PrivacyMeter: Designing and Developing a Privacy-Preserving Browser Extension

Oleksii Starov and Nick Nikiforakis

Stony Brook University, Stony Brook NY 11794, USA,
ostarov@cs.stonybrook.edu, nick@cs.stonybrook.edu

Abstract. Anti-tracking browser extensions are popular among web users since they provide them with the ability to limit the number of trackers who get to learn about their browsing habits. These extensions however are limited in that they ignore other privacy signals, such as, the presence of a privacy policy, use of HTTPS, or presence of insecure web forms that can leak PII. To effectively inform users about the privacy consequences of visiting particular websites, we design, implement, and evaluate PrivacyMeter, a browser extension that, on-the-fly, computes a relative privacy score for any website that a user is visiting. This score is computed based on each website's privacy practices and how these compare to the privacy practices of other pre-analyzed websites. We report on the development of PrivacyMeter with respect to the requirements for coverage of privacy practices, accuracy of measurement, and low performance overhead. We show how relative privacy scores help in interpreting results as different categories of websites have different standards across the monitored privacy parameters. Finally, we discuss the power of crowdsourcing for privacy research, and the existing challenges of properly incorporating crowdsourcing in a way that protects user anonymity while allowing the service to defend against malicious clients.

1 Introduction

The modern web is home to many online services that request and handle sensitive private information from their users. For example, most of the popular websites require personal information to create an account, including one's email address, name, and date of birth, or even ask users to provide similar information in order to just submit a contact form. Unfortunately, this sensitive data is not always collected and handled in the most confidential and secure way possible. Previous research has shown how websites may leak user information, either due to poor programming practices [6, 13, 18, 26], or through the intentional outsourcing of functionality to third-party services [6, 26]. One of the most intrusive scenarios is the leakage of a user's personally identifiable information (PII) to web trackers, which can then match it with existing pseudonymous user identifiers (such as cookies and browser/device fingerprints) and thus deanonymize the user across browsing sessions and websites.

Despite the magnitude of this problem, users today have few, if any, options, for protecting their PII against accidental and intentional leakage. Generic anti-tracking extensions, such as Ghostery [9], Disconnect [8], and uBlock Origin [29] or other ad blockers [4, 5, 28], operate solely using manually-curated blacklists which, due to their

reactive nature, are destined to be always out of date [19]. Moreover, these anti-tracking extensions only account for domains belonging to tracking companies and thus cannot account for non-tracking-related third-party domains which happen to receive a user's PII due to the poor programming practices of the first-party website with which the user interacts. In addition, current anti-tracking extensions are rather myopic in the sense that they only care about the third-party trackers available on a webpage, but ignore other signals (such as the presence of a privacy policy, or the use of HTTPS) which are directly correlated with the confidentiality of user data. Finally, next to the lack of technology, websites are becoming more and more hostile towards any kind of client-side tracking-blocking tools forcing users to whitelist them before they can get access to their content [11, 12, 20], taking away what little control users had regained.

As such, the sole focus on blocking trackers may not be sufficient to preserve the privacy of web users, and more attention should be given to improve their overall awareness about the privacy-related consequences of visiting any given website. For example, in addition to a list of detected trackers, a privacy-preserving browser extension can convey to non-technical users, which of those trackers pose a serious threat to their privacy — and therefore must be blocked — and which are reliable and can be allowed. Similarly, such a tool should take into account and evaluate as many additional privacy practices on a website as possible, in order to provide user with a complete picture of the expected privacy guarantees.

To effectively inform users about the privacy consequences of visiting particular websites, we propose, design, implement, and evaluate PrivacyMeter, a browser extension that, on-the-fly, computes a relative privacy score for any website that a user is visiting. This score is computed based on each website's privacy practices (e.g., reputation of trackers, amount of third-party content or presence of insecure "leaky" web forms) and how these compare to the privacy practices of other pre-analyzed websites. In addition to the overall risk score, PrivacyMeter also provides users with contextual information about the discovered privacy issues (e.g., "many aggressive trackers", or "many inputs are submitted to third parties"), and what actions are advised. This is a clear departure from virtually all other available browser extensions which merely inform the user about the total number of trackers on any given page and assume that the user is somehow capable of using this information in a constructive manner.

In this paper we provide details on the design and development of PrivacyMeter, including the covered privacy practices and encountered implementation challenges, as well as additional envisioned features, which may be implemented in later versions. Development of such a privacy-preserving browser extension is not trivial as each privacy-related feature must be reliably and accurately retrieved, the overall privacy scores calculated in the same fashion over all of the websites in order to guarantee their comparability, and the results must be presented on time with low performance overhead. At the same time, there is another level of challenges, which lie in the research and crowdsourcing nature of the tool.
The key contributions of this paper are as follows:

- We describe our case study of implementing a complex privacy-preserving browser extension, PrivacyMeter, which provides a real-time privacy quantification for the web, and has strong requirements for coverage, accuracy, and performance.

- We identify and group important privacy features of websites in order to design an informative user interface, which combines comparable privacy values with immediate unconditional warnings.
- We show how relative privacy score helps in improving the privacy awareness as different categories of websites have different privacy standards across the monitored privacy parameters.
- We measure the performance overhead from PrivacyMeter and compare it to Ghostery, a state-of-the-art commercial anti-tracking extension.
- We describe the benefits of crowdsourcing as a necessary feature both for the quality of PrivacyMeter, as well as for the further research benefits. At the same time, we discuss the challenges in adopting crowdsourcing in any privacy-preserving tool due to: (a) higher anonymity and transparency requirements when collecting the data; (b) potential of a polluted data from malicious clients.

