Contents lists available at ScienceDirect

Computer Communications

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

A comparison of web privacy protection techniques

Johan Mazel [a,*], Richard Garnier [b], Kensuke Fukuda [c]

[a] National Institute of Informatics, JFLI, ANSSI, Japan
[b] ENSIMAG, France
[c] National Institute of Informatics, Sokendai, Japan

ARTICLE INFO

Keywords:
Web privacy protection
Measurements

ABSTRACT

Tracking is pervasive on the web. Third party trackers acquire user data through information leak from websites, and user browsing history using cookies and device fingerprinting. In response, several privacy protection techniques (e.g. the Ghostery browser extension) have been developed. To the best of our knowledge, our work is the first study that proposes a reliable methodology for privacy protection comparison, and extensively compares a wide set of privacy protection techniques. Our contributions are the following. First, we propose a robust methodology to compare privacy protection techniques when crawling many websites, and quantify measurement error. To this end, we reuse the privacy footprint and apply the Kolmogorov–Smirnov test on browsing metrics. This test is likewise applied to HTML-based metrics to assess webpage quality degradation. To complement HTML-based metrics, we also design a manual analysis. Second, we study the overlap of blocking resources between most popular browser extensions, and compare the performances using the proposed methodology. We show that protection techniques have vastly different performances, and that the best of them exhibit a wide overlap. Next, we analyze the impact of privacy protection techniques on webpage quality. We show that automated HTML-based analysis sometimes fails to expose quality reduction perceived by users. Finally, we provide a set of usage recommendations for end-users and research recommendations for the scientific community. Ghostery and uBlock Origin provide the best trade-off between protection and webpage quality. Ghostery however requires a configuration step which is difficult for users. The RequestPolicy Continued and NoScript extensions exhibit the best performances but reduce webpage quality. Ghostery and uBlock Origin use manually built blocking lists which are cumbersome to maintain. Research efforts should focus on improving existing approaches that do not rely on blocking lists (such as Privacy badger or MyTrackingChoices), and automatically building reliable blocking lists.

1. Introduction

The huge growth of the Internet comes along with an ever-increasing advertising market. Internet users access content provided for free by publishers. Consequently, publishers monetize their audience through advertisement. Companies thus buy online exposure to promote their products. In order to maximize advertisement efficiency, advertisers tailor ads to users regarding their interests. To this end, advertisers leverage context (e.g. visited website) or previous browsing interests.

Advertisers use techniques such as cookies to identify users across websites and build their browsing history. Other techniques have also been developed to allow advertising actors to communicate with each other (such as cookie syncing [1]), or circumvent cookie removal by respawning cookies using diverse types of data storage inside the browser (e.g. using Flash [2]). Browser fingerprinting [3] allows a tracking entity to follow a user across websites without any in-browser data storage. In response to these techniques, several counter-measures were designed. We can here quote the Do Not Track HTTP header [4] by which a user can ask not to be tracked. Browsers can also block some or all cookies. Finally, many browser extensions hinder third party tracking by preventing cookie creation and/or blocking requests to tracking services.

End-users thus have many techniques available but have trouble picking one that offers good protection. Similarly, privacy researchers often want to assess protection efficiency but do not want to test all available tools. Our goal is to compare existing privacy protection techniques and provide efficiency-based recommendations. Some work previously tried to compare privacy protection techniques [5–12]. Krishnamurthy et al. [5] provide the first comparison of privacy protection methods but do not evaluate most of the browser extensions available today because they appeared after the publication of the paper. Balebako et al. [7] focus on a very specific use case: behavioral advertising. Mayer et al. [6], Hill [8,9] and Traverso et al. [12] evaluate

* Corresponding author.
E-mail address: johanmazel@gmail.com (J. Mazel).

https://doi.org/10.1016/j.comcom.2019.04.005
Received 24 January 2018; Received in revised form 16 November 2018; Accepted 14 April 2019
Available online 4 May 2019
0140-3664/© 2019 Published by Elsevier B.V.

4, 4, 4 and 7 browser extensions on a limited set of websites (respectively Alexa Top 500, 45, 84 and 100 italian URLs). Wills et al. [10] crawl a thousand of websites but use only six browser extensions. Merzdovnik et al. [11] use five browser extensions. Furthermore, these studies do not use the same metrics, it is thus difficult to compare their results. Our extensive coverage improves the state of the art regarding measurement methodology and error, and evaluated techniques.

Our contributions are the following. First, we propose a robust methodology to compare privacy protection techniques against third party tracking when crawling many websites. To this end, we reuse the privacy footprint [13] and apply the Kolmogorov–Smirnov test on browsing metrics. This test is likewise applied on HTML-based metrics to assess webpage quality degradation. We also design a manual analysis to complement HTML-based metrics. Second, we analyze the blocked resource overlap between privacy protection techniques, and compare their performances. Third, we assess the impact of privacy protection techniques on website quality. Finally, we provide recommendations for end-users and scientific community. These experiments are performed using the Firefox browser in the context of OpenWPM [14].

2. Background

This section presents an overview of third party web tracking and existing privacy protection techniques. We refer the reader to [15,16] for a more complete description of these aspects.

2.1. Third party web tracking

Websites massively rely on advertisement to monetize user visit. Advertisers purchase ads for their products directly from publishers or ad exchanges. When users access websites, they communicate with all these actors. From the users' viewpoint, these entities belong to two categories: first and third parties. First parties are the entities that users intend to reach, here, publishers. Third parties can be advertisers, ad exchanges or others actors that provide services to first parties (such as web analytics). Using techniques such as cookies, local data storage or fingerprinting, third parties can identify users across websites. Combining HTTP referrer field and user identification, they can reconstruct users' browsing history. Using cookie syncing, trackers can also exchange user identification and thus improve data collection.

2.2. Privacy protection techniques

Several techniques have been designed to protect users from third party tracking. Network-based techniques use DNS filtering or proxy. They however exhibit several shortcomings: proxies cannot analyze encrypted traffic while DNS filtering only blocks entire domains [11]. User agent spoofing browser extensions may also improve privacy by hindering fingerprinting, but exhibit poor performance in practice [17]. The next subsections provide a breakdown of browser-related techniques that we compare in this work. Unless specified otherwise, these extensions are open source.

2.2.1. Extensions

The selection of privacy protection techniques is crucial in our work. Covering all existing techniques is however extremely difficult (e.g. some minor extensions may only be available on GitHub, and needs to be manually installed which is difficult [31]). Furthermore, as shown in [32], the number of users of such extensions is negligible compared to that of most used ones. Instead, we cover the most common ones according to [32] and existing literature [5–12].

We classify these extensions regarding third party tracking impediment methods: blocking lists, heuristics, indiscriminate blocking, or other. We chose this classification because it exposes extension characteristics that plays a major role in performance (see Section 6).

Table 1
Privacy protection technique characteristics. Heu.: heuristics ; Ind.: Indiscriminate ; Popularity (K): popularity in thousands.

	Name	License	User number (K)	Version
Blocking lists	AdBlock Plus [18]	GPL v3	13,917	2.7
	uBlock Origin [19]	GPL v3	3,759	1.6
	Ghostery [20]	Proprietary	966	5.4.10
	Disconnect [21]	GPL v3	204	3.15.3
	NoTrace [22]	–	0.8	2.4
	DoNotTrackMe/Blur [23]	Prop.	86	6.0.2091
	BeefTaco [24]	Apa. 2.0	10	1.3.7.1
Heu.	Privacy Badger [25]	GPL v3	205	1.0.6
	MyTrackingChoices [24]	–	0.1	1.0
Ind.	NoScript [26]	GPL v2	1,765	2.9
	RPC [27]	GPL v3	6	1.0
Other	HTTPSEverywhere [28]	GPL v2	353	5.2.5
	Decentraleyes [29]	MPL 2.0	69	1.2.2
	WebOfTrust [30]	GPL v3	315	–

The following browser extensions use blocking lists made of regex-based rules on domain names. These lists are usually community maintained. Ghostery [20] is a proprietary, privacy-focused extension that uses a specific tracker blocking list. It has recently been bought by the privacy-focused browser Cliqz [33]. Ghostery requires a configuration step to select categories of trackers to block. uBlock Origin [19] is a general purpose blocker. It can also understand the syntax used by the famous ad-blocker AdBlock Plus. uBlock Origin includes by default: EasyList, EasyPrivacy, Peter Lowe's Adservers, Malware domains and some specific lists. Disconnect [21] is another blocker. Blur [23] (also known as Do Not Track Me) is a proprietary extension owned by Abine. It blocks trackers, protects mail addresses and passwords. NoTrace [22] uses a wide range of techniques in order to enhance privacy on the Internet. NoTrace needs to be configured after installation. One can also block tracking by setting opt-out cookies that block domains from setting standard cookies. Several entities [34,35] provide opt-cookie lists. BeefTaco [36] creates opt-out cookies in the browser. Another privacy protection approach uses ad-blockers to hinder tracking in the same way they block advertisements. AdBlock Plus [18] is the most popular ad blocker. Using specific lists, it can block trackers, social widgets, or malwares. Since 2011, AdBlock Plus is commercially exploited by Eyeo which monetizes domain whitelisting [37]. In addition to the ad-focused EasyList [38] that is used by default in AdBlock Plus, there is a privacy-focused list called EasyPrivacy [38].

Unlike previous extensions relying on blocking list, some use heuristics to block trackers. Privacy Badger [25] is developed by the Electronic Frontier Foundation and its behavior is further described in Section 5.2. By default, Privacy Badger uses the Do Not Track HTTP header [4] and strips the referrer field in HTTP requests. MyTrackingChoices [24] is an advertisement friendly privacy protection extension. It uses an hybrid approach which leverages both heuristics similar to Privacy Badger and blocking list for bootstrapping. Users can allow tracking for some website categories.

Some extensions indiscriminately block resources NoScript [26] disables JavaScript. As a side effect, this also disables some tracking. RequestPolicy Continued [27] blocks all third party requests.

