Web Tracking: Mechanisms, Implications, and Defenses

Tomasz Bujlow, *Member, IEEE,* Valentín Carela-Español, Josep Solé-Pareta, and Pere Barlet-Ros

arXiv:1507.07872v1 [cs.CY] 28 Jul 2015

Abstract—This articles surveys the existing literature on the methods currently used by web services to track the user online as well as their purposes, implications, and possible user's defenses. A significant majority of reviewed articles and web resources are from years 2012 – 2014. Privacy seems to be the Achilles' heel of today's web. Web services make continuous efforts to obtain as much information as they can about the things we search, the sites we visit, the people with who we contact, and the products we buy. Tracking is usually performed for commercial purposes. We present 5 main groups of methods used for user tracking, which are based on sessions, client storage, client cache, fingerprinting, or yet other approaches. A special focus is placed on mechanisms that use web caches, operational caches, and fingerprinting, as they are usually very rich in terms of using various creative methodologies. We also show how the users can be identified on the web and associated with their real names, e-mail addresses, phone numbers, or even street addresses. We show why tracking is being used and its possible implications for the users. For example, we describe recent cases of price discrimination, assessing financial credibility, determining insurance coverage, government surveillance, and identity theft. For each of the tracking methods, we present possible defenses. Some of them are specific to a particular tracking approach, while others are more universal (block more than one threat) and they are discussed separately. Apart from describing the methods and tools used for keeping the personal data away from being tracked, we also present several tools that were used for research purposes – their main goal is to discover how and by which entity the users are being tracked on their desktop computers or smartphones, provide this information to the users, and visualize it in an accessible and easy to follow way. Finally, we present the currently proposed future approaches to track the user and show that they can potentially pose significant threats to the users' privacy.

Index Terms—web tracking, tracking mechanisms, tracking implications, defenses against tracking, user identification, tracking discovery, future of tracking.

I. INTRODUCTION

It is widely known that service providers (e.g., YouTube, elPais), content providers (e.g., Google, Facebook, and Amazon), and other third parties (e.g., DoubleClick) collect large amounts of personal information from their users when browsing the web. The large scale collection and analysis of personal information constitutes the core business of most of these companies, which use it for commercial purposes. Our personal information can be used, for example, for precise targeting

The authors are with the Broadband Communications Research Group, Department of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, 08034, Spain.

E-mails: tomasz@bujlow.com (T. Bujlow), vcarela@ac.upc.edu (V. Carela-Español), pareta@ac.upc.edu (J. Solé-Pareta), pbarlet@ac.upc.edu (P. Barlet-Ros).

of ads [1], [2], price discrimination [3], [4], assessing our health and mental condition [5], [6], or assessing financial credibility [7]–[9]. Apart from that, the data can be accessed by government agencies and identity thieves. Some affiliate programs (e.g., pay-per-sale [10]) require tracking to follow the user from the website where the advertisement is placed to the website where the actual purchase is made [11].

Personal information in the web can be voluntarily given by the user (e.g., by filling web forms) or it can be collected indirectly without their knowledge through the analysis of the IP headers, HTTP requests, queries in search engines, or even by using JavaScript and Flash programs embedded in web pages. Among the collected data, we can find information of technical nature, such as the browser in use, the operating system, the IP address, or even details about the underlying hardware, but also much more sensitive information, such as the geographical location of the users, their preferences or even the history of the web pages they visit. Unfortunately, the collected data do not stop here. For example, webmail services are known for scanning and processing user's e-mails, even if they are received from a user who did not allow any kind of message inspection.

This article surveys the literature on the methods currently used to track the user online by web services as well as their purposes, implications, and possible user's defenses. A significant majority of reviewed articles and accessed web resources are from years 2012 – 2014. The selected material covers online blogs, original research papers, as well as small surveys about some particular tracking methods. The topic is treated by us holistically and systematized according to defined criteria and hierarchy, which aim at providing easy understanding and comparison of the approaches to track the users' online activity. A high-level non-technical overview of the most common tracking techniques is presented in Chapter 7 of [12].

Our survey can be useful for a broad range of readers. At first, it is going to increase the awareness of Internet users and make them able to both better protect their private data and to understand which data is collected from them and how the obtained information is being used. At second, it is, according to our best knowledge, the first work that surveys this topic holistically in a comprehensive way. Therefore, it can be used as a help for students and other researchers working on this topic. At third, we expect this survey to start a discussion about the privacy issues between the online providers, advertisers, Internet users, and regulatory agencies. The discussion should lead to developing some regulations (either external or self-regulations) and to throw out from the

market unfair businesses that use offensive tracking or use the collected data for malicious purposes. That would strengthen both the Internet users and those companies that comply with acceptable tracking practices. Otherwise, if no regulations are being developed, the current online advertisement model is going to collapse. Namely, when users become aware that they are tracked in probably offensive ways, they will presumably start to use ad blocking tools, destroying the web economy and causing severe loss to themselves (e.g., by missing a potentially interesting offer directed to them or the necessity to pay for reading a news portal due to insufficient income from ads), to the ad providers, and to the services that use ads as the main income source.

This survey is structured in three main parts. At first, we present the known tracking mechanisms (Sections II – VI), afterwards we show their purposes and implications (Section VIII), and at the end, we discuss the possible defenses (Section IX).

Sections II – VI present 19 main groups of methods used for user tracking. The timeline of the first reported occurrences of the particular tracking methods is shown in Figure 1. The technologies used to track the user by different tracking methods are shown in Table I, while their tracking scope is summarized in Table II. Section II describes the less intrusive techniques, which can track the user only during a single browsing session (e.g., session identifiers). Section III presents the most widely known generation of tracking techniques, which are based on a persistent storage on a user's computer; cookies fall into this category. They can be easily blocked by a wide range of available tools. However, as we show later, these mechanisms were also exploited to invent more invasive techniques, such as cookie leaks or syncing, and the aggregation of trackers into advertising networks. The explicit web-form authentication, with and without the support from cookies, is yet another technique widely used over the Internet. Nevertheless, it does not pose major privacy concerns, because it is accepted (or even chosen) by the user who must sign up to a service. There are, however, many intrusive techniques, which were designed and are intentionally used just to overcome the privacy protection mechanisms incorporated in the browsers. Such techniques include the use of plugin storages and standard HTML5 storages. Tracking methods exploiting the browser's cache are shown in Section IV. The newly emerged fingerprinting techniques, which are able to identify the user without relying on any client-based storage and regardless of the private browsing mode enabled are shown in Section V. The last group of tracking methods presented in Section VI include the ones that are less expected by the user and that raise numerous ethical concerns. The tracking mechanisms are described together with the methods, which can be used for discovering and mitigating the particular threats.

Section VII shows how the users can be identified on the web and associated with their real names, e-mail addresses, phone numbers, or even street addresses.

Section VIII shows why tracking is being used and what the possible implications of it are for the users, e.g., by describing previously observed examples of price discrimination,
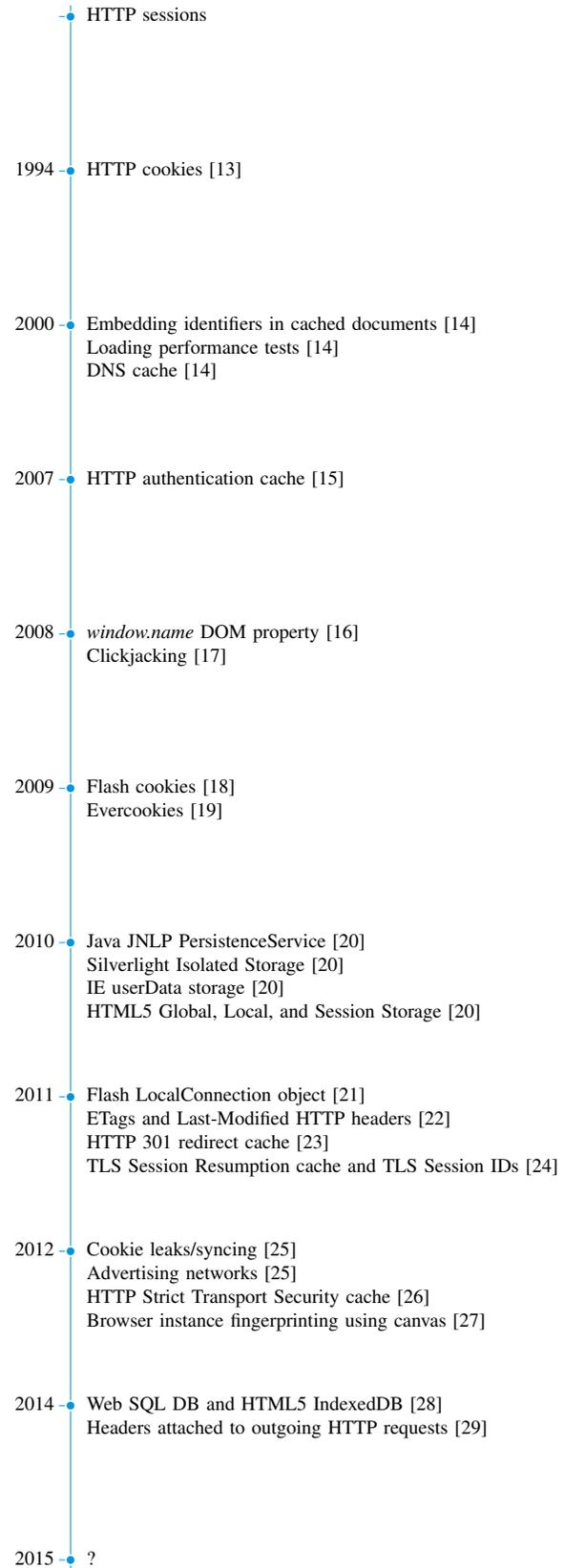
HTTP sessions

1994 — HTTP cookies [13]

2000 — Embedding identifiers in cached documents [14]
Loading performance tests [14]
DNS cache [14]

2007 — HTTP authentication cache [15]

2008 — *window.name* DOM property [16]
Clickjacking [17]

2009 — Flash cookies [18]
Evercookies [19]

2010 — Java JNLP PersistenceService [20]
Silverlight Isolated Storage [20]
IE userData storage [20]
HTML5 Global, Local, and Session Storage [20]

2011 — Flash LocalConnection object [21]
ETags and Last-Modified HTTP headers [22]
HTTP 301 redirect cache [23]
TLS Session Resumption cache and TLS Session IDs [24]

2012 — Cookie leaks/syncing [25]
Advertising networks [25]
HTTP Strict Transport Security cache [26]
Browser instance fingerprinting using canvas [27]

2014 — Web SQL DB and HTML5 IndexedDB [28]
Headers attached to outgoing HTTP requests [29]

2015 — ?

Figure 1. Tracking mechanisms timeline based on their first documented occurrence

Table I
TECHNOLOGIES USED TO TRACK THE USERS

Section	Tracking mechanisms	Technologies
II	**Session-only**	
A	*Session identifiers stored in hidden fields*	Web-server session
B	*Explicit web-form authentication*	Web-server session
C	*window.name DOM property*	HTML5, JavaScript
III	**Storage-based**	
A	*HTTP cookies*	HTTP headers, JavaScript
B	*Flash cookies and Java JNLP PersistenceService*	Flash / Java
C	*Flash LocalConnection object*	Flash
D	*Silverlight Isolated Storage*	Silverlight
E	*HTML5 Global, Local, and Session Storage*	HTML5, JavaScript
F	*Web SQL Database and HTML5 IndexedDB*	HTML5, JavaScript
G	*Internet Explorer userData storage*	JavaScript
IV	**Cache-based**	
A	***Web cache***	
1	Embedding identifiers in cached documents	HTML5, JavaScript
2	Loading performance tests	Server-side measurements, JavaScript
3	ETags and Last-Modified headers	HTTP headers
B	***DNS lookups***	JavaScript
C	***Operational caches***	
1	HTTP 301 redirect cache	HTTP headers
2	HTTP authentication cache	HTTP headers, JavaScript
3	HTTP Strict Transport Security cache	HTTP headers, JavaScript
4	TLS Session Resumption cache and TLS Session IDs	Web-server session
V	**Fingerprinting**	
A	*Network and location fingerprinting*	IP address, server-based geolocation techniques, HTTP headers, HTML5, JavaScript, Flash, Java
B	*Device fingerprinting*	IP address, TCP headers, HTTP headers, JavaScript, Flash
C	*Operating System instance fingerprinting*	JavaScript, Flash, Java, ActiveX
D	*Browser version fingerprinting*	HTML5, JavaScript, CSS
E	*Browser instance fingerprinting using canvas*	HTML5, JavaScript
F	*Browser instance fingerprinting using web browsing history*	Server-side measurements, HTTP headers, JavaScript
G	*Other browser instance fingerprinting methods*	HTTP headers, JavaScript, Flash
VI	**Other tracking mechanisms**	
A	*Headers attached to outgoing HTTP requests*	HTTP headers
B	*Using telephone metadata*	Smartphone malware
C	*Timing attacks*	HTML5, JavaScript, CSS
D	*Using unconscious collaboration of the user*	HTML5, JavaScript, CSS, Flash
E	*Clickjacking*	HTML5, JavaScript, CSS
F	*Evercookies (supercookies)*	Web-server session, HTTP headers, HTML5, JavaScript, Flash, Silverlight, Java

assessing financial credibility, determining insurance coverage, government surveillance, and identity theft. We also show that in-website tracking can be used for non-invasive purposes as web analytics and usability tests, which are made in order to improve the user browsing experience.

Section IX discusses the defense techniques and tools that can be used against more than one tracking method. The most popular are blocking the advertisement services, hiding the IP address by VPNs, the use of proxies or TOR, and using the private browsing mode. In addition to these, there exist more sophisticated methods, e.g., modification of the data sent over the network and execution blocking. The user can also choose to use privacy-focused search engines, anonymous mailing services and mail aliases, and can decide to frequently clear the browser cache and history. Apart from describing the methods and tools used for keeping the personal data away from being tracked, we also present here several tools that can be used for research purposes – their main goal is to discover how and by what the user is being tracked on his desktop computer or smartphone, provide the information to the user, and visualize it in an accessible and easy to follow way.

Section X presents the currently proposed future approaches to track the user and show that they can potentially increase the threats to the users' privacy. An example can be the solution being prepared by the Future of the Cookie Working Group from the Interactive Advertising Bureau (IAB), which relies on device-inferred, client generated, network-inserted, server-issued, and cloud-synchronized identifiers. Other examples of identifiers designed to have broad coverage are Google AdID and Microsoft Device Identifier.

Section XI concludes the survey and summarizes the main findings.

II. SESSION-ONLY TRACKING MECHANISMS

In this section, we start to introduce several groups of tracking techniques. The historically first known tracking mechanisms relied on sessions. As we see in Table I and Table II, the methods are quite simple and do not pose significant threats to the users.

A. Session identifiers stored in hidden fields

Before cookies became available in 1994 [13], the only way to track a user except using the IP address was to pass his identifier to another website in the URL (GET method) or as a value in a hidden field of a web form (POST method) [30]. The identifier can be any string that is able to uniquely track the user during a single browsing session. It can be composed, for example, from a timestamp and a random number. The

Table II
KNOWN TRACKING SCOPE

Section	Tracking mechanisms	Scope
II	**Session-only**	
A	*Session identifiers stored in hidden fields*	Session id
B	*Explicit web-form authentication*	User id
C	*window.name DOM property*	Session id
III	**Storage-based**	
A	*HTTP cookies*	Browser instance id
B	*Flash cookies and Java JNLP PersistenceService*	Operating system instance id
C	*Flash LocalConnection object*	Operating system instance id
D	*Silverlight Isolated Storage*	Browser instance id
E	*HTML5 Global, Local, and Session Storage*	Browser instance id
F	*Web SQL Database and HTML5 IndexedDB*	Browser instance id
G	*Internet Explorer userData storage*	Browser instance id
IV	**Cache-based**	
A	*Web cache*	
1	Embedding identifiers in cached documents	Browser instance id, browsing history
2	Loading performance tests	Browsing history
3	ETags and Last-Modified headers	Browser instance id
B	*DNS lookups*	Browsing history
C	*Operational caches*	
1	HTTP 301 redirect cache	Browser instance id
2	HTTP authentication cache	Browser instance id
3	HTTP Strict Transport Security cache	Browser instance id
4	TLS Session Resumption cache and TLS Session IDs	Browser instance id
V	**Fingerprinting**	
A	*Network and location fingerprinting*	IP address, user's country, city, and neighborhood
B	*Device fingerprinting*	Device id, IP address (entire or a part), operating system, screen resolution, timezone, list of system fonts, web browser, information about hardware (mouse, keyboard, accelerometer, multitouch capability, microphone, camera), TCP timestamps
C	*Operating System instance fingerprinting*	Operating system instance id, operating system version and architecture, system language, user-specific language, local timezone, local date and time, list of system fonts, color depth, screen dimensions, audio capabilities, access to the user's camera, microphone, and hard disk, printing support, hard disk identifiers, TCP/IP parameters, computer name, Internet Explorer product id, Windows Digital Product Id, installed system drivers, operating system instance id stored by a Java privileged applet
D	*Browser version fingerprinting*	Detailed browser version
E	*Browser instance fingerprinting using canvas*	Browser instance id
F	*Browser instance fingerprinting using web browsing history*	Browser instance id, browsing history
G	*Other browser instance fingerprinting methods*	Browser instance id, detailed browser version, supported formats of images and media files, preferred and accepted languages, list of browser plugins, browser user's language, browser dimensions, Flash version, screen resolution, color depth, timezone, system fonts, IP address, accepted HTTP headers, cookies enabled, supercookies limitations
VI	**Other tracking mechanisms**	
A	*Headers attached to outgoing HTTP requests*	Customer's id
B	*Using telephone metadata*	Health (including mental) condition, religious believes, and addictions of a specific real person
C	*Timing attacks*	Boolean values dependent on the look of the website (e.g., if the user is logged in to a particular service), stealing any graphics embedded or rendered on the screen
D	*Using unconscious collaboration of the user*	Browsing history, browser instance id, user's location
E	*Clickjacking*	User's email and other private data, Paypal credentials, spying on a user by a webcam
F	*Evercookies (supercookies)*	Operating system instance id, browser instance id

values submitted by a web form using POST, contrary to the values appended to the URL using GET, are neither persisted on the disk nor preserved in the browser history [11]. Although this technique still works, its usefulness is limited to a single browsing session. However, the identifier can be passed to third parties [31] when no client-side programing language (e.g., JavaScript) is involved into the submission of the web form.