2 Design and Interface

PrivacyMeter is developed as a browser extension which, upon installation, enhances each tab of a web browser with new privacy-related information. Whenever a user visits a new website, PrivacyMeter computes a numerical privacy score for that website, which is an overall count of detected privacy issues, and warns the user about privacy risks by changing the color of its icon from green to yellow or red depending on the severity of the identified risks. Each issue can be a *definite* privacy risk (e.g., if a web page does not use HTTPS or does not have a privacy policy), or a *potential* relative privacy risk (e.g., if a web page has 17% more third-party trackers than average for similar websites). PrivacyMeter augments, in a non-intrusive fashion, the UI of the browser to add the computed number of privacy issues to its icon in the right top corner of the URL bar, and gives users the ability to find the rationale behind this score as well as advice on how to proceed by clicking it.

Figure 1-a shows an example of the extension's popup with details on the discovered privacy-related issues at the `money.cnn.com` page. The interface consists of the two main parts: (1) box plots to show the relative privacy risks; (2) text warnings with definite privacy risks (or with additional details about relative risks). In both cases, we use the aforementioned three-color scheme to highlight the severity of the threat, where green is used as "safe", yellow as "potentially dangerous", and red as "dangerous" level. For instance, if the web page contains too many trackers (in comparison with other similar sites), the corresponding box plot will be colored with yellow or red. Similarly, if the page contains any number of low-reputation trackers, the corresponding message will be highlighted with red. PrivacyMeter also gives users the ability to get additional details, whether by hovering the mouse over the particular box plot, or by clicking on each text warning.

The main functionality of PrivacyMeter is to provide users with understandable privacy scores of the websites that they visit. To be usable, these scores must reflect the privacy practices of websites in such a way, that users intuitively understand and agree with them. As such, in order to evaluate the severity of a particular relative privacy issue we compare the value for the current web page to the mean value over

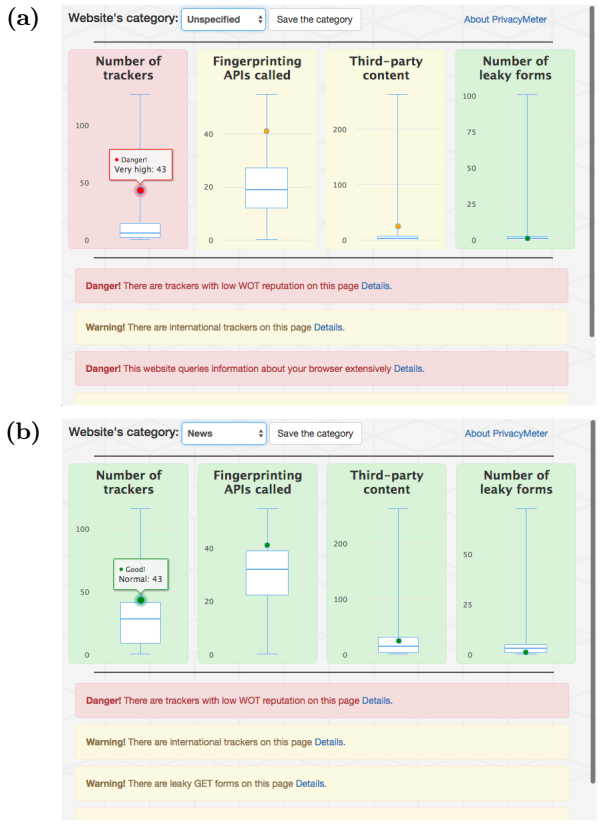

Fig. 1: *UI and information provided by* PrivacyMeter *on the example of visiting* money.cnn.com *page: (a) comparing privacy features to* any *other websites; (b) comparing privacy features to other* news *websites.*

other websites: if it is greater than the mean plus one standard deviation, we mark it as "potentially dangerous," and if it is greater than the mean plus two standard deviations, we mark it as a "dangerous" issue. The overall privacy score presented to a user is the number of all the privacy issues found on the page, which is colored with red if at least one of the issues falls into the "dangerous" category.

Despite our best efforts to score privacy issues in an objective way, we anticipate that a fraction of advanced users may not entirely agree with the default scoring thresholds or with the set of privacy features that comprise our privacy score. To maintain the usability of our tool, we allow users to remove some factors from the score calculation on the extension's setting page to better reflect their privacy preferences. Similarly, PrivacyMeter provides an option to to compare relative factors to median and the third quartile of a distribution instead of the mean-based comparison, as well as to change the base threat level (whether to relax it by counting considerable risks as safe, or to consider any number of trackers or leaky web forms as dangerous).

The text warnings assist users with the decision whether to limit (or stop) the browsing of a web page. For instance, when a leaky form is discovered, the corre-

sponding message advises user to refrain from trusting the website with their PII. We decided to use box plots in addition to the text warnings with the assumption that visual information is easier and quicker to understand. Particularly, we chose box plots for the type of plot as those clearly show how a single data point compares to the overall distribution of values. We argue that with a little training, even non-technical users are able to read such visualizations faster then text explanations, and develop an intuition about relative privacy risks.

As an additional functionality, PrivacyMeter's interface provides a control to select a category for the current website. In this case, all the relative privacy risks will be compared to other known web pages of only the selected category. For instance, Figure 1-b shows how the picture changes for `money.cnn.com` if we consider only news websites. Given that news sites tend to have more trackers than other types of websites (we quantify this in Section 3), the 43 trackers of `money.cnn.com` are not considered an outlier and therefore PrivacyMeter's warnings regarding the number of trackers disappear. We argue that this ability to compare with other similar websites helps web users to translate raw numbers into the privacy expectations, i.e., to immediately gauge whether their website of interest stands out in a positive (i.e. more private than average) or negative (i.e. less private than average) direction.