Finally, some extensions use other mechanisms to protect users' privacy. HTTPSEverywhere [28] tries to replace HTTP connections with HTTPS ones if HTTPS is supported. WebOfTrust (WOT) [30] provides website rating regarding trustworthiness and child safety. It was temporally removed from extensions stores when it was revealed that WOT broke privacy rules of browser developers [39]. Decentraleyes uses local files to emulate resources, e.g. freely available JavaScript libraries, hosted on centralized entities. This blocks tracking from these entities. Table 1 provides popularity (measured as the number of users for Firefox), source code license and version of the extensions used in our work.

2.2.2. Browsers

Some browsers also have built-in privacy protection features. DoNotTrack [4] is a field in the HTTP header that asks the destination not to track the sender. It was proposed in 2009 and is currently under standardization in W3C. This feature is turned off by default in Firefox. The last proposed amendments to the European Union's ePrivacy Regulation hints at an increase interest of policy makers towards DoNotTrack [40]. Firefox [41] can block cookies: either all or only third parties' ones or only third party cookies from previously visited domains. A tracking protection [42] has also been added in Firefox version 42. Brave [43] and Cliqz [44] are browsers that emphasize their privacy protection features.

3. Related work

3.1. Tracking on the internet

The seminal contribution in the field of web privacy was by Krishnamurthy et al. They first studied the diffusion of privacy information [13], then, analyzed the impact of counter-measures on this diffusion and webpage quality [5], and observed the increase of user-data aggregation by a small number of entities [45]. Several works then tried to analyze this phenomenon with more details by classifying tracking [46] or analyzing geographical variations of tracking [47,48].

While early studies focused on cookie-based tracking, two main new tracking techniques trends emerged: resilient in-browser data storage [2] and browser fingerprinting [3]. Local data storage allows one to store data in multiple locations (for example, Flash Local Storage Object, HTML5 or ETags among other means) on a device to bypass standard cookie removal. This technique thus provides resilient local identifier storage. By using browser fingerprint, a tracker can completely avoid using and storing an identifier inside the browser and instead recognize a browser, or the device it is installed on, using its characteristics such as fonts [49], battery [50], canvas [51], hardware level features [52] or browser extensions [53]. Several studies then tried to detect fingerprinting [54], or both local storage and fingerprinting [1,14]. Furthermore, Lerner et al. [55] show that local storage fingerprinting increased between 1996 and 2016. Another privacy threat is the practice of sharing user identification across tracking entities. This is called ID-sharing, cookie-syncing or cookie-matching. While the phenomenon has been documented for some time [56], recent work analyzed its prevalence in the wild and its impact on user browsing history sharing among trackers [1,14,57].

Finally, the feasibility of user surveillance through bulk passive network traffic monitoring-based cookie observation has also been analyzed [58].

3.2. Automated blocking list building

Many privacy protection tools have been proposed (see Section 2.2). Most of these tools use manually maintained blocking lists (see Table 1). Some proposals automatically build tracking blocking lists using specific keys in URL that correspond to user identifying data [59], machine learning on DOM structure [60], Javascript [61,62] or user browsing behavior [63]. The same machine-learning-based approach was also applied to ad-blocking list building using network traffic features [64].

3.3. Privacy protection techniques comparison

While some extensions are used in almost all studies (e.g AdBlock Plus [18] and Ghostery [20]), many of them are seldom employed. Similarly, some works compare between four and seven extensions [6–12] while another focuses on two [5]. Table 2 describes how existing web privacy-related work used or analyzed existing privacy protection techniques.

Features used to compare privacy protection techniques are very diverse and thus complicate result comparison across work. Some works compare privacy protection techniques in terms of HTTP request number [6,8,9,11], cookie number [6–9], domain number [8,10,12], private information diffusion [5], and occurrence of behavioral advertising [7]. Some counter-measure proposals are also comparing their approaches to others regarding tracker blocking [42,68,69] or impact on browser performance [24]. Table 3 lists metrics and features leveraged by existing web privacy-related studies.

Our work is close to [5–12] but improves the state of the art along four axes: protection techniques, target websites, metrics, and reliability. *First*, we compare more protection techniques (15 and additional combination of blocking lists), than Krishnamurthy et al. [5] (2), Mayer et al. [6] (4), Balebako et al. [7] (4), Hill [8,9] (4), Wills et al. [10] (6), Merzdovnik et al. [11] (5) and Traverso et al. [12] (7). It is difficult to compare the extensions covered by our work with Krishnamurthy et al. [5] since the ecosystem was much simpler at the time (AdBlock Plus and NoScript were the only extensions available). We did not use some extensions that were addressed in previous studies because they are either, not supported anymore (e.g. TACO which was owned by Abine [7,31]), or not available for Firefox (AdBlock [10,24,67], Superblock Adblocker [24], Adremover [24] and Adblock Pro [24]). Furthermore, AdBlock (resp Firefox Tracking Protection [42]) uses the same blocking list as AdBlock Plus (resp. Disconnect). By analyzing AdBlock Plus and Disconnect, we thus provide performance bounds on these two techniques. Achara et al. [24] and Wills et al. [10] also evaluated the performance of an ad-blocking extension that we do not address in this study: AdGuard Adblocker. *Second*, we us more websites than most existing work. We use the Alexa Top 1000 while Mayer et al. [6], Hill [8,9] and Traverso et al. [12] use the Alexa Top 500, 45, 84 and 100 URLs, and 100 italian URLs. Balebako et al. discussed five browsing scenarios that use a small number of websites to assess the occurrence of behavioral advertising, we thus cannot compare the websites we use. We crawled a number of websites similar to Wills et al. [10] but analyzed many more protection techniques. We use a smaller number of crawled websites than Merzdovnik et al. [11] (who uses Alexa Top 100000) because some of our metrics (such as number of cookies) require a stateful crawl and thus forbid parallelism. *Third*, we use more metrics (see Table 3) than any existing work. We are thus able to provide a better description of techniques' performance. We did not compute the host number metric used by Hill [8,9] because it is very similar to the number of domains. This metric actually reflects the internal network architecture of trackers, which is not relevant for privacy protection comparison. *Finally*, none of these work performs several measurements on each website to remove measurement error, except Mayer et al. [6] who perform three crawls for each URL but does not provide any justification for this number. To the best our knowledge, we here propose the first measurement error analysis for privacy protection technique comparison (see Section 5.1).

4. Methodology

This section presents data collection and used metrics for privacy protection and webpage quality.

4.1. Data collection

We use OpenWPM [14], an open-source framework written in Python that relies on Selenium for browser automation. OpenWPM supports Firefox and provides some extensions. With minor modifications, we add extensions presented in Table 1 and Section 2.2. This study has been conducted with Firefox 45.

We crawl the highest-ranked websites by Alexa [72]. Unless stated otherwise, we perform measurements in August 2017, using IP addresses located in Japan. Typically, a crawl on the Alexa Top 1000 (Section 6.1) takes about three days with commodity hardware.

Table 2
Extensions used in previous publications (· for usage in the wild, ○ for ground-truth, ⊙ for ground-truth and usage, + for comparison, ⊕ for comparison and ground-truth, ⋆ for ML bootstrapping , ∼means that we did not evaluate these techniques but that their results can be derived from our work: AdBlock uses the same default blocking list as AdBlock Plus and Firefox Tracking Protection uses the same blocking list as Disconnect). The "ground-truth" here means that the considered study uses the blocking list of considered extensions as reference to classify a domain as related to tracking.

Type	Authors & references	Extensions																	Browsers	
		AdBlock [65]	AdBlock Plus EL [18,38]	AdBlock Plus EP [18,38]	uBlockOrigin [19]	Ghostery [20]	Privacy Badger [25]	Disconnect [21]	NoTrace [22]	WebOfTrust [30]	DoNotTrackMe/Blur [23]	MyTrackingChoices [24]	RPC [27]	Decentraleyes [29]	NoScript [26]	DNT [4]	Opt-out cookies [34–36]	HTTPSEverywhere [28]	Firefox [42]	Cliqz
Web privacy analysis	Krishnamurthy et al. [13]		○																	
	Castelluccia et al. [47]		○			○														
	Fruchter et al. [48]		○	○																
	Carrascosa et al. [66]															+				
	Metwalley et al. [67]	·	·			⊙				·	⊙				·					
	Englehardt et al. [58]					+										+		+	+	
	Englehardt et al. [14]		○+	○+		+													+	
Privacy protection	Malandrino et al. [68]		+			+			+				+		+					
	Malandrino et al. [69]		+			+			+				+		+					
	Kontaxis et al. [42]		+																+	
	Achara et al. [24]		+		+	+						+							+	
Tracker detection	Gugelmann et al. [64]		+⋆	+⋆																
	Metwalley et al. [59]					○					○									
	Wu et al. [61]					⋆														
	Yu et al. [70]							+											+	+
	Ikram et al. [62]		+			+	+	+							+					
User ana.	Leon et al. [31]		+			+										+	+			
	Malandrino et al. [71]								+											
Comp.	Krishnamurthy et al. [5]		+												+					
	Mayer et al. [6]		+	+		+											+			
	Balebako et al. [7]		+			+										+	+			
	Hill [8]		+			+	+	+												
	Hill [9]		+		+	+		+												
	Wills et al. [10]		+	+	+	+		+			+									
	Merzdovnik et al. [11]		+		+	+	+	+												
	Traverso et al. [12]		+	○	+	○+	+	○+			+		+							
Our work	–	∼	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	∼	

4.2. Privacy protection

In this section, we present our robust methodology to compare privacy protection techniques across several websites and using several metrics. We also present privacy footprint [13].

