com>.

and third-parties that perform web analytics. It should also be noted that the website includes three iframes, two of which belong to Facebook and are possibly embedded pages that communicate and share information with the well-known social network.

On the other hand, we check that the analyzed website contains many elements that may threaten the privacy of users when surfing the net. We can see that they are 8 persistent cookies, one of which is a cookie from a third-party (it is marked in red in Fig. 3). This persistent third-party cookie indicates that this website actually tracks the user. In addition, the third-party that uses this cookie is an advertising third-party that also inserts a web beacon on the website to carry out the user tracking.

Persistent Cookies						
Name	Domain/path	Value	Expiration date	Secure	Http	
<span style="color: red;">▲</span> mbox	.nycgo.com //	session#800415da935a468b9ea00da6a5fa0cde#1574270873PC#800415da935a468b9ea00da6a5fa0cde.26_25#1637513813	21/11/2021 17:58	No	No	
<span style="color: blue;">●</span> _ga	.nycgo.com //	GA1.2.1472729786.1574269021	19/11/2021 17:58	No	No	
<span style="color: red;">▲</span> _mibtv	www.nycgo.com //	anon-1574269012086-5399010228_7551	18/11/2021 17:58	Yes	No	
<span style="color: red;">▲</span> B_au	.nycgo.com //	1.1.87421647.1574269008	18/02/2020 17:57	No	No	
<span style="color: red;">▲</span> _micpn	www.nycgo.com //	esp-1:1574269012086	24/11/2019 17:58	Yes	No	
<span style="color: blue;">●</span> _gid	.nycgo.com //	GA1.2.498334023.1574269021	21/11/2019 17:58	No	No	
<span style="color: blue;">●</span> test_cookie	.doubleclick.net //	CheckForPermission	20/11/2019 18:13	No	No	
<span style="color: blue;">●</span> _gat	.nycgo.com //	1	20/11/2019 17:59	No	No	

Session Cookies					
Name	Domain/path	Value	Secure	Http	
check	.nycgo.com //	true	No	No	

**Fig. 3.** Cookies from the website <https://www.nycgo.com> analysis

By observing the cookies obtained in Fig. 3, we verify that the majority of persistence cookies are long-term cookies and that the website uses the known tracking cookies *\_ga*, *\_gid*, and *\_gat*.

Based on the use of web beacons, social media iframes, long-term cookies and third-parties of web analytics, advertising and social networks, we can conclude that this website puts the user's privacy at risk.

## 6 Limitations

Some limitations of the current solution have been detected when carrying out the testing phase, namely, we can mention:

- Websites that require acceptance of a web certificate cannot be analyzed because the current version of PWS does not include a certificate repository. As a result, a handshake exception is thrown.

- The system cannot consult all the resources of the websites that use dynamic module load frameworks, such as RequireJs<sup>19</sup>. These frameworks do not include inline scripts. Instead of that, they load the script require.js that dynamically loads the other scripts.
- We have not implemented any mechanism to solve the problem of script code obfuscation. As a result, it may happen that no fingerprinting variables or calls to other scripts are detected.
- A Chrome browser driver has been used. Therefore, the analysis depends on this browser. Not all the browsers manage web pages in the same manner, so using another browser might give slightly different results.

## 7 Conclusions and Future Work

Tracking and privacy issues are a concern for Internet users. However, for many users is difficult to know whether when they are accessing a website there are risks for their privacy. To cope with this issue, several web scanners that check privacy issues have been developed. However, many of them do not analyze the main sources of privacy risks and the information that they offer might be overwhelming for end-users without technical expertise. To address this issue, we have designed and developed PrivacyWebScanner, which is a web scanner for privacy risk identification that presents the information in a simple and understandable way for end-users. PWS is able to detect the use of Web beacons, cookies, iframes, Javascript, and HTML5 storage for tracking purposes. We have also shown how this information is depicted in a simple and attractive way.

This solution is developed to be in the beta permanent state since there are issues that require not only more development but also more research as is the case of detecting the use of Javascript or canvas for tracking. Thus, our future work will focus on studying detection mechanisms for these issues and performing a users' evaluation.

**Acknowledgements.** This work has been sponsored by the Spanish Ministry of Economy and Competitiveness through the PERSEIDES (contract TIN2017-86885-R), Spanish Ministry of Science, Innovation and Universities, grant number RTI2018-095855-B-I00, and European Union's Horizon 2020 research and innovation program under grant agreement No. 786725 (OLYMPUS project).

## References

1. Alidoost Nia, M., Ebrahimi Atani, R., Ruiz-Martínez, A.: Privacy enhancement in anonymous network channels using multimodality injection. *Secur. Commun. Netw.* **8**(16), 2917–2932 (2015)
2. 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.* **69**, 497–520 (2018)

<sup>19</sup> <https://requirejs.org>.

3. Boerman, S.C., Kruijkemeier, S., Borgesius, F.J.Z.: Online behavioral advertising: a literature review and research agenda. *J. Adver.* **46**(3), 363–376 (2017)
4. Cozza, V., Tsiatsikas, Z., Conti, M., Kambourakis, G.: Why snoopy loves online services: an analysis of (lack of) privacy in online services, February 2017
5. Estrada-Jiménez, J., Parra-Arnau, J., Rodríguez-Hoyos, A., Forné, J.: Online advertising: analysis of privacy threats and protection approaches. *Comput. Commun.* **100**, 32–51 (2017)
6. Estrada-Jiménez, J., Parra-Arnau, J., Rodríguez-Hoyos, A., Forné, J.: On the regulation of personal data distribution in online advertising platforms. *Eng. Appl. Artif. Intell.* **82**, 13–29 (2019)
7. Maass, M., Wichmann, P., Pridöhl, H., Herrmann, D.: PrivacyScore: improving privacy and security via crowd-sourced benchmarks of websites. In: Schweighofer, E., Leitold, H., Mitrakas, A., Rannenber, K. (eds.) *APF 2017*. LNCS, vol. 10518, pp. 178–191. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-67280-9\\_10](https://doi.org/10.1007/978-3-319-67280-9_10)
8. Mazel, J., Garnier, R., Fukuda, K.: A comparison of web privacy protection techniques. *Comput. Commun.* **144**, 162–174 (2019)
9. Merzdovnik, G., et al.: Block me if you can: a large-scale study of tracker-blocking tools. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 319–333, April 2017
10. Palos-Sanchez, P., Saura, J.R., Martin-Velicia, F.: A study of the effects of programmatic advertising on users’ concerns about privacy overtime. *J. Bus. Res.* **96**, 61–72 (2019)
11. Ruiz-Martínez, A.: A survey on solutions and main free tools for privacy enhancing web communications. *J. Netw. Comput. Appl.* **35**(5), 1473–1492 (2012)
12. Starov, O., Nikiforakis, N.: Extended tracking powers: measuring the privacy diffusion enabled by browser extensions. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, pp. 1481–1490 (2017)
13. Zimmeck, S., Li, J.S., Kim, H., Bellovin, S.M., Jebara, T.: A privacy analysis of cross-device tracking. In: Proceedings of the 26th USENIX Conference on Security Symposium, SEC 2017, Vancouver, BC, Canada, pp. 1391–1408. USENIX Association (2017)