User Profiling by Network Observers

Roberto Gonzalez
roberto.gonzalez@neclab.eu
NEC Labs Europe
Spain

Claudio Soriente
claudio.soriente@neclab.eu
NEC Labs Europe
Spain

Juan Miguel Carrascosa
jm.carrascosa@tyrceo.com
TYRCEO Data Solutions
Spain

Alberto Garcia-Duran
agaduran@gmail.com
Atinary Tech.
Switzerland

Costas Iordanou
costas.iordanou@eecei.cut.ac.cy
Cyprus University of Technology
Cyprus

Mathias Niepert
mathias.niepert@neclab.eu
NEC Labs Europe
Germany

ABSTRACT

Targeted online advertising is a multi-billion dollar business based on the ability of profiling and delivering targeted ads to a wide range of users. Due to the privacy erosion associated with such business, researchers are trying to understand how profiling works and anti-tracking applications are becoming popular among users. Both research and privacy-enhancing apps, however, target ad-networks or over-the-top providers that have unrestricted access to users' online activity. There seems to be little interest in potential profiling activities by "network observers" like ISPs or VPN providers. On the one side, this may be explained by the pervasiveness of TLS that secures connections end-to-end. On the other side, TLS does leak some information, and it is not clear what an eavesdropper can learn about a user, despite her traffic being encrypted.

In this paper, we show that a network observer can build accurate user profiles notwithstanding the limited visibility due to TLS. In particular, we introduce a technique based on representation learning algorithms that can build profiles by only using the hostnames of URLs requested by users. To evaluate the accuracy of the profiles built with our technique, we setup an experiment where we serve personalized ads to more than one thousand real users over a period of one month. We compare the click-through rate of ads served by our system with the one of ads served by ad-networks. We empirically show that the quality of profiles that a network observer could build is comparable to the quality of profiles available to ad-networks and over-the-top providers. This is particularly worrisome since current anti-tracking mechanisms cannot counter profiling activities by network observers, whereas effective mechanisms like TOR incur in a performance and usability penalty.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; • **Information systems** → **Display advertising**; **Content match advertising**; • **Networks** → *Network measurement*.

CoNEXT '21, December 7–10, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-9098-9/21/12…$15.00
https://doi.org/10.1145/3485983.3494859

KEYWORDS

privacy, advertising, user profiling

ACM Reference Format:
Roberto Gonzalez, Claudio Soriente, Juan Miguel Carrascosa, Alberto Garcia-Duran, Costas Iordanou, and Mathias Niepert. 2021. User Profiling by Network Observers. In *The 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '21), December 7–10, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3485983.3494859

1 INTRODUCTION

Online targeted advertising is a multi-billion dollar business grounded in a complex network of stakeholders that collect, buy, and sell user data with the goal of assembling accurate user profiles. Demographic data, income and, ultimately, the history of visited websites are used as proxies of user interests. However, the ways in which such personal data is mapped to specific interests and, ultimately, personalized ads remains largely unknown.

The widespread collection of user data and the erosion of privacy associated with it, has fostered research on *data transparency* with multiple initiatives that try to shed light on the ways user data is collected and used for targeted advertising [1]. At the same time, apps and browser extensions to limit tracking and data collection have become popular among users.

Both research activities on data transparency and privacy-enhancing apps (e.g., ad-blockers) try to overcome the tracking mechanisms of ad-networks and Over The Top (OTT) providers that have unfettered access to the online activities of a user. However, there seems to be little interest in profiling activities by "network observers" such as ISPs or VPN providers. On the one side, there may be little to fear as the pervasiveness of TLS only allows eavesdroppers to learn the hostname requested by the user—leaked by the `client_hello` message of the TLS handshake. On the other side, profiling activities by a network eavesdropper cannot be prevented with ad-blockers and alike. Further, it is not clear how much the requested hostnames tell about a user and, to the best of our knowledge, no previous work has tried to assess the effectiveness of user profiling from the perspective of a network eavesdropper. Motivated by these considerations, we set out to answer two important questions.

Question 1. Is it possible to build user profiles from the perspective of a network eavesdropper, despite the limited view due to TLS?

Question 2. Are user profiles built by a network eavesdropper more or less accurate than user profiles created by ad-networks and OTTs?

To answer the first question, we propose a profiling technique that solely uses the hostnames of URLs that users visit. Even if hostnames are available to network eavesdroppers, hostnames by themselves may not be very useful to profile users. The main challenge is to find a mapping from a hostname to a set of semantically meaningful "topics" or "categories". For example, hostnames like *booking.com* may be mapped to topics such as "travels", whereas hostnames like *espn.com* may be mapped to "sports". Ontologies for this task are available and have been used in previous work [2–4], Google Adwords being the most popular one. The problem with an ontology is its "coverage" since many ontologies simply do not categorize a large fraction of the Web. One may even think of building its own ontology by, e.g., crawling webpages and analyzing their content [5]. Yet, when a user visits a website, a network observer is likely to see requests for CDNs and API services and it may not be feasible to infer the actual website. In order to mitigate the limited coverage due to ontologies and the information loss due to the perspective of a network observer, we propose an algorithm based on a neural network to learn relations among hostnames. Intuitively, our algorithm is able to learn similarities among different hostnames by understanding the temporal relations between user requests to those hostnames. Once we are able to group similar hostnames, it is sufficient to obtain the topic of one of them (e.g., by using an ontology), in order to categorize the remaining ones.

Regarding the second question, we note that there is no popular metric to measure the accuracy of a profile or to compare the output of two profiling techniques. We therefore leverage the Click-Through Rate (CTR) as a meaningful proxy to compare different approaches. CTR is defined as the ratio of displayed ads that are clicked by users and is a well-known metric to measure the quality of ad campaigns. We use CTR to indirectly measure the accuracy of profiles built by a network eavesdropper that uses our neural network algorithm, and compare it with the accuracy of profiles available to ad-networks.

We build user profiles and assess their accuracy by means of a one month experiment involving 1329 users across 17 countries. The experiment leverages our Chrome extension available at https://privacyaware.nlehd.de/CoNext21/captureData.crx. The extension has full access to browsing sessions. We use it to collect only the hostnames of URLs visited by users and to manipulate ads they see on the screen (i.e., to replace an ad served by ad-networks with one of our choice). Results show that CTR of ads picked according to the profiles output by our algorithm are comparable to CTR of ads served by ad-networks. We conclude that profiles built by a network observer may be quite accurate despite the limitations due to TLS.

We remark that we have no evidence of profiling activities by network observers such as ISPs or VPN providers. Yet, we provide evidence of what a network observer can learn about a user by looking at her (encrypted) traffic. Network observers could even use techniques different from ours. Some of them, like ISPs may use additional customer data (e.g., gender, age, residential address) to improve profiling. Profiles may be sold to third-parties or could be used for advertising by other means (e.g., ads sent via email or SMS). Further, we argue that profiling activities by network eavesdroppers are particularly worrisome because countermeasures like ad-blockers cannot prevent profiling by network observers, whereas tools like TOR incur in a performance and usability penalty that may not be tolerable by all users or applications.