4.2.1. Browsing metrics

As explained in Section 2.1, privacy leakage occurs through communications with trackers, and local data storage (e.g. cookie) allows trackers to easily identify users across website. It is impossible to construct the exhaustive set of efficient metrics for privacy protection techniques comparison. Furthermore, this task is beyond the scope of our work. Instead, we carefully survey the metrics used in the past literature, and select commonly used and efficient metrics as shown in Table 3. As later demonstrated in Section 6.1.1, used metrics are able to compare and discriminate privacy protection techniques. We consider five simple browsing-related metrics that reflect the ability of protection techniques to hinder these two phenomena. We first focus on HTTP requests. We separately count the requests that are made to the accessed domain (*the number of first party requests*), and the requests that are made to other domains (*the number of third party requests*). To this end, we use the second level domain name obtained from the Public Suffix List [73]. These metrics give a rough estimation on the performance of a particular extension. A protection technique is effective if the number of first party requests is not impacted, and the number of third party requests decreases.

The third metric is *the number of third party domains* accessed during a crawl. This is complementary to the number of third party requests. It provides a metric on the number of blocked entities. There may be a few domains that generate a lot of requests, or a many domains that produce a few requests. Efficient protection techniques reduce the number of third party domains.

This is however not sufficient since third party requests are not always used to track users, they can also provide content to users (e.g. media resources) or contact non-tracking third parties (e.g. website analytics). The fourth metric is *the number of the profile cookies* obtained from a stateful crawl.

Protection techniques with good performances diminish the number of cookies.

The last metric is *the total amount of data transferred*. It is especially important in low-bandwidth situations for example on mobile. This metric is directly related to tracking and advertisement. Advertisement often generates considerable traffic during browsing. However, as we will see in Section 5.1, this metric shows high variability when crawling the same website. We thus did not use this metric in our analysis.

4.2.2. Kolmogorov–Smirnov test-based browsing metrics comparison

While the metrics introduced in Section 4.2.1 provide a good estimation of the privacy protection for a particular website, summarizing this aspect for a set of websites is non-trivial. Fig. 1 presents two empirical cumulative distribution functions (ECDFs) built on the mean number of third party requests sent, for each website of the Alexa Top 1000 world. We use the mean of each metrics for ten crawls to reduce measurement error (see Section 5.1). We here use two Firefox configurations: one without any extension (bare) and one with uBlock Origin. To determine whether one Firefox configuration exhibits the

Table 3
Metrics used for privacy protection technique comparison (○ for metric used and ⊙ for metric used with breakdown, ⋆ uses total # domains = #3rd party domains + 1 for the publisher).

Type	Authors & references	Privacy protection										Webpage quality							Browser		
		# 3rd party domains	# hosts	# cookies	# third party cookies	# requests	# first party requests	# third party requests	TP/FP/FN/TN	Privacy footprint	# script	# image	Data	Script data size	Image data size	Manual analysis	Loading time		Browser CPU usage	Browser memory usage	Use-case specific metric
Web privacy analysis	Krishnamurthy et al. [13]									○											
	Fruchter et al. [48]				○			○													
	Carrascosa et al. [66]																				○
	Englehardt et al. [58]																				○
	Englehardt et al. [14]				○			○			○										○
Privacy protection	Malandrino et al. [68]								○								○			○	
	Malandrino et al. [69]												⊙								
	Kontaxis et al. [42]							○					○				○				
	Achara et al. [24]																○		○	○	
Tracker detection	Yu et al. [70]							○													
	Ikram et al. [62]										○										
Comparison	Krishnamurthy et al. [5]	⊙											○								
	Mayer et al. [6]			○		○															
	Balebako et al. [7]			○																	○
	Hill [8]	○⋆	○	○		○					○		○								
	Hill [9]			○		○					○								○		
	Wills et al. [10]	○																			
	Merzdovnik et al. [11]							○													
	Traverso et al. [12]	⊙											○				○				
Our work	–	○		○		⊙	○	○		○	○	○	○	○	○	○					

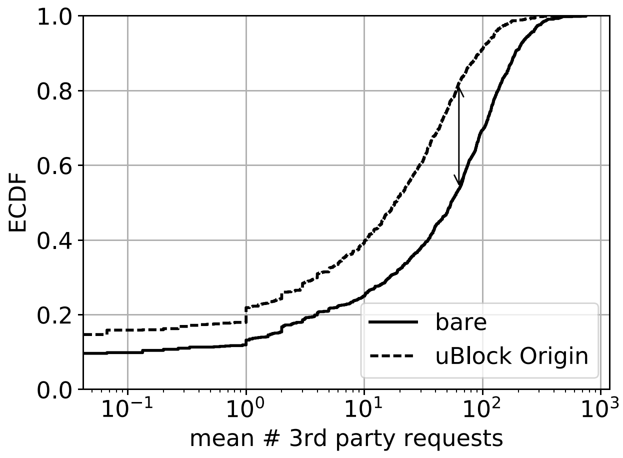


Fig. 1. Example of ECDF of the mean number of third party requests for all websites in the Alexa Top 1000 with two configurations: default Firefox (bare) and Firefox with uBlock Origin. The arrow represents the KS statistic.

same performance as another, we use the Kolmogorov–Smirnov (or KS) test. It is a non-parametric test that does not make any assumption on the underlying distributions which fit our context. This test relies on the KS-statistic which is the maximum difference between two cumulative distributions. The KS statistic is represented by the arrow on Fig. 1. The null hypothesis is that both ECDFs belong to the same distribution. In our case, this means that both configurations have the same performance. We arbitrarily choose a significance level α of 0.05 which is in line with common practice and the seminal work of Fisher [74]. If the p-value of the KS-test is smaller than α, we consider the two ECDFs as distinct. In other words, the two considered configurations have performances that are statistically different.

4.2.3. *Privacy footprint*

In this work, we also use the privacy footprint proposed by Krishnamurthy et al. [13]. The privacy footprint represents interactions between first parties and third parties (i.e. potential trackers) in a graph. For each third party accessed by the user when visiting a first party, an edge is added between the nodes representing the considered first and third parties. Privacy footprint thus captures both the potential information leaking from first parties, and the aggregating behavior of third parties. Krishnamurthy et al. [13] used the second-level domain name of the authoritative DNS server of third parties to group domains in the same entity on the same node in the graph. This method is called *ADNS*. Krishnamurthy et al. [45] then noticed that unrelated third parties located on the same hosting service or Content Delivery Network (CDN) are grouped together by *ADNS* because they share authoritative DNS servers. They thus develop a new method called *root*. With *root*, if a third party is located in a hosting service or a CDN, the second-level domain-name of the third party domain is the node identifier. Otherwise, as with *ADNS*, the second-level domain name of the authoritative DNS server of the considered third party is the node.

We use three metrics that are built on the graph. (1) the *number of third parties* reveals tracking breadth. (2) the *mean number of third parties per first party* corresponds to tracking intensity. (3) the *number of first parties associated with the top 10 third parties* estimates how much tracking is concentrated on the most prominent third parties.

4.3. *Webpage quality*

Privacy protection techniques prevent privacy information leakage by hindering of communications with trackers and blocking cookie creation, among other techniques. This may however have negative side-effects on webpage quality. Blocked third party domains may actually host images or JavaScript that are needed to render the webpage correctly. We first analyze the impact of privacy protection on browsing data in an automated fashion. We then perform a manual analysis to assess privacy protection repercussion on rendered webpage.

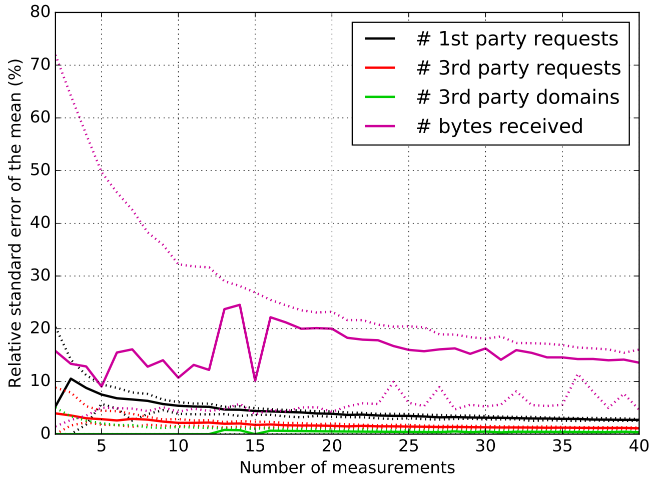


Fig. 2. Relative standard error of the mean of browsing metrics in function of the number of measurements when accessing www.yahoo.jp. Solid lines represent the median, dashed lines correspond to 5 and 95-centiles.

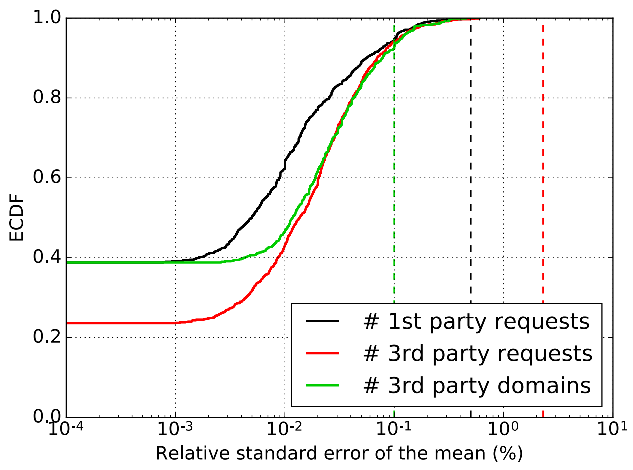


Fig. 3. ECDF of the relative standard error of the mean of browsing metrics on the Alexa Top 1000 websites crawled ten times. Vertical dashed lines represent the relative standard error for www.yahoo.jp.

4.3.1. HMTL-based metrics

Our goal is to detect layout differences or missing elements that may reduce webpage quality both in terms of rendering quality and functionality. We crawl HTML data using OpenWPM. We devise several metrics that analyze the browsed page. The first metric is the *HTML page size*. We then derive two metrics from the crawled HTML data itself: *number of images and scripts*. The last two metrics are the *total size of images and scripts*. For all metrics, a reduction is associated with a webpage quality decrease. We here reuse the metric comparison method presented in Section 4.2.2.