Monitored Privacy Practices

To provide a representative privacy score, PrivacyMeter must account for different privacy practices, web security vulnerabilities, and other factors on a website. As we explained above, factors which are *potential* privacy risks can be represented by comparative metrics, whereas factors which are *definite* risks are usually represented using binary values. PrivacyMeter deploys mechanisms to test a web page against the following four main groups of privacy risks (each group has a correspond box plot on the interface in Figure 1):

- **Third-party trackers.** To measure the potential privacy risk due to third-party trackers, we first calculate the comparative metric on how the number of trackers differs from the average number of trackers for similar websites. The result is visualized on the corresponding box plot. Second, we answer additional questions that may also correlate with the potential privacy risks, such as what is the number of trackers served from international servers. Similarly, the severity of the warning message depends on how this quantity deviates from the average but it is, by default, marked as "potentially dangerous" even if only one international tracker is present. Finally, an example of a definite privacy risk marked as "dangerous" is the presence of trackers with low reputation (according to the Web-of-Trust score [30]).
- **Fingerprinting activity.** Fingerprinting activity may correlate with intrusive tracking, either from an advanced third-party tracker or the first-party website. The privacy threat lies in the fact that fingerprinting is immune to the deletion of stateful identifiers (such as cookies) and thus harder to avoid. The main metric we calculate is a relative number of unique called APIs (as utilized by Lerner et al. [15]), which are known to be used for fingerprinting browsers and devices. The corresponding box plot visualizes the comparison with the sample distribution.

In addition, we show a separate warning for the cumulative number of API calls, which may correlate with continuous fingerprinting attempts.

- **Third-party content.** If a web page is populated with third-party content, a user may be easily tricked into interacting and sharing sensitive information with parties other than the ones expected. For instance, a website may load a large number of iframes with third-party ads completely overlaying the page, or it may incorporate a third-party form widget. PrivacyMeter answers the question on how the number of third-party widgets differs from the average for similar websites. In future versions, we can also warn about signs of visually overlapping iframes.
- **Leaky web forms.** PrivacyMeter has ability to identify web forms that may leak entered information, such as forms implemented with HTTP GET method and thus exposing the entered values in URL and HTTP Referer headers, forms submitting without HTTPS or forms directly submitting to a third-party domain. Next to warning messages about the presence of such forms, we provide a comparison on the number of leaky forms in the corresponding box plot. The rationale for that lies in the fact that some categories of websites may include numerous unsafe search forms, or other forms that do not necessary request PII or other sensitive private information. In our current version of PrivacyMeter, we assign a dangerous level to a leaky web form if it contains a large number of visible inputs (such as the ones used during account creation), or one or more sensitive fields (such as a password field).

In addition, PrivacyMeter collects features that can be used as "proxies" of the level of privacy and security awareness of the website owners. For example, PrivacyMeter checks whether a P3P privacy policy is available to the users. Even though we are fully aware that such a policy is not a guarantee of proper privacy measures, we argue that its absence is a strong negative indicator. We also detect signs of mixed HTTP/HTTPS inclusions, and out-of-date versions of web servers, which quantifies the extent to which website administrators update their software. The overall privacy score is calculated as a sum of issues found among these parameters, and users are also notified about each separate issue as shown in Figure 1. Note that PrivacyMeter is unique in that it goes above and beyond the measuring of standard privacy practices, by also measuring security issues which could be abused to compromise the confidentiality of a user's private information (such as a web server getting compromised and the user's data exfiltrated, because of the use of out-of-date software with known exploitable vulnerabilities).

Architectural Challenges

In the previous section, we provided a list of factors that PrivacyMeter uses to calculate the privacy score of a website. The extension part of PrivacyMeter is responsible for retrieving them using a combination of static and dynamic techniques, as those are allowed by modern browser-extension frameworks. Even though some of these factors can be straightforwardly detected, e.g., version of a web server or mixed HTTP/HTTPS requests, others could pose technical challenges. For instance,

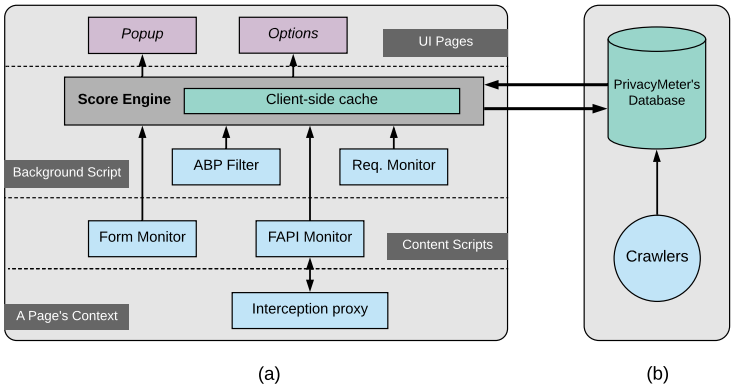

Fig. 2: PRIVACYMETER*'s architecture: (a) client-side modules inside browser extension; (b) server-side support with the central database and automatic crawler.*

a privacy policy may be present in a wide range of locations (making it hard to automatically locate) and trackers may use stealthy tracking techniques to avoid being easily detected. As such, the current architecture of PRIVACYMETER's extension, shown on Figure 2-a, already includes several modules running in the contexts of extension's background script, content scripts, as well as inside the web page context. In order to identify privacy risks, each module runs a set of tests on every newly visited web page, and analyzes every new web request.

Given the goal of building a user-friendly and reliable privacy-enhancing system, we can distinguish the following three challenges in the realization of the client-side logic.