1.1 Ethical considerations

We consider this study to potentially raise ethical concerns because of the usage of data and profiling of real users. Therefore, we have taken precautions to mitigate the impact of the data collection and processing phases. First, the Data Protection Officer available at one of our institutions reviewed and approved the study. Second, we resorted to the Spanish[1] association of internet users (https://www.aui.es/). The association has advised us throughout the whole experiment by, e.g., reviewing the extension and all the forms that users were asked to accept in order to participate in our study. In addition, the association has controlled the distribution of the extension and its updates, and has obtained explicit user consent to collect and process their data. Further, we did not manage/store any personal user information (e.g., name, email, IP address, etc.) but we only assigned a random ID to each installation of the extension. Finally, communication between the plugin and our back-end server happened over TLS. Collected data was periodically transferred to and stored on a machine in a locked room and with no network access. Access to the room and to the machine is only allowed to authorized personnel.

2 RELATED WORK

Online Advertising Ecosystem. Lerner *et. al.,* [6] study the evolution of web tracking over time (1996-2016) by analyzing the total number of third-party trackers embedded in popular first-party domains. The authors discover that third-party tracking on the Web has increased in prevalence and complexity since 1996. The authors of [7] conduct a large measurement study by crawling 1M popular websites and by analyzing the presence of different tracking techniques, such as, stateful (cookie-based) and stateless (fingerprinting-based), and the exchange of tracking data based on cookie-synching. Mayer *et. al.,* [8] study the policy debate surrounding third-party web tracking and the relevant technology around it. The authors of [9] use an ISP dataset with more than 3M subscribers to measure the extent of ad-related traffic in mobile networks. On a similar path, the work of [10] use a mobile app to analyze traffic at the user device. The authors observe that sharing harvested data among tracking entities for user profiling purposes is the norm. The work of [11] studies the tracking behavior of more than 950K mobile apps and shows that applications related to news and children are among the most privacy-invasive ones. Binns *et. al.,* [12] compares the web and mobile version of different online services with respect to web tracking and privacy. Vallina *et. al.,* [13] studied the tracking and advertising in porn websites and Pachilakis *et. al.,* [14] measured the Header Bidding ecosystem.

User Profiling. The methods used to profile users by the advertising industry are in general non-public. While we are not aware of any previous work that creates interest-related profiles from network traffic, a handful of previous work has tried to profile users in different contexts. Previous work attempted to build user profiles by leveraging social network activity [15, 16]. Kumar *et. al.,* [17] builds user profiles for personalized news delivery, by using traces of news read in online newspapers. Different from ours, their approach requires access to the text of the news read by users. Alotibi *et. al.,* [18] profiles users from network data. Their approach is based on the

[1] Some of the authors of this paper are located in Spain

theory that users interact with Internet applications in a unique manner and the system they propose is designed to identify users in the context of insider misuse detection. Our previous work [4] leverages network traces (in particular, packet sizes) to identify the exact URL visited by a user, even when the connection goes over https. As we argued before, once a profiler obtains the URL requested by a user, (lack of) coverage by available ontologies may hinder the process of building a user profile. Further, the system in [4] detects URLs of depth 1 (i.e., URLs of pages linked on the main page of a website) and it is not clear how it would work to detect pages at arbitrary depth.

3 BACKGROUND

Parties. Websites where ads are displayed are known as "publishers" (e.g., *nytimes.com*) that sell advertising estate on their webpages. The visual part of an ad is referred to as its "creative" that could be text, a video or an image (e.g., an image of a car). The ad is linked to a "landing page", i.e., the page that opens when the user clicks on the ad and where the object of the ad is offered (e.g., the webpage of the car manufacturer). Behind the curtains, a complex network of stakeholders—Ad-networks, ad-exchanges, Demand and Supply Side Platforms (DSPs, SSPs), Data Management Platforms (DMPs), etc.—act as brokers between demand (companies willing to advertise their products) and offer (webpages offering ad estate) [19]. This brokerage activity leverages user profiles (created, in turn, by other companies specialized in "tracking" users and collecting their data) to ensure that ads are served to the right audience. Often, publishers, advertisers, brokers and trackers are owned by one single firm, making the online advertising ecosystem extremely complex and blurry. We refer the reader to Pastor et al.[20] for a comprehensive description of the online advertising ecosystem.

Ad types. Ads may be divided in two main types. "Premium" ads typically promote important advertisers willing to pay premium prices to show their brand on top publisher websites. These ads are served to all users visiting a given website within a time-frame. For instance, *Coca-Cola* (the advertiser) may pay *espn.com* (the publisher) to show its creative to all users that browse that website on a given day. Furthermore, premium ads are placed on a prominent part of the webpage, sometimes even over the content of the webpage itself. "Programmatic" ads are the ones that are served by taking into account the profile of an audience. In particular, programmatic ads include "retargeted" ads, i.e., ads based on a product seen by a user in a recent browsing session, "contextual" ads, i.e., ads based on the demographic properties of an audience or on the topic of the website where they are displayed, and "targeted" ads, i.e., ads based on a user profile.

Business model. Displaying an ad to a user on a website is referred to as ad "impression". Publishers are usually compensated by the number of impressions. The quality of an advertising campaign is usually measured by so-called Click Through Rate (CTR), i.e., the ratio of the number of impressions that were clicked by the audience (and that led to the landing page) over the total number of impressions. An accurate user profile allows the ad ecosystem to understand on which ads that user will most likely click and, ultimately, improve the click-through rate of an advertising campaign. Yet, depending on the product, some campaigns may be optimized to increase the expected revenue rather than the click-through rate. For example, the advertiser may prefer fewer users that eventually buy its products, rather than a larger number of user that visit the landing page without making purchases. Once again, having accurate user profiles is key to understanding which audience is likely to buy specific products.

4 USER PROFILING FROM BROWSING ACTIVITY

Little is known about the methods currently deployed to map browsing activity to profiles and, ultimately, to targeted ads. Recent work has attempted to shed light on such profiling activities to bring some transparency to the matter. In particular, previous work has found large correlations between webpages that users visit and the kind of ads they receive during their browsing activity [3, 21]. The common theme of those studies were to map both webpages (i.e., their URLs) and ads (i.e., their landing pages) to a set of "topics" and to find correlations between the topics of visited webpages and the ones of received ads. In other words, webpages and ads are labeled with a set of topics drawn from a common universe; a user profile is described by a set of topics defined by the webpages that user visits; a user (apparently) receives ads labeled with topics that match the ones in her profile.