4.3.2. Manual analysis

By analyzing HTML, we are able to determine how many elements and how much data is missing when privacy protection is used. This, however, does not assess how well the webpage is rendered inside the browser. We thus perform a manual analysis to determine webpage quality obtained with privacy protection techniques. Webpage screenshots are captured using OpenWPM. First, we want to determine whether privacy protection techniques impact webpage layout. Our first question thus is: "Please rate layout similarity between Bare (left) and Extension (right)". Good layout similarity, however, does not guarantee lack of missing elements. The next step is to compare webpage elements when privacy protection is used and not used. We want to avoid comparing elements of the same webpages that may change for different users or crawling time (e.g. news items). We thus ask user to focus on webpage elements that are part of the user interface. Our second question thus is: "Please rate the proportion of elements (image, text frame, widgets, etc.) of the user interface (e.g.: login button, tabs, etc, but not news items or pictures) in Bare (left) that are also in Extension (right)". For both questions, users give a rating between 0 and 10, 0 being the worst and 10 the best. Our manual analysis was performed by six users.

5. Measurement parameters

We address two measurement parameters: the impact of crawling parameters on measurement error, and the training of Privacy Badger.

5.1. Impact of crawling parameters on measurement error

During our preliminary experiment, we notice that measurement results are not stable. Querying twice a website usually yields two different metric values. To determine the appropriate number of measurements that generates a small error, we conducted a detailed study on the website showing the highest variability in our preliminary experiment: www.yahoo.jp. These measurements were performed is November 2016.

Varying the number of measurements, we computed the relative standard error of the mean of four browsing metrics.

Fig. 2 shows that the relative standard error decreases when the number of measurements increases. Performing ten crawls reduces the relative standard error on the number of first party requests, third party requests, and third party domains to less than 5%. The number of bytes received, however, still has a large error, we thus discard this metric. One possible reasons for this measurement error is the webpage advertisement churn. Guha et al. [75] analyze ad churn during page reloads. They show that the number of unique new ads increases quickly during the first ten reloads but only linearly thereafter. This further justifies our choice to use ten measurements. An existing comparison of privacy protection techniques [6] uses three crawls for each website and extension. They however do not motivate this choice nor provided measurement error analysis. We perform ten crawls because this value is a reasonable trade-off between measurement error and duration.

Fig. 3 is the ECDF of the relative standard error of the mean of the number of first and third party requests, and of the number of third party domains for all websites in the Alexa Top 1000 crawled 10 times. Most websites have a relative standard error smaller than that of www.yahoo.jp (here in dashed lines). Overall, 99% of websites have a relative standard error smaller than 1% for all observed features.

5.2. Privacy badger training

Privacy Badger employs heuristics to determine if a domain is performing tracking (see Section 2.2.1). Privacy Badger measures the number of times that a domain reads a cookie as a third party. When this number is greater than three, the considered domain is blocked. On top of this, cookies are also whitelisted using heuristics. This behavior causes a freshly installed Privacy Badger not to block any domain. As the user browses websites, more and more domains are blocked. To fairly evaluate Privacy Badger, we analyze the impact of Privacy Badger training. In other words, we intend to quantify how many websites need to be accessed to ensure that Privacy Badger is completely trained.

If we were to determine the number of websites regular users need to browse to train Privacy Badger, we should use a browsing model such as AOL search logs [76] as Roesner et al. [46] do. Our use-case here is however to ensure that Privacy Badger is trained for a specific set of websites. We thus measure Privacy Badger's performance on the

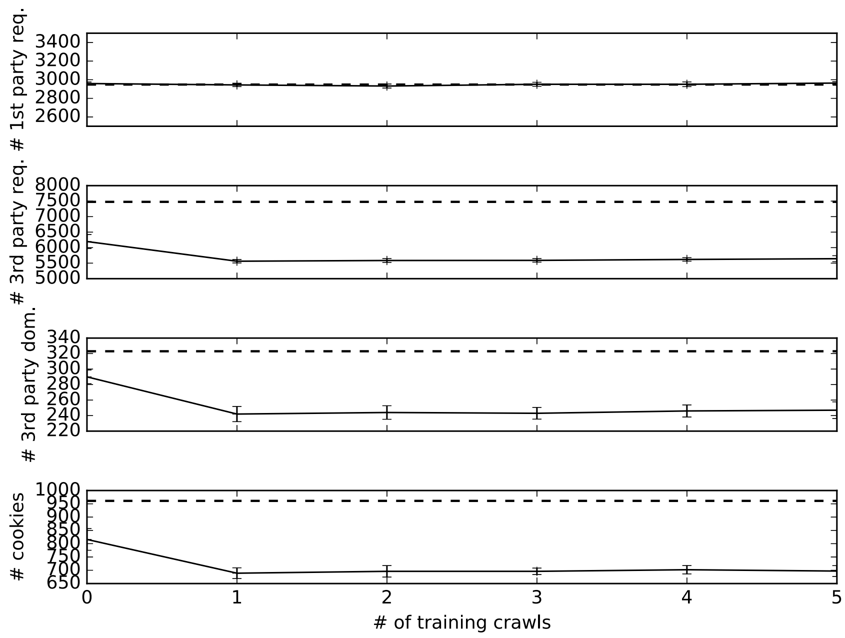
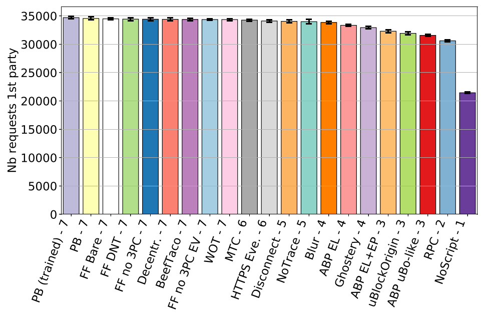
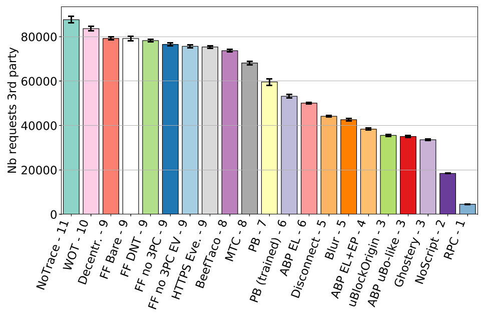


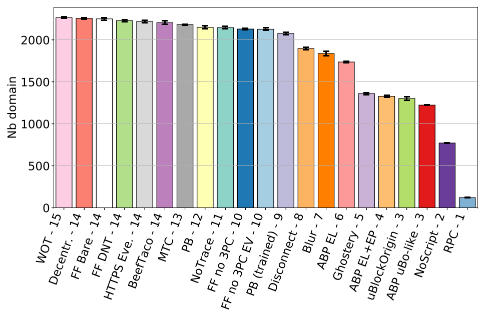
Fig. 4. Performances of Privacy Badger depending on the number of crawl performed for its training. The horizontal dashed line represents a default Firefox as reference.



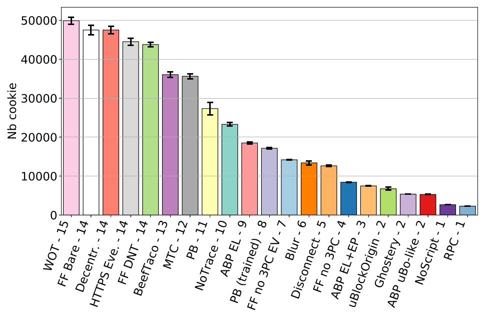
(a) # first party requests



(b) # third party requests



(c) # third party domains



(d) # cookies

Fig. 5. Performance comparison of protection techniques regarding the mean # first party request, mean # third party requests, mean # third party domains and # cookies. Each metric value corresponds to the total obtained by crawling the Alexa Top 1000 websites. The mean and standard deviation are computed on ten crawls. The number on top of technique name is the KS-based rank. Bare: Firefox alone, MTC: MyTrackingChoices, PB: Privacy Badger, FF no 3PC: Firefox no 3rd party cookie, FF no 3PC EV: Firefox no 3rd party cookie blocking except from previously visited domains, ABP: AdBlockPlus, EL: EasyList, EP: EasyPrivacy, RPC: RequestPolicy Continued . AdBlock Plus uBo-like uses all uBlock Origin's lists except uBlock Origin's specific ones; it thus loads EasyList, EasyPrivacy, Peter Lowe's Ad Server list and Malware domains.

Alexa Top 100 websites after training. These crawls were performed is November 2016. This training consists of several repeated crawls targeting the same set of websites and performed without reinitializing the user profile, i.e. Privacy Badger continues its training. We use zero to five successive training crawls and then perform a single measurement crawl.