Coverage of privacy factors. PRIVACYMETER incorporates open source code from Adblock Plus to implement our tracker filter, and uses an up-to-date EasyPrivacy list to identify trackers. As we mentioned earlier, such blacklists, by definition, cannot guarantee detection of all trackers. Thus, PRIVACYMETER deploys another module to detect indirect signs of intrusive tracking, namely fingerprinting activity. On Figure 2-a, the FAPI monitor reports to the main score engine all the calls to fingerprinting-related APIs. In order to listen API calls from a web page, the module injects proxy code to the context of each visited web page, which intercepts more than 60 APIs of interest (e.g., navigator.plugins to attempt enumerating plugins or HTMLElement.getBoundingClientRect involved in font enumeration via JavaScript [21]). For a complete list of APIs see Appendix B.

Comparability of the values. PRIVACYMETER compares many parameters among different websites in order to calculate a relative privacy scores. To make this comparison representative and fair, the tests must be performed in the same fashion across all the web pages. For instance, one challenge is to unify the duration of measurements as users spend different time on different websites. We can imagine a situation where a new tracking script is dynamically loaded on the page after a one-minute delay, or after a particular user action. While this does not affect the client-side functionality of the extension as the privacy score shown to the user is dynamically updated, on the back end we take into account that one URL may naturally have different

measurements, and we want to record the duration of each test. Another affected factor is the fingerprinting activity, though in this case later API calls may be a legitimate result of the user's actions on a page, and should not be attributed to the tracker's script. To allow the comparison of fingerprinting APIs we decided to record fingerprinting activity only during the first five seconds on the page, assuming that the most tracking activity happens immediately after loading tracking scripts. Naturally this threshold is configurable and can be changed to fit different use cases.

Performance side-effects. An important requirement of PrivacyMeter is that of low performance overhead as this directly affects the usability of the tool. As such, we use the state-of-the-art code by Adblock Plus to detect tracking requests as fast as possible, and optimize each custom monitoring module to perform less comparisons and function calls when analyzing new requests, web forms, and iframes. Moreover, we deploy the following strategies:

- **Lazy-loading and batch processing.** When a web page is loading, new trackers, web forms, iframes and API calls appear one by one. In real-time, each new item has to be processed by the corresponding PrivacyMeter's module, and the overall privacy score has to be updated. A naive approach would be to process each item separately and immediately request a redrawing of the privacy score. Practically, this results in high performance overhead as numerous messages are generated passing between different contexts of the browser extension. Moreover, information such as a tracker's Web-of-Trust score is requested from the back-end via HTTP requests, and it can therefore be expensive to issue a separate request for each tracker. Similarly, a call to the function that provides geolocation based on a tracker's IP address is also time-consuming. As such, we decided to use batch-processing after lazy loading of new items. Technically, we launch a periodical update event which, once every 0.5 seconds, surveys all the modules about newly gathered information, and re-calculates the privacy issues and score. This helps to decouple the extension's UI from the score calculations, and keep it responsive to other user actions.
- **Client-side and server-side caching.** Even with batch processing, we issue at least one bulk request to the back-end per visited URL in order get additional information like Web-of-Trust scores for encountered trackers. This may be expensive in terms of bandwidth, as well as unnecessary as reputation of trackers may not change that often. To reduce the number of requests to the back-end, we keep an internal client-side cache for tracker information, empirically setting each record's TTL to one week. Similarly, we cache the ground truth with statistics about relative privacy parameters, received from the centralized database, as well as EasyPrivacy lists for two days. Finally, we keep a server-side cache to avoid overloading APIs that are external to our infrastructure, such as those provided by the Web-of-Trust [30].

Invisibility of the extension. Finally, we do not want trackers or websites to find out the presence of PrivacyMeter in the user's browser. Previous research has shown that browser extensions may be discovered through a variety of methods [23,24],

including detection via their DOM side-effects [25]. PrivacyMeter has to modify a page's DOM in order to intercept the fingerprinting API calls. Specifically, PrivacyMeter injects a script into page before any other script runs, which overrides all the fingerprinting APIs. To hide this DOM modification, we immediately remove the script tag after all APIs are patched and mask possible signs of the overridden functions, e.g., by ensuring that when the `toString` method is called, our extension returns the appropriate output as if the method was never overridden.

To calculate a relative privacy score, PrivacyMeter must have information about the privacy practices of popular websites on the web. For example, in order to be able to provide a meaningful relative privacy score of a new sports page, PrivacyMeter needs data about the privacy practices of other popular sports pages. Thus our architecture includes a central database with cached privacy statistics as presented on Figure 2-b. To ensure that our collected data remains up-to-date, we deploy a separate crawler, which runs continuously and analyzes a large fraction of popular websites. Given the flexibility of our design, a crawler is implemented as an automated browser with the same PrivacyMeter extension installed, which is instructed to visit a range of websites and report the calculated privacy statistics so that they can be included in our centralized database.

3 PrivacyMeter's Evaluation

In this section, we report on the PrivacyMeter's evaluation as a privacy preserving browser extension. First, we show the benefits of a relative privacy score, which is one of the key functionalities of PrivacyMeter, comparing privacy practices across different website categories. Second, we test the performance overhead added by the extension to the daily browsing of users.

Relative privacy score. In this work, we argue that the ability to compare privacy practices of a website to privacy practices of other similar websites is a necessary function of a modern privacy preserving tool. First of all, it gives users more fine-grained options in order to protect their privacy. For example, with traditional blockers a non-technical user has only two options: either to block all the trackers, or to allow them all. In this case, when a content publisher deploys an anti-blocker solution [11, 12, 20], the user ends up with the choice whether to allow all tracking or leave the website. While savvy users may be able to find a third choice (e.g. whitelist enough of the trackers to be able to utilize the website), this approach is out-of-reach for the majority of web users. With PrivacyMeter's relative privacy comparison, we equip *all* users with sufficient information to assist them in making a decision to keep on trusting the website or seek a different website of the same category.