Working in such a setting requires defining the universe of topics and a reliable labeling of webpages and ads. Previous work has essentially explored two alternatives. On the one side, a URL (be it a webpage or the landing page of an ad) can be mapped to topics by analyzing its text [5]. On the other side, labeling can be done via an ontology, Google Adwords [22] being the most popular one. We note that both options are ill-suited for user profiling from the perspective of a network observer. Analyzing the content of a webpage requires knowing the full URL requested by the user, which is not available when one can only observe the hostnames of TLS requests. Moreover, analyzing the content of a page may require time (and resources) and may not even be feasible if the URL refers to a content delivery network (CDN) or an API service. In fact, using only the hostname of a URL pointing to a CDN or API is likely to return no results or to return the homepage of the CDN provider. During our experiments, 67% of the 470K hostnames visited by our users returned an error/empty page when we tried to download the website content. Ontologies represent a lightweight alternative to content-based labeling. The main problem with ontologies is their coverage. For example, Google Adwords classifies only 10.6% of the hostnames in our dataset.

4.1 User Profiling using hostnames

The main challenges in profiling users from the perspective of a network observer lie in the limited coverage of available ontologies and the coarse-grained information obtained from TLS requests. We tackle such challenges by leveraging an unsupervised machine learning approach that ultimately allows us to assign vector representations to hostname sequences. The proposed solution is inspired by representation learning, typically used in Natural Language Processing to assign vector representations (also known as embeddings) to words carrying information about their usage and meaning. Our system learns vector representations of hostnames based on sequences of hostname requests observed in the network. In the same way the

meaning of a word can be inferred from the context it is frequently used in, the profiling-relevant information a hostname carries can be inferred from other hostnames it is frequently co-requested with. Based on the learned hostname representations, we can assign a vector representation to hostname sequences and use those to construct an accurate user profile.

For instance, while it is a-priori challenging to assign labels to an API request such as *api.bkng.azure.com*, having the request co-occur in the network, across numerous user sessions, with hostnames for which we do have known labels such as *hotels.com*, allows us to learn that *api.bkng.azure.com* is probably a travel-related API endpoint. Learning these representations across hostnames allows us to assign profiling-relevant information to users even if, at inference time, only API calls are observed. The details of the algorithm are presented below. We would like to emphasize that the algorithm is fully parallelizable and can be scaled up to requirements, allowing traffic analysis at line rate.

User Profiling Algorithm. The input data to the proposed representation learning algorithm are hostname request sequences across users in the network over a time interval. At training time, the algorithm learns a vector representation for hostnames that carries information about the *contexts*, that is, the other hostnames it has been co-requested with. The resulting hostname representations carry information about its use and similarity to all other hostnames requested in the network. In a second step, the hostname representations are used to construct a hostname sequence representation which is then categorized with a k-nearest neighbor algorithm. The reasonable assumption of the proposed approach is that *some* hostnames do have a unique categorization (from the ontology) assigned to them. It is these categories that are leveraged in the kNN algorithm.

We build our representation learning approach on the SkipGram model [23], which can be directly related to matrix factorization methods [24]. While SkipGram was originally proposed to learn word representations from sets of sentences, we learn representations of hostnames from sets of sequences of hostnames visited by a particular user. Intuitively, instead of estimating the likelihood of sequences of words appearing in a corpus, we aim to estimate the likelihood of sequences of hosts collected at the network level. The learning mechanism behind SkipGram leads the learned representations for elements in a sequence to be useful for predicting surrounding elements. In applying this learning framework to sequences of hosts visited by users, one would expect the learned representation for, e.g., *facebook.com* to be predictive for the hostname *twitter.com*, as it is natural for a user to check all her social networks one after the other [25].

Let H be the set of all the hosts, and let $f : H \rightarrow \Re^d$ be the mapping function, defined as a matrix W of size $|H| \times d$, from hosts to feature representations (traditionally called embeddings) we aim to learn. d is an hyperparameter of the model which is related to the dimensionality of the feature representations. Therefore for every hostname $h \in H$ we can define its embedding as $\mathbf{h} = \texttt{one\_hot}(h)W$, where $\texttt{one\_hot}(h)$ is a vector of size $|H|$ whose hth entry is 1 and all other entries are 0. Given a set of sequences of hosts, a window of size $2m + 1$ is then moved over such sequences and, for each hostname h_c at the center of the window, the negative log likelihood

$$-\log P(h_{c-m}, \cdots, h_{c-1}, h_{c+1}, \cdots, h_{c+m} \mid h_c)$$

of the hosts in the window given h_c is minimized. The modeling assumption is that the hosts in the context window are mutually independent given the central hostname h_c. This conditional independence assumption yields the following optimization problem

$$P(h_{c-m}, \cdots, h_{c-1}, h_{c+1}, \cdots, h_{c+m} \mid h_c) = \prod_{-m \le i \le m, i \neq 0} P(h_{c+i} \mid h_c). \tag{1}$$

Negative sampling, which is closely related to noise contrastive estimation [26], assumes the probability $P(h \mid h_c)$ is proportional to the dot product $\mathbf{h}_c^t \mathbf{h}'$, where $\mathbf{h}'$ is the context feature representation of a surrounding host. Similarly, for every hostname $h \in H$ we can define its context embedding as $\mathbf{h}' = \texttt{one\_hot}(h)W'$. By simply maximizing the L2-norm of the embeddings one could maximize the dot product for all (context, central) hostname pairs contained in the window. To avoid this type of degenerate solutions, the negative sampling objective tries to maximize the likelihood for observed (context, central) hostname pairs while minimizing the likelihood for randomly sampled negative (context_N, central) hostname pairs.

Therefore, for each of the windows of size $2m + 1$ we seek to minimize the following log loss function:

$$\sum_{j=0, j\neq m}^{2m} \Big(\log\ \sigma(\mathbf{h}_c^t \mathbf{h}'_{c-m+j}) + K \sum_{h_k \sim P_D} \log\ \sigma(-\mathbf{h}_c^t \mathbf{h}'_k) \Big), \tag{2}$$

where $\mathbf{h}', \mathbf{h} \in \mathbb{R}^d$ are the context and central representations of hostname h, respectively. K is the number of negative sampled hosts, which are drawn according to a empirical unigram distribution P_D [23], and σ is the sigmoid function. All parameters of the objective (namely, context and central representations of hosts) are learned with stochastic gradient descent.

Once the representation learning stage has finalized, we leverage the resulting feature representations, in conjunction with a subset of hosts $H_L \subseteq H$ for which their categorization is known, to generate a user profile for each session. Usually, the number of hostnames for which categories from the given ontology are known H_L is small compared to all known hostnames H. Therefore, for all hosts $h \in H_L$ we know their related categories $\mathbf{c}^h = [c_1^h, \cdots, c_i^h, \cdots c_C^h]$, wherein $c_i^h \in [0, 1]$ refers to the importance of the category i in the hostname h[2], and C is the number of categories.