The results of this experiment are shown in Fig. 4. The dashed line corresponds to a reference: crawling results of a bare browser. KS test groups the six measurements with Privacy Badger together, but separates them from the bare browser. Without training, Privacy Badger is already able to provide partial protection. After a single training pass, the number of third party requests is reduced by 10.3%. The results however do not improve when the number of training crawl increases. We also ran a single training crawl on the Alexa Top 500 websites and observed similar performance as with a single training crawl. The training is thus complete with less than one crawl (here

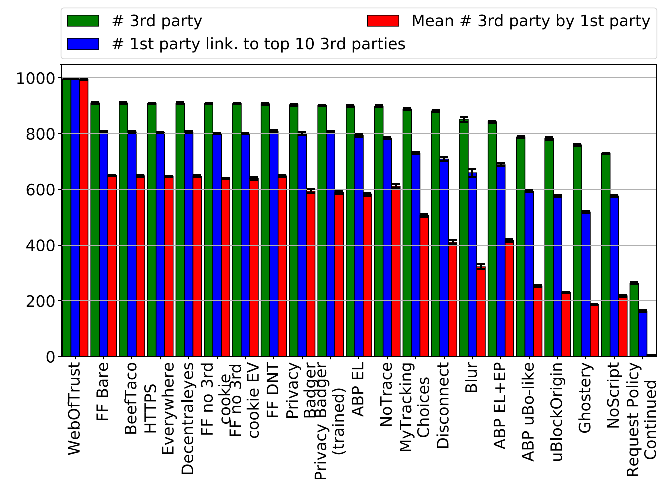


Fig. 6. Privacy footprint [13,45] for the Alexa Top 1000 world. The mean and standard deviation are computed on the metric value of the ten crawl. EV: Firefox no 3rd party cookie blocking except from previously visited domains, ABP: AdBlockPlus, EL: EasyList, EP: EasyPrivacy.

one hundred websites). MyTrackingChoices uses the same principle but without Privacy Badger's cookie data-based white-listing heuristics. One can thus expect MyTrackingChoices to train faster. Moreover, MyTrackingChoices is provided with a small bootstrapping tracker list which should further reduce training time. For the remainder of this paper, we train Privacy Badger using a single crawl on the considered Alexa Top websites. Out of the four existing work that uses Privacy Badger [8,11,12,62], only two actually perform some training [8,11]. We here provide the first estimation of how much website needs to be visited to complete Privacy Badger's training. In the remainder of the paper, we use a single training pass.

6. Results

6.1. Privacy protection

We compare protection techniques, first in terms of performance and then, regarding their blocking overlap.

6.1.1. Browsing metrics

We perform a general comparison of privacy protection techniques. We crawled the Alexa Top 1000 World using 21 different browser configurations. One configuration is the default setup of the Firefox browser (Bare). One configuration is Firefox setup with the DoNotTrack HTTP header. Two configurations use Firefox's ability to block cookies: third party ones or third party cookies except from previously visited websites. We do not use complete cookie blocking because we hypothesize that users want to keep using cookie for some domains. Three configurations set up AdBlock Plus with various lists: EasyList, EasyList and EasyPrivacy and a filter set similar to uBlock Origin (EasyList, EasyPrivacy, Peter Lowe's Ad Server list and Malware domains). One configuration uses NoTrace with the high preset (the default configuration has no effect on our metrics). The last 14 configurations are extensions with their default setting. We do not use the Firefox Tracking Protection [42] because OpenWPM does not support it yet. However, this technique uses the Disconnect blocking list. This means that by analyzing Disconnect, we provide a performance bound on Firefox Tracking Protection's performances. The results are shown in Fig. 5. The number on top of technique name is the KS-based rank (see Section 4.2.2).

All techniques have a limited impact on first party requests. The most aggressive technique is NoScript [26]. This is due to the fact that Javascript is pervasive on Internet. Other techniques exhibit a limited impact on th enumber of blocked first party requests.

Except for the number of cookies, the two most effective extensions are RequestPolicy Continued and NoScript which reduce the number of third party HTTP requests by 96% and 87%. However, as shown in Section 6.2 and other studies [5,69], they strongly impact webpage quality.

Among other techniques, Ghostery [20] and uBlock Origin [19] provide the best performances. They reduce the number of third party HTTP requests by 58% and 55%. Ghostery however needs to be configured which is difficult for users [31]. AdBlock Plus [18] uBO-like has similar performances with uBlock Origin and Ghostery. Our results show that users can manually setup AdBlock to obtain good results, unfortunately this is difficult [31]. Furthermore, AdBlock Plus used with EasyList is much more CPU and memory-hungry than uBlock Origin [24].

Another group of techniques provides average performances: Disconnect (and Firefox Tracking Protection), AdBlock Plus (with EasyList and EasyList + EasyPrivacy), Privacy Badger, Blur, BeefTaco and NoTrace. Their results are mostly consistent across third party requests and domains, and cookies. Privacy Badger [25], Blur [23], Disconnect [21] and AdBlock Plus with the EasyList block a relatively large number of third party HTTP requests (between 33% and 48%), but a fairly small number of third party domains (between 8% and 33%). We hypothesize that these techniques block the main trackers but fail to impact the tracking long-tail [14]. MyTrackingChoices performances are worse than untrained Privacy Badger. We hypothesize that this is due to a GUI bug that forbids users from customizing website categories where blocking occurs. MyTrackingChoices thus always block tracking on 13 websites categories out of 32. BeefTaco has a noticeable impact on cookies but no impact on other metrics. This is consistent with previously observed tracking long tail [14]. This phenomenon makes the opt-out cookie maintenance very difficult while the protection that they offer is limited to cookie blocking. NoTrace [22] provides average protection despite being setup at its highest level. NoTrace increases users' online privacy awareness [71], but unfortunately, users cannot reliably setup the extensions [31] which is required.

Remaining techniques provide poor protection. Decentraleyes [29] has almost no effect on the results. We hypothesize that blocking library loading requests only marginally impacts HTTP traffic. The DoNotTrack HTTP header [4] also has barely no effect. If trackers complied with DoNotTrack, the number of cookies would significantly drop. We thus conclude that DoNotTrack is not respected by most trackers. WebOfTrust and HTTPSEverywhere have no effect.

We note that WebOfTrust and NoTrace both yield significantly more third party requests than Firefox without any extensions. WebOfTrust also contacts more third party domains and increases the number of cookies. WebOfTrust annotate hyperlinks in webpages with trust rating. We hypothesize that this rating is obtained from WebOfTrust servers and thus impact our metrics. We do not have any explanation regarding NoTrace's behavior.

When we block third party cookies in Firefox, there is a reduction in the number of third party domains. Blocking cookies also slightly reduces the number of third party requests. As hypothesized in [14], this may be due to impeded cookie-syncing interactions. However, contrary to Englehardt et al. [14], third party cookie blocking here has poor performances. This may be due to the fact that, unlike [14], we perform a stateful crawl. In a previous study [58] that uses stateful crawl and OpenWPM, third party cookie blocking also exhibits poor performances. The metric used in this study is however extremely specific, it thus is very difficult to compare our results with theirs. Furthermore, some third parties from the Alexa Top 1000 are first party at some point (e.g. Twitter, Facebook). Their cookies are thus created as first parties.

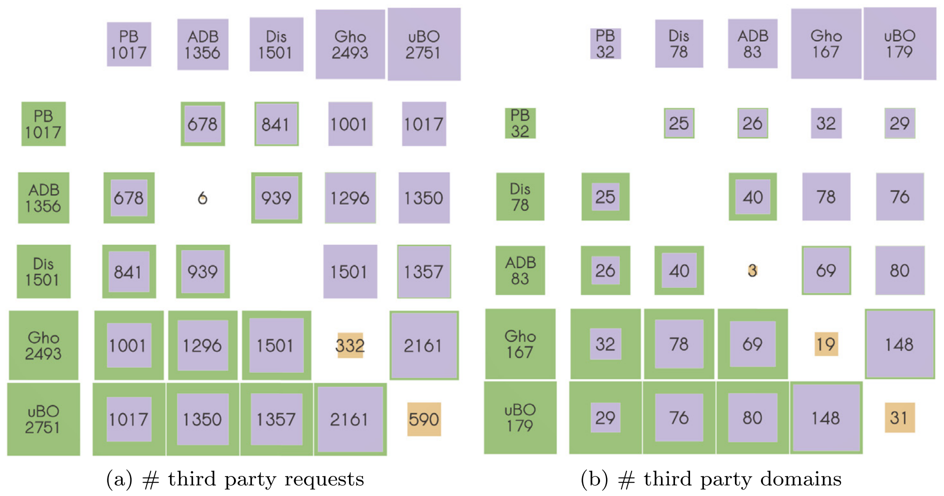


Fig. 7. Overlap of third party requests and domains blocked by five extensions (PB: Privacy Badger, Dis: Disconnect, ADB: AdBlock Plus, Gho: Ghostery, uBO: uBlock Origin). Square surfaces represent the metric value for the extension on the row.

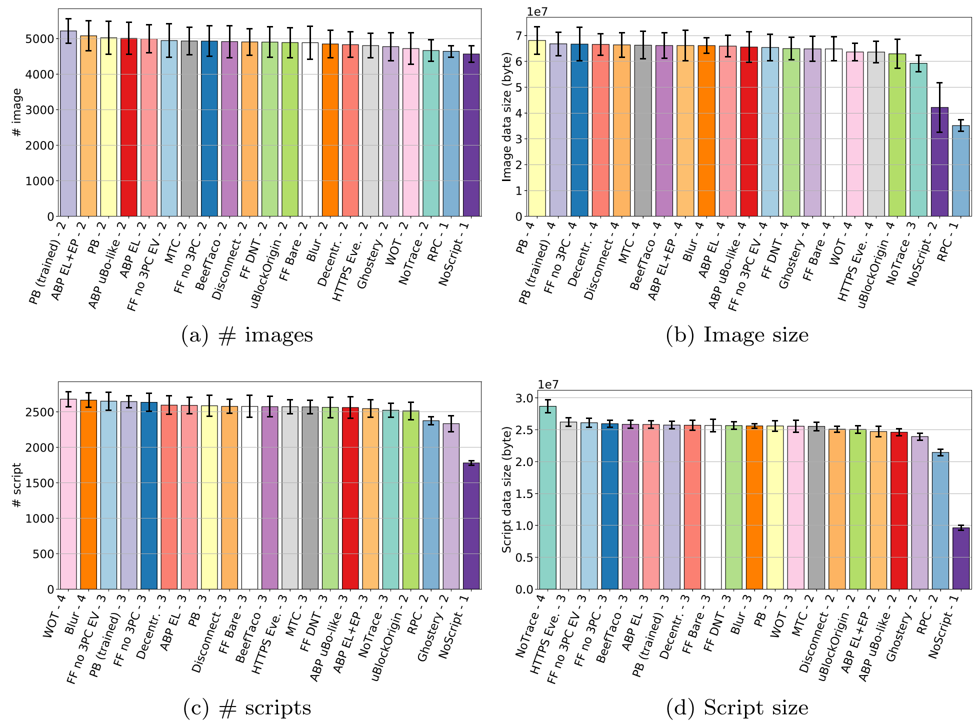


Fig. 8. Comparison of protection techniques' website quality regarding: number of image and scripts and size of images and scripts. We crawled the Alexa Top 100 websites. The mean and standard deviation are computed on ten crawls. The number on top of technique name is the KS-based rank. Bare: Firefox alone, FF: Firefox feature, MTC: MyTrackingChoices, PB (trained): Privacy Badger (trained), FF no 3rd coo. EV: Firefox no 3rd party cookie blocking except from previously visited domains, ABP: AdBlockPlus, EL: EasyList, EP: EasyPrivacy, RPC: Request Policy Continued. AdBlock Plus uBo-like uses all uBlock Origin's lists except uBlock Origin's specific ones; it thus loads EasyList, EasyPrivacy, Peter Lowe's Ad Server list and Malware domains.