Therefore, the next logical question that arises is whether different groups of websites and web pages have different privacy practices, making PrivacyMeter's relative comparison between websites a useful and desirable feature. To answer this question, we compare the four main relative privacy features, currently supported by the PrivacyMeter extension, across Alexa's 17 categories of websites, such as, "Sports" and "News." By crawling Alexa's top 500 websites per each category we

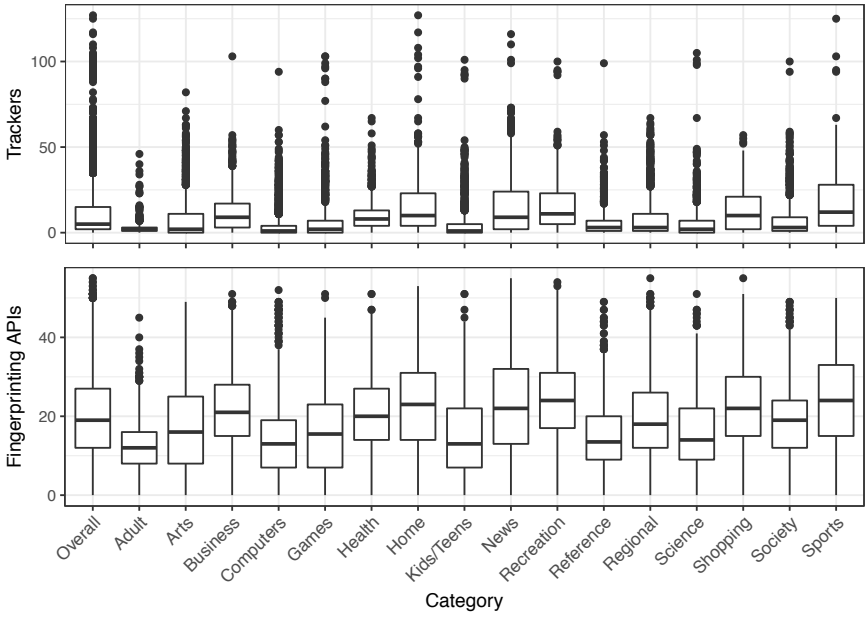

Fig. 3: *Box plots for number of trackers and number of fingerprinting API calls per each of 17 Alexa's website categories, as well as the overall distribution.*

populate PrivacyMeter's database with a total of 6,580 distinct URLs on the 6,049 TLD+1 domains with known categories. Figure 3 compares the resulting box plots on the number of trackers and fingerprinting APIs.

One can notice that the distributions of the number of trackers and fingerprinting APIs indeed vary across website categories. For example, web pages belonging to the adult category tend to have almost no trackers, and hence lower fingerprinting activity, presumably due to the sensitivity of the service and with the goal to earn the trust of the users. Similarly, websites for kids and teens also deploy less tracking as a response to stricter privacy regulations. Contrastingly, news and sports pages (followed by shopping and recreation pages) tend to include more trackers, and tend to be involved more in fingerprint their users' browsers. At the same time, web pages belonging to the "Computers" category appear to not be utilized a large number of third-party trackers. This can be partially explained by the fact that sufficiently large companies, such as, Apple, Google, and Facebook, tend to utilize their own, in-house, tracking solution rather than rely on third parties.

In terms of third-party iframes and leaky web forms, only some distribution parameters, such as median, are different (see Figure 5 in Appendix A). For example, news and sports pages clearly have more third-party iframes than other web pages. This could be because these types of pages rely on advertising for monetization and therefore are likely to be utilizing a large number of distinct iframes where ads are rendered. It is worth noting that PrivacyMeter will count each loaded third-party iframe, even if it was substituted with another one. Next to these general trends, we can spot categories with strongly-pronounced outliers especially on the "leaky" forms

box plot. For example, news websites have many outliers with websites containing up to 200 leaky web forms (our crawler counts each instance of a leaky form since the more present a form is on a website, the more likely a user is to interact with it).

Overall, our results support the premise that incorporating similar box plots to the PrivacyMeter's interface and comparing privacy scores of a particular web page to them, can reveal whether the current website is an outlier or not, in terms of other sites of the same category.

Performance overhead. In order to test the performance of PrivacyMeter, we decided to compare it to Ghostery, a state-of-the-art commercial blocker which, according to prior work [19], detects more trackers than competing extensions. Ghostery, like PrivacyMeter, does not, by default, block trackers but instead presents the list of detected trackers, which makes the performance comparison appropriate. We visited the top 50 websites according to Alexa's ranking with and without each extension. During each visit we recorded times of the `DOMContentLoaded` and `Load page` events. Each test was repeated 10 times with caching disabled in order to retrieve the average timing. For the measurements, we utilized a laptop with 8GB of RAM, Intel's i3 CPU, Ubuntu 14.04 and the latest version of Google Chrome. We present the performance overhead as the time difference between the average page load time with an extension present and absent.

Even with the additional modules of PrivacyMeter (described in Section 2), our extension introduces 0.138 seconds of delay to the DOM loading, and 1.217 to the overall page loading, while Ghostery adds 1.412 and 2.546 seconds correspondingly. Figure 4 compares the distribution of delays introduced by each extension for both of the load events. Note that the negative values mean that, for those particular websites, the overhead from PrivacyMeter is less than the loading variance due to network conditions and system load.

According to Figure 4-a and Figure 4-b, PrivacyMeter is faster for most sites. At the same time, PrivacyMeter slows down only two particular websites more than Ghostery, in terms of the page load event. Namely, those are `www.yahoo.com` (12.7 seconds) and `www.cnn.com` (18.3 seconds). If in the first case we encountered two long delays out of 10 attempts, which increased the average of our measurements (the delays were likely due to increased size of remote content and previously unseen trackers). The second case can be clearly explained by the large number of trackers, fingerprinting calls, and third-party iframes loaded. Note that for the similarly resource-heavy `www.nytimes.com`, both extensions add more than 8 seconds of delay. It is worth pointing out that a 10 second delay does not mean that the user needs to wait for 10 seconds before being able to consume the content of the page. Browsers start rendering content immediately therefore users can start interacting with a website much earlier than the firing of the `DOMContentLoaded` and `Load page` events.