We define the session $s_u^T = [h_1, \cdots, h_n]$ as the sequence of hosts visited by user u in the last window of length T. T can then refer to either a number of hosts (in which case $n = T$) or to a time interval. In general, for a given s_u^T we know the related categories for a number of hosts. In the following, we refer to the set of hostnames contained in s_u^T for which we know their categorization as labeled set $L \subseteq H_L \subseteq H$, and unlabeled set U to those hosts for which we do not know their categories. Note s_u^T cannot be an empty set since the profiling algorithm is only executed for users that are currently browsing the Internet. Moreover, if a host was visited more than one time during the last window, the algorithm only takes into account the first visit. This is done to avoid the impact of interactive services (i.e., video or audio streaming)—where the browser connects to multiple times—in the final profile.

We propose a method to assign labels to user session that is both simple and effective. We compute the vector representations

[2]Note that $\mathbf{c}^h$ is not a probability distribution (i.e., it does not sum up to 1).

of a session s_u^T by applying an aggregation function g to the set of vector representations of the sessions' requested hostnames. Let $\mathbf{s}_u^T = g(\{\mathbf{h} \mid h \in s_u^T\})$ be the aggregated representation of s_u^T, our method computes the N (in this work we set $N = 1000$) hostname representations most similar to $\mathbf{s}_u^T$ according to a similarity metric such as cosine similarity. In other words, we use a simple N-nearest neighbor approach to determine a profile for a given session representation. We refer to this set of N hostnames as $H_{s_u^T}$. For each hostname $h \in H_{s_u^T} \cup L$ (in the following referred as to $H^L_{s_u^T}$) a weight α_u^h is computed as follows

$$\alpha_u^h = \begin{cases} 1 & \text{if hostname } h \in L \\ \left[\dfrac{\mathbf{h}^t \mathbf{s}_u^T}{||\mathbf{s}_u^T|| ||\mathbf{h}||} \right]_+ & \text{otherwise,} \end{cases} \quad (3)$$

where $[x]_+$ is the positive part of x.

Once the weights α_u^h for all $h \in H^L_{s_u^T}$ are computed, the importance of category c_i in session s_u^T, denoted as $c_i^{s_u^T}$, is computed as follows:

$$c_i^{s_u^T} = \frac{\sum_{h \in H^L_{s_u^T} \cap H_L} \alpha_u^h c_i^h}{\sum_{h \in H^L_{s_u^T} \cap H_L} \alpha_u^h}. \quad (4)$$

Since $c_i^h \in [0, 1]$, the resulting values $c_i^{s_u^T}$ will be also in $[0, 1]$. Finally, the session s_u^T is profiled as

$$c^{s_u^T} = [c_1^{s_u^T}, \cdots, c_i^{s_u^T}, \cdots, c_C^{s_u^T}].$$

5 EXPERIMENT

To the best of our knowledge, there is no public metric to assess the accuracy of a profile or to compare different user profiling techniques. We therefore resort to the Click Through Rate (CTR)—a standard metric to measure the quality of an advertising campaign, defined as the percentage of ad impressions that are clicked by users. We use the CTR of ads picked according to a given profiling algorithm as a proxy of the quality of the profiles built by that algorithm. We, therefore, empirically compare the CTR of ads picked according to our algorithm with ads served by ad-networks.

5.1 Design

A full assessment of our profiling technique would require a complex setup including (at least) a network observer, such as an ISP, and some online advertising companies. We design our experiment to mimic to the maximum extent the real scenario. To this end, we have developed a Google Chrome extension that is able to monitor and manipulate browsing sessions, and a back-end that takes care of profiling users and sending ads selected according to their profiles.

In a nutshell, the extension collects sequences of hostnames visited by users—as a network observer would do—and reports them to our back-end. The latter profiles users and select relevant ads by using the algorithm of Section 4.1. At times, the extensions replace ads served by ad-networks with the ones received by the back-end, and reports on which ads the user has clicked.

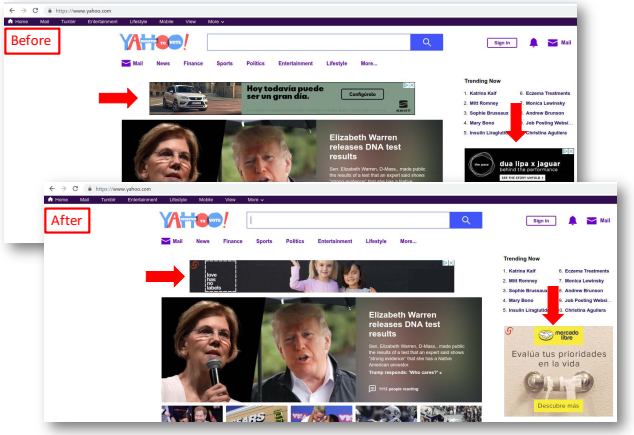


Figure 1: The extension detects ads served by ad-networks and can replace them with ads received from our back-end.

5.2 Execution

Recruitment. We started our recruitment phase by uploading the extension to the Google Chrome Store and by creating a website. The study was announced by the Spanish internet user association (AUI in the following) of the country of one of our institutions, on popular online forums and via Facebook advertising. The declared goal of the study was to better understand online advertising and its privacy implications. Participants were requested to be at least 18 years old and to use Google Chrome as their main browser. As a compensation, participants were given the opportunity of entering a monthly raffle for an iPhone X as long as they would browse the Web with our extension installed on their browsers. By the end of the study, 5 iPhones were awarded.

All participants were informed (and gave explicit consent) of the data collected, and the implications of our experiment (user profiling and injection of ads on visited websites). All forms provided to participants as well as the extension were examined and approved by AUI. The recruitment phase ended after reaching 1000 participants. It lasted 2 months. We witnessed further installations of the extension after the recruitment phase ended, and we totaled 1329 installations by the end of the experiment.

Data Collection. Right after the recruitment campaign, we started a data collection phase that lasted for three months. During this phase we mainly collected (i) the sequence of hostnames visited by the users and (ii) the ads they received.

The sequences of hostnames were used to train the machine learning model presented in 4.1 and fine tune the whole process for the User Profiling phase.

The ads collected during this phase populated the database of ads/creatives to be used in the final phase of the experiment. They were manually filtered to remove ads not properly downloaded (at times, the extension failed to capture a creative using dynamic HTML5) or offensive (e.g., ads of porn websites). After filtering, we were left with a database of roughly 12K ads.

User Profiling. The final and most important phase of the experiment lasted one month. During this time we (i) profiled users from the hostnames they requested, (ii) served ads according to those profiles, and (iii) measured the CTR of displayed ads.

During the last phase, the extensions periodically reported to the back-end the sequence of hosts visited by the user during the last 10 minutes. The back-end generated a profile with the sequence of hostnames visited by that user in the past 20 minutes, and used our ad database to create a list of the most relevant ads for that profile. The list was sent to the extension. (More details about profiling and ad selection can be found in Section 5.4.) Finally, the extension replaced some of the ads served by ad-networks with ads from the received list, during the following 10 minutes. Figure 1 shows a webpage where two ads were replaced. The extension also reported to the back-end the ads that were clicked by the user.