B. Explicit web-form authentication

Another possibility to identify the user is to ask or require him to register on the website. Then, the resources provided by the website (e.g., webmail) are available only to the user who logs in. That make the user identification very easy and accurate. This method is independent of the web browser, operating system, or computer used, as well as of the place where the user is connected to the Internet. There are, however, two important issues connected to this method. At first, the user must authenticate every time he uses the website, which can be seen as burdensome to him. At second, the authentication is valid only within the current session after the user logs in. At third, it is impossible to track the user transparently – the user is always aware that he is logged in and that everything he does can be recorded.

C. window.name *Document Object Model (DOM) property*

The W3C Document Object Model (DOM) [32] is a cross-platform interface for accessing and interacting with the content, structure, and style of web documents (i.e., HTML, XHTML, and XML). It is language-independent and it organizes all the objects in a tree structure. Each object (e.g., a *window*) has a number of properties (e.g., a *name*).

The *window.name* DOM property [33] can hold up to 2 MB of data as a single string value. By using a JSON *stringifier* [34], it is possible to pack multiple variables into one string, which can be subsequently stored on the user's computer. The *window.name* property is resistant to page reloads and it is accessible from other domains as well, which gives the third-party content an opportunity to exchange the information with the first-party or with another third-party content. An implementation of cookieless session variables using this DOM property is shown in [16].

III. STORAGE-BASED TRACKING MECHANISMS

The next group of tracking mechanisms depend on explicit storing data on the users' computers. These methods seem to be the most commonly used ones. Generally, they are much more advanced than the session-based methods (see Table I) and their abilities are also higher, as they are able to recognize the particular instance of a browser or an operating system, depending on the mechanism (see Table II). Each of these mechanisms presented the biggest threat to users' privacy directly after it was invented. Over time, browsers started to implement clearing of these storages on the user's request.

A. HTTP cookies

The most currently known method to identify a user is by using cookies – small pieces of data (each limited to 4 KB) placed in a browser storage by the web server [35]. When a user visits a website for the first time, a cookie file with a unique user identifier (could be randomly generated) is stored on the user's computer. Then, the website can retrieve this identifier each time the user visits it, unless the user delete the cookie from his computer. This method of identification is very fast, does not require any kind of interaction from the user, and is completely transparent to him as the browser does not show any notifications when the cookies are set or read. However, its accuracy is limited to the situations when the user allows using cookies, does not clean the cookies cache in the browser, and always uses the same web browser to visit the particular website. There are two basic kinds of cookies: session cookies and persistent cookies. Session cookies expire when the user closes the web browser, while persistent cookies expire after a specified amount of time [36].

Cookies are set on the user's computer in 2 ways: either by a JavaScript using an API call, or by HTTP responses containing the *Set-Cookie* header. Cookies can be read by the services in 2 ways as well. At first, they are automatically attached to the HTTP requests made to the domain to which the cookies belong using *Cookie* headers. At second, they can be explicitly requested by a JavaScript API and then sent to the server in any way [25].

Usually, tracking and non-tracking cookies can be distinguished based on their expiration time and length of the value field. Experiments described in [37] show that more than 90 % of tracking cookies have their lifetime greater than 1 day, while such lifetimes concern only 20 % of non-tracking cookies. As the value contained by a cookie must be long enough to be able to distinguish each user, 80 % of tracking cookies have their values longer than 35 characters, while that concerns only 20 % of non-tracking cookies [37]. In order to avoid this detection, websites can try to split the user's identifier in multiple cookies, but such behavior can be detected by taking into account the sum of the value lengths from all the cookies from the same third-party website [37]. The authors of [37] showed that based only on these attributes, their tracking recognition tool is characterized by precision of 99.4 % and recall of 100.0 %.

Some browsers or browser add-ons (for the list of defense techniques look at Section IX) try to increase the users' privacy level by blocking third parties from setting and reading cookies (usually only from setting, however, Firefox also forbids reading). This limitation can be easily bypassed by redirecting the user by a JavaScript to the third-party website that will set or read the cookies; from this website, the user is redirected back to the one he originally visited. That way, the third-party content appears as coming from first parties. For this purpose, instead of redirecting the user, the website can use popup windows in which the third-party content is shown as coming from the first party [38]. As only Firefox is known from forbidding reading cookies by third parties by default, in other cases, it is enough just to set the cookies by a popup or

a redirection in the first-party context, while they can be read directly by the third-party tracker [25].

Research from 2011 shows that only around 30 % of users delete their cookies (both first party and third party) within a month from their acquisition [39].

HTTP cookies can be used as a tracking mechanism alone or combined with other techniques.

1) Explicit web-form authentication and cookies: This method combines the web-form authentication (Section II-B) and cookies in order to make the authentication more friendly to the user. The user registers on the website and logs in to be able to use the service as in the previous method. The user credentials are, however, stored in cookies on the user's computer. Thanks to that, the user does not need to log in again to the service as long as he allows using cookies, does not clean the cookies cache in the browser, and uses the same web browser. The authentication performed by the services owned by Google is a good example of this method.

2) Cookie leaks / syncing: During the experiment described in [25], the authors observed a large number of cookie leaks. It means that cookies from one domain were passed to another domain, for example, as parameters of a request. This method is used, for example, by Microsoft to track the users on its domains (e.g., *bing.com*, *microsoft.com*, *msn.com*, *live.com*, and *xbox.com*) and other websites which include resources from these domains. The cookies between these domains are exchanged regardless if the user visits both sites during one browser session or completely separately [25], [40]. Cookie syncing, used for example by Google [41] to better facilitate targeting and real-time bidding, enables third parties to exchange information about a user. This is possible by making requests containing the user identifier from one domain to another.

3) Advertising networks: Some websites embed a limited number of trackers, which act as aggregators for tracking services. For example, a very frequent tracker *admeld.com* is known from making requests to other trackers, e.g., *turn.com* and *invitemedia.com*. The requests contain the visited website and the identifier assigned to that user by the aggregator [25].

B. Flash cookies and Java JNLP PersistenceService

Local Shared Objects (LOSes) are used by Adobe Flash to store data on users' computers [42]. They can store 100 KB of data by default, which makes their use for tracking more proper than using HTTP cookies, which can store only 4 KB of data. Moreover, it is harder for the user to erase them than HTTP cookies, and they are accessible from all the browsers installed in the system, as all the instances of Adobe Flash plugins share the same storage directory. That makes them able to track the users across different browsers [43]. Apart from that, Flash cookies do not expire by default. They are stored in a disk directory as *.sol* files.

Apart of the Local Shared Objects, Flash also supports Remote Shared Objects (RSOs). The cooperating Flash objects from the same domain can access the contents of locally persisted objects stored in a disk directory as *.sor* files [38].

Since version 10.3, Adobe Flash supports *ClearSiteData* API, which removes the shared Flash objects. This API call is currently invoked by all main browsers while clearing the usual HTTP cookies. Additionally, the new Flash manager allows to block storing any Flash objects on the local computer [44].

Another option to store data is to use Java JNLP Persistence-Service [45], which facilitates storing data locally on the client system, also for applications that are running in the untrusted execution environment.

C. Flash LocalConnection object

The LocalConnection object supported by Flash can be used to communicate between different SWF files running on the same computer at the same time. Therefore, it also can be used to communicate and exchange values between Flash instances running in the normal and private browsing windows [21]. This approach can be also combined with Flash cookies to pass the values from the cookies accessible in a normal browsing window to a Flash instance running in a private browsing window.

D. Silverlight Isolated Storage

This storage allows to save 100 KB of information per site in the user's profile (as Flash cookies). However, this storage is disabled in private mode. The last version of Silverlight was announced by Microsoft to expire in 2021 [44]. This storage can be cleaned only manually (by deleting files from a hidden folder in the file system or by using storage options in the Silverlight application). The Silverlight application also offers the possibility to disable the Isolated Storage [44].

E. HTML5 Global, Local, and Session Storage

HTML5 Global Storage [46] introduced by an early HTML5 draft offered some possibilities to store data by websites, but it was not implemented by any main browser due to the violation of the same-origin policy [44].

HTML5 Local Storage [47], which obeys the same-origin policy, provides yet another possibility to track the user. Placing the objects (key-value pairs) in the storage does not require any plugin. The objects are stored permanently (there is no automatic expiration) – they persist until they are removed by the website or by the user. Additionally, an object can be as big as 5 MB, which gives a significant advantage over both HTTP and Flash cookies [48]. The content of the Local Storage can be shared between different browser windows [44]. The Local Storage is automatically emptied at the time when the cookies are cleared. The research performed in [42] showed that 17 among the top QuantCast.com ranking websites in 2011 used HTML Local Storage to store 60 key/value pairs. In several cases, the values stored in HTML Local Storage matched the HTTP cookies: *twitter.com*, *foxnews.com*, *nytimes.com*, and *cnn.com*. The match was usually seen with third-party websites, as *meebo.com* (now being a part of Google), *kissanalytics.com*, or *polldaddy.com*.

HTML5 Session Storage [49] is very similar to the Local Storage: it preserves the same-origin policy and the stored objects can be as big as 5 MB. However, the objects are available only to the current browser window and are deleted when the window is closed [44].

F. Web SQL Database and HTML5 IndexedDB

Web SQL Database [50] uses SQLite instead of the local file system to store data on the client side. Although it was implemented by all main browsers, its development was stopped in favor of HTML5 IndexedDB [51]. HTML5 database storages operate under the same conditions as the local storage, thus, the privacy impact is the same.

The first report of using HTML5 IndexedDB to rebuild Flash and HTTP cookies is from 2014 [28]. The authors found that a script from *weibo.com* stores an item in this database, which directly matches the content of the Flash cookie *coosimg.sinajs.cn/stonecc_suppercookie.sol* and the corresponding HTTP cookies from *weibo.com* and *sina.com.cn*.

G. Internet Explorer userData storage

Internet Explorer userData storage [52] is a proprietary technique introduced in Internet Explorer 5.5 and declared obsolete in Internet Explorer 7, however, it still works in Internet Explorer 11. It allows to store up to 64 KB of data in the XML format [44].

IV. CACHE-BASED TRACKING MECHANISMS

Another group of tracking methods also use client-based storage. But in contrast to the previous group that used storages explicitly designed for preserving data, this group exploits possibilities to identify browser instances and determine the previously visited websites (see Table II) by the use of various caches.

A. Web cache

Before 2010, the browsing history could be easily obtained using DOM API in an automatic way. The attacker could set the color used by the browser to display visited and non-visited links according to his wish. Then, JavaScript could be used to generate links pointing to a specific destination (e.g., *http://www.cnn.com*), which the attacker would like to check in the browsing history. The color of the generated links could be read by JavaScript and compared with the color previously set to be used by both the visited and non-visited links. Using this technique, an attacker could test 10 000 to 30 000 links for presence in the browser history [53], [54]. This threat, however, was fixed in all popular browsers after a solution was proposed in [55]. The fix relies on returning the style of all hyperlinks by API calls as non-visited regardless if the links are visited or not.

Despite fixing this threat, web browsing history still can be obtained by a number of ways. As it is shown in the following paragraphs, the most known automatic techniques use browser caching. When a browser downloads an object (e.g., an image), it is usually stored in the browser cache for faster display when the user visits the website again. Therefore, when a user browses a website, the website can easily determine if this user visited it before (so that this object is pulled from the cache) or not (so that the object is downloaded from the server). When an advertiser has his objects on many websites, he can easily

compare them with the cached copies and determine which pages were visited [42], [56]. Exploiting the web cache can be done in several ways.

1) Embedding identifiers in cached documents: The first possibility to identify a user is to make the user request an HTML file, which contains the embedded identifier. The identifier can be stored in an invisible *div* and subsequently read from the browser cache using the *div id* property [38]. As the cached files can be included by any website, the values stored in them can be used across multiple services [14].

2) Loading performance tests: Websites can use JavaScript to detect the time of loading any object (e.g., an image) from any URL. The loading time can be measured by JavaScript and reported to the host service, which can evaluate if the object is present in the browser cache or not. That way, by testing on an object that is always accessed when a user uses the particular website (e.g., company logo), the script can assess if the website was previously accessed [14]. In case it is not possible (e.g., JavaScript is blocked), the offensive service can try to use another technique: load at first a file from itself, then a file from the website whose presence in the browser cache is tested, and then again a file from itself. The time used to load the file from the tested website can be calculated by subtracting the times between loading the files from the offensive service itself [14].

3) ETags and Last-Modified HTTP headers: To enable user identification using the web cache, entity tags (ETags) [57] or *Last-Modified* HTTP headers [57] can be used. The HTTP header provided with the first download of an object contains the following fields: *Last-Modified*, *ETag*, *Cache-Control*, and *Expires*. The *ETag* field can be as long as 81864 bits and, therefore, it is sufficient to store the user identifier, which became used by the tracking companies. The *Last-Modified* header [57] is shown to accept any random string, not only a valid date [22], which also started to be used for tracking purposes. When a user visits the website again, the browser sends the *If-Modified-Since* and *If-None-Match* headers as a part of the HTTP request (these headers contain values stored in the *Last-Modified* and *ETag* fields from the previously cached document, respectively). The web server then checks if the cached copy is not outdated [56]. If the cached copy is still valid, the server returns a very short response with the *HTTP 304 Not Modified* status. In case if it is outdated, a new document is returned.

The first use of ETags to track the users was observed in 2011, when hulu.com employed KISSmetrics service to rebuild HTTP and HTML5 cookies. To avoid tracking by ETags, the user must clear the browser cache before each visit on the website that compares the particular ETag. Tracking is also possible during a single private browsing session, as the cache is kept until the last browser window is closed [42]. The method using the *Last-Modified* headers does not have any problem with bypassing web proxies, which can be problematic for ETags [57].

B. DNS cache

Yet another automatic method use the possibility of JavaScript to indirectly cause a DNS lookup and measure its

time. In case the website was previously accessed, the corresponding entry exists in the DNS cache, which significantly reduces the lookup time [14].

C. Operational caches

Operational caches are components used to store information related to operations made by the web browsers, rather than to store the copies of the downloaded elements. Such stored information includes permanent redirects, authentication credentials, or a list of domains that must be used together with the HTTP Strict Transport Security (HSTS).

1) HTTP 301 redirect cache: The *HTTP 301 redirect* mechanism was designed to tell the browser that the particular queried resource is permanently available at another URL. The browser caches the redirect and uses it instead of the original URL during the following attempts to visit the website. That could be used if an existing page was moved somewhere else or if the initial link was a shortened URL, generated to conserve space or provide better visibility. However, this mechanism can be used to track the user by third parties. The first time the user visits a service (which can be observed by the service by the format of the URL used to access it), the service generates a *HTTP 301 redirect* to the same URL as initially used, but with the attached identifier for the particular user. As this request is cached, the browser is going to use this URL (together with the attached user identifier) during the next attempts to access the service [23]. The redirection can be made in an iframe, so it is transparent to the users [44].

2) HTTP authentication cache: There are two commonly used HTTP authentication mechanisms: *Basic access authentication* and *Digest access authentication* [11]. When a user enters his credentials on a website using one of these mechanisms, the website stores the credentials temporarily, so they can be automatically submitted in the HTTP *authorization header* to the server upon next requests, so the session can be identified. Jeremiah Grossman on his website [15] presents several methods that use JavaScript to force the browser to authenticate to the web server in a way that no popup with an authentication request is presented to the user. When the browser is authenticated and it receives *401 (Unauthorized)* status code, it will cache the credentials and send them with each subsequent HTTP request.

3) HTTP Strict Transport Security cache: HTTP Strict Transport Security (HSTS) [58] is yet another mechanism which can be used to create a cookie-like storage. Mikhail Davidov on his blog [26] described a possible implementation of such a storage. The current standard of HSTS specifies that the server can ask over a secure connection to make all the future connections to its domain using HTTPS instead of HTTP. The server indicates when the HSTS mode will expire and whether the sub-domains should be covered by the policy. Wildcard SSL certificates can be issued for a server to cover multiple subdomains within the same top-level domain. When a user visits websites in the HSTS mode, for each of the subdomains, a single entry is made in the HSTS database on the user's computer. The website is able to encode the user's identifier character-by-character in different subdomain names,

for example, an identifier "*DBAS*" can be encoded into *1-D.addomain.com*, *2-B.addomain.com*, *3-A.addomain.com*, and *4-S.addomain.com*. When some resources are requested from these domains (e.g., a pixel image) using HTTPS while HSTS is on for the domain, these subdomains are stored into the HSTS database. Then, during the later user's visit on the website, a JavaScript code can use a brute-force method to query iteratively by HTTP the resources from all the possible combinations of the subdomains, e.g., *1-A.addomain.com*, *1-B.addomain.com*, *1-C.addomain.com*, and *1-D.addomain.com*; in this case, during the first 3 tries, the URL will not get translated into HTTPS, as no entry in the HSTS database exist. However, when asking for the resource from *1-D.addomain.com*, HTTP is going to be rewritten by HTTPS, which indicates that the first character of the user's identifier is *D*. Mikhail also claims that Google Chrome clears the HSTS database at the same time as the cookies are deleted, while Firefox 4 stores the HSTS database in a separate place, which is not easy to clear.

4) TLS Session Resumption cache and TLS Session IDs: TLS session resumption cache is used to store TLS/SSL session IDs [59], which are sent by the server to the client during the *hello* message, in order to use them when connecting to the host the next time. That avoids full TLS handshakes, which reduces latencies and the CPU usage [60]. It is shown in [24] that both the TLS session resumption cache and TLS/SSL session IDs can be used for third-party tracking and, thus, their implementation was fixed in Tor Browser.