6.1.2. Privacy footprint

We then compared extensions using the privacy footprint (see Section 4.2.3 and [13]). Fig. 6 presents these results. They are similar to browsing metrics' Fig. 5. Six extensions exhibit much better results than the rest: RequestPolicy Continued, NoScript, Ghostery, uBlock Origin and AdBlock Plus with uBlock Origin's list. For RequestPolicy Continued, the mean number of third parties per first party is very small while the total number of third parties is much higher. This means that some third parties are detected but that they are present on a limited set of first parties. NoScript results are here close to Ghostery's. The hierarchy between Ghostery, uBlock Origin and the customized AdBlock Plus is here very clear and follows this order. We hypothesized that the uniqueness of Ghostery and uBlock Origin blocking lists (see Section 6.1.3) is the main factor of their superior performances.

6.1.3. Overlap

Next, we measure the overlap between the five extensions among the most popular ones regarding the number of blocked third party requests and blocked third party domains using a matrix view. We here data crawled on the Alexa Top 100 websites in November 2016. We do not analyze NoScript [26] (resp. RequestPolicy Continued [27]) here because it indiscriminately blocks Javascript (resp. third party requests). Privacy Badger has been trained on the Alexa Top 100 websites and each extension is used with its default blocking list. AdBlock

Plus uses the EasyList. uBlock Origin loads EasyList, EasyPrivacy, Peter Lowe's Ad Server list, Malware domains, and some uBlock specific lists. Ghostery is set up with all its filter categories, while Disconnect is employed with its own filters.

Let us consider two extensions. Let E_i be the extension on the row i, and E_j the extension on the column j. The resources blocked by E_i (resp. E_j) are noted B_i (resp. B_j). On the left (resp. top), the green (resp. purple) square on row i (resp. column j) represents the total number of resource blocked by E_i (resp. E_j). On each line i of the matrix, the surface of the square represents the total number of resources blocked by E_i: $|B_i|$. The inner purple square on row i and column j represents the intersection between E_i and E_j and its surface is $|B_i \cap B_j|$. The diagonal orange square surface displays resources that are only blocked by E_i: $|B_i - \{B_k \forall k \neq i\}|$.

As an example, we see AdBlock Plus on the second line of Fig. 7(a). Square surfaces on this line represent the number of third party requests blocked by AdBlock Plus: 1356. The first square on the considered line represents the intersection with Privacy Badger which is encoded by the surface of the inner purple square, here 678. The green surface outside the purple square describes the resources blocked by AdBlock Plus but not by Privacy Badger. The next square on this line is located on the diagonal, and represents the resources blocked by AdBlock Plus only. Remaining squares on this line depicts intersections with Disconnect, Ghostery and uBlock Origin in the same fashion as the intersection with Privacy badger is represented (see above).

The overlap of the third party requests (Fig. 7(a)) shows that there is a big overlap across all extensions. All third party requests blocked by Privacy Badger are also blocked by uBlock Origin. Similarly, Ghostery almost completely covers Privacy Badger. Despite using two different blocking techniques, lists and heuristics, these extensions exhibit a significant overlap. This proves that the Privacy Badger's heuristics are reliable.

Results on the third party domains on Fig. 7(b) are similar to Fig. 7(a). The major difference is that Privacy Badger's impact is smaller. This is consistent with the heuristics behavior (see Section 5.2) that only block third party domains that are seen across several websites. These heuristics thus have trouble blocking less prominent trackers [14].

Both figures illustrate that Ghostery and uBlock Origin block many specific resources that other extensions do not block. We speculate that this is due the fact that their blocking list settings are unique. This may explain why they exhibit the best performance (see Section 6.1.1).

Some resources are only blocked by AdBlock Plus. It however uses EasyList which is also employed by uBlock Origin. The ten crawls made with AdBlock Plus may have reached some third parties that were not contacted during the ten measurements made with uBlock Origin. We thus hypothesize that this observation is another side-effect of the ad churn observed by Guha et al. [75].

Summary

The most popular extensions show a wide overlap. Ghostery and uBlock Origin block specific resources that are not affected by other extensions. In terms of overall privacy protection, RequestPolicyContinued and NoScript show the best performances. Ghostery and uBlock Origin protect users slightly less. Remaining techniques provide average to low protection. The DoNotTrack HTTP header provides almost no protection.

6.2. Webpage quality

We finally analyze how privacy protection techniques impact website quality. We use an HTML-based automated approach and perform a manual analysis.

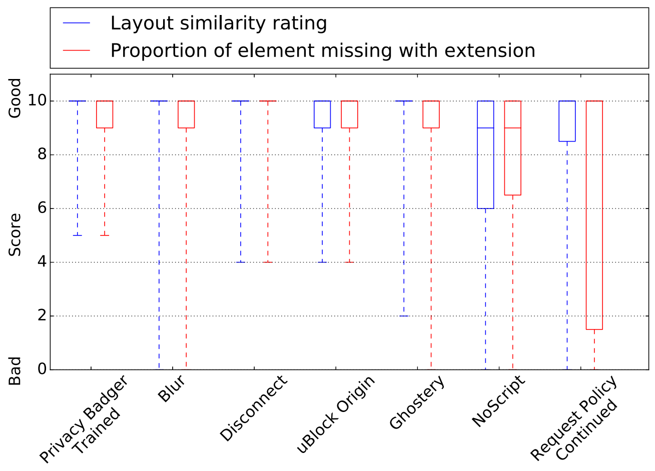


Fig. 9. Webpage quality manual analysis. Users are presented two screenshots obtained with Firefox alone and with a privacy protection technique. The first question (blue) asks user to rate layout similarity. The second question (red) corresponds to the proportion of elements observed with Firefox alone also present when a privacy protection technique is used. Screenshots are captured on the Alexa Top 50. Boxplots extend to the upper and lower quartile and contain a line that represents the median. Dashed lines show min/max range. When not visible, median is located on top of the upper quartile. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6.2.1. *HTML metrics*

We crawl the Alexa Top 100 world on the 21 different browser configurations of Section 6.1.1. We use the metrics presented in Section 4.3.1 and the ranking method exposed in Section 4.2.2. Fig. 8 presents the results. We discard the HTML size metric because there is no significant change across protection techniques. NoScript has a noticeable impact on all metrics. This is especially visible for the script number, and image and script size. RequestPolicy Continued's impact is smaller than NoScript's except for image size. We hypothesize that RequestPolicy Continued blocks third party image hosting services, and thus impacts many websites. NoScript greatly reduces the total size of scripts which is consistent with its goal: blocking JavaScript on webpages. Other privacy protection techniques appear to have a limited impact of website quality. Compared to other techniques, Ghostery exhibits a smaller impact on images and a higher reduction of the number of script and their total size. This is consistent with its tracking protection goal. This extension actually does not intend to block media such as advertisements.

6.2.2. *Manual analysis*

The previous subsubsection analyzes webpages quality in terms of HTML and associated data. This automated approach does not necessarily reflect the quality of the rendered webpage. We perform a manual analysis to cope with this limitation. We gather screenshots of the Alexa Top 50 without pages from the same entity but with distinct localization (e.g. Google and Amazon). These screenshots were gathered in November 2016. We use the seven techniques that provided the best privacy protection (see Section 6.1.1): Privacy Badger trained, Blur, Disconnect, uBlock Origin, Ghostery, NoScript and RequestPolicy Continued. For each webpage and protection technique, we display a picture that contains two screenshots side-to-side to the user: the left one captured with Firefox alone, the other with Firefox used with an extension. We use the questions about layout similarity and missing elements presented in Section 4.3.2. Fig. 9 exposes our results. The blue (resp. red) boxplot corresponds to the first (resp. second) question. Overall, Privacy Badger, Blur, Disconnect, uBlockOrigin and Ghostery have a very small impact on rendered webpages. On the other hand, RequestPolicy Continued and NoScript significantly impact on pages both in terms of layout and missing elements. This is consistent with

Section 6.2.1 and previous results [5,69]. The manual analysis uncovers RequestPolicy Continued's strong impact on webpage quality which was not exposed by HTML-based metrics.

Summary

Both studies show that RequestPolicyContinued and NoScript reduce webpage quality. Other extensions do not have such impact.

6.3. Summary

Fig. 10 summarizes our findings. Each metric-based synthetic index is build as the mean of previously used metrics (see Sections 4.2.1 and 4.3.1) normalized using the metric value obtained with Firefox used alone. Manual analysis-based index is the mean of the mean rating of both questions. Fig. 10(a) shows that our manual analysis captures the negative effects of RequestPolicy Continued that are not visible with HTML-based metrics. Figs. 10(b) and 10(c) demonstrate RequestPolicy Continued and NoScript show the best protection performances but impact browsing experience. Ghostery and uBlock Origin protect users slightly less but have a smaller impact on webpage quality. Remaining techniques provide average protection.