5.3 Ads

Throughout the experiment we have witnessed more than 600M connections to more than 470K unique hostnames, and almost 2.4M ad impressions.

During the last month of our experiment users were shown two types of ads. In some cases, users were shown "Original" ads, that is, those served by ad-networks. Note that we are not aware of the algorithms used by ad-networks to serve a particular ad and, in particular, we do not know whether user profiles built by ad-network include data such as IP address, time, or other information gathered thorough other sources. We also stress that we did not ask user to clear cookies at any time throughout the experiments (hence, they may have done it if there were used to do so) to avoid any interference with the profiling activities of ad-networks. In other cases users were served "Eavesdropper" ads, that is, those chosen by our system—according to the user profile—from the database of ads we had built during the data collection phase.

For each report received from the extension, our back-end served 20 eavesdropper ads. For each ad detected, the extension replaced it with an eavesdropper ad only if one of the ads in the replacement list had a size similar to the size of the original ad. If no ad had similar size, the original creative would not be replaced.

5.4 Relevant ad selection

Finally, we describe the process we followed to obtain the most relevant ads from a sequences of hosts visited by a user. The core of the process, that is, the general algorithm used to profile users is explained in Section 4.1. However, in this section we describe the design decisions that were taken specifically for this experiment.

Mapping hostnames to topics. Our profiling algorithm requires an initial set of labelled hostnames—referred to as H_L in Section 4.1. Similar to previous work [3, 4], we used the Display Planner tool of Google Adwords for this task. We instrumented a Web Browser using Selenium to query the Display Planner for the topics associated to the hostnames visited by our users. In total, we collected the topics associated to roughly 50K of the hostnames visited by the users or included in the landing page of one of the ads they received.

Google Adwords provided us with 1397 different categories/topics associated in a hierarchy that goes from more general categories to the most specific ones. The number of levels of the hierarchy is different for each one of the top categories. For instance, category *Telecom* only has two subcategories, while category *Computers & Electronics* has 123 subcategories organized in a 5-level hierarchy.

In order to harmonize the number of sub-categories and to reduce the overall number of categories taken into account, we decided to use only categories up to the second level of the hierarchy. As a result, 328 categories –this is the set C of Section 4.1– are used to generate the categorization $\mathbf{c}^h$ for each hostname $h \in H_L$.

Filtering hostnames. A first inspection to the hostnames usually visited by the users showed that some of the most popular ones belong to advertisers or tracking companies. Roughly 50 of the top 100 hostnames we witnessed are known to belong to one of those companies.

We decided not to use those hostnames for profiling since they add noise without providing any valuable information about the interests of a user. In order to identify those hostnames we used three different lists designed to block tracking and advertising traffic. Those lists are provided by adaway.org[3], hosts-file.net[4] and yoyo.org[5].

Roughly 3K different hostnames included on these lists were visited by our users. Moreover, 6.1M out of the more than 75M connections witnessed during the user profiling phase were to one of those hostnames. It represents more than 8% of the connections captured by the extension.

Training the algorithm. As described in Section 4.1, our algorithm for session profiling is built on the SKIPGRAM model. Our aim is to demonstrate the usability of the whole system and not the fine tuning of the model that may work best with different hyperparameters in difference circumstances (i.e., we expect the need of a bigger window size in a fixed network where users use a browser, compared to a mobile network scenarios where users tend to use more apps), thus, we use the default hyperparameter values of the popular implementation GENSIM [27]: the embedding dimension d is set to 100, the window size to 5 ($m = 2$) and the number of negative sampled hosts K to 5. For all other hyperparameter values, please refer to such implementation.

We update our model every day. To this end, we obtain from our database the sequence of hosts visited by all the users during the whole previous day. (The amount of data used for training is configurable, however, one day of data has provided good empirical results for different use cases.) We use all that sequences to train a new model that we immediately start using to calculate profiles.

Selecting the best ads. Finally, we execute our profiling algorithm every time a user sends new data to the server. After the new data is included in the database, we obtain the sequence of hosts visited by the user during the last T minutes. As in the case of the amount of data used for training, the amount of data used for profiling is also configurable. For this experiment we set $T = 20$ minutes. This value was empirically tested as a good trade-off between very short sessions that may led to non meaningful profiles and very long ones that may include topics that are not relevant anymore for the user.

Our profiling algorithm characterizes the session s_u^T with a value between 0 and 1 for each one of the 328 possible categories. The categorization of the session, denoted as $c^{s_u^T}$, is used to retrieve a number of related ads a user might be interested in. To do so, we compute the 20-nearest neighbors of $c^{s_u^T}$ (according to Euclidean

[3] https://adaway.org/hosts.txt

[4] https://hosts-file.net/ad_servers.txt

[5] https://pgl.yoyo.org/adservers/serverlist.php?hostformat=hosts&showintro=0&mimetype=plaintext

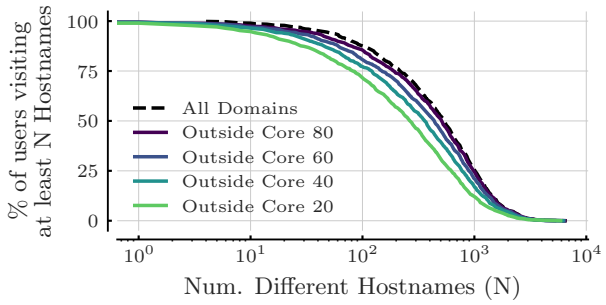


Figure 2: User diversity (hostnames). We identify *cores* of hostnames visited by large fractions of users (e.g., Core 80 is the set of hostnames visited by at least 80% of the users) and show the CCDF of the number of visited hostnames outside of each core. The dashed line shows the CCDF of the total number of visited hostnames. For reference, we also report the size of each core: Core 80, 60, 40 and 20 have sizes 30, 120, 271 and 639, respectively.

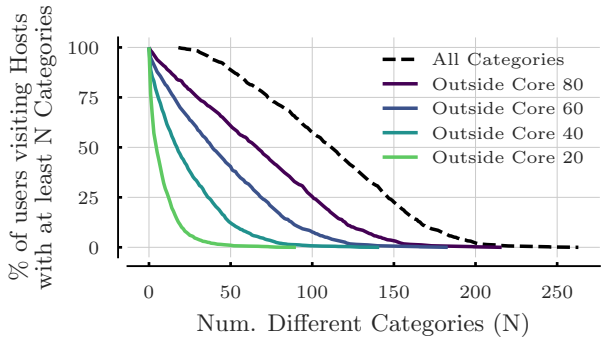


Figure 3: User diversity (categories). We identify *cores* of categories assigned to large fractions of users (e.g., Core 80 is the set of categories assigned to at least 80% of the users) and show the CCDF of the number of visited categories outside of each core. The dashed line shows the CCDF of the total number of assigned categories. For reference, we also report the size of each core: Core 80, 60, 40 and 20 have sizes 47, 80, 124 and 177, respectively.

distance) from the pool of hosts for which we know their categorization, previously denoted as H_L. We then select ads for each of the closest hosts and serve such ads to the user for the next 10 minutes.