V. FINGERPRINTING

Fingerprinting is a group of methods using a broad range of technologies (see Table I), which is able to discover the widest spectrum of properties from all the methods (see Table II).

A fingerprint (a unique identifier of a device, operating system, browser version or instance) can consist of one or more values which can be read by the web service when the user browse its website. That way, the user can be tracked on multiple different websites belonging to different entities, which is not directly possible to be done using cookies. No cookies are generated and the user does not need to log in, so the tracking is transparent to the user and it works regardless if the browser accepts cookies or not. Therefore, an average user does not have any means to know if he is tracked and the user does not know how to prevent that (although, it is possible in some extent by disabling the support of JavaScript, Java, and Flash, however, it does not prevent passive fingerprinting). A study from 2010 made by Yahoo [61] reveals that only around 1 % of users disabled the support of JavaScript in their browsers. Some of the properties used for fingerprinting can be discovered passively from the network traffic (IP address, operating system, user agent, language, HTTP accept headers), while others require (or can be also obtained) by active approaches (operating system, CPU type, user agent, timezone, clock skew, display settings, installed fonts, installed plugins, enabled plugins, supported MIME types, cookies enabled, third-party cookies enabled) [62]. There are numerous websites which show us extensive amount of information that

can be automatically collected from the user's computer [63]–[66]. According to [67], 40 out of top 10 000 Alexa websites, including *skype.com*, use fingerprinting scripts from BlueCava [68], Iovation [69], or ThreatMetrix [70]. The most popular website categories that use fingerprinting were found to be porn (15 %) and dating (12.5 %). The next sections are going to present different categories of fingerprinting methods, define their scope, coverage, and potential defenses.

A. Network and location fingerprinting

One of the easiest features to be determined based on the headers of incoming HTTP requests are the global network address and the IP-based geographical location of the user. By using network tools, the service is able to identify the name of the domain and the user's Internet Service Provider. The presence of a proxy server can be detected by asking about that explicitly in an HTTP request [71]. Proxies set in the browser can be bypassed by Flash applets to discover the real IP address of the client [67], which is done by Iovation [69] and ThreatMetrix [70], two of the most popular fingerprinting companies. Therefore, anonymity-providing applications, e.g., Tor Browser, tend to block Flash. Several important features can be returned by JavaScript functions, as the internal IP address [72] and GPS coordinates [73]. The connection performance (e.g., the download and upload speeds, round-trip time, and jitter) can be calculated either by JavaScript [74] or by Flash [75]. Finally, Java applets can be used to detect the presence of a firewall [76].

Traditionally, the user's location can be inferred from the IP address. There are many free databases (e.g., IP To Country Database [77]), which facilitate discovering the country and city where the IP address should be placed. A comprehensive survey on the possible geolocation mechanisms that can be used based on the IP address is shown in [78]. In [79], it was shown that the IP-based location is not accurate for mobile networks. The authors claimed that over 90 % of mobile devices in Seattle have IP addresses located more than 600 miles from Seattle. Apart from that, VPN or Tor can be used to hide the real global IP address, as it is masked by the IP address of the VPN gateway or a Tor exit relay [80]. Therefore, geolocation is a research topic and there are many existing papers tackling with this problem by different methods. For example, a geo-inference method, based on the history and cache of visited websites, is shown in [81].

B. Device fingerprinting

A novel cross-browser device fingerprinting method based only on JavaScript and using passively collected information is shown in [82]. The user identifier is derived from the first two octets of the IP address, version of the operating system, screen resolution, timezone, and the list of the standard fonts available in the system, which are universal for all the web browsers. Apart from on the user identifier, the authors collected for research purposes some other variables, as the *User-Agent* string and the real IP address. The list of system universal fonts was obtained solely by JavaScript. Thanks to such a design, the authors achieved browser independence (the user identifier

does not depend on any browser-specific feature), plugin independence, and cross-domain tracking (given that the user has its IP address assigned from the same pool). Furthermore, using the visual inspections, they were able to recognize returning users who used multiple browsers, changed their screen resolutions, timezones, or the IP addresses.

The Turkish Ministry of National Education [83] was shown in [84] to be the only known *.gov* website, which uses extensive device fingerprinting. A Flash object embedded on the website extracts and sends back to the server the detailed information about the mouse, keyboard, accelerometer, multi-touch capability, microphone, camera, and system fonts.

A remote device fingerprinting method was described in [85]. This method is based on the clock skew on fingerprinted machines and it uses TCP timestamps. Therefore, it is able to recognize the device regardless of the distance from the device, location of the device (e.g., cable or wireless network), or the presence or absence of NATs, which is an advantage over methods based on MAC or IP/ICMP. However, although the clock skews are constant during time, they are not enough unique to be able to identify a device with a high probability, as many different machines can have the same skew, especially, when they use NTP or other time synchronization methods. It is why this method can be used to deny the existence of the particular machine in a network at the certain time, but it cannot be used to confirm that the particular machine at this time was present in the network. The authors tried to de-anonymize a previously anonymized Caida trace with 11 862 IP addresses. The identification accuracy based on the clock skew was 57.66 % or 87.65 % depending on the maximal TCP timestamp option clock skew differences given as a parameter.

C. Operating System instance fingerprinting

The version and architecture (32/64 bit) of the operating system can be identified both by JavaScript and Flash. JavaScript also facilitates recognizing the system language and system user-specific language, local timezone [86], and the local date and time up to 1 millisecond [87]. The list of installed fonts can be detected both by JavaScript [88] and Flash, which is performed by scripts and Flash objects of three popular large fingerprinting companies [67]: BlueCava [68], Iovation [69], and ThreatMetrix [70]. The color depth and screen dimensions can be detected both by JavaScript [89] and Flash [90], [91]. Flash can also detect whether the system has audio capabilities, whether the access to the user's camera and microphone has been prohibited or allowed, whether the system does or does not support printing, and whether read access to the user's hard disk has been prohibited or allowed [90], [91]. Standard Java applets can open, read, and save files on the client, thus, they can be used to store tracking information. Privileged Java applets can run outside the security sandbox and have extensive capabilities to access the client, obtaining all the information available to a standard system application [76]. Another possibility used to read the values from the operating system is to use ActiveX controls. Research shown in [67] demonstrated that BlueCava [68] and Iovation [69] use them to read the hard disk identifier,

TCP/IP parameters, computer name, Internet Explorer product identifier, Windows installation date, Windows Digital Product Id, and the installed system drivers.

D. Browser version fingerprinting

The detection of the web browser by the self-reported *User-Agent* field in HTTP headers is considered to not be reliable, as various browsers and other applications are able to obfuscate this string or claim to be a particular browser [92]–[94].

In [92], the authors introduced new browser version fingerprinting methods, called CSS and HTML5 fingerprinting. These methods are able to recognize the family and version of the web browser, however, they cannot be used to identify the particular browser instance. CSS fingerprinting uses the differences between the implemented CSS properties, CSS selectors, and CSS filters to distinguish one browser version from another. The overview of CSS supported properties by different web browsers can be found in [95]. The support of the particular property can be confirmed by JavaScript in two ways: by asking if the property exists or trying to set the particular property and then trying to read its value [92]. HTML5 fingerprinting relies on the differences between how the web browsers implement the standard. In [92], 242 new tags, attributes, and features of HTML suitable for fingerprinting were detected. New HTML tags introduced in HTML5 amounted for 30 of them, while the rest consisted of new features and attributes for already existing tags.

The JavaScript engine was used for browser identification in [93]. As the methods relying on CSS or HTML5, it is able to detect the family and version of the browser, but not the particular instance. ECMAScript *test262* is a test suite built to check how the JavaScript implementations follow the ECMA-262 standard of the JavaScript language. It consists of 11552 tests and it executes in around 10 minutes on a desktop PC or 45-60 minutes on a mobile device. The particular tests are passed or failed depending on the browser. The authors propose two methods for the identification of various web browsers: finding a minimal subset of *test262* that is able to distinguish one web browser from another, or building a decision tree based on the results of the tests. The second approach allows to include more tests in the comparison while reducing the script execution time.

The performance of the JavaScript engine was used in [94] to detect the web browser version, the operating system, and even the architecture of the computer. The authors constructed a suite of 39 JavaScript tests. Each test was separately executed 5 times and its execution time was recorded. A delay of 800 ms was added between every launching of these script to minimize the effect of any cleanup processes made by the browser. The minimal execution times were taken into account while creating the browser profiles, so that the differences in the use of the system resources by other processes could not impact the results.

E. Browser instance fingerprinting using canvas

HTML5 introduced an area of the screen, which can be used to draw text or images programmatically. Currently, there are two defined graphics contexts: *2d* and *webgl*. Using the *2d* context, the browser can use *fillRect*, *lineTo*, and *arc* methods to draw basic figures or *fillText* to draw the text. The text can be formatted using styling methods similar to the ones found in CSS [27].

A special kind of fingerprinting is *canvas fingerprinting*, which relies on using the browser canvas API to draw invisible graphics. There are several methods to retrieve the produced graphics. The *getImageData* of the *2d* context returns an object containing the RGBA values (as integers) for every pixel of the requested rectangular region of the canvas. The second method, *ToDataURL*, returns a Base64 encrypted string encoding the entire content of the canvas in the requested format (e.g., PNG) [27]. As every browser instance draws the graphic differently, the Base64 encrypted string differs as well from one browser to another, which results in the possibility of extracting a unique hash from the string and using it as a browser identifier. The rendered image depends on the operating system, installed fonts, graphics card and its drivers, and the browser itself, due to font rasterization, anti-aliasing, smoothing, API implementations, and the physical display [27]. According to [28], canvas fingerprinting is the most commonly used fingerprinting method, present on more than 5.5 % of top 100 000 Alexa websites. The authors found out that the examined JavaScripts used for fingerprinting the following methods: *fillText* and *strokeText* to write the text, and *ToDataURL* to read the image data. There are also some other methods which could be used for fingerprinting (but not directly): *MozFetchAsStream*, *getImageData*, and *ExtractData*, however, they were not investigated in the paper. A JavaScript was considered as fingerprinting in case if both methods (writing the image and reading the image data) came from the same script (URL), if the image contained more than one color and was bigger than 16x16 pixels, and if the image was not requested in a lossy compression format, which could remove the details. In case more images were drawn by the script, the dimension criteria concerned the sum of the dimensions of all the images. The rendered image can be accessed only by the same origin as drew it, otherwise, a *SecurityError* exception is returned.

fingerprintJS [96], a free JavaScript library, also supports canvas fingerprinting and the returned value is the same in the normal and private browsing modes [97].

The fingerprinting scripts from *addthis.com* were responsible for the majority (95 %) of the fingerprinting attempts on the analyzed websites [28]. This script used the perfect pangram "*Cwm fjordbank glyphs vext quiz*" as the drawn text string. The text was printed two times, once, using a particular font, second time, using the default browser font (the script requests an non-existing font). Additionally, the script checks for drawing Unicode support (it tries to print the *U+1F603* smiling face character). It also checks for canvas *globalCompositeOperation* support (it sets or returns how a new image are drawn onto an existing one) and it draws two rectangles to check if a specific point is in the path by the *isPointInPath* method.

The same font (e.g., Arial) looks differently at different operating systems. To make the website use the same font

regardless of the hardware and software used by the computer, the website can use WebFonts instead. A CSS3 rule called *@font-face* allows us to load any web font. We can also use a WebFont Loader [98] library designed to load web fonts using only JavaScript. The web fonts can also be used to draw on the browser canvas.

Normally, web browsers use the GPU for rendering 3D or even 2D graphics. In [27], the authors proved that machines with the same browser, operating system, and graphics card, always render the image in the same way and the produced fingerprint is identical. However, various combinations of them render the image differently, which can be used to identify the browser, OS, and graphics card on the fingerprinted computer.

Tor Browser is the only known browser that protects the user from canvas fingerprinting by asking the user every time a canvas function which can be used to read image data is invoked. If not, an empty string is returned [99]. In addition to that, Tor Browser takes actions against using the Web Graphics Library (WebGL) to obtain a fingerprint [100]. At first, WebGL can reveal the underlying driver and optimizations. At second, this library can be used for performance fingerprinting. This document also mentions keystroke fingerprinting as a kind of a biometric fingerprinting method, which relies on measuring the key strike and flight times.

A possible defense scheme for fingerprinting would be to use hardware acceleration to produce the pixels visible on the screen for performance reasons, but to use software rendering libraries such as Pango to return platform-independent strings whenever the site wants to read the produced image [27].

F. Browser instance fingerprinting using web browsing history

Methods for obtaining web history were described in Section IV-A and Section VI-D.

Research performed on 368 284 Internet users [101] revealed that it is possible to fingerprint 42 % of them by testing which websites out of 50 pre-defined ones were visited. In case if the test considered a set of 500 websites, it was possible to fingerprint 70 % of the users. The websites included in the test were taken out of 500 top Alexa ranked websites and from the top 4 000 QuantCast popular websites. The authors also converted the history profiles of visited websites by each user to category profiles. Around 88 % of the profiles appeared to be unique. It was shown that the history profiles are stable during the time. Even after clearing the browser data, the new profiles were in 38 % of cases strongly correlated to the old ones.

Another approach to identify and track users based on the history of visited websites is shown in [54]. All the major social networks allow their members to be organized in groups: public or closed ones. Every group is associated with a numerical identifier, which is a part of the URL pointing to the user's activity in the group, as browsing the posts. This URL is saved in the browsing history. The authors found that the groups to which a user belongs can be used not only to fingerprint the particular user, but also to identify his real name. The built-in functionality of social networks can be used to crawl various groups and list their members. During the 23 days of the experiment, the authors managed to crawl over 43.2 million group members from 31 853 Facebook groups. The same experiment was repeated on Xing; 1.8 million users from over 6 500 groups were crawled. Based on that, it is possible to create a database of the users and the associated groups. When a malicious script launches a web history attack and discovers which groups were visited, it is possible to correlate the outcomes with the list of groups to which the particular users belong. For the Xing network, 42.06 % of members had an unique fingerprint based on the groups to which they belong. For the rest of them, the fingerprint allows to narrow down the list of potential users. In the real-world experiment and launching a test website, the authors were able to de-anonymize 1 207 (37.3 %) out of 3 717 Xing users [54].

G. Other browser instance fingerprinting methods

A lot of browser instance-specific information is known to be detectable by using HTTP requests and the analysis of the responses [71]: the detailed version of the web browser, supported formats of images and media files, or preferred and accepted languages. Apart from the web browser version, JavaScript can list the browser plugins [102] and identify the browser user's language. Browser dimensions and Flash version can be recognized both by JavaScript [89], [103] and Flash [90], [91]. fingerprintJS [96] is a free JavaScript library which is able to fingerprint the web browser based on its agent string, screen resolution, color depth, installed plugins with supported mime types, timezone offset, local storage, and session storage. Studies performed in [104] show that the installed plugins, screen resolution, timezone, system fonts, and user-agent strings yielded altogether an entropy of 18.1 bits. In [105], the authors claim that the *User-Agent* string (which is supposed to be store the detailed information about the used browser version) alone has the entropy of 11.59 bits, while combined with the IP address results in 20.29 bits of entropy. It was also shown in [104] that if a user changes a fingerprinted property (as updates the plugin or changes the screen resolution), his current fingerprint can be matched to the earlier one by a heuristic algorithm with the accuracy of around 99 %.

In [67], the authors evaluated mechanisms used to identify the browser's instance by 3 popular large fingerprinting companies: BlueCava [68], Iovation [69], and ThreatMetrix [70]. They discovered that all the companies use both JavaScript and Flash for fingerprinting.

1) Panopticlick project: The browser fingerprinting project Panopticlick [65] uses different features to create and check the fingerprint of the particular instance of a web browser. After we got our browser fingerprinted by Panopticlick, the fingerprint appeared to be unique among 4 561 261 browsers tested so far. A quite surprising is the fact that the user agent "*Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36*" was common for one of 10 834.35 browsers tested so far, the set of installed browser plugins was common for one of 5 715.87 browsers, while the set of installed system fonts was common for one of 4 561 261 tested browsers. Other

information taken into account while computing the browser fingerprint (accepted HTTP headers, timezone, color depth, screen size, cookies enabled, and supercookies limitations) were common for 1–31 tested web browsers. Summarizing, our browser was assessed to have a fingerprint that conveys at least 22.12 bits of identifying information. As shown, these easy to obtain properties are sufficient to accurately identify the concrete instance of the web browser [104], [106].

VI. Other tracking mechanisms

The last group contains different methods, which use various methodologies (see Table I) to discover various properties (see Table II).

A. Headers attached to outgoing HTTP requests

Normally, when a web browser follows a link, the *Referer* field in the HTTP header contains the URL of the currently browsed website. If the URL contains any appended data using the GET technique (e.g., search terms, login), the information is also passed in the HTTP request.

Verizon Wireless is known from attaching a special header *X-UIDH* to every outgoing HTTP request [29]. This header is static for a long time; the average changing time was reported to be around one week. Thanks to the header send from all the mobile devices using Verizon's wireless (including mobile) networks, all the websites are able to track the users even if they use very enhanced privacy settings. The web services then use Verizon's API to buy information about the subscribers' demographic and geographic segments.

B. Using telephone metadata

According to the studies shown in [107], the telephone metadata (e.g., call logs possessed by default by governments of most countries) is highly sensitive. The authors developed an Android application, which was capturing phone metadata. Using the call logs, the researchers were able to determine the health (including mental) condition of the person, the religious believes, and addictions. As many users place their phone numbers in publicly accessible directories (e.g., Google+, Facebook), the scientists could associate around 18 % of the private numbers with the real users' identities. Summarizing, mobile applications, being granted permissions to access the call logs and other sensitive data, are a serious threat to the users' privacy.