To quantify the frequency with which PrivacyMeter queries our back-end for the information about trackers, we calculated how many unique tracking domains a user may encounter while visiting 10, 100 and 1000 different URLs in a row. For the simulation we analyzed the same 6,580 URLs of Alexa's top websites across all the categories. Table 1 presents the resulting statistics. If a user visits 100 different websites during a week (the life time of the client-side cache), PrivacyMeter will make approximately

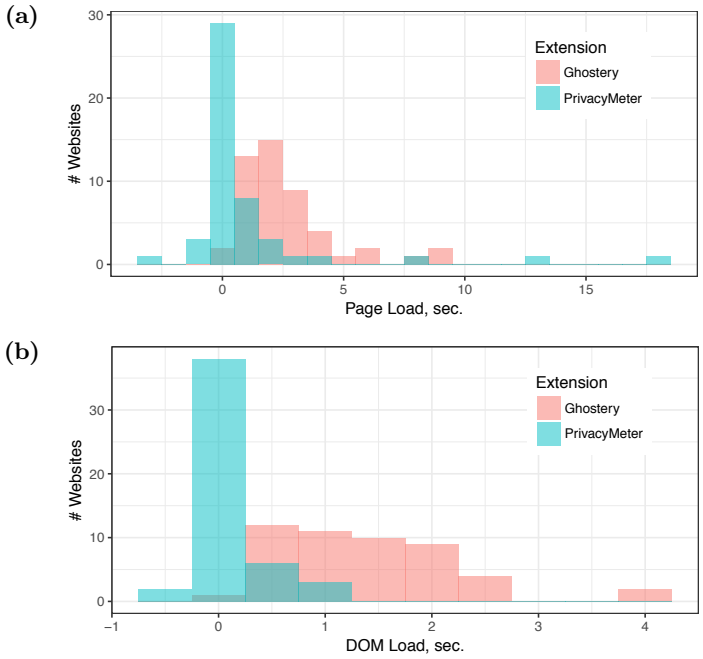

Fig. 4: *Comparison of the performance overhead by* PrivacyMeter *and* Ghostery *extensions based on top 50 websites: (a) overall page load overhead; (b) delay to the DOMContentLOaded event.*

65 requests for the additional information about newly encountered trackers. If a user visits 1000 websites, PrivacyMeter will initiate at most 495 batch requests.

4 Crowdsourcing

An additional component of PrivacyMeter's architecture is the ability to collect privacy scores and privacy statistics of web pages that users browse. If users opt-in to our crowdsourcing mechanism, PrivacyMeter will, after generating a page's privacy score, communicate that score and its individual parameters back to our central server. This crowdsourcing mechanism will enrich our database with entries for websites that the active users of PrivacyMeter find relevant, and will help the crawler to keep available statistics up-to-date. Consequently, the accuracy of the privacy score will be continuously adjusted to provide proper, up-to-date values.

To protect the anonymity of users who opt-in to crowdsourcing, PrivacyMeter does not collect any PII. Instead, for users who have opted-in, the extension reports to its backend the following privacy practices per web page: list of present trackers (their URLs), fingerprinting APIs called (and their counts), list of third-party iframes, and the list of "leaky" web forms. Our extension does not utilize any stateful/stateless identifiers that would allow us to reconstruct a user's session. Finally, in order to be fully transparent, the code of PrivacyMeter is made available as open source and the extension itself does not utilize any obfuscated JavaScript code.

# Websites	# Trackers			# New encounters		
	Min	Avg	Max	Min	Avg	Max
10	4	65	188	3	8	10
100	154	271	365	48	65	83
1000	824	925	1043	397	445	495

Table 1: *Simulated numbers of encountered trackers while browsing popular websites*

Next to the improving of the tool itself, the ability to collect crowdsourced statistics about the privacy practices of websites will help to drive future privacy research that will benefit end users. For instance, the collected data can allow us to understand how privacy risks evolve and inform policies and future technical countermeasures. Even with our currently small user base (17 users at the time of this writing, most of whom are researchers participating in the Data Transparency Lab initiative [7]), we already see the benefits of crowdsourcing, which gives us ability to discover more trackers when considering other URLs of the same website in addition to its front page. Namely, these PrivacyMeter users contributed privacy reports for an additional 7K URLs (belonging to both popular and less popular websites) which were not part of our crawling efforts on which 1,015 previously unseen tracking domains were detected.

5 Future work

In terms of future work, our next step is to conduct user studies (using either online platforms such as Amazon Mechanical Turk or recruiting students from our institute) to quantify how much more helpful users find the output of PrivacyMeter, compared to traditional output of existing browser extensions, such as, Ghostery. In this paper we decided to focus on the engineering and implementation challenges of building a privacy-preserving browser extension which are separate from followup user studies.

Next to user studies, we plan on adding detection capabilities for identifying malicious web clients. As with any system supported by user-provided data, malicious users can attempt to poison PrivacyMeter's central database by submitting false reports. To the best of our knowledge no privacy browser extension deploys a client-side protection against that. We plan to mitigate this kind of abuse through a combination of client-side and server-side techniques including using proof-of-work algorithms [10] at the client side (to slow down automated submissions) and IP-address-based majority voting at the server side (to filter-out reports containing outliers).