6 RESULTS

In this section we analyze the results obtained during the one-month profiling period of our study. During this phase, we witnessed 75M connections to 470K hostnames; users received 270K ads and we replaced 41K of them.

6.1 User Diversity

The first question we try to answer is whether visited hostnames provide any insights on user profiles. During our data collection, a few hostnames (e.g., ones related to Google or Facebook) were extremely popular. If all users visit the same set of hostnames, then browsing habit is probably not a good discriminant for user interests. To shed light on this matter, we try to identify *cores* of hostnames visited by large fractions of users, and then compute how many users visit hostnames outside of those cores. In a nutshell, hostnames in a core are essentially *background noise* while hostnames outside of a core are the ones that allow a profiler to tell user interests. Figure 2 shows the CCDF (survival function) of the number of visited hostnames outside of a set of cores. We use "Core XX" to denote the set of hostnames visited by at least XX% of the users; for example, "Core 80" is the set of hostnames visited by at least 80% of the users. Further, the dashed line of Figure 2 shows the CCDF of the total number of visited hostnames. The figure shows that 75% of the users (the dashed black line represents all the users) visit at least 217 hostnames and one fourth of them visit up to 1015 different hostnames. Roughly 25% of the users have visited at least 985 hostnames outside Core 80; similarly, 75% of them have visited at least 191 hostnames outside Core 80.

We repeat the experiment on cores by using categories in place of hostnames. This is because profiles are eventually computed from website categories—hence heterogeneity of user profiles should be assessed based on website categories rather than websites. Similar to the previous experiment we identify cores of categories—that is, categories assigned to a large fraction of users—and try to understand how many users are assigned categories outside of such cores. Figure 3 shows the CCDF of the number of categories visited by users outside of different cores.[6] That figure shows that all users in our experiment are assigned the same 14 categories. Such categories have no profiling value as they provide no information to discriminate one user from another. Those common categories are the ones assigned to very popular hostnames such as *google.com* or *facebook.com*. Further, 50% of the users are assigned with the same 113 categories out of the 348 possible. Finally, 1,5%, 5,2%, 11,1% and 23,2% of the users are not assigned with a single category outside the cores 80, 60, 40 and 20, respectively.

6.2 Qualitative analysis of the embeddings

The accuracy of the profiling system heavily relies on the algorithm described in Section 4.1. Here we provide insights on the results produced throughout our profiling experiment.

For the sake of visibility, we focus on data (i.e., visited hostnames) collected during a single day. Further, we only use second-level domain names instead of complete hostnames. For example, given hostnames *mail.google.com* or *ds-aksb-a.akamaihd.net*, we only consider *google.com* or *akamaihd.net*, respectively. With such design choices, we reduce the number of points in our space from roughly 470K to less than 3K. We stress that the strategy just described is only used to improve the readability of the current section (and its figures), whereas all other experiments used the complete one-month dataset and considered full hostnames.

[6] Differently from Figure 2, in Figure 3 we use a linear scale for the X axis. This is because when mapping hostnames to categories, we shrink the set from more than 400K hostnames to 328 categories.

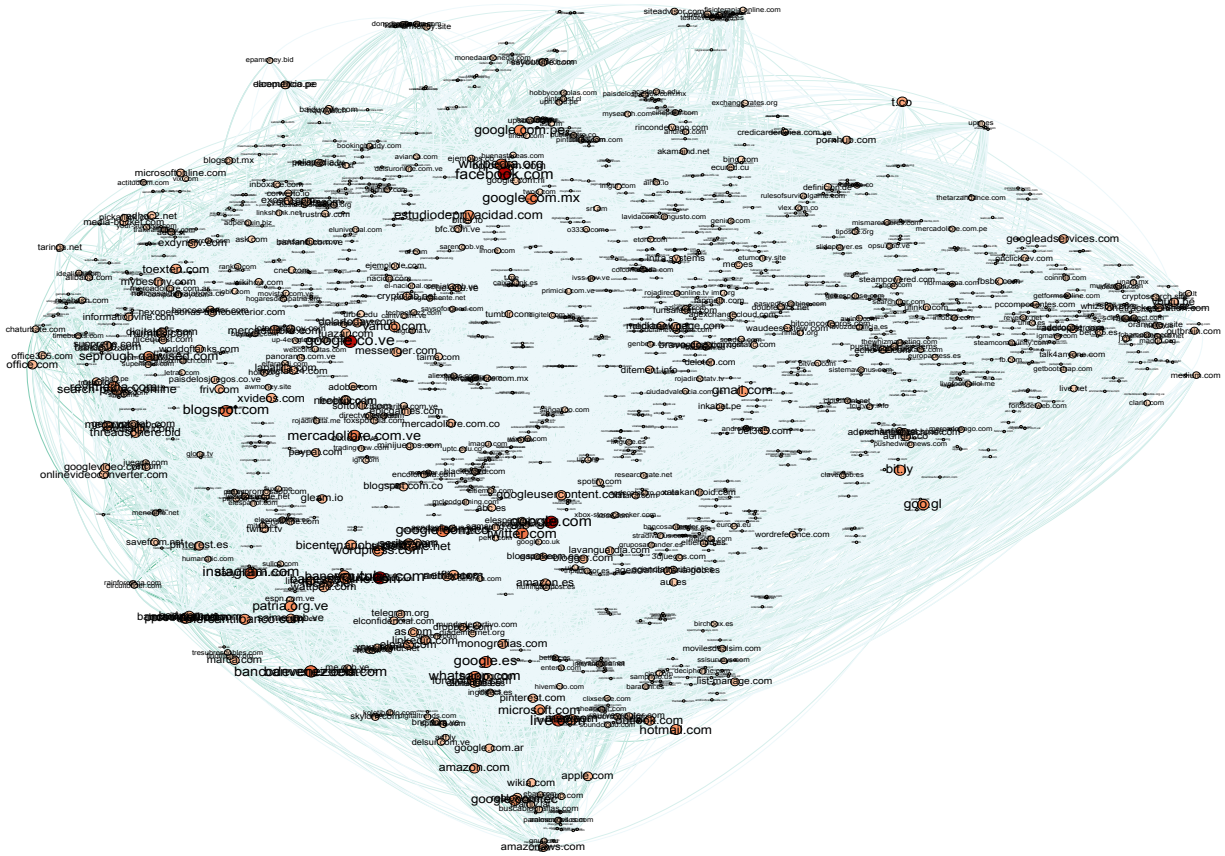


Figure 4: t-SNE (2 dimensions) representation of the embeddings obtained for each of the hostnames visited by users. A zoomable version is available at https://bit.ly/2LMOEP2.