7. Discussions

7.1. Recommendations

Our analysis shows that RequestPolicy Continued and Noscript are the most effective privacy protection techniques. Similarly to previous work [5,69], we confirm that they affect webpage quality. Users that want to avoid website breakage can instead use uBlock Origin or Ghostery. The latter however needs to be configured which is difficult for users [31]. We note that previous work [11,12] also recommended uBlock Origin and Ghostery. Their results were however noisy (percentiles are overlapping in [12] and some techniques actually observed more fingerprinting than the default browser in [11]) due to the use of single crawl to each website. Our robust measurement methodology does not suffer from the same weakness, and show that uBlock Origin or Ghostery exhibit statistically better performance than other techniques.

Previous network traffic monitoring studies [67,77] show that users are more preoccupied by blocking ads than tracking. Malloy et al. [78] show that ad-blockers usage vary between 18% and 37% in analyzed countries (US, UK, Germany, France, and Canada). Metwalley et al. show [67] that 10 to 18% of users had installed AdBlock Plus while less than 3.5% (resp. 2%) were using DoNotTrackMe/Blur (resp. Ghostery). Similarly, Pujol et al. [77] speculate that out of the 19.7% of users that install AdBlock Plus, less than 15% of them setup the EasyPrivacy list. We show that the default setup of AdBlock Plus has average performances but that it can be configured to achieve good results. This task is however difficult for users [31]. We thus advise ad-blocking users that want to improve their privacy to directly change to more efficient extensions (see above).

Extensions that do not rely on manually maintained filters, Privacy Badger and MyTrackingChoices, exhibit average performances. It however is not clear how their performances may be improved. Analyzing extensions' behavior against existing blocking lists would help to improve their results. Blocking list-based protection techniques are efficient but their scalability is limited due to human intervention. Current efforts [59–62,64] to automatically build blocking list are thus extremely relevant.

Ghostery and uBlock Origin block a significant amount of resources that are not blocked by other protection techniques. Both should thus be considered as relevant when assessing users' ability to protect themselves.

7.2. Limitations

In this work, we estimate the impact of privacy protection techniques on webpage quality in terms of missing elements and data (using HTML-based metrics Section 4.3.1 and alteration to the webpage layout and elements (using a user manual analysis Section 4.3.2). We however do not address the functionality loss. We intend to explore this aspect in future work along two axes. We plan to explore each website beyond its homepage to improve website coverage.

Several work [46,59–62,64] automatically build blocking lists. We did not evaluate them because produced lists are not publicly available.

Evaluated approaches aim at blocking tracking. One side effect is advertisement blocking. These approaches thus threaten the current economic model of many publishers on the Internet. Techniques examined in this work however do not have the same impact on third party tracking. For example, Ghostery and MyTrackingChoices can block tracking for specific website categories. Similarly, Firefox's third party cookie blocking only partially impacts advertisement techniques (such as Real-Time Bidding, RTB) that rely on user interests. Olejnik et al. [57] show that new users (i.e. users without any browsing history) are less valued by advertisers. A user that blocks all third parties cookies thus reduces his customer value for advertisers, and, consequently, publisher's revenue. This however does not completely block tracking and thus, advertisement transactions (e.g. bidding in RTB). Indiscriminate third party blocking techniques (such as RequestPolicy Continued [27]) however have a much bigger impact on advertising. They for example block interactions between ad exchange and bidders in the case of RTB. All privacy protection techniques presented in this work may thus have very different impact on advertising. We intend to address this aspect in future work.

Ghostery [20] and MyTrackingChoices [24] can block tracking for specific website categories, and thus, allow monetization for others. This is obviously less aggressive than most approaches addressed in this work. Achara et al. [24] however report that 30% of users block the tracking for all categories. These users thus have no reason to use these two techniques instead of other ones.

7.3. Future work

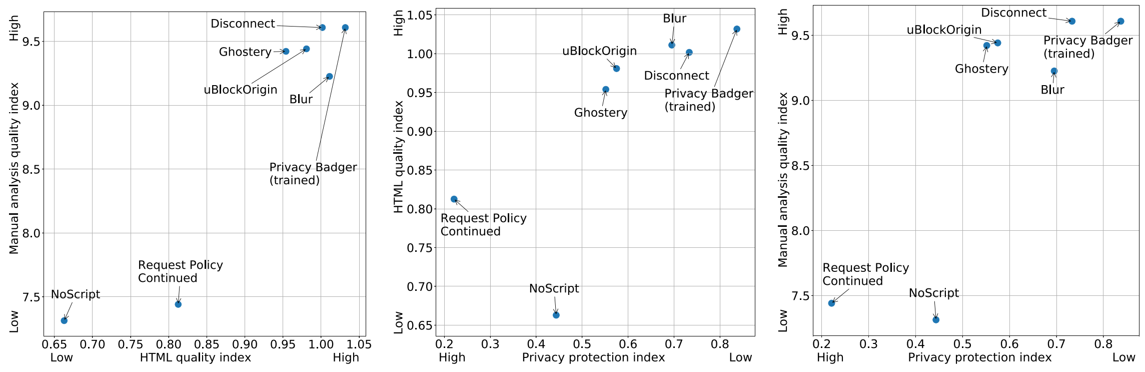
As shown in Section 3, existing work uses a vast array of metrics. Some of them are tailored to very specific use cases (such as behavioral advertising [7,66] and cookie-based mass-surveillance [58]). Similarly, data-related metrics present an obvious interest in a mobile context. We intend to add new metrics to this work. We also want to analyze the impact of all these protection techniques on specific attacks (browser fingerprinting [3,14], cookie syncing [1,14,57,79]).

Analyzing extensions inside other browsers, such as Chrome, Internet Explorer or Opera, would provide a wider picture of privacy protection techniques in the wild. Selenium, the browser automation framework used by OpenWPM, supports these browsers. OpenWPM, however, currently only supports Firefox. Furthermore, browsers include more and more privacy protection (e.g. Firefox [42], Brave [43], Cliqz [70]). Cliqz and Brave are not supported by Selenium, which make both browsers impossible to use with OpenWPM (see Section 4.1). We however intend to add them later on.

While some works use domain name-based heuristics [57,80] to identify third parties, others use blocking lists-based extensions as ground-truth (see Table 2). When the latter method is used, tracking's long tail [14] may cause many false negatives. Beyond our analysis in Section 6.1, we intend to investigate blocking lists more thoroughly.

8. Conclusions

This work extensively compares existing privacy protection techniques against third party tracking and provides four contributions.



(a) HTML-based vs manual analysis quality

(b) Privacy protection vs HTML-based quality

(c) Privacy protection vs manual analysis

Fig. 10. Scatter plots of synthetic index for privacy protection, HTML-based quality and manual analysis-based quality.

First, we propose a robust methodology to compare privacy protection techniques when crawling many websites, and quantify measurement error. We use the privacy footprint [13] and apply the Kolmogorov–Smirnov (KS) test on browsing metrics to evaluate privacy protection. This test is likewise applied to HTML-based metrics to assess the impact on webpage quality. We also design a manual analysis to complement HTML-based metrics. Our second contribution is a privacy protection techniques comparison in terms of overlap and performances. We show that the most popular privacy protection techniques exhibit a common blocking baseline. We also highlight the fact that some extensions, namely Ghostery and uBlock Origin, have a very specific behavior and are the only ones to block some domains. Our third contribution is a comparison of the impact of privacy protection techniques on webpage quality. Our fourth contribution is a set of usage recommendations for end-users, and research recommendation for the scientific community. Ghostery and uBlock Origin provide the best trade-off between protection and webpage quality. Ghostery however requires a configuration step which is difficult for users [31]. The RequestPolicy Continued and NoScript extensions exhibit the best performances but reduce webpage quality. Ghostery and uBlock Origin use manually built blocking lists which are cumbersome to maintain. Research efforts should focus on improving existing approaches that do not rely on blocking lists (such as Privacy Badger or MyTrackingChoices) and automating blocking list building.

Acknowledgment

This research has been supported by the Japanese Society for Promotion of Science under grant 15H02699.

References

[1] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, C. Diaz, The web never forgets: Persistent tracking mechanisms in the wild, in: CCS 2014, pp. 674–689.

[2] A. Soltani, S. Canty, Q. Mayo, L. Thomas, C.J. Hoofnagle, Flash cookies and privacy, in: SSRN, 2009, pp. 158–163.

[3] P. Eckersley, How unique is your web browser?, in: PETS 2010, pp. 1–18.

[4] DoNotTrack, 2017, URL https://tools.ietf.org/html/draft-mayer-do-not-track-00. (Accessed 30 July 2017).

[5] B. Krishnamurthy, D. Malandrino, C.E. Wills, Measuring privacy loss and the impact of privacy protection in web browsing, in: SOUPS 2007, pp. 52–63.

[6] J.R. Mayer, J.C. Mitchell, Third-party web tracking: Policy and technology, in: SP 2012, pp. 413–427.

[7] R. Balebako, P. Leon, R. Shay, B. Ur, Y. Wang, L. Cranor, Measuring the effectiveness of privacy tools for limiting behavioral advertising, in: W2SP 2012.

[8] R. Hill, Comparative benchmarks against widely used blockers: Top 15 most popular news websites, GitHub Reposit. (June) (2014) URL https://github.com/gorhill/httpswitchboard/wiki/Comparative-benchmarks-against-widely-used-blockers:-Top-15-Most-Popular-News-Websites. (Accessed 30 July 2017).

[9] R. Hill, Ublock and others: Blocking ads, trackers, malwares, GitHub Reposit. (May) (2015) URL https://github.com/gorhill/uBlock/wiki/uBlock-and-others%3A-Blocking-ads%2C-trackers%2C-malwares. (Accessed 30 July 2017).

[10] C.E. Wills, D.C. Uzunoglu, What ad blockers are (and are not) doing, in: HotWeb 2016, pp. 72–77.

[11] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, E. Weippl, Block me if you can: A large-scale study of tracker-blocking tools, in: EuroSP 2017.