C. Timing attacks

Differences in time taken to render different DOM trees can be used to determine boolean values, e.g., if the user has an account on a tested website [108]. The authors found that WebKit renders the website by traversing the entire tree of *RenderElements* from the root to leaves in order to resolve the pixel depth and it paints the element from the furthest away to the nearest. The users' login status (or any other state) can be determined by checking the color of particular pixels. The color of the pixel can be checked based on the time taken for the shader to output the final value [108]. The same paper

proposes a pixel-stealing attack using CSS, which was shown to have high accuracy. This kinds of attacks, which can be used to read the browser history or even steal any graphics embedded on the website, are broadly described in [109].

D. Using unconscious collaboration of the user

Apart from the automatic solutions, the browsing history can be detected in cooperation with the user, who is unaware of that fact. Four easily implementable methods were proposed in [110]. The ideas rely on producing images pretending to be CAPTCHAs, which the user need to type into a field below the image and submit to the service before some facilities or actions on the website are possible. The images are composed of words, letters, or other font-based symbols, which are in fact links to the destinations we are going to check if they were previously visited by the user. The color of the particular symbol is set to be the same as the background in case if the link was visited; otherwise, the color is set to be distinct than the background, so that the symbol can be easy recognized by the user. The produced CAPTCHA image is usually covered by a transparent image to make it more difficult to the user to recognize that this CAPTCHA is composed of hyperlinks. The first of the proposed techniques use word CAPTCHA; each word is a link to one tested website. The second solution uses LCD-like characters. Every character is composed of 4 different symbols with a nearly transparent gray background covering each other. Each of these symbols (except one, which is always black) is drawn black if the corresponding link is visited or white if non-visited. That way, depending on the combination of visited and non-visited links, the symbols compose a different character. The third solution use a gaming approach. A drawn chessboard grid is filled with pawns, which are in fact images rendered by text or SVG shapes (so their color can be controlled by CSS), which color depends on that if the corresponding link was visited or not. The user is asked to click on the visible pawns. As the website does not know on how many pawns the user needs to click, a time-out must be set after which the user is able to proceed further. The fourth presented method requires the user to select two puzzle images from which the shown resulting image can be composed. The resulting composite image is created using four SVG shapes, whose color depends on if the user visited the associated hyperlinks.

Additionally to these four universal and easy to implement solutions, a method based on a web camera is shown. Contrary to the previous ones, this method is limited to be used on websites, where the people normally uses their web cameras. The background color of the website depends on the condition if the user previously visited the tested website or not. The light of the background is reflected by the user and captured by the web camera. This method is limited to day-time estimations (as the paper proved, the accuracy is low in case if it is too dark in the room) and testing one particular website at a time. Although the background can change to test various websites, it cannot change too frequently to annoy the user [110].

There are also other techniques, which can be used to track the users and require their intervention in order to work. Such

techniques use HTML5 protocol and content handlers, and W3C Geolocation API [111].

E. Clickjacking

Clickjacking is a method to present a sensitive website element out of context, so the user acts out of context. Several approaches to clickjacking, which can lead to compromising user's anonymity, stealing user e-mails and private data, and spying on a user by webcam are shown in [112]. These techniques were used to conduct Tweetbomb [113] and Like-jacking [114] attacks. The clickjacking attacks were divided into 3 categories based on what is compromised: target display, pointer, or temporal integrity. The first method (compromising target display integrity) usually uses either hiding the target element by setting the opacity to 0 [115] or covering the target by an unclickable opaque decoy [116], so that the user clicks the attackers link. Alternatively, a partial overlay can be made [117], [118] or the target can be cropped [118]. The second method (compromising pointer integrity) uses the CSS *cursor* property to hide the default cursor and draw a fake cursor somewhere else [119] or use a custom cursor icon [120] to trick the user that the cursor is in another position. The third method (compromising temporal integrity) use the fact that people require a few hundred milliseconds to react [121], [122] in order to manipulate the target, e.g., move it after the users hovers the cursor over a button in order to click it [123], [124].

There exist several approaches to protect the user from clickjacking [112]. Target elements (e.g., Facebook *Like* buttons) started to require the user to confirm its decision [125]. Randomizing the user interface layout [126], e.g., Paypal *Pay* button, makes it more difficult to discover where it is located in order to cover it by another element. Finally, the target element can disallow to be embedded in an *iframe* by using JavaScript [127] or *X-Frame-Options* [128].

F. Evercookies (supercookies)

In [19], it was shown that in 2009, 54 out of the 100 most popular Quantcast websites used Flash cookies to rebuild removed HTTP cookies. The authors found that 41 of the Flash cookies had their content identical to the corresponding HTTP cookies. In [42], it was found that ETags and the HTML5 Local Storage are also used to rebuild deleted HTTP cookies. In [28], the authors found that 10 out of 200 the most popular Alexa websites used Flash cookies to rebuild HTTP cookies. Flash cookies are shared between different browsers by using Adobe Flash plugin and, therefore, they can be used to expand the current user's identity and rebuild HTTP cookies even in a newly installed web browser. They also detected the cases where an HTTP cookie rebuilt by one domain is passed to another domain. The *evercookie* project [20] shows how to construct a resilient JavaScript evercookie (also called a supercookie), which is using various storages in order to survive, rebuild after deletion, or even reproduce in other browsers used at the same computer. The following storages are used by this special evercookie: HTTP Cookies, Local Shared Objects (Flash Cookies) [129], Silverlight Isolated Storage [130], web history [131], ETags [57], web cache,

window.name DOM property [33], Internet Explorer userData storage [52], HTML5 Session Storage [49], HTML5 Local Storage [47], HTML5 Global Storage [46], HTML5 Database Storage via SQLite [50], HTML5 IndexedDB [51], Java JNLP PersistenceService [45], Java CVE-2013-0422 exploit (applet sandbox escaping) [132], and finally, storing cookies in RGB values of auto-generated, force-cached PNGs using HTML5 Canvas tag to read pixels (cookies) back out.

The authors also considered introducing support for caching in the HTTP Authentication [133], using Java to produce a unique key based on the NIC info, and Google Gears [30], which was discontinued by Google in 2011. Although, in these cases, a user intervention is required for the method to work.

Tor Browser Design and Implementation Draft [100] reveals various places, which can be potentially used for storing the cookies and, therefore, the data is erased while the user switches to another identity: searchbox and findbox text, HTTP auth, SSL state, OCSP state, site-specific content preferences (including HSTS state), content and image cache, offline cache, cookies, DOM storage, DOM local storage, the safe browsing key, and the Google WiFi geolocation token.

An article from 2011 [134] claimed that KISSmetrics [135], a third-party service, is using HTTP cookies, Flash cookies, ETags, the userData storage, and the HTML5 local storage to create a persistent evercookie being able to recreate the missing parts if deleted. However, the official response from the company to the published article denied these facts [136]. The company declares that it does not track users across different websites and the only mechanism used for tracking are first-party HTTP cookies. The use of ETags or any other persistent tracking object is denied. Moreover, the company is shown to respect the Do Not Track header, by avoiding tracking any information about the user, even during a single browsing session.

VII. IDENTIFICATION OF THE TRACKED USER

Being able to detect and track the digital identity of a user is valuable for many reasons. But even more profitable could be the ability to associate the digital identity of a user with the real identity (name, surname, social security number, etc). As shown in [137], it is possible to unambiguously identify 87 % of the USA population based on only 3 attributes: the date of birth, gender, and ZIP code. That can cause a severe security threat to the users. If a user decides to reveal his true identity to one service (e.g., during the registration of an e-mail account), he risks being recognized by all the other entities with which the database of user data is shared based on the formally anonymous data, as the date of birth and gender. Five ways of identifying the user and associating him with his online activities, which are the base for our classification, were proposed in [138].

A. Legitimate first-party services tracking as third parties

The tracking third party can be also a first party. Such a situation exists when we are browsing websites while we are logged into services as Facebook or Google. The Facebook *Like* or Google *+1* buttons embedded on various websites report the visit to Facebook or Google, respectively.

B. Leaking information to third parties

The identifier can leak from first parties to third parties by a number of ways. The identifier or the e-mail address can be provided in the *Referer* field in HTTP headers. The *Request-URI* header can also contain sensitive information as gender, ZIP code, or interests. Usernames or real names can be embedded in page titles or contained in shared cookies resulting from hidden third-party servers. The information collected by all these different means can be combined by a third party. For example, if a request to the third party contains a user identifier or an e-mail address in the *Referer* field in HTTP headers and attached third-party cookie, the third party can easily associate the cookie with the particular identifier or e-mail address. In [62], the authors observed several forms of leaking the identifying information to third parties as a parameter in a URL: Ads contained on the Home Depot website sent the first name and e-mail address of the user to 13 companies; when a user entered a wrong password on the Wall Street Journal Website, his e-mail address was sent to 7 companies; finally, video-sharing website Metacafe sent user's first name, last name, birthday, e-mail address, physical address, and phone numbers to 2 companies when the user changed his user settings.

In [139], the results from examining over 100 popular non-social networking websites were shown. Fully 56 % of them directly leaked personal information to third parties. When considering leaking the user identifier, the number of leaking websites grew to 75 %, as the user identifier was passed by 48 % of the websites. The e-mail address was leaked by 13 % of them. The research was based on inspecting the HTTP requests, responses, and POST requests against predefined pieces of data entered previously to the user profiles, e.g., e-mail addresses, names, and zip codes. Most of the data leaked by the *Referer* field in HTTP headers and by passing automatically cookies to third-party web servers operating under the domain of the first party.

C. Selling information to third parties

The first party can sell the identity to a third party. For example, an advertising data provider Datalogix buys data from consumer marketing companies and data compilers [140]. These data include names, postal addresses, email addresses, and demographic and behavioral information, such as past purchases. The data are used to create profiled audiences and measure the efficiency of advertising. Datalogix also claims to receive and collect information about consumers' visits to websites, such as the browser types, IP addresses, types of ads served, and dates of ad deliveries [140].

D. Using web hacks

A third-party content on the first-party website can use hacks in order to get the sensitive information. In 2010, it was demonstrated how to obtain the real name of the people associated with a particular Google account using Google Docs (this vulnerability was quickly fixed after being announced [141]). As a document edited online using Google Docs displays the names of the people who have the document opened, it was sufficient to create a fake website, which contained an invisible iframe (dimensions 0x0) embedding a publicly available Google document. If the victim was logged into his Google account, he could be identified by a script running on the fake website and looking who was recently attached to the list of people viewing the document.

E. Intended deanonymization

In 2013, a professor from Harvard University was able to re-identify 42 % of anonymous people participating in the Genome Project using census data and public records [142]. Scientists also identified 30.8 % of Twitter and Flickr users using data correlation between online accounts [143]. A Stanford University student discovered that almost 50 % of the 185 biggest websites share usernames and other sensitive information with online advertisers by putting them into the URL, which is sent back to the advertisers. That includes YouTube and Twitter [144]. The Stanford Security Lab conducted a larger scale research and found out that a popular dating website, OkCupid, leaks information to advertising networks Lotame and BlueKai about the drinking, smoking, drug habits, ethnicity, kids, pets, location, and more [145]. The Center for Internet and Society at Stanford Law School discovered that Epic Marketplace, a member of the self-regulatory Network Advertising Initiative (NAI) is stealing the users' history by a JavaScript [146]. The list of checked URLs contained 15 511 entries, including pages related to getting pregnant and fertility, menopause, repairing bad credit, and debt relief.

An approach to link multiple user accounts by using only their usernames is shown in [147]. The uniqueness of a username (the probability that it refers to the same user across different online services) was estimated based on its entropy. The model was extended to cover also matching two different usernames to the same person. The ground truth was obtained by Google Profiles, on which the users are able to link their other accounts on other services. A Markov-Chain classifier trained on over 10 million of usernames gathered from eBay and Google showed that the entropy of both distributions is higher than 35 bits. The authors also revealed how the users construct their usernames. Over 70 % of the usernames contained the first name and/or the last name of the user. Around 30 % of the usernames are result of the concatenation of the first and last names without adding any additional character.

KISSmetrics [135], introduced by us in Section VI-F, is an analytic service, which claims to be able to link all the possessed data with real users, even if they use multiple devices and browsers. It advertises itself as being able to track all the activities done by the users. Examples of the information contained by KISSmetrics databases are: the real name, gender, age, social networks logins and IDs, which websites were browsed and when, which e-mails were read, shared content in social networks, and purchased items in web stores. However, as mentioned before in Section VI-F, KISSmetrics use only first-party cookies and, therefore, the tracking cannot span among multiple services.

VIII. PURPOSES AND IMPLICATIONS OF TRACKING

Tracking is commonly used by various entities for a wide range of purposes. The obtained information is not always used directly by the tracker - a very common practice is that the collected data are sold to other parties (e.g., insurance companies or online stores) or accessed by government agencies and identity thieves. Next, we describe a few examples of the most frequently shown applications of web tracking.

A. Online advertising

Web tracking was initially developed in order to better facilitate marketing and increase sales profit. Within the time, it emerged into sophisticated techniques as behavioral tracking, audience segmentation, and targeting. For these purposes, tracking is used nowadays by majority of websites. In this section, we will show several examples of how tracking is implemented.

GMail uses words from the sent and received email messages to display targeted ads. The received e-mails are scanned despite that the sender did not give an explicit the permission to do that. GMail scans the content of the entire inbox to identify the themes and trends for ad targeting [1]. According to the Google Terms of Service: "*Our automated systems analyze your content (including emails) to provide you personally relevant product features, such as customized search results, tailored advertising, and spam and malware detection. This analysis occurs as the content is sent, received, and when it is stored*". [148].

AdStack [2] facilitates including advertisements in marketing e-mails. When a user opens a specially prepared e-mail, the advertisement is downloaded in real-time based on the information known about the specific user.

Affiliate programs (e.g., pay-per-sale [10]) require tracking to follow the user from the website where the advertisement is presented to the website where the actual purchase is made [11].

B. Web analytics and usability tests

Web tracking can also be done for some internal purposes of the website host. There are two common applications of the internal tracking of a user by a website: web analytics and usability tests [11]. These services, however, do not impose any threat to the user, as all the user's activity data are kept within a single website. The usability tests can be used to record and play back cursor movement paths, measuring the time spent by the user on the particular positions in a questionnaire, or other issues, which are important in order to improve the website in the future [11].

In [149], the authors performed a study of keyboard and mouse tracking on the Alexa top 1300 websites and they identified 115 websites, which were suspicious of user tracking. They found that 7 out of the 10 top identified websites (e.g., *microsoft.com*) append additional event-handlers to track the clicks and send the information to a third-party server. A behavior tracking software from *tynt.com* was used on 7 out of the 115 identified websites. The software, apart from its intended job of appending the URL of the website to the content copied by the user to the clipboard, sends the copied content to *tynt.com*, which allows to profile the user according to the used information.

C. Assessing financial credibility

Many companies that assess creditworthiness use the online activity of the user as one of the criteria [7]. For example, Lenddo determines if our Facebook friends are late with payment of their loans. If so, our credibility is automatically lowered proportionally to the degree of intimacy in the relation between the friends. Kreditech uses data from our Facebook, eBay, or Amazon accounts, and the user's location to assess if the user is credible or not. Kabbage requires the borrowers to grant it access to their PayPal, eBay, and other payment accounts. In addition to that, the creditworthiness can be significantly improved by linking the Facebook and Twitter accounts, which gives sense that the social network contacts are taken into account by the algorithm.

In 2009, Kevin Johnson reported to have his credit limit in American Express of 10 800 $ lowered to 3 800 $ after he shopped online in Walmart. American Express claimed that it was due to the fact that many other Walmart customers have problem with paying the credit back [8].

Schufa, the largest German ranking company, wanted to use the data from Facebook, LinkedIn, and Twitter accounts to determine the connection between different people and to track changes of their locations [9].

D. Price discrimination

Tracking also can be used to modify the advertised price of products according to the estimated financial situation of the potential customers.

In [3], it was proved that the displayed price differs based on the geographical location of the user visiting a website (up to 166 %), on the affluence of the user (up to 400 %), and on the referrer – the website from which the user accessed the website selling the product (up to nearly 50 %). The authors collected data during 20 days, performing some actions on websites belonging to 200 online vendors. No discrimination was noticed based on the operating system or browser used by the customer.

An article [4] claims that the interest rates on credit cards offered by several companies vary depending on who is looking at the credit card offer on the web. For example, it was observed that the Chase Sapphire card was advertised with interest rates 13.24 % and 12.24 %. Capital One Financial Corporation uses the calculations made by a company called [x+1] to decide which credit cards should be showed to their visitors in the first place. [x+1] extensively uses various online tracking techniques to obtain and assemble information about the Internet users. [x+1] claims that it can access and analyze thousands of pieces of information about a single user, of which the most valuable are the ZIP code and the date of birth, as people living in the same region tend to have similar income and living habits [150].

In 2010, Devin found out that Capital One Financial Corporation differentiates the interests for car loans based on the browser used by the prospective customer (3.5 % for Firefox, 2.7 % for Safari, 2.3 % for Chrome, and 3.1 % for Opera) [151].

Orbitz Worldwide Inc. differently sorts out the hotel advertisements depending on the type of computer used by the customer. Orbitz found that Mac users tend to spend around 30 % more on hotel bookings than PC users. Using that fact, more expensive hotels are advertised to Mac users, while the cheaper ones to the PC users. Furthermore, the search results are sorted according to the estimated top amount the user is going to pay for the room [152].

Matt Ilardo was shown different prices when booking a car by Hotwire from his work (88 $) and home (117 $) computers. The prices were stable in time on both computers [153], which proves that the discrimination was made based on the user's identity associated with each browser, not on the time the booking was made.

E. Determining the insurance coverage

Our online activity tells a lot about our lifestyle, interests, habits, and hobbies. That information can be used by numerous companies and institutions to assess the risk of providing us an insurance.

Allfinanz and TCP LifeSystems develop software, which is able, through different marketing data, to assess what is the risk that a certain person gets cancer soon or becomes a victim of an accident. The data is sourced from product warranties, consumer surveys, magazine subscriptions, and credit-card spending. Based on the assumed lifestyle, some insurance companies modify the frequency of medical checks of their customers [5].

Acxiom Corporation, one of the biggest data-collecting companies, is known from buying data from online publishers about the types of online articles read by subscribers. These data is correlated with publicly available information from social networks and other online profiles. Acxiom is presenting to have in average 1500 pieces of data about around 96 % of Americans and 700 millions of people globally [6].

F. Impact on the job market

It is estimated that around 90 % of employers in the US make background check of their employees (or job applicants). At the same time, the data found online is very often of bad quality, outdated, and confusing different persons having the same name [154]. Finland's Data Protection Ombudsman banned employers from googling the job applicants after the case when an employer refused to hire a person on the ground that he participated in a mental health conference [155].

G. Government surveillance

The data obtained by user tracking is a valuable source of information for government agencies and law enforcement authorities. Between January and June 2014, the US government made 12 539 requests for 21 576 person's information from Google, including search history, and Google complied with 84 % of them [156]. Google also actively scans the images that pass through GMail accounts to see if they match up with known child pornography [157].

Unique identifiers contained by tracking cookies make the government surveillance capabilities more powerful. The government agents, granted access to Internet links, are able to use cookies in order to distinguish flows originated by different users sharing the same Internet connection. The flow of the unique cookies exposes among others how the users change their locations during time. They also can be used to denounce the users who granted access to the network in a non-authorized way [158]. According to the internal NSA presentations revealed by Edward Snowden in 2013 [159], the American NSA and British GCHQ use this technique to investigate the online activity of the users. Google *PREFID* cookie was used for this purpose, as it is set by Google when a user visits its properties directly or indirectly (e.g., by an advertisement or a *+1* button provided by Google services). The presentation also shows that Special Source Operations (SSO), a division of NSA, collaborates with private companies to collect data from their systems as well as from the Internet backbone. The logins, cookies, and Google *PREFIDs* obtained by SSO were later shared with Tailored Access Operations, another division of NSA, which engages in offensive hacking operations, and with the British intelligence agency GCHQ [159]. Another presentation revealed by Snowden showed that NSA uses Doubleclick cookies to identify TOR users. That technique was assessed to be more efficient than decrypting TOR tunnels, for which the government does not have sufficient computational power. On the other hand, many TOR users seem to not clean cookies when they switch their browser to communicate with TOR, so their identity can be easy exposed by matching the cookies sent during the TOR session with the cookies sent when the communication was not encrypted [159].

Another NSA program, HAPPYFOOT, was designed to map Internet addresses to their physical locations in a precise way. Mobile devices are able to determine their location in many ways: by using GPS, WiFi, or cellular towers. The location is sent to Google or other companies for tailored advertising. By capturing the Internet traffic, NSA gathers almost 5 billion records a day on the locations of cellphones around the world. That also allows NSA to track how particular people travel and gain the knowledge about their mutual relations by revealing co-travelers [160].

Citizenfour [161], a documentary thriller based on the life of Edward Snowden and the NSA spying scandal, had its US and UK premieres in October, 2014. The name of the movie originated from a nickname of Snowden (*Citizen Four*) used by him for the purpose of mailing with journalists before revealing the compromising information.

H. Identity theft

Finally, the collected information can be used to steal our identity.

Carnegie Mellon University studies showed that data revealed by people in the Internet, when combined together,

are in many cases sufficient to predict the social security number of these people. That opens the doors to steal someone's identity, as the social security number is widely used to authenticate on sensitive websites (e.g., banking or loan services) [162].

The 2012 Identity Fraud Report revealed that LinkedIn, Google+, and Facebook users are much more likely to become victims of a fraud than other people. The reason could be that 68 % of them share their date of birth, 63 % share their high school name, 18 % their phone number, and 12 % the name of their pet [163]. LinkedIn users or people who frequently check in using their GPS position were shown to be more than twice prone to be a fraud victim than the others [164].

I. Prevalence of tracking

Third-party tracking is a form of tracking performed by resources from other services that the one explicitly visited by the user. For example, when a user browses *DR.dk*, his activity is tracked by Facebook by the mean of *Like* buttons embedded on the sites belonging to *DR.dk*. Third-party trackers (e.g., Doubleclick) are considered as a serious privacy threat, as they can collect and accumulate browsing statistics through many different websites. In [37], the authors found that around 46 % of home pages of the websites from the top 10 000 Alexa ranking are monitored by at least one third-party tracker. In particular, one third of the requests sent to third-party websites was sent to a tracker. Google was responsible for tracking on 25 % of examined websites, Facebook on 13 %, and Twitter on 5 %. A newly released study [165] showed that the presence and dominance of local tracking third parties is different in different geographical regions. In Europe, East Asia, Oceania, and South America, the distribution of third parties is almost even, while in Turkey and Israel, the local third parties have a much stronger market position. German and Russian trackers are shown to be present among the top sites in every country. Websites located in the Middle East contain mostly European and American trackers. The highest number of third-party trackers was observed on websites from Qatar (814), Korea (769), and Hong Kong (726), while the lowest on the websites from the United Kingdom (397), Jordan (330), and Belgium (274), out of 6817 trackers instances noticed over 6497 examined pages in total.

A comprehensive study about the various characters of the third-party tracking domains is shown in [166]. The authors specify 4 different types of trackers. The first type uses only third-party cookies, e.g., *doubleclick.net*. The second type uses first-party cookies together with a JavaScript code used to interrogate the cookies, which is included at the website. An example of such kind of a tracker is *Google Analytics*. It was shown that nearly 60 % of all examined first-party servers used first-party cookies set by a third-party JavaScript. The third type of trackers combines both previous approaches and use both third-party cookies and first-party cookies together with a JavaScript code. An example can be *quantserve.com*. The last, fourth type, is just used to serve advertisements, while the actual tracking process is performed by another service. An example is *2mdn.net*, which is supplied with information by *doubleclick.net*.

IX. Methods and tools for avoiding and auditing tracking

There are several strategies, methods, and tools, which can be used to discover and defend against several different tracking techniques. The possible defense strategies against each of the known tracking techniques is shown in Table III. Many of these strategies make use of the existing tools (summarized in Table IV), while the rest require from the user taking individual actions (summarized in Table V). In this section, we compare and evaluate the possible solutions. Another useful list of methods that can be used against tracking is shown in [167].

A. Blocking advertisement services

So far, third-party tracking was defended mainly by using blacklists, as Microsoft Tracking Protection List (TPL), which is a part of the newest Internet Explorer [168]. According to [37], this list is characterized by precision of 96.3 % and recall of 72.2 %. It is possible to avoid third-party tracking by blocking connections to all third-party websites, however, only 37 % of the requests to third-party websites are directed to a tracker.

TPLs are designed to block only third-party content. When the user directly navigates to a website from a TPL, it can be accessed without any problems, as it is seen as a first-party website. This can be used by scripts to load the third-party content as the first-party content. The trick is to redirect a user by a JavaScript to the third-party domain, from which he will be after a quick time redirected back to the first-party website. It is also possible to use popup windows for this purpose – the loaded third-party content will also appear as coming from the first party [38].

Facebook admits that while we browse a website that contains the *Like* button, the information about our Facebook identifier is sent to the server in order to show us who of our friends like the page as well [169]. At the same time, some additional browser-related information is sent to Facebook together with the address of the website we browse [169]. The information collected by Atlas [170] (the underlying tracking mechanism) from various sources (including the Facebook profiles) are used to sell advertisements by Facebook on properties which are owned by other entities [171]. ShareMeNot [172], incorporated from July 2014 into Privacy Badger [173], is able to detect and selectively block third-party buttons (e.g., Google *+1* or Facebook *Like*) from tracking the user unless the user clicks them. That way, the functionality of the websites in not limited, but the user's privacy is protected from being tracked by the buttons on all the visited websites. ShareMeNot uses multiple approaches to achieve this goal: provides replacement buttons instead of the original ones, disables requests to the trackers, or optionally removes cookies from request made while real buttons are loaded. When a user chooses to click on a button, ShareMeNot allows the button provider to identify the user, so the initial functionality of the button is preserved.

A very similar addon for a Firefox web browser is Request-Policy [174]. By default, it blocks all the cross-site requests

Table III
TRACKING DEFENSE STRATEGIES

Section	Tracking mechanisms	Defense strategies
II	**Session-only**	
A	*Session identifiers stored in hidden fields*	No defense known; although, not needed
B	*Explicit web-form authentication*	Do not authenticate
C	*window.name DOM property*	Block JavaScript execution
III	**Storage-based**	
A	*HTTP cookies*	Disallow cookies (all, third-party, or selectively)
B	*Flash cookies and Java JNLP PersistenceService*	Disallow Flash cookies, block Flash execution / block Java applets
C	*Flash LocalConnection object*	Block Flash execution
D	*Silverlight Isolated Storage*	Disable Isolated Storage, block Silverlight execution
E	*HTML5 Global, Local, and Session Storage*	Block JavaScript execution, disallow cookies (HTML5 storages run under the same policy as cookies)
F	*Web SQL Database and HTML5 IndexedDB*	Block JavaScript execution, disallow cookies (these storages run under the same policy as cookies)
G	*Internet Explorer userData storage*	Block JavaScript execution, disable the userData storage in IE
IV	**Cache-based**	
A	*Web cache*	
1	Embedding identifiers in cached documents	Block JavaScript execution
2	Loading performance tests	Partly by blocking JavaScript execution, but server-side measurements are still possible
3	ETags and Last-Modified headers	Web proxies able to remove or rewrite the headers
B	*DNS lookups*	Block JavaScript execution
C	*Operational caches*	
1	HTTP 301 redirect cache	No defense known; tracking can be minimized by frequent clearing of browser caches
2	HTTP authentication cache	No defense known; tracking can be minimized by frequent clearing of browser caches
3	HTTP Strict Transport Security cache	No defense known; tracking can be minimized by frequent clearing of browser caches
4	TLS Session Resumption cache and TLS Session IDs	Use Tor Browser
V	**Fingerprinting**	
A	*Network and location fingerprinting*	Tor, VPNs, anonymous web proxies, clearing of browser web cache
B	*Device fingerprinting*	Partly by blocking JavaScript and Flash execution, Tor, VPNs, anonymous web proxies, clearing of browser web cache
C	*Operating System instance fingerprinting*	Block JavaScript, Flash, Java, and ActiveX execution
D	*Browser version fingerprinting*	Block JavaScript execution
E	*Browser instance fingerprinting using canvas*	Block JavaScript execution, use Tor Browser
F	*Browser instance fingerprinting using web browsing history*	Partly by blocking JavaScript execution, but server-side measurements are still possible
G	*Other browser instance fingerprinting methods*	Partly by blocking JavaScript and Flash execution
VI	**Other tracking mechanisms**	
A	*Headers attached to outgoing HTTP requests*	Tor, VPNs, or remote web proxies configured to remove the additional HTTP headers
B	*Using telephone metadata*	Install mobile applications only from known and trusted sources
C	*Timing attacks*	Block JavaScript execution
D	*Using unconscious collaboration of the user*	No defense known; tracking can be minimized by disabling JavaScript and Flash, however, most techniques do not use any of them
E	*Clickjacking*	No defense known; tracking can be minimized by disabling JavaScript and Flash
F	*Evercookies (supercookies)*	Partly by disallowing cookies, Flash cookies, blocking JavaScript, Flash, Java, and Silverlight execution, disabling the userData storage in IE, and frequent clearing of all browser caches

Table IV
TRACKING DEFENSE TOOLS

Section	Name	Type	Description
IX			
A	Microsoft Tracking Protection List	Part of IE	Blocks third-party tracking based on a blacklist.
A	Privacy Badger	Browser extension for Google Chrome and Mozilla Firefox	Blocks third-party websites which appear to track the user based on the number of first-party websites embedding the third party. Replaces social network buttons, which blocks the social networks from tracking the user. At the same time, the button functionality is intact.
A	Request-Policy	Browser extension for Mozilla Firefox	Blocks all third-party requests by default, by the user can maintain a whitelist.
A	Adblock Plus	Browser extension for Mozilla Firefox, Google Chrome, Internet Explorer, Opera, and Safari	Blocks all unwanted ads by default, e.g., video ads on YouTube, Facebook ads, flashy banners, pop-ups, and pop-unders. Adblock Plus is also able to identify acceptable ads. It can be configured to block domains known to spread malware or to disable all known tracking. It also allows to remove all social media buttons from every website.
B	Zend2.com, KPROXY, etc.	External anonymous proxy servers	Offer anonymization services, which rely among others on hiding the IP address of the user or/and removing cookies from HTTP requests. Some of them allow bypassing censorship as the user connects with the proxy server instead of with the banned remote website.
B	SecurityKISS, CyberGhost, USA IP PPTP/L2TP/OpenVPN VPN service, VPNReactor.com, etc.	Virtual Private Networks (VPNs)	Extend the private network to a remote destination through a secure tunnel. VPN nodes are logically hosts in a single network sharing the same subnet, regardless of where the nodes are physically located.
B	Tor	Set of standalone tools for Window Linux, and Mac OS X	Protects the user's IP address by routing the traffic through a chain of Tor relays and, finally, releasing it to the global Internet by the Tor exit relay. Tor Browser is a self-sufficient package for browsing the web anonymously, which contains and embedded Tor client and a browser that disables all possible known kinds of tracking techniques.
C	Privoxy	Standalone HTTP and SSL proxy server for Windows, Linux, OpenWrt, DD-WRT, Mac OS X, OS/2, AmigaOS, BeOS, and most Unix distributions	Makes it able to alter, filter, or strip the information requested by the user. Blocks advertisements, banners, popups, and filters traffic, abusive content or cookies. HTTP headers can be modified on the fly.
I	NoScript	Browser extension for Mozilla Firefox	Allows JavaScript, Java, Flash, and other plugins to be executed only by trusted web sites of the user's choice.
I	Flashblock	Browser extension for Mozilla Firefox	Blocks all Flash content from loading leaving instead placeholders on the webpage. The user can explicitly request loading the specific Flash object.
K	Vanish	Browser extension for Mozilla Firefox	Creates self-destroying file systems based on DHT networks (e.g., Vuze, Kademlia, OpenDHT), which can be used for storing sensitive data encapsulated on a website or attached to e-mails send by GMail.
M	Disconnect	Standalone commercial tool for Windows, Mac OS X, Android, and IOS	Routes all the traffic through and encrypted tunnel to many selectable destinations, making the user able to avoid censorship and data wiretapping. Blocks around 5000 of trackers, sources of malware and identity theft. Anonymizes search queries made in web browsers. Provides the graphical visualization of tracking entities.
M	Meddle	Standalone tool for Android and IOS	Blocks, shapes, filters, or modifies the traffic from mobile devices. Supports ad blocking, tunneling all the traffic through VPN, and provides the visibility of all connections established by mobile phones. The users are informed which personal information is accessed by the particular applications and where the information is sent. They are able to decide which information should be blocked from being sent over the network or how the sent information should be altered.

(requests to the third parties), but the user can maintain a whitelist.

Adblock Plus is an open source popular browser extension created in 2006, which reached from that time more than 50 million users. It blocks all unwanted ads by default, e.g., video ads on YouTube, Facebook ads, flashy banners, pop-ups, and pop-unders. Adblock Plus is also able to identify acceptable ads. It can be configured to block domains known to spread malware or to disable all known tracking. It also allows to remove all social media buttons from every website [175]. In December 2014, the French division of the Interactive Advertising Bureau (IAB) [176] decided to fill a complaint to the court against the software allowing hiding advertisements, especially Adblock, which is the market leader.

B. Hiding the IP address

The most popular method to hide the IP address from the remote site is to use anonymous proxy servers, Virtual Private Networks (VPNs), or Tor [11].

Proxy servers are intermediary entities in communication between two parties. Some of the proxy servers offer anonymization services, which rely among others on hiding the IP address of the user or/and removing cookies from HTTP requests. At the same time, some of the proxy services allow the user to bypass censorship as the user connects with the proxy server instead of with the banned remote website. There are many web-based anonymous proxy services offered by free, e.g., Zend2.com [177] and KPROXY [178].

VPNs are designed to extend the private network to a

Table V
OTHER PRIVACY ENHANCEMENT TECHNIQUES

Section	Technique	Description
IX		
D	Opt-out cookies	Unpopular (used by less than 1% of web browsers) and inconvenient method: - for every third party, a new opt-out cookie must be set - limited lifetime (must be periodically manually renewed) - opt-out cookies are deleted while cleaning all browser cookies Finally, they usually do not stop tracking itself, but only displaying advertisements based on tracking.
E	Do Not Track (DNT) browser setting	Although adopted by all main browsers, it does not work in practice as it is supported by a low number of trackers.
F	Using privacy-focused search engines (e.g., DuckDuckGo, Startpage, Ixquick)	Contrary to the standard search engines (e.g., Google), they neither collect nor pass the searched terms to the destination websites.
G	Private browsing mode	In Firefox, Chrome, and Internet Explorer, it creates a new clean environment where the previously accumulated data (e.g., browsing history, cache, cookies, persistent storages) are not accessible. However, Safari makes the cookies, history, and HTML5 accessible in the private browsing mode, thus, it does not protect the user from being tracked.
H	Clearing the browser cache and history	Protects the user from being tracked by cache-based and storage-based mechanisms on the condition that the user does not authenticate in web services (e.g., Facebook) which extends tracking beyond this limitation.
J	Using e-mail aliases (offered by, e.g., Inbox Alias, 33Mail, Jeetable, Mailexpire, TrashMail, Guerrilla Mail)	Prevents tracking by e-mail aliases provided on different Internet websites. Decreases the number of spam messages, which also prevents tracking by entities of questionable credibility.

remote destination through a secure tunnel. VPN nodes are logically hosts in a single network sharing the same subnet, regardless of where the nodes are physically located. The tunnel connecting the particular parts of a VPN is usually secured by one of the following protocols: Point-to-Point Tunneling Protocol (PPTP), Layer 2 Tunneling Protocol (L2TP), Internet Protocol Security (IPsec), Secure Sockets Layer (SSL), or OpenVPN. SecurityKISS [179], CyberGhost [180], USA IP PPTP/L2TP/OpenVPN VPN service [181], and VPNReactor.com L2TP/PPTP/OpenVPN VPN service [182] are examples of widely-known VPN services.

Tor bundles [183] consist of the onion routing software and Tor Browser, which is a self-sufficient package for browsing the web anonymously. The users can select to install only the core onion routing package and use it together with their own web browsers, although it is not recommended, as Tor Browser protects the user's privacy by disabling all possible known kinds of tracking techniques. The traffic is routed through a chain of Tor relays and, finally, leaves the network by the Tor exit relay, which protects the real user's IP address from being known to the remote parties. Tor packages for running Tor non-exit and exit relays are also accessible on the developers website.

C. Modification of data sent over the network

Privoxy [184] is an HTTP and SSL proxy server, which is able to alter, filter, or strip the information requested by the user. The main functionality of this proxy is concentrated on blocking advertisements, banners, and popups, and filtering tracking or abusive content or cookies. Privoxy can also modify the HTTP headers on the fly.

D. Opt-out cookies

Users can opt-out from behavioral advertising using opt-out cookies. However, according to [185], less than 1 % of web browsers are reported to use them. There are several grounds for such a state of affairs. At first, for every third party, a new tracking cookie must be set. At second, these cookies have a limited lifetime, so they must be periodically manually renewed. At third, the cookies are lost when the user cleans the cookies from his web browser [62]. At fourth, the opt-out does not block the tracking itself, but only providing the advertisements based on the tracking [167].

E. Do Not Track

Do Not Track (DNT) [186] is a new type of HTTP header field, which can be appended to outgoing requests to inform the service that the user does not want to be tracked. DNT also defines an HTML DOM property, which is accessible to scripts running on the website, and APIs to register site-specific exceptions granted by the user. The websites can include the *Tk* header in the HTTP response to communicate whether and how they honor a received preference [186]. DNT was widely adopted by all the main web browsers. It was aimed at becoming the standard technique to opt-out web tracking by different services, however, it requires the tracker compliance and there is no technical or lawful possibility to enforce this request [25].

F. Using privacy-focused search engines

Privacy-focused search engines (e.g., DuckDuckGo [187], Startpage Search Engine [188], and Ixquick Search Engine [189]) claim that they do not collect any private information. Normally, when following the link from a web browser, the *Referer* field in the HTTP header contains also the searched phrases. However, the privacy-focused search engines use some techniques to prevent forwarding the search results to the linked website, mainly by obfuscating the *Referer* field in the HTTP header, or by using POST instead of GET requests.

G. Private browsing mode

All the main browsers developed a private browsing mode in order to provide the user more anonymity. A comprehensive comparison of the implementations of private browsing by various modern browsers is shown in [190]. All the main browsers (Firefox, Safari, Chrome, and Internet Explorer) provide quite good defense against the local attacker, who is, therefore, not able to determine which websites were visited by the user on the condition that the attacker did not have any prior access to the computer and the user did not bookmark any websites or saved any files on his hard drive. Regarding the web attacker, the situation is quite different. Safari, contrary to the other browsers, ignores this threat by making its earlier cookies, history, and HTML5 storages accessible in the private browsing mode. Another threat are the fingerprinting techniques, for example, used by fingerprintJS [97], which return the same values in the normal and private modes. Table VI shows how using the private browsing mode influences the different tracking techniques.

H. Clearing the browser cache and history

This method, on the condition that the user properly cleans every storage in which the browser and its plugins can keep information, behaves almost in the same way as private browsing. The only exception is Safari, which in the private browsing mode continuously serves cookies and history from the earlier browsing activities.

I. Execution blocking

There are several browser extensions, which are able to block the execution of JavaScript (e.g., NoScript [191]), Flash (e.g., Flashblock [192]), and other script content. However, other tracking methods are not impacted by using these extensions.

J. E-mail aliases

One of the most frequently revealed on the websites sensitive information is the e-mail address. If the e-mail address is publicly accessible (for example, revealed in the user's public profile), it allows easy association of different accounts to the same user. Furthermore, publicly available e-mail addresses are harvested by spam bots and used as the target for unsolicited advertising and, thus, further tracking of the user. It is why the protection of e-mail addresses is very important. However, it is not practical to create multiple e-mail accounts, which need to be frequently checked. A solution for this problem is to use e-mail aliases, from which the incoming messages are automatically forwarded to the user's real account. Many providers (e.g., Inbox Alias [193], 33Mail [194], Jetable [195], Mailexpire [196], TrashMail [197], and Guerrilla Mail [198]) offer free e-mail aliases, which lifetime can be configured from 5 minutes, through 1 month, to being infinite. They can also be manually deleted if not needed anymore. That prevents spam and protects the users' privacy, as the e-mail aliases can be safely provided to any website which requires a working e-mail address, or used to be displayed in a public web profile.

K. Self-destroying file systems

Sometimes, it is desired to place on a web or send by e-mail some content, which has a limited lifetime. That especially concerns sensitive data that should not be accessible in any way after the time specified by the owner, which prevents extracting the sensitive information at a later time from the collected data. Self-destroying file systems are the response to this need.

In [199], the authors presented Vanish – a novel self-destroying file system, which can be used for sharing sensitive data. This system is based on distributed DHT networks (e.g., Vuze, Kademlia, OpenDHT), which are used to store the encapsulated user's data. The authors implemented prototypes of Vanish clients as a Firefox plugin for sending and reading destroyable e-mails from GMail, and a browser extension, which allows to integrate encapsulated destroyable elements on a website.

L. Discovering how tracking works

A summary of available traffic auditing tools is shown in Table VII. Regardless of the mechanism, tracking can be automatically discovered by correlation between the sources of information associated with the user (as the list of visited websites or received e-mails) and the advertisements presented to him. XRay – an automatic system for discovering these correlations – is shown in [200]. The system was tested and evaluated separately on GMail, Amazon, and YouTube. Additionally, the authors assessed its precision and recall across services (YouTube to GMail). XRay was shown to be able to correlate GMail advertisements, products suggested by Amazon, and videos recommended on YouTube to e-mails in GMail, Amazon wishlists, and previously watched videos on YouTube. The correlation is based on creating a number of shadow accounts, which are used to perform various activities and to compare the produced output. For example, if an advertisement is seen on multiple accounts which contain the same e-mail while this ad is never shown on accounts lacking this particular e-mail, it is highly probable that this ad is directly associated with the content of this particular e-mail. The authors showed that XRay needs only a logarithmic number of accounts in the number of data inputs. The experiments showed that all the e-mails related to various diseases (except AIDS) result in the advertisements

Table VI
PRIVATE BROWSING MODE

Section	Tracking mechanisms	Tracking a normal-mode user identity in a private browsing mode
II	**Session-only**	
A	*Session identifiers stored in hidden fields*	Not applicable
B	*Explicit web-form authentication*	Not applicable
C	*window.name DOM property*	Not applicable
III	**Storage-based**	
A	*HTTP cookies*	Yes - in Safari No - in Chrome, Firefox and IE
B	*Flash cookies*	Yes - in Safari No - in Chrome, Firefox and IE
B	*Java JNLP PersistenceService*	Yes
C	*Flash LocalConnection object*	Yes
D	*Silverlight Isolated Storage*	Yes - in Safari No - in Chrome, Firefox and IE
E	*HTML5 Global, Local, and Session Storage*	Yes - in Safari No - in Chrome, Firefox and IE
F	*Web SQL Database and HTML5 IndexedDB*	Yes - in Safari No - in Chrome, Firefox and IE
G	*Internet Explorer userData storage*	No
IV	**Cache-based**	
A	*Web cache*	
1	Embedding identifiers in cached documents	Yes - in Safari No - in Chrome, Firefox and IE
2	Loading performance tests	Yes - in Safari No - in Chrome, Firefox and IE
3	ETags and Last-Modified headers	Yes - in Safari No - in Chrome, Firefox and IE
B	*DNS lookups*	Yes
C	*Operational caches*	
1	HTTP 301 redirect cache	Yes - in Safari No - in Chrome, Firefox and IE
2	HTTP authentication cache	Yes - in Safari No - in Chrome, Firefox and IE
3	HTTP Strict Transport Security cache	Yes - in Safari No - in Chrome, Firefox and IE
4	TLS Session Resumption cache and TLS Session IDs	Yes - in Safari No - in Chrome, Firefox and IE
V	**Fingerprinting**	
A	*Network and location fingerprinting*	Yes
B	*Device fingerprinting*	Yes
C	*Operating System instance fingerprinting*	Yes
D	*Browser version fingerprinting*	Yes
E	*Browser instance fingerprinting using canvas*	Yes
F	*Browser instance fingerprinting using web browsing history*	Yes - in Safari No - in Chrome, Firefox and IE
G	*Other browser instance fingerprinting methods*	Yes
VI	**Other tracking mechanisms**	
A	*Headers attached to outgoing HTTP requests*	Yes
B	*Using telephone metadata*	Not applicable
C	*Timing attacks*	Yes - in Safari No - in Chrome, Firefox and IE
D	*Using unconscious collaboration of the user*	Yes - in Safari No - in Chrome, Firefox and IE
E	*Clickjacking*	Not applicable
F	*Evercookies (supercookies)*	Yes - in Safari No - in Chrome, Firefox and IE

specific to the topic. For example, e-mails related to depression cause numerous *shamanic healing* ads, e-mails related to Alzheimer disease cause appearance of assisted living services ads, while cancer-related e-mails cause ads related to fight-cancer campaigns. Other topics, which highly correspond to the high number of associated ads are pregnancy, divorce, race, and homosexuality. It was shown that e-mails related to pregnancy result in showing baby-related ads, while e-mails related to homosexuality result in advertisements of gay-friendly hotels.

TaintDroid [201] is an extension to Android, which tracks in real time how different applications use our private data, especially, which data is leaked out of the mobile device. It labels data coming from privacy-sensitive sources and, thus, it can monitor when such data leave the system. The authors evaluated the accuracy of TaintDroid by installing 30 random popular applications from Google Play store. The test revealed that 15 out of the 30 applications reported users' locations to remote advertising servers. Seven of the examined applications collected the device identifier, while some of them also collected the SIM card serial number and the phone number.

Lightbeam for Firefox [202] enables the user to see the requests made to the first and third party sites. By providing interactive visualizations, Lightbeam shows the relationships between these parties.

FPDetective [84] is a framework for identifying and analyzing of web-based fingerprinting. It consists of a crawler, a parser, a proxy, a decompiler, and a central database. The crawler is based on two approaches: JavaScript fingerprinting data is collected by PhantomJS driven by CasperJS, while Flash fingerprinting data is collected by Chromium driven by Selenium WebDriver. WebKit (the engine used by both browsers) was modified to intercept and log access to various properties using API calls, which can be used for fingerprinting. All the traffic was passed by *mitmproxy*, where *mitmdump* logged all the HTTP traffic and *libmproxy* parsed and extracted all Flash files, which were decompiled to their original Action-Script source code using *JPEXS Free Flash Decompiler*. A Flash file was considered to be fingerprinting when it enumerated system fonts, collected information about the devices, and sent the information to a remote server by socket connections (*sendAndLoad*, *URLLoader* calls) or JavaScript asynchronous calls (*ExternalInterface.Call*, *addCallback*). A JavaScript file was considered to be fingerprinting when it loaded more than 30 system fonts, enumerated plugins or mime types, detected screen or browser properties, and sent the data back to the server.

MindYourPrivacy [203] is a tool for the visualization of collectors of private information, which is based on Deep Packet Inspection techniques. Thanks to that, the authors achieved browser and device independence. The results of the analysis are provided through a web-based user-friendly interface. This tool provide a wide range of identification techniques. At first, it analyzes and visualizes the graph of HTTP *Referers*. The tracking domains are aggregated as either second or third-level domains. The system consists of an application-level packet analyzer, NoSQL DB for data storage,

tracking analyzer, results generator, and an HTTP server. The main outputs of the system are: most-referred URLs, sites referring most-referred URLs, and the graph of the HTTP referrers.

$heriff [204] is a service for discovering price discrimination over the Internet. It is available as an extension for Google Chrome. $heriff allows the user to select the price of a particular product on the website of the online store and check what is the price of this product if the website is accessed by various web browsers and from different geographical locations. For that purpose, a query is issued to the $heriff server, which issues further queries with different *User-Agent* field and from different geographical locations using the PlanetLabs infrastructure.

M. Combined tools

Disconnect [205] is a commercial tool combining multiple approaches shown above. It routes all the traffic through an encrypted tunnel to many selectable destinations, making the user able to avoid the censorship and data wiretapping. The VPN blocks around 5000 of trackers, sources of malware and identity theft. The search queries made in web browsers are anonymized, so the browser cannot associate them with the particular user. This tool also provides the graphical visualization of tracking entities. A free version of Disconnect exists, however, its capabilities are limited to the visualization of trackers.

Meddle [206], described in detail in a technical report [207], is a tool for Android and IOS designed to block, shape, filter, or modify the traffic from mobile devices. It supports ad blocking, tunneling all the traffic through VPN, and provides the visibility of all connections established by mobile phones. The users are informed which personal information is accessed by the particular applications and where the information is sent. They are able to decide which information should be blocked from being sent over the network or how the sent information should be altered. All the modifications are made by a chain of middle boxes.

X. THE FUTURE OF TRACKING

A. The Future of the Cookie Working Group

The Future of the Cookie Working Group [208] of the Interactive Advertising Bureau (IAB) [176] is designing a new technology to track users across multiple sessions and devices. The group consists of 90 member companies, including [x+1], Adobe, Amazon, AOL, eBay, Google, PayPal, Quantcast, Verizon, and Yahoo [208]. The IAB, the host association, consists of more than 650 leading media and technology companies that are responsible for selling, delivering, and optimizing digital advertising, or marketing campaigns. It claims to account for 86 % of online advertising in the US [176].

In the whitepaper published in January 2011 [209], the group mentions that the current tracking techniques are not sufficient. Third-party cookies are often disabled by privacy-concerned users (and by Firefox, by default). The users are considering legislation approaches to regulate current tracking

Table VII
TRACKING AUDITING TOOLS

Section	Name	Type	Description
IX			
L	XRay	Browser extension for Google Chrome	Discovers the correlations between the sources of information associated with the user (e.g., the list of visited websites or received e-mails) and the advertisements presented to him.
L	TaintDroid	Android application	Tracks in real time which private data (e.g., user's location, device identifier, SIM card and phone numbers) are leaked out of the mobile device and which applications are responsible for the leaks.
L	Lightbeam	Browser extension for Mozilla Firefox	Provides interactive visualization and shows the relationships between tracking third parties as well as all the requests made to the first and third party sites.
L	FPDetective	Analytical framework shipped as a VirtualBox virtual machine	Identifies and analyzes web-based fingerprinting by JavaScript and Flash.
L	MindYourPrivacy	Standalone tool for Linux	Uses Deep Packet Inspection techniques to analyze and visualize collectors of private information based on HTTP Referers.
L	Sheriff	Browser extension for Google Chrome	Allows the user to select the price of a particular product on the website of the web store and check what is the price of this product if the website is accessed by various web browsers and from different geographical locations.
L	Disconnect free	Standalone tool for Windows, Mac OS X, Android, and iOS	Visualizes third-party trackers on Internet websites.

policies. All of these are causing loss of revenue to the advertising services. The authors claim that cookies have too less power: there can be accessed only by the same domain as set it up, they are specific for a device, a login to the device, and each installed web browser. The report indicates that the web services cannot use the information collected from a user from his laptop (as the location set for the weather forecast) in order to provide the same experience on the mobile device, unless the users logs in (which, in our opinion, should be the only way the user has its webspace personalized). The aim of this new tracking project is to provide full transparency of the consumer. In order to achieve that, five solution classes were proposed:

- Device-inferred, using an identifier created by using statistical algorithms based on the information about the device, browser, application, or the operating system. The identifier should be tied to the specific device and known to multiple third parties if they use the same statistical algorithm. However, the authors noticed that it is not easy to keep the identifier stable, as users change their network or IP addresses, update their devices or operating systems, and change their browser add-ons.
- Client-generated, using a static identifier, which is passed to all the third parties together with the browser, application, and operating system settings, which can be used to customize the advertising experience. The methodology is similar to Apple's AdID and identifiers used by Google and Microsoft.
- Network-inserted, using an identifier assigned by a third-party intermediary server on the path between the user's device and the publishers' servers. That can be achieved by the use of content distribution networks, Wi-Fi or wireless proxy servers, and ISPs. All the users' devices could be associated with a single identifier if the devices are registered within the same ISP as belonging to the same user. A network identifier could also be used across different networks in case of network partnerships.

- Server-issued, using cookies.
- Cloud-synchronized, using an identifier assigned by a centralized service for all third parties.

B. Google AdID

According to an article from September 2013 [210], Google is developing AdID, an anonymous identifier, which can be used for advertising and replace third-party cookies. This identifier is intended to be transmitted to the parties which agree to some guidelines giving the users more privacy and control over the tracking. The AdID is planned to be reset every year. Different browsing profiles will be associated with different AdIDs, giving the users possibility to keep their private sessions disconnected from their normal browsing activities. Additionally, the users will be able to block specific advertisers by the control panel in the browsers.

C. Microsoft Device Identifier

An article from October 2013 [211] reveals that Microsoft is working on developing its own tracking technology using a device identifier. The technology is designed to span desktop, mobile, and xbox. The permission to use the identifier will be given to Microsoft by the user when accepting the user agreement or terms of service.

D. Data Transparency Lab

Data Transparency Lab (DTL) [212] is a community, which aims at bringing clarity to the flow and usage of personal data online and in exploring ways towards a transparent and respectful data trade in the future. DTL consists of universities, businesses, and other institutions, which discover new tracking methods, analyze their impact, and design tools, to make the users able to control the use of their personal information on the web.

XI. Conclusion

The aim of this survey was to familiarize the reader with the different tracking mechanisms he, or she, encounters when browsing the web on a regular basis. The presented mechanisms have different scopes, coverages, and purposes. We show that the methods evolved over time from being privacy-friendly (as HTTP cookies, which can be easily removed by the users in case if not desired by the user) to be more and more invasive. Some of the recent techniques (e.g., fingerprinting, or using special identifiers inserted by ISPs) are not easy to avoid by regular users. Even worse, the currently designed methods are potentially even more harmful. However, during the last few years, it is observed a subsequent substantial growth of tools and services that can be used to protect the users' privacy. The potential defense approaches were described and evaluated against different tracking methods. We also showed a number of tools that can be used to discover which private data and by which application are leaking from a computer or a mobile phone.

Acknowledgment

References

[1] "Get the Facts – Don't Get Scroogled! – Scroogled," 2014. [Online]. Available: http://www.scroogled.com/mail/GetTheFacts

[2] "AdStack | Email Optimization Platform," 2014. [Online]. Available: http://adstack.com

[3] J. Mikians, L. Gyarmati, V. Erramilli, and N. Laoutaris, "Detecting price and search discrimination on the internet," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks (Hotnets'12)*. ACM New York, Seattle, Washington, USA, October 2012, pp. 79–84.

[4] "Credit card issuers watch online how you shop, customize offers – CreditCards.com," 2011. [Online]. Available: http://www.creditcards.com/credit-card-news/credit-card-issuers-watching-online-shopping-offers-1273.php

[5] "Insurance data: Very personal finance | The Economist," 2012. [Online]. Available: http://www.economist.com/node/21556263

[6] "Insurers Test Data Profiles to Identify Risky Clients – WSJ," 2010. [Online]. Available: http://online.wsj.com/news/articles/SB10001424052748704648604575620750998072986

[7] "Facebook friends could change your credit score," 2013. [Online]. Available: http://money.cnn.com/2013/08/26/technology/social/facebook-credit-score/index.html?hpt=hp_t2

[8] "'GMA' Gets Answers: Some Credit Card Companies Financially Profiling Customers – ABC News," 2009. [Online]. Available: http://abcnews.go.com/GMA/TheLaw/gma-answers-credit-card-companies-financially-profiling-customers/story?id=6747461

[9] "Outrage as credit agency plans to mine Facebook data – Tech News and Analysis," 2012. [Online]. Available: https://gigaom.com/2012/06/07/credit-agency-mines-facebook-data

[10] "Pay Per Sale Affiliate Programs :: Affiliate Directory," 2015. [Online]. Available: http://www.affiliateseeking.com/pay-per-sale.html

[11] N. Schmücker, "Web tracking," in *SNET2 Seminar Paper. Berlin University of Technology*, 2011, p. 9.

[12] K. Tarnay, S. Imre, and L. Xu, *Research and Development in E-business Through Service-oriented Solutions*. IGI Global, 2013.

[13] "Giving the Web a Memory Cost Its Users Privacy – NYTimes.com," 2001. [Online]. Available: http://www.nytimes.com/2001/09/04/technology/04COOK.html

[14] E. W. Felten and M. A. Schneider, "Timing attacks on web privacy," in *Proceedings of the 7th ACM conference on Computer and communications security*. ACM, 2000, pp. 25–32.

[15] "Jeremiah Grossman: Tracking users with Basic Auth," 2007. [Online]. Available: http://jeremiahgrossman.blogspot.com.es/2007/04/tracking-users-without-cookies.html

[16] "Javascript session variables without cookies – thomasfrank.se," 2008. [Online]. Available: http://www.thomasfrank.se/sessionvars.html

[17] "You don't know (click)jack," 2008. [Online]. Available: http://www.securityfocus.com/news/11535/

[18] "You Deleted Your Cookies? Think Again | WIRED," 2009. [Online]. Available: http://www.wired.com/2009/08/you-deleted-your-cookies-think-again/

[19] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle, "Flash cookies and privacy," in *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[20] "evercookie – virtually irrevocable persistent cookies," 2010. [Online]. Available: http://samy.pl/evercookie

[21] "LocalConnection – Adobe ActionScript® 3 (AS3) API Reference," 2011. [Online]. Available: http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/net/LocalConnection.html

[22] "Persistent and unblockable cookies using HTTP headers," 2011. [Online]. Available: https://www.nikcub.com/posts/persistant-and-unblockable-cookies-using-http-headers

[23] "Tracking users that block cookies with a HTTP redirect – Elie Bursztein," 2011. [Online]. Available: http://elie.im/blog/security/tracking-users-that-block-cookies-with-a-http-redirect

[24] "#4099 (Disable TLS Session resumption and Session IDs) – Tor Bug Tracker & Wiki," 2011. [Online]. Available: https://trac.torproject.org/projects/tor/ticket/4099

[25] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 12–12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228315

[26] "The Double-Edged Sword of HSTS Persistence and Privacy | Leviathan Security Group," 2012. [Online]. Available: http://www.leviathansecurity.com/blog/the-double-edged-sword-of-hsts-persistence-and-privacy

[27] K. Mowery and H. Shacham, "Pixel perfect: Fingerprinting canvas in html5," *Proceedings of W2SP*, 2012.

[28] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The web never forgets: Persistent tracking mechanisms in the wild," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS 2014)*, 2014.

[29] "How Verizon's Advertising Header Works | Web Policy," 2014. [Online]. Available: http://webpolicy.org/2014/10/24/how-verizons-advertising-header-works

[30] K. McKinley, "Cleaning Up After Cookies," iSEC PARTNERS, Tech. Rep., December 2008, accessible: https://www.isecpartners.com/media/11976/isec_cleaning_up_after_cookies.pdf.

[31] "HTTP access control (CORS) – HTTP – Mozilla Developer," 2014. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS

[32] "W3C Document Object Model," 2005. [Online]. Available: http://www.w3.org/DOM/

[33] "HTTP cookie: window.name – Wikipedia, the free encyclopedia," 2014. [Online]. Available: http://en.wikipedia.org/wiki/HTTP_cookie#window.name

[34] "JSON stringify revisited – thomasfrank.se," 2006. [Online]. Available: http://www.thomasfrank.se/json_stringify_revisited.html

[35] "RFC 6265 – HTTP State Management Mechanism – IETF Tools," 2011. [Online]. Available: http://tools.ietf.org/html/rfc6265

[36] M. Nasir, "Tracking and Identifying Individual Users in a Web Surfing Session," Computer and Network Security, Middlesex University, London, Tech. Rep., January 2014, accessible: https://www.academia.edu/4214725/Tracking_and_Identifying_Individual_Users_in_a_Web_Surfing_Session.

[37] T.-C. Li, H. Hang, M. Faloutsos, and P. Efstathopoulos, "TrackAdvisor: Taking back browsing privacy from Third-Party Trackers," in *Proceedings of the 16th Passive and Active Measurement Conference (PAM 2015), Proceedings Series: Lecture Notes in Computer Science 8362*. Springer International Publishing Switzerland, New York, USA, March 2015, pp. 1–12.

[38] G. Fleischer, "Implementing web tracking," in *Black Hat USA 2012 Conference Briefings*. Black Hat, July 2012, pp. 1–37. [Online]. Available: https://media.blackhat.com/bh-us-12/Briefings/Fleischer/BH_US_12_Fleischer_Implementing_Web_Tracking_gfleischer_WP.pdf

[39] "The Impact of Cookie Deletion on Site-Server and Ad-Server Metrics in Australia. Whitepaper." 2011. [Online]. Available: http://www.comscore.com/content/download/7251/125689/file/Impact+of+Cookie+Deletion+Australia_January+2011.pdf

[40] "Tracking the Trackers: Microsoft Advertising – Stanford Center for Internet and Society," 2011. [Online]. Available: http://cyberlaw.stanford.edu/node/6715

[41] "Cookie Matching – Google Developers," 2014. [Online]. Available: https://developers.google.com/ad-exchange/rtb/cookie-guide

[42] M. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle, "Flash cookies and privacy ii: Now with html5 and etag respawning," *Social Science Research Network*, 2011.

[43] S. Mittal, "User privacy and the evolution of third-party tracking mechanisms on the world wide web," Master's thesis, Stanford University. Department of Computer Science, Stanford, California 94305, USA, 2010, accessible: http://purl.stanford.edu/hw648fn9717.

[44] P. Verleg, M. van Eekelen, and H. Vranken, "Cache cookies: searching for hidden browser storage," 2014, accessible: http://www.cs.ru.nl/bachelorscripties/2014/Patrick_Verleg___3049701___Cache_Cookies_searching_for_hidden_browser_storage.pdf.

[45] "PersistenceService (JNLP API Reference 1.7.0_71)," 2014. [Online]. Available: https://docs.oracle.com/javase/7/docs/jre/api/javaws/jnlp/javax/jnlp/PersistenceService.html

[46] "DOM Storage guide – Web developer guide | MDN – Mozilla," 2014. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Storage#globalStorage

[47] "Web Storage: dom-localstorage – World Wide Web Consortium," 2014. [Online]. Available: http://dev.w3.org/html5/webstorage/#dom-localstorage

[48] B. Lawson and R. Sharp, *Introducing HTML5*. New Riders, 2011.

[49] "Web Storage: the-sessionstorage-attribute – World Wide Web Consortium," 2014. [Online]. Available: http://dev.w3.org/html5/webstorage/#the-sessionstorage-attribute

[50] "Web SQL Database – World Wide Web Consortium," 2014. [Online]. Available: http://dev.w3.org/html5/webdatabase

[51] "Indexed Database API," 2014. [Online]. Available: http://www.w3.org/TR/IndexedDB

[52] "userData Behavior – MSDN – Microsoft," 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/ms531424(VS.85).aspx

[53] A. Janc and L. Olejnik, "Web browser history detection as a real-world privacy threat," in *Computer Security–ESORICS 2010*. Springer, 2010, pp. 215–231.

[54] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *2010 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2010, pp. 223–238.

[55] L. D. Baron, "Preventing attacks on a user's history through css: visited selectors," *[Online]. Available: http://dbaron.org/mozilla/visited-privacy*, 2010.

[56] A. Juels, M. Jakobsson, and T. Jagatic, "Cache cookies for browser authentication," in *Security and Privacy, 2006 IEEE Symposium on*. IEEE, May 2006, pp. 5 pp.–305.

[57] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – http/1.1," 1999.

[58] C. Jackson, A. Barth, and J. Hodges, "Http strict transport security (hsts)," 2012.

[59] T. Dierks, "The transport layer security (tls) protocol version 1.2," 2008.

[60] "The sad state of server-side TLS Session Resumption implementations – Tim Taubert," 2014. [Online]. Available: https://timtaubert.de/blog/2014/11/the-sad-state-of-server-side-tls-session-resumption-implementations

[61] "How many users have JavaScript disabled? | ydnfourblog – Yahoo," 2010. [Online]. Available: https://developer.yahoo.com/blogs/ydnfourblog/many-users-javascript-disabled-14121.html

[62] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *2012 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2012, pp. 413–427.

[63] "MyBrowserInfo.com," 2014. [Online]. Available: http://www.mybrowserinfo.com

[64] "SystemDetails," 2014. [Online]. Available: http://www.systemdetails.com

[65] "Panopticlick," 2014. [Online]. Available: https://panopticlick.eff.org

[66] "BrowserSPY.dk," 2014. [Online]. Available: http://browserspy.dk

[67] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: Exploring the ecosystem of web-based device fingerprinting," in *Security and Privacy (SP), 2013 IEEE Symposium on*. IEEE, 2013, pp. 541–555.

[68] "BlueCava | Cross-Screen Audience Association," 2014. [Online]. Available: http://bluecava.com

[69] "Iovation," 2014. [Online]. Available: https://www.iovation.com

[70] "Advanced Cybercrime Prevention Solutions | ThreatMetrix Online Fraud Protection," 2014. [Online]. Available: http://www.threatmetrix.com

[71] "HTTP/1.1: Header Field Definitions," 2014. [Online]. Available: http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

[72] "Network IP Address via ipcalf.com," 2014. [Online]. Available: http://net.ipcalf.com

[73] "HTML5 Geolocation – W3Schools," 2014. [Online]. Available: http://www.w3schools.com/html/html5_geolocation.asp

[74] "API | SpeedOf.Me, HTML5 Speed Test | JavaScript," 2014. [Online]. Available: http://speedof.me/api.html

[75] "Speedtest.net by Ookla – The Global Broadband Speed Test," 2014. [Online]. Available: http://www.speedtest.net

[76] "What Applets Can and Cannot Do," 2014. [Online]. Available: http://docs.oracle.com/javase/tutorial/deployment/applet/security.html

[77] "IP to Country Database (IPV4 and IPV6) – Software77.net," 2014. [Online]. Available: http://software77.net/geo-ip

[78] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards street-level client-independent ip geolocation." in *NSDI*, 2011.

[79] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian, "Where's that phone?: geolocating ip addresses on 3g networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 294–300.

[80] "'Revolution 2.0': How Social Media Toppled A Dictator," 2012. [Online]. Available: http://www.npr.org/2012/02/08/145470844/revolution-2-0-how-social-media-toppled-a-dictator

[81] Y. Jia, X. Dong, Z. Liang, and P. Saxena, "I know where you've been: Geo-inference attacks via the browser cache," *Internet Computing, IEEE*, 2014, dOI: 10.1109/MIC.2014.103.

[82] K. Boda, Á. M. Földes, G. G. Gulyás, and S. Imre, "User tracking on the web via cross-browser fingerprinting," in *Information Security Technology for Applications*. Springer, 2012, pp. 31–46.

[83] "Ministry of National Education – TC MİLLÎ EĞİTİM BAKANLIĞI," 2014. [Online]. Available: http://meb.gov.tr

[84] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel, "Fpdetective: Dusting the web for fingerprinters," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1129–1140.

[85] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 93–108, 2005.

[86] "Automatic Timezone Detection Using JavaScript," 2014. [Online]. Available: http://pellepim.bitbucket.org/jstz

[87] "JavaScript Dates," 2014. [Online]. Available: http://www.w3schools.com/js/js_dates.asp

[88] "How to detect if a font is installed (only using JavaScript)," 2014. [Online]. Available: http://remysharp.com/2008/07/08/how-to-detect-if-a-font-is-installed-only-using-javascript

[89] "Detecting user's screen size," 2014. [Online]. Available: http://www.javascriptkit.com/howto/newtech3.shtml

[90] "System – Adobe ActionScript® 3 (AS3) API Reference," 2014. [Online]. Available: http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/system/System.html

[91] "Capabilities – Adobe ActionScript® 3 (AS3) API Reference," 2014. [Online]. Available: http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/system/Capabilities.html

[92] T. Unger, M. Mulazzani, D. Fruhwirt, M. Huber, S. Schrittwieser, and E. Weippl, "Shpf: Enhancing http (s) session security with browser fingerprinting," in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. IEEE, 2013, pp. 255–261.

[93] M. Mulazzani, P. Reschl, M. Huber, M. Leithner, S. Schrittwieser, E. Weippl, and F. C. Wien, "Fast and reliable browser identification with javascript engine fingerprinting," in *Web 2.0 Workshop on Security and Privacy (W2SP)*, vol. 5, 2013.

[94] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, "Fingerprinting information in javascript implementations," *Proceedings of W2SP*, 2011.

[95] "Can I use... Support tables for HTML5, CSS3, etc," 2014. [Online]. Available: http://caniuse.com

[96] "Fingerprintjs by Valve – valve's – GitHub Pages," 2014. [Online]. Available: http://valve.github.io/fingerprintjs

[97] "Valve/fingerprintjs · GitHub," 2014. [Online]. Available: https://github.com/Valve/fingerprintjs

[98] "Web Font Loader – Google Fonts – Google Developers," 2014. [Online]. Available: https://developers.google.com/fonts/docs/webfont_loader

[99] "#6253 (Prompt before allowing HTML5 Canvas image extraction) – Tor Bug Tracker & Wiki," 2012. [Online]. Available: https://trac.torproject.org/projects/tor/ticket/6253

[100] "The Design and Implementation of the Tor Browser [DRAFT]," 2013. [Online]. Available: https://www.torproject.org/projects/torbrowser/design

[101] L. Olejnik, C. Castelluccia, A. Janc *et al.*, "Why johnny can't browse in peace: On the uniqueness of web browsing history patterns," in *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, 2012.

[102] "Plugin detection with PluginDetect.js," 2014. [Online]. Available: http://www.pinlady.net/PluginDetect/All

[103] "SWFObject: Javascript Flash Player detection and embed script | deconcept," 2014. [Online]. Available: http://blog.deconcept.com/swfobject

[104] P. Eckersley, "How unique is your web browser?" in *Proceedings of the 10th international conference on Privacy enhancing technologies (PETS'10)*. Springer-Verlag, Berlin, Germany, July 2010, pp. 1–18.

[105] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: Privacy and security implications." in *NDSS*, 2012.

[106] "A Primer on Information Theory and Privacy | Electronic Frontier Foundation," 2010. [Online]. Available: https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy

[107] "MetaPhone: The Sensitivity of Telephone Metadata | Web Policy," 2014. [Online]. Available: http://webpolicy.org/2014/03/12/metaphone-the-sensitivity-of-telephone-metadata

[108] R. Kotcher, Y. Pei, P. Jumde, and C. Jackson, "Cross-origin pixel stealing: timing attacks using css filters," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 1055–1062.

[109] P. Stone, "Pixel perfect timing attacks with html5," *Context Information Security (White Paper)*, 2013.

[110] Z. Weinberg, E. Y. Chen, P. R. Jayaraman, and C. Jackson, "I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks," in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 147–161.

[111] "Geolocation API Specification – W3C – World Wide Web Consortium," 2014. [Online]. Available: http://dev.w3.org/geo/api/spec-source.html

[112] L.-S. Huang, A. Moshchuk, H. J. Wang, S. Schecter, and C. Jackson, "Clickjacking: Attacks and defenses." in *USENIX Security Symposium*, 2012, pp. 413–428.

[113] "Explaining the "Don't Click" Clickjacking Tweetbomb," 2009. [Online]. Available: http://softwareas.com/explaining-the-dont-click-clickjacking-tweetbomb

[114] "Clickjacking – Wikipedia, the free encyclopedia," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Clickjacking#Likejacking

[115] "Clickjacking – SecTheory," 2008. [Online]. Available: http://www.sectheory.com/clickjacking.htm

[116] "Clickjacking with pointer-events – JS Bin," 2009. [Online]. Available: http://jsbin.com/imuca/1

[117] "Clickjacking Details ha.ckers.org web application security lab," 2008. [Online]. Available: http://ha.ckers.org/blog/20081007/clickjacking-details

[118] "sirdarckcat: About CSS Attacks," 2008. [Online]. Available: http://sirdarckcat.blogspot.com.es/2008/10/about-css-attacks.html

[119] "Cursorjacking again - the world. according to koto," 2012. [Online]. Available: http://blog.kotowicz.net/2012/01/cursorjacking-again.html

[120] "Proof of Concept – CursorJacking – Vulnerability.fr." [Online]. Available: http://static.vulnerability.fr/noscript-cursorjacking.html

[121] "Jesse Ruderman » Race conditions in security dialogs," 2004. [Online]. Available: http://www.squarefree.com/2004/07/01/race-conditions-in-security-dialogs

[122] "lcamtuf's blog: On designing UIs for non-robots," 2010. [Online]. Available: http://lcamtuf.blogspot.com.es/2010/08/on-designing-uis-for-non-robots.html

[123] "Clickjacking Attacks Unresolved | mayscript," 2011. [Online]. Available: http://mayscript.com/blog/david/clickjacking-attacks-unresolved

[124] "162020 – pop up XPInstall/security dialog when user is about to click," 2014. [Online]. Available: https://bugzilla.mozilla.org/show_bug.cgi?id=162020

[125] "Facebook adds speed bump to slow down likejackers," 2011. [Online]. Available: https://nakedsecurity.sophos.com/2011/03/30/facebook-adds-speed-bump-to-slow-down-likejackers

[126] B. Hill, "Adaptive user interface randomization as an anti-clickjacking strategy," 2012. [Online]. Available: http://www.thesecuritypractice.com/the_security_practice/papers/adaptiveuserinterfacerandomization.pdf

[127] G. Rydstedt, E. Bursztein, D. Boneh, and C. Jackson, "Busting frame busting: a study of clickjacking vulnerabilities at popular sites," *IEEE Oakland Web*, vol. 2, p. 6, 2010.

[128] "IE8 Security Part VII: ClickJacking Defenses - MSDN Blogs," 2009. [Online]. Available: http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx

[129] "Local shared object – Wikipedia, the free encyclopedia," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Local_shared_object

[130] "Isolated Storage – MSDN – Microsoft," 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/bdts8hk0(v=vs.95).aspx

[131] "CSS History Knocker," 2014. [Online]. Available: http://samy.pl/csshack

[132] "CVE-2013-0422 – Common Vulnerabilities and Exposures," 2014. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-0422

[133] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "Http authentication: Basic and digest access authentication," 1999.

[134] "Flash Cookies and Privacy II – Ashkan Soltani," 2011. [Online]. Available: http://ashkansoltani.org/2011/08/11/respawn-redux-flash-cookies

[135] "KISSmetrics Customer Intelligence & Web Analytics," 2014. [Online]. Available: https://www.kissmetrics.com

[136] "Official KISSmetrics Response to Data Collection Practices," 2011. [Online]. Available: https://blog.kissmetrics.com/official-kissmetrics-response-to-data-collection-practices

[137] L. Sweeney, "Computational Disclosure Control: A primer on data privacy protection," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Massachusetts, USA, 2001, accessible: http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/sweeney-thesis-draft.pdf.

[138] "There is no such thing as anonymous online tracking – Stanford Center for Internet and Society," 2011. [Online]. Available: http://cyberlaw.stanford.edu/node/6701

[139] B. Krishnamurthy, K. Naryshkin, and C. Wills, "Privacy leakage vs. protection measures: the growing disconnect," in *Proceedings of the Web*, vol. 2, 2011, pp. 1–10.

[140] "Privacy | Datalogix," 2014. [Online]. Available: http://datalogix.com/privacy

[141] "How Google Docs Leaks Your Identity | 33 Bits of Entropy," 2010. [Online]. Available: http://33bits.org/2010/02/22/google-docs-leaks-identity

[142] L. Sweeney, A. Abu, and J. Winn, "Identifying participants in the personal genome project by name," *Harvard University. Data Privacy Lab. Available at SSRN.*, April 2013.

[143] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Security and Privacy, 2009 30th IEEE Symposium on*. IEEE, 2009, pp. 173–187.

[144] "Sex, drugs, and Internet advertising | ITworld," 2011. [Online]. Available: http://www.itworld.com/it-managementstrategy/212973/sex-drugs-and-internet-advertising

[145] "Tracking the Trackers: Where Everybody Knows Your Username | Center for Internet and Society at Stanford Law School," 2011. [Online]. Available: http://cyberlaw.stanford.edu/node/6740

[146] "Tracking the Trackers: To Catch a History Thief – Stanford Center for Internet and Society," 2011. [Online]. Available: http://cyberlaw.stanford.edu/node/6695

[147] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils, "How unique and traceable are usernames?" in *Privacy Enhancing Technologies*. Springer, 2011, pp. 1–17.

[148] "Google Terms of Service -- Privacy & Terms -- Google," 2014. [Online]. Available: http://www.google.com/policies/terms

[149] D. Jang, R. Jhala, S. Lerner, and H. Shacham, "An empirical study of privacy-violating information flows in javascript web applications," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 270–283.

[150] "On the Web's Cutting Edge, Anonymity in Name Only – WSJ," 2010. [Online]. Available: http://online.wsj.com/article/SB10001424052748703294904575385532109190198.html

[151] "Capital One Made Me Different Loan Offers Depending On Which Browser I Used – Consumerist," 2010. [Online]. Available: http://consumerist.com/2010/11/capital-one-made-me-different-loan-offers-depending-on-which-browser-i-used.html

[152] "On Orbitz, Mac Users Steered to Pricier Hotels – WSJ," 2012. [Online]. Available: http://online.wsj.com/news/articles/SB10001424052702304458604577488822667325882

[153] "Are online travel agencies quoting higher rates because of your Web cookies? – Elliott," 2009. [Online]. Available: http://elliott.org/blog/are-online-travel-agencies-quoting-higher-rates-because-of-your-web-cookies

[154] "AP IMPACT: When your criminal past isn't yours – Yahoo Finance," 2011. [Online]. Available: http://finance.yahoo.com/news/ap-impact-criminal-past-isnt-182335059.html

[155] "Finnish Employers Cannot Google Applicants," 2006. [Online]. Available: http://blogs.law.harvard.edu/infolaw/2006/11/15/finnish-employers-cannot-google-applicants

[156] "United States — Google Transparency Report," 2014. [Online]. Available: http://www.google.com/transparencyreport/userdatarequests/US

[157] "Google scans everyone's email for child porn | The Verge," 2014. [Online]. Available: http://www.theverge.com/2014/8/5/5970141/how-google-scans-your-gmail-for-child-porn

[158] "How the NSA Piggy-Backs on Third-Party Trackers | Stanford Center for Internet and Society," 2013. [Online]. Available: http://cyberlaw.stanford.edu/publications/how-nsa-piggy-backs-third-party-trackers

[159] "NSA uses Google cookies to pinpoint targets for hacking – The Washington Post," 2013. [Online]. Available: http://www.washingtonpost.com/blogs/the-switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking

[160] "NSA tracking cellphone locations worldwide, Snowden documents show – The Washington Post," 2013. [Online]. Available: http://www.washingtonpost.com/world/national-security/nsa-tracking-cellphone-locations-worldwide-snowden-documents-show/2013/12/04/5492873a-5cf2-11e3-bc56-c6ca94801fac_story.html

[161] "Home | CITIZENFOUR," 2014. [Online]. Available: https://citizenfourfilm.com

[162] "Face Recognition Study – FAQ – Heinz College – Carnegie Mellon University," 2011. [Online]. Available: http://www.heinz.cmu.edu/~acquisti/face-recognition-study-FAQ

[163] "Identity Fraud Rose 13 Percent in 2011 According to New Javelin Strategy & Research Report," 2012. [Online]. Available: https://www.javelinstrategy.com/news/1314/92/Identity-Fraud-Rose-13-Percent-in-2011-According-to-New-Javelin-Strategy-Research-Report/d,pressRoomDetail

[164] "Using LinkedIn? You're Twice As Likely To Report ID Theft – Business Insider," 2012. [Online]. Available: http://www.businessinsider.com/linkedin-users-are-over-twice-as-likely-to-report-identity-theft-2012-3

[165] M. Falahrastegar, H. Haddadi, S. Uhlig, and R. Mortier, "Anatomy of the third-party web tracking ecosystem," *arXiv preprint arXiv:1409.1066*, 2014.

[166] B. Krishnamurthy and C. Wills, "Privacy diffusion on the web: a longitudinal perspective," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 541–550.

[167] "Tracking the Trackers: Self-Help Tools – Stanford Center for Internet and Society," 2011. [Online]. Available: http://cyberlaw.stanford.edu/node/6730

[168] "Internet Explorer 9 Tracking Protection Lists – Microsoft," 2014. [Online]. Available: http://ie.microsoft.com/testdrive/Browser/TrackingProtectionLists

[169] "What information does Facebook get when I visit a site with the Like button or another social plugin?" 2014. [Online]. Available: https://www.facebook.com/help/186325668085084

[170] "Atlas Solutions," 2014. [Online]. Available: http://atlassolutions.com

[171] "Facebook Will Use Facebook Data to Sell Ads on Sites That Aren't Faceboo," 2014. [Online]. Available: http://recode.net/2014/09/28/facebook-will-facebook-data-to-sell-ads-on-sites-that-arent-facebook

[172] "ShareMeNot," 2014. [Online]. Available: http://sharemenot.cs.washington.edu

[173] "Privacy Badger," 2014. [Online]. Available: https://www.eff.org/privacybadger

[174] "RequestPolicy – Firefox addon for Privacy and Security," 2014. [Online]. Available: https://requestpolicycontinued.github.io

[175] "Adblock Plus – Surf the web without annoying ads!" 2014. [Online]. Available: https://adblockplus.org

[176] "IAB – About the IAB," 2014. [Online]. Available: http://www.iab.net/about_the_iab

[177] "Zend2.com," 2014. [Online]. Available: https://www.zend2.com

[178] "KPROXY – Free Anonymous Web Proxy – Anonymous Proxy," 2014. [Online]. Available: https://www.kproxy.com

[179] "SecurityKISS – Free VPN Service," 2014. [Online]. Available: http://www.securitykiss.com/index.php

[180] "CyberGhost VPN: Surf anonymously," 2014. [Online]. Available: https://www.cyberghostvpn.com/en_us

[181] "USA IP, American IP, VPN, VPS, iTunes," 2014. [Online]. Available: http://www.usaip.eu/en/index.php

[182] "VPNReactor.com," 2014. [Online]. Available: https://www.vpnreactor.com/default.php

[183] "Tor Project: Anonymity Online," 2014. [Online]. Available: https://www.torproject.org

[184] "Privoxy – Home Page," 2013. [Online]. Available: http://www.privoxy.org

[185] "Web Tracking and User Privacy Workshop, 29 April 2011," 2011. [Online]. Available: http://w3.org/2011/04/29-w3cdnt-minutes.html

[186] "Tracking Preference Expression (DNT) – World Wide Web Consortium (W3C)," 2014. [Online]. Available: http://www.w3.org/2011/tracking-protection/drafts/tracking-dnt.html

[187] "DuckDuckGo," 2014. [Online]. Available: https://duckduckgo.com

[188] "Startpage Search Engine," 2014. [Online]. Available: https://startpage.com/eng

[189] "Ixquick Search Engine," 2014. [Online]. Available: https://www.ixquick.com

[190] G. A. E. Bursztein, C. Jackson, and D. Boneh, "An analysis of private browsing modes in modern browsers," in *Proceedings of the 19th USENIX Security Symposium*, 2010.

[191] "NoScript Security Suite :: Add-ons for Firefox – Mozilla Add-ons," 2014. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/noscript

[192] "Flashblock :: Add-ons for Firefox – Mozilla Add-ons," 2014. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/flashblock

[193] "Inbox Alias – Protect Yourself," 2014. [Online]. Available: http://inboxalias.com

[194] "33Mail.com – Simple free disposable email address service," 2014. [Online]. Available: http://www.33mail.com

[195] "Jetable.org – Home," 2014. [Online]. Available: http://www.jetable.org

[196] "create a new mailexpire alias – avoid spam," 2014. [Online]. Available: http://www.mailexpire.com

[197] "TrashMail – Disposable email addresses," 2014. [Online]. Available: https://trashmail.com

[198] "Guerrilla Mail – Disposable Temporary E-Mail Address," 2014. [Online]. Available: http://www.guerrillamail.com

[199] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data." in *USENIX Security Symposium*, 2009, pp. 299–316.

[200] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu, "Xray: Enhancing the web's transparency with differential correlation," *arXiv preprint arXiv:1407.2323*, 2014.

[201] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information flow tracking system for real-time privacy monitoring on smartphones," *Communications of the ACM*, vol. 57, no. 3, pp. 99–106, 2014.

[202] "Lightbeam for Firefox," 2014. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/lightbeam

[203] Y. Takano, S. Ohta, T. Takahashi, R. Ando, and T. Inoue, "Mindyourprivacy: Design and implementation of a visualization system for third-party web tracking," in *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*. IEEE, 2014, pp. 48–56.

[204] "$heriff – Detecting Price Discrimination," 2014. [Online]. Available: http://sheriff.dynu.com/views/home

[205] "Disconnect | Online Privacy & Security," 2014. [Online]. Available: https://disconnect.me

[206] "Meddle: Take Control of Your Mobile Traffic," 2014. [Online]. Available: http://www.meddle.mobi

[207] A. Rao, A. Molavi Kakhki, A. Razaghpanah, A. Tang, S. Wang, J. Sherry, P. Gill, A. Krishnamurthy, A. Legout, A. Mislove, and D. Choffnes, "Using the Middle to Meddle with Mobile," CCIS, Northeastern University, Tech. Rep., December 2013, accessible: http://www.meddle.mobi/papers/meddle-main.pdf.

[208] "The Future of the Cookie Working Group – IAB," 2014. [Online]. Available: http://www.iab.net/member_center/committees/working_groups/Future_of_the_Cookie_Working_Group

[209] "Privacy and Tracking in a Post-Cookie World. Whitepaper." 2014. [Online]. Available: http://www.iab.net/media/file/IABPostCookieWhitepaper.pdf

[210] "Google may ditch 'cookies' as online ad tracker," 2013. [Online]. Available: http://www.usatoday.com/story/tech/2013/09/17/google-cookies-advertising/2823183

[211] "Microsoft Cookie Replacement to Span Desktop, Mobile, Xbox," 2013. [Online]. Available: http://adage.com/article/digital/microsoft-cookie-replacement-span-desktop-mobile-xbox/244638

[212] "Data Transparency Lab," 2014. [Online]. Available: http://datatransparencylab.org
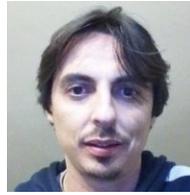
Pere Barlet-Ros received the M.Sc. and Ph.D. degrees in Computer Science from the Universitat Politècnica de Catalunya (UPC) in 2003 and 2008, respectively. He is currently an Associate Professor with the Computer Architecture Department of UPC and co-founder of Talaia Networks, a University spin-off that develops innovative network monitoring products. His research interests are in the fields of network monitoring, traffic classification, and anomaly detection. See more: http://people.ac.upc.es/pbarlet.

Tomasz Bujlow received the M.Sc. and Ph.D. degrees in Computer Science from the Silesian University of Technology in 2008 and Aalborg University in 2014, respectively. He is currently a Postdoctoral Researcher at the Broadband Communications Research Group (CBA) that belongs to the Computer Architecture Department (DAC) at the UPC BarcelonaTech. His research interests are in the field of traffic analysis and network measurements, focusing on network traffic classification. He is a participant on numerous scientific projects related to traffic classification and he holds the Cisco Certified Network Professional (CCNP) certification since 2010. See more: http://tomasz.bujlow.com.

Valentín Carela-Español received a B.Sc. degree in Computer Science from the Universitat Politècnica de Catalunya (UPC) in 2007, a M.Sc. degree in Computer Architecture, Networks, and Systems from UPC in 2009, and a Ph.D. degree from UPC in 2014. He is currently a Postdoctoral Researcher at the Broadband Communications Research Group (CBA) that belongs to the Computer Architecture Department (DAC) at the UPC BarcelonaTech. His research interests are in the field of traffic analysis and network measurements, focusing on network traffic classification. His key research area is the study of the identification of applications in network traffic based on Machine Learning and Deep Packet Inspection techniques and the aspects related to the application of those techniques in backbone networks. See more: http://people.ac.upc.es/vcarela.

Josep Solé-Pareta obtained his M.Sc. degree in Telecom Engineering in 1984, and his Ph.D. in Computer Science in 1991, both from the Universitat Politècnica de Catalunya (UPC). In 1984 he joined the Computer Architecture Department of the UPC. Currently he is Full Professor with this department. He did a Postdoc stage (summers of 1993 and 1994) at the Georgia Institute of Technology. He is co-founder of the UPC-CCABA (http://www.ccaba.upc.edu). His publications include several book chapters and more than 100 papers in relevant research journals (>25), and refereed international conferences. His current research interests are in Nanonetworking Communications, Traffic Monitoring and Analysis and High Speed and Optical Networking, with emphasis on traffic engineering, traffic characterization, MAC protocols and QoS provisioning. He has participated in many European projects dealing with Computer Networking topics. See more: http://people.ac.upc.es/pareta.