Finally, we apply the t-SNE[7] [28] algorithm over the embeddings in order to reduce the dimensionality of our space from 100 to 2 dimensions.

Figure 4 shows a 2D representation of the embeddings. Each circle represents the embedding of a second level domain and its center is the position of that embedding in the space after applying t-SNE. The size of a circle is proportional to the number of users that visited the corresponding domain. A link between two circles means that the two domains were *co-requested*, i.e., visited one after the other by at least one user; the thickness of the link is proportional to the number of users that co-requested the two domains. A zoomable version of Figure 4 can be found at https://bit.ly/2LMOEP2.

Figure 5 magnifies three areas of Figure 4. We have chosen those three areas as they serve as clear examples of how the algorithm is able to learn similarities among hostnames.

The top rectangle (marked with a red 1) focuses on a set of very clustered hostnames. A closer look reveals that they are porn-related websites. Our algorithm identifies those hostnames as similar even when most of them were not co-requested (i.e., *spankwire.com* and *livejasmin.com* were co-requested, but they were never co-requested with any other of the hostnames in the cluster.).

The second rectangle (marked with a red 2) focuses on a small area in the middle of our representation space that includes hostnames such as *rojadirecta.me*, *arenavision2018.tk*, *directvplay.com* or *foxplay.com*. All these hostnames stream sport events (both in a legal or illegal way). We speculate that our algorithm could be used to identify websites hosting illegal streaming websites as those services frequently move to new hostnames in order to evade justice.

Finally, the third rectangle (marked with a red 3) presents an intricate case. It magnifies an area in the center of the representation space with lots of hostnames close to each other. The right part of the area shows websites like *atrapalo.com*, *skyscanner.es*, *ryanair.com*, *vueling.com* or *lastminute.com*. All of them are traveling-related websites. However, in this area we also find websites not related to traveling like *betfair.es* (a betting website) or *enterat.com* (a Spanish web portal). A closer look at the data reveals that such result is an artefact of dimensionality reduction via t-SNE: the similarity matrix for the area under examination—using all of the 100 dimensions—shows that travel-related hostnames are far from the other hostnames. Finally, the third rectangle also shows that our algorithm can cluster

[7]The t-SNE algorithm is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.

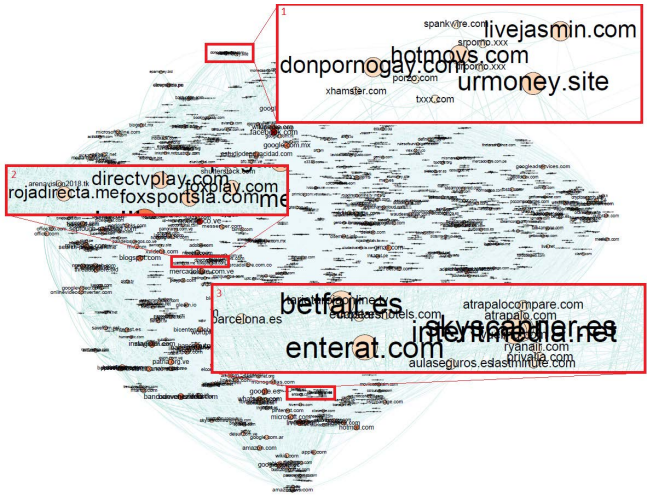


Figure 5: Example of clusters of hostnames present in *Figure 4*: cluster 1 includes porn-related websites, cluster 2 includes websites used to watch sport events, and cluster 3 includes travel-related websites.

embeddings of hostnames visited by specific user demographics (in this case, Spanish users).

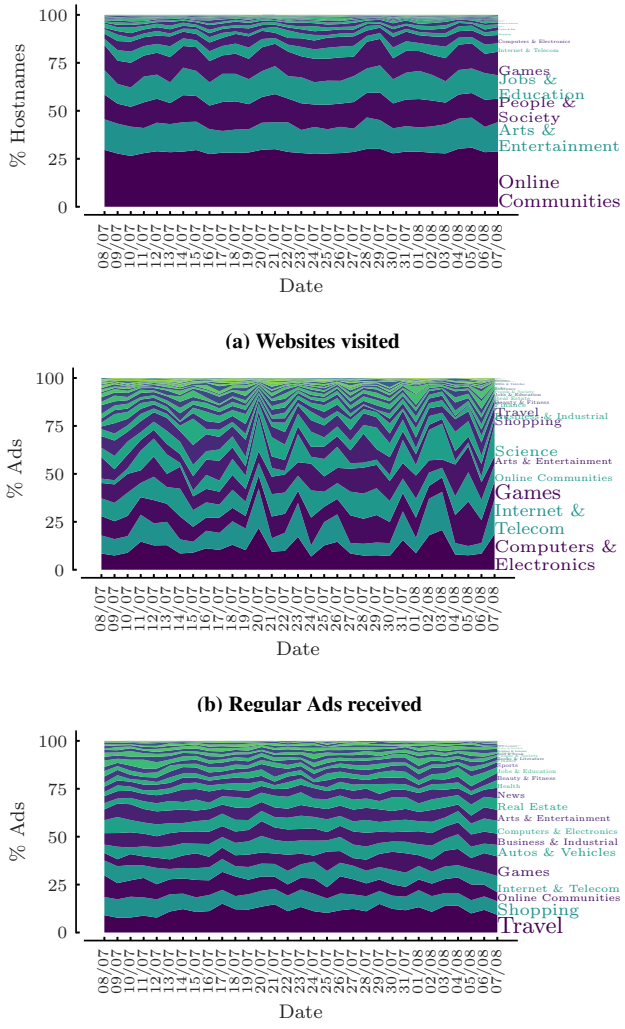
6.3 Hostnames and Ads Analysis

We analyze the topics of the hostnames requested by our users and compare them with the topics of the ads served by both our system and ad-networks. We only take into account hostnames or ads for which Google Adwords returned an answer (roughly 50K out of 470K among hostnames of visited webpages and ads).

Figure 6 shows the percentage of hostnames requested and ads received for each topic across one month. As explained in Section 5.4, we use the first two levels of the hierarchy provided by Google Adwords, thereby considering 328 different topics. Nevertheless, Figure 6 only uses top-level topics (34 in total) to ease readability.

Figure 6a shows that hostname topics like *Online Communities*, *Arts & Entertainment*, *People & Society* or *Jobs & Education* are very prominent and stable across time. A closer look at our data reveals that very popular websites like Facebook or YouTube are labelled with those topics. While very popular, those hostnames carry little information for a network observer because such websites provide very diverse services and attract a wide range of user types. For example, on YouTube one may find food recipes as well as reviews of electronic goods. The sole indication that a user is browsing YouTube is simply not enough to tell that user interests.