[12] S. Traverso, M. Trevisan, L. Giannantoni, M. Mellia†, H. Metwalley, Benchmark and comparison of tracker-blockers:should you trust them?, in: TMA 2017.

[13] B. Krishnamurthy, C.E. Wills, Generating a privacy footprint on the internet, in: IMC 2006, pp. 65–70.

[14] S. Englehardt, A. Narayanan, Online tracking: A 1-million-site measurement and analysis, in: CCS 2016.

[15] T. Bujlow, V. Carela-Español, J. Solé-Pareta, P. Barlet-Ros, A survey on web tracking: Mechanisms, implications, and defenses, Proc. IEEE PP (99) (2017) 1–35.

[16] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, J. Forné, Online advertising: Analysis of privacy threats and protection approaches, Comput. Commun. (2016).

[17] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, Cookieless monster: Exploring the ecosystem of web-based device fingerprinting, in: SP 2013, pp. 541–555.

[18] AdBlock Plus, 2017, URL https://adblockplus.org/. (Accessed 30 July 2017).

[19] uBlock Origin, 2017, URL https://github.com/gorhill/uBlock. (Accessed 30 July 2017).

[20] Ghostery, 2017, URL https://www.ghostery.com/. (Accessed 30 July 2017).

[21] Disconnect, 2017, URL https://disconnect.me/. (Accessed 30 July 2017).

[22] NoTrace, 2017, URL http://www.isislab.it/projects/NoTrace/. (Accessed 30 July 2017).

[23] Blur, 2017, URL https://dnt.abine.com. (Accessed 30 July 2017).

[24] J.P. Achara, J. Parra-Arnau, C. Castelluccia, MyTrackingChoices: Pacifying the ad-block war by enforcing user privacy preferences, in: WEIS 2016.

[25] Privacy Badger, 2017, URL https://www.eff.org/fr/privacybadger. (Accessed 30 July 2017).

[26] NoScript, 2017, URL https://noscript.net/. (Accessed 30 July 2017).

[27] RequestPolicyContinued, 2017, URL https://requestpolicycontinued.github.io/. (Accessed 30 July 2017).

[28] HTTPS Everywhere, 2017, URL https://www.eff.org/fr/https-everywhere. (Accessed 30 July 2017).

[29] Decentraleyes, 2017, URL https://decentraleyes.org. (Accessed 30 July 2017).

[30] WebOfTrust, 2017, URL https://www.mywot.com/. (Accessed 30 July 2017).

[31] P. Leon, B. Ur, R. Shay, Y. Wang, R. Balebako, L. Cranor, Why Johnny Can't Opt out: A usability evaluation of tools to limit online behavioral advertising, in: CHI 2012, pp. 589–598.

[32] Firefox privacy and security extensions, 2017, URL https://addons.mozilla.org/fr/firefox/search/?category=privacy-security&sort=users&type=extension. (Accessed 30 July 2017).

[33] J. Vincent, Ghostery has been bought by the developer of a privacy-focused browser, 2017, URL http://www.theverge.com/2017/2/15/14622484/ghostery-ad-tracking-plug-in-cliqz. (Accessed 30 July 2017).

[34] Network advertising industry, 2017, URL http://www.networkadvertising.org/choices/. (Accessed 30 July 2017).

[35] Digital advertising alliance, 2017, URL http://www.aboutads.info/choices/. (Accessed 30 July 2017).

[36] BeefTaco, 2017, URL https://jmhobbs.github.io/beef-taco/. (Accessed 30 July 2017).

[37] Allowing acceptable ads in adblock plus - Agreements, 2017, URL https://adblockplus.org/acceptable-ads-agreements. (Accessed 30 July 2017).
[38] EasyList/EasyPrivacy, 2017, URL https://easylist.to/. (Accessed 30 July 2017).
[39] V.S. Eckert, J. Klofta, J.L. Strozyk, "Web of Trust" späht Nutzer aus, 2016, URL https://www.tagesschau.de/inland/tracker-online-103.html. (Accessed 30 July 2017).
[40] L. Olejnik, Proposed amendments to ePrivacy Regulation are great, 2017, URL https://blog.lukaszolejnik.com/proposed-amendments-to-eprivacy-regulation-are-great. (Accessed 30 July 2017).
[41] Firefox, 2017, URL https://www.mozilla.org/en-US/firefox. (Accessed 30 July 2017).
[42] G. Kontaxis, M. Chew, Tracking protection in firefox for privacy and performance, in: W2SP 2015.
[43] Brave, 2017, URL https://brave.com/. (Accessed 30 July 2017).
[44] Cliqz, 2017, URL https://cliqz.com/en (Accessed 30 July 2017).
[45] B. Krishnamurthy, C. Wills, Privacy diffusion on the web: a longitudinal perspective, in: WWW 2009, pp. 541–550.
[46] F. Roesner, T. Kohno, D. Wetherall, Detecting and defending against third-party tracking on the web, in: NSDI 2012, pp. 12–26.
[47] C. Castelluccia, S. Grumbach, L. Olejnik, Data harvesting 2.0: from the visible to the invisible web, in: WEIS 2013.
[48] N. Fruchter, H. Miao, S. Stevenson, R. Balebako, Variations in tracking in relation to geographic location, in: W2SP 2015.
[49] D. Fifield, S. Egelman, Fingerprinting web users through font metrics, in: FC 2015, pp. 107–124.
[50] Ł. Olejnik, G. Acar, C. Castelluccia, C. Diaz, The leaking battery, in: DPM 2015, pp. 254–263.
[51] K. Mowery, H. Shacham, Pixel perfect: Fingerprinting canvas in HTML5, in: W2SP 2012.
[52] Y. Cao, S. Li, E. Wijmans, (Cross-)browser fingerprinting via OS and hardware level features, in: NDSS 2017.
[53] O. Starov, N. Nikiforakis, XHOUND: Quantifying the fingerprintability of browser extensions, in: SP 2017, pp. 941–956.
[54] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, B. Preneel, FPDetective: dusting the web for fingerprinters, in: CCS 2013, pp. 1129–1140.
[55] A. Lerner, A.K. Simpson, T. Kohno, F. Roesner, Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016, in: Usenix Security 2016, pp. 997–1013.
[56] A. Ghosh, A. Roth, Selling privacy at auction, in: EC 2011, pp. 199–208.
[57] L. Olejnik, T. Minh-Dung, C. Castelluccia, Selling off privacy at auction, in: NDSS 2014, pp. 104–114.
[58] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan, E.W. Felten, Cookies that give you away: The surveillance implications of web tracking, in: WWW 2015, pp. 289–299.
[59] H. Metwalley, S. Traverso, M. Mellia, Unsupervised detection of web trackers, in: GLOBECOM 2015, pp. 1–6.
[60] J. Bau, J. Mayer, H. Paskov, J.C. Mitchell, A promising direction for web tracking countermeasures, in: W2SP 2013.
[61] Q. Wu, Q. Liu, Y. Zhang, G. Wen, Trackerdetector: A system to detect third-party trackers through machine learning, Comput. Netw. 91 (2015) 164–173.
[62] M. Ikram, H.J. Asghar, M.A. Kaafar, A. Mahanti, B. Krishnamurthy, Towards seamless tracking-free web: Improved detection of trackers via one-class learning, in: PETS 2017.
[63] V. Kalavri, J. Blackburn, M. Varvello, K. Papagiannaki, Like a pack of wolves: Community structure of web trackers, in: PAM 2017, pp. 42–54.
[64] D. Gugelmann, M. Happe, B. Ager, V. Lenders, An automated approach for complementing ad blockers' blacklists, in: PETS 2015, pp. 282–298.
[65] AdBlock, 2017, URL https://getadblock.com/. (Accessed 30 July 2017).
[66] J.M. Carrascosa, J. Mikians, R. Cuevas, V. Erramilli, N. Laoutaris, I always feel like somebody's watching me. Measuring online behavioural advertising, in: CoNext 2015.
[67] H. Metwalley, S. Traverso, M. Mellia, S. Miskovic, M. Baldi, The online tracking horde: a view from passive measurements, in: TMA 2015, pp. 111–125.
[68] D. Malandrino, A. Petta, V. Scarano, L. Serra, R. Spinelli, B. Krishnamurthy, Privacy awareness about information leakage: Who knows what about me?, in: WPES 2013, pp. 279–284.
[69] D. Malandrino, V. Scarano, Privacy leakage on the web: Diffusion and countermeasures, Comput. Netw. 57 (14) (2013) 2833–2855.
[70] Z. Yu, S. Macbeth, K. Modi, J.M. Pujol, Tracking the trackers, in: WWW 2016, pp. 121–132.
[71] D. Malandrino, V. Scarano, R. Spinelli, How increased awareness can impact attitudes and behaviors toward online privacy protection, in: SocialCom 2013, pp. 57–62.
[72] Alexa top 500 sites on the web, 2017, URL http://www.alexa.com. (Accessed 30 July 2017).
[73] Public suffix list, 2017, URL https://publicsuffix.org/. (Accessed 30 July 2017).
[74] R.A. Fisher, Statistical methods for research workers, 1925.
[75] S. Guha, B. Cheng, P. Francis, Challenges in measuring online advertising systems, in: IMC 2010, pp. 81–87.
[76] G. Pass, A. Chowdhury, C. Torgeson, A picture of search, InfoScale 152 (2006) 1.
[77] E. Pujol, O. Hohlfeld, A. Feldmann, Annoyed users: Ads and ad-block usage in the wild, in: IMC 2016, pp. 93–106.
[78] M. Malloy, M. McNamara, A. Cahn, P. Barford, Ad blockers: Global prevalence and impact, in: IMC 2016, pp. 119–125.
[79] M. Falahrastegar, H. Haddadi, S. Uhlig, R. Mortier, Tracking personal identifiers across the web, in: PAM 2016, pp. 30–41.
[80] M. Falahrastegar, H. Haddadi, S. Uhlig, R. Mortier, The rise of panopticons: examining region-specific third-party web tracking, in: TMA 2014, pp. 104–114.