6 Related Work

The modern market of privacy-preserving browser extensions is mainly represented by anti-tracking blockers [4, 5, 8, 9, 22, 28, 29]. While some of them attempt to provide additional information like categories of trackers, such as advertisement or analytics [8,9], the majority are general blockers, which just show users the list of discovered tracking domains. Similarly, to the best of our knowledge, PrivacyMeter is the

first browser extension to evaluate a range of privacy practices of visited web pages in a single solution, as well as to calculate a relative privacy score, comparing each site with other similar sites.

Leon et al. in 2011 evaluated the usability of nine blocking tools including Adblock Plus and Ghostery [14]. The study reports many issues revolving around the configuration and usage of these tools. Malandrino et al. also point out issues with user awareness and effectiveness of blocking tools [17]. As such, other privacy preserving extensions attempt to deceive trackers, e.g. AdNausem [1] automatically clicks on all blocked ads in an effort to confuse trackers about a user's true interests. Similarly, TrackMeNot [3] simulates dummy search queries, and BrowsingFog [27] obfuscates browsing history against extension-level trackers. Chameleon [2] attempts to unify fingerprinting features of the Chrome browser (similar to the Tor Browser), in order to break fingerprintability. Despite their benefits, none of the aforementioned tools provide feedback to users about the privacy practices of each visited website.

The work that is the closest to ours is the PrivacyScore website by Maass et al. [16], which deploys automated scanning of websites and allows its users to get security and privacy features for websites of their interest. A major difference is the vantage point of these two tools since, in our work, PrivacyMeter is a browser extension running on the client-side and therefore having access to all of the content that a server-side crawler cannot access (such as content behind registration walls). Moreover, PrivacyMeter's calculates a privacy score dynamically which means that the score that the user sees is always representative of the current state of the website, and is not a previous score from the last time that the site was crawled.

7 Conclusion

As companies seek to collect more and more data about our online activity, it is imperative that users develop an understanding of privacy issues on the web, rewarding responsible websites with their visits while shunning away from websites employing intrusive privacy practices. In this paper, we described the design and implementation of PrivacyMeter, a browser extension (with a server back-end) which aims to provide users with actionable information about a website's privacy-related practices and how it compares to other sites of the same category. We demonstrated that PrivacyMeter's performance overhead is less than that of popular alternatives while offering more functionality. We have open-sourced PrivacyMeter and we hope that, in addition to helping users online, our work can be used as a case-study for building privacy-preserving tools.

Availability: The demo of the PrivacyMeter extension is available at Chrome Store: `http://bit.ly/PrivacyMeter`

8 Acknowledgments

We thank the reviewers for their valuable feedback. This work was support by the National Science Foundation under grants CNS-1527086 and CNS-1617593 as well as by the Data Transparency Lab.

References

1. AdNauseam. http://adnauseam.io/
2. Chameleon. https://github.com/ghostwords/chameleon
3. TrackMeNot. https://cs.nyu.edu/trackmenot/
4. Adblock. https://getadblock.com/
5. Adblock plus. https://adblockplus.org/
6. Chaabane, A., Ding, Y., Dey, R., Ali Kaafar, M., Ross, K.: A Closer Look at Third-Party OSN Applications: Are They Leaking Your Personal Information? In: Passive and Active Measurement conference (2014). Springer, Los Angeles, United States (Mar 2014), https://hal.inria.fr/hal-00939175
7. Data Transparency Lab. http://datatransparencylab.org/
8. Disconnect — Online Privacy & Security. https://disconnect.me/
9. Ghostery. https://www.ghostery.com/
10. Hashcash: Proof-of-work algorithm. http://www.hashcash.org/
11. Hruska, J.: Forbes forces readers to turn off ad blockers, promptly serves malware. http://www.extremetech.com/internet/220696-forbes-forces-readers-to-turn-off-ad-blockers-promptly-serves-malware (2016)
12. Iqbal, U., Shafiq, Z., Qian, Z.: The ad wars: Retrospective measurement and analysis of anti-adblock filter lists. In: Proceedings of the 2017 Internet Measurement Conference. IMC '17 (2017)
13. Krishnamurthy, B., Naryshkin, K., Wills, C.E.: Privacy leakage vs. protection measures: the growing disconnect. In: Web 2.0 Security and Privacy Workshop (2011)
14. Leon, P., Ur, B., Shay, R., Wang, Y., Balebako, R., Cranor, L.: Why johnny can't opt out: A usability evaluation of tools to limit online behavioral advertising. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 589–598. CHI '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2207676.2207759, http://doi.acm.org/10.1145/2207676.2207759
15. Lerner, A., Simpson, A.K., Kohno, T., Roesner, F.: Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In: USENIX Security Symposium (2016)
16. Maaß, M., Wichmann, P., Pridöhl, H., Herrmann, D.: Privacyscore: Improving privacy and security via crowd-sourced benchmarks of websites. In: Schweighofer, E., Leitold, H., Mitrakas, A., Rannenberg, K. (eds.) Privacy Technologies and Policy - 5th Annual Privacy Forum, APF 2017, Vienna, Austria, June 7-8, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10518, pp. 178–191. Springer (2017). https://doi.org/10.1007/978-3-319-67280-9_10, https://doi.org/10.1007/978-3-319-67280-9_10
17. Malandrino, D., Petta, A., Scarano, V., Serra, L., Spinelli, R., Krishnamurthy, B.: Privacy awareness about information leakage: Who knows what about me? In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society. pp. 279–284. WPES '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2517840.2517868, http://doi.acm.org/10.1145/2517840.2517868
18. Mayer, J.R., Mitchell, J.C.: Third-party web tracking: Policy and technology. In: IEEE Symposium on Security and Privacy. pp. 413–427. IEEE Computer Society (2012), http://dblp.uni-trier.de/db/conf/sp/sp2012.html\#MayerM12
19. Merzdovnik, G., Huber, M., Buhov, D., Nikiforakis, N., Neuner, S., Schmiedecker, M., Weippl, E.: Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools. In: Proceedings of the 2nd IEEE European Symposium on Security and Privacy (IEEE EuroS&P) (2017)

20. Mughees, M.H., Qian, Z., Shafiq, Z.: Detecting anti ad-blockers in the wild. Proceedings on Privacy Enhancing Technologies **2017**(3), 130–146 (2017)
21. Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., Vigna, G.: Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In: Proceedings of the 34th IEEE Symposium on Security and Privacy (IEEE S&P). pp. 541–555 (2013)
22. Privacy Badger — Electronic Frontier Foundation. `https://www.eff.org/privacybadger`
23. Sanchez-Rola, I., Santos, I., Balzarotti, D.: Extension breakdown: Security analysis of browsers extension resources control policies. In: 26th USENIX Security Symposium. pp. 679–694 (2017)
24. Sjösten, A., Van Acker, S., Sabelfeld, A.: Discovering browser extensions via web accessible resources. In: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy. pp. 329–336. ACM (2017)
25. Starov, O., Nikiforakis, N.: Xhound: Quantifying the fingerprintability of browser extensions. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 941–956 (May 2017). https://doi.org/10.1109/SP.2017.18
26. Starov, O., Gill, P., Nikiforakis, N.: Are you sure you want to contact us? quantifying the leakage of PII via website contact forms. PoPETs **2016**(1), 20–33 (2016), `http://www.degruyter.com/view/j/popets.2016.2016.issue-1/popets-2015-0028/popets-2015-0028.xml`
27. Starov, O., Nikiforakis, N.: Extended tracking powers: Measuring the privacy diffusion enabled by browser extensions. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1481–1490. WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017). https://doi.org/10.1145/3038912.3052596, `https://doi.org/10.1145/3038912.3052596`
28. ublock. `https://www.ublock.org/`
29. ublock origin. `https://chrome.google.com/webstore/detail/ublock-origin/cjpalhdlnbpafiamejdnhcphjbkeiagm`
30. Safe Browsing Tool — WOT (Web of Trust). `https://www.mywot.com/`

Appendix A

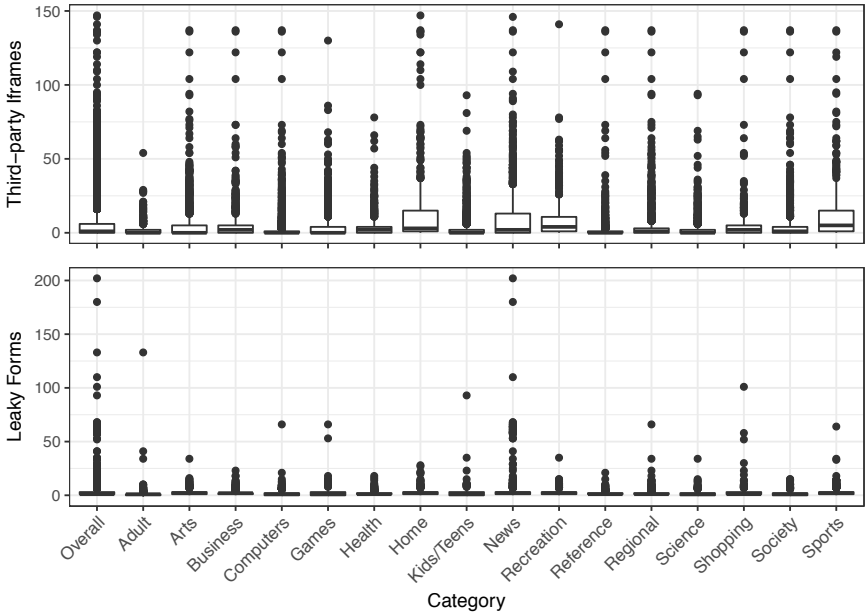

Fig. 5: *Box plots for number of third-party iframes and number of leaky web forms per each of 17 Alexa's website categories, as well as the overall distribution.*

Appendix B

The list of fingerprinting-related APIs currently intercepted by PrivacyMeter:

- window._phantom
- window.callPhantom
- window.chrome.webstore
- window.devicePixelRatio
- window.domAutomation
- window.domAutomationController
- window.indexedDB
- window.localStorage
- window.mozRTCPeerConnection
- window.RTCPeerConnection
- window.RunPerfTest
- window.sessionStorage
- window.TouchEvent
- window.webdriver
- window.webkitRTCPeerConnection
- navigator.appCodeName
- navigator.appName
- navigator.appVersion
- navigator.cookieEnabled
- navigator.cpuClass
- navigator.doNotTrack
- navigator.hardwareConcurrency
- navigator.javaEnabled
- navigator.language
- navigator.languages
- navigator.maxTouchPoints
- navigator.mediaDevices
- navigator.mimeTypes
- navigator.onLine
- navigator.platform
- navigator.plugins
- navigator.product
- navigator.productSub
- navigator.userAgent
- navigator.vendor
- navigator.vendorSub
- screen.availHeight
- screen.availLeft
- screen.availTop
- screen.availWidth
- screen.colorDepth
- screen.height
- screen.orientation
- screen.pixelDepth
- screen.width
- HTMLCanvasElement.getContext
- HTMLCanvasElement.getImageData
- HTMLCanvasElement.toDataURL
- CanvasRenderingContext2D.fillText
- CanvasRenderingContext2D.getImageData
- CanvasRenderingContext2D.strokeText
- WebGLRenderingContext.getExtension
- WebGLRenderingContext.getParameter
- WebGLRenderingContext.getShaderPrecisionFormat
- HTMLElement.addBehavior
- HTMLElement.getBoundingClientRect
- HTMLElement.offsetHeight
- HTMLElement.offsetWidth
- Date.getTimezoneOffset
- document.createEvent("TouchEvent")
- document.addEventListener("mousemove")