Figure 6b and Figure 6c shows that ads served by our system and those served by ad-networks belong to different categories. This may partially be explained by the fact that ads served by ad-networks include also premium ads, retargeting, massive campaigns, etc. Also, the fact that topics of ads by ad-networks change over time may be explained by the occurrence of ad campaigns. Finally we note that topics that are prevalent in Figure 6a are not so popular in Figure 6b and Figure 6c. This is because Figure 6a shows the number of connections to a webpage and one visit to pages like YouTube generates many connections.



(a) Websites visited

(b) Regular Ads received

(c) Ads selected by our algorithm

Figure 6: Topics of the websites visited or the ads received by users per day.

6.4 Click Through Rate

Finally, we compare the CTR of ads served by our system with the one of ads served by ad-networks. Ads served by our system show a CTR of 0.217% whereas ads served by ad-networks have a CTR of 0.168%.

One may wonder whether a CTR around 0.2% is any good. We note that little is known about actual CTRs of ad-campaigns. Several specialized blogs report CTRs between 0.07% and 0.84 [29–31]. Our results (for both ads served by our systems and ads served by ad-networks) are within the lower part of this range. There could be different reasons for this. One possible reason is the bias induced by our sample of the population (mainly young people willing to install an extension on their browser). Another reason may be the pervasiveness of ad-fraud [32] that amplifies CTRs of ad-campaigns.

Statistical significance. We use hypothesis testing to determine whether our results bear statistical significance. As our study par-

ticipants received both types of ads (the ones sent by ad-networks and the ones picked by our profiling algorithm) we used a two-tailed paired t-test with $p < .05$ to assess the mean difference of CTRs. Resulting p-value was .11333 so we conclude that there is no statistical difference between the mean CTR of ads provided by ad networks and the mean CTR of ads shown according to our profiling algorithm. In other words—and if we assume CTR to be a meaningful proxy of profiling quality—we can argue that profiling activity by a network observer may produce profiles that are as "good" as the profiles available to ad-networks or OTT.

7 DISCUSSION

7.1 Limitations

Our sample of participants may not be representative of the population of Internet users—a common problem of studies with real users. Yet, we consider our results to constitute evidence of the profiling ability of networks eavesdropper.

Another limitation of our experiment is the set of ads we used. The latter was built during the data collection phase and was used to serve ads during the profiling phase. Hence, the set was static and some ads could have become outdated at the time they were served. Differently, the set of ads served by ad-networks is ever-changing and up-to-date.

Finally, we used CTR to compare profiles that could be built by an eavesdropper and the ones available to ad-networks. Nevertheless, some ad-campaigns could be optimized to improve metric like final revenue rather than CTR.

7.2 Real world observations

Our study assumes that a network observer can obtain all hostnames a user visits. Here we discuss to which extent such assumption holds in real-world deployments.

HTTPS and QUIC. Both HTTPS and QUIC leak to a network observer the hostname requested by the user in the Server Name Indication (SNI) field. Even if the SNI field is sent during the handshake and the connection may be long lasting, an eavesdropper may obtain the hostname of the server (by tracking the TCP flow in HTTPS or checking the UDP datagrams of QUIC). Note the algorithm used in our experiment does not consider multiple requests to the same hostname, thus, our experiment perfectly mimics the information a network observer could obtain when by observing HTTPS or QUIC traffic. New protocols like TLS 1.3 may use encrypted SNI but do not hide the IP address that may be used by the profiling algorithm.

Multiple Users. If requests for hostnames generated by different users are ascribed to a single one, the profiling accuracy is clearly affected. The ability of an eavesdropper to tell apart traffic generated by multiple users depends on its position in the network. A WiFi provider could match each requested hostname to a device by using MAC addresses. A mobile provider would separate traffic per user by leveraging MSISDN or IMSI identifiers. Differently, a landline ISP may not be able to tell apart traffic generated by multiple users behind the same NAT device (e.g., a domestic router).

DNS providers. A DNS provider may actually act as a profiler since it learns the hostnames requested by a user via DNS requests. Techniques like DoH or DoT limit the visibility of network and VPN providers to the IPs connected by the user unless they use some complex system[33]. However, they would not prevent the DNS provider itself from learning the hostnames a user visits.

7.3 Using profiles

Even if TLS prevents an eavesdropper from injecting ads into the connection, we argue that profiling users could still be a lucrative business for network observers. The ad industry is fragmented in hundreds of companies that do business even without a direct channel with the final user. Profiles could be sold to third parties or direct ads could be sent via email or SMS messages. Finally, we note that many ISPs are entering the online advertising business [34, 35] and may leverage their unique perspective to create accurate users profiles.

7.4 Countermeasures

Most anti-tracking applications try to fight the profiling activities of ad-networks and over-the-top providers. This is usually achieved by blocking connections and cookies towards domains that belong to trackers and other stakeholders of the targeted advertising ecosystem. This mechanism is not effective against potential profiling activity of a network eavesdropper that simply leverages the hostnames a user requests. This information is leaked despite TLS and, as we have argued above, upcoming patches like encrypted SNI are not likely to solve the issue. We do not consider VPNs as a valid solution as it simply shifts the threat from the WiFi provider or ISP to the VPN provider. Preventing leaks to a network observer requires tools like TOR. Nevertheless, TOR is not immune to weaknesses that expose the privacy of its users [36, 37] and it has a performance/usability penalty that not all users or application may tolerate.

8 CONCLUSIONS

User profiling for targeted advertising is raising concern among researchers and Internet users. However, most research activity and privacy-enhancing apps like ad-blockers focus on profiling by ad-networks and over-the-top providers. There seems to be little interest in what is leaked about a user browsing activity to network observers, and this may be because of the pervasiveness of TLS.

In this paper we have shown that a network observer can effectively build user profiles by leveraging information leaked by TLS connections. In particular, we introduce an algorithm inspired by natural language processing to infer the relevant topics of a website, even when its hostname is not categorized by available ontologies. Further, we have provided evidence that the quality of the profiles that can be built using only network information is comparable to the quality of profiles available to ad-networks. This task was carried out by means of a 6 month experiment involving more than 1.3K users.

Our findings show that user profiling by network eavesdroppers is effective despite TLS. This is especially worrisome as anti-tracking tools such ad-blockers are not effective against a network observer whereas countermeasures like TOR incur in a performance and usability penalty.

ACKNOWLEDGMENTS

We thank anonymous reviewers and our shepherd for their valuable comments and suggestions. We also thank Rosa Lillo for her help. This work has been partially supported by the European Union

through the H2020 PIMCity (871370) project. Costas Iordanou acknowledges support by the TV-HGGs project (OPPORTUNITY/0916/ERC-CoG/0003), co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation.

REFERENCES

[1] "Data Transparency Lab." http://datatransparencylab.org, 2015.

[2] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan, "Adreveal: improving transparency into online targeted advertising," in *Twelfth ACM Workshop on Hot Topics in Networks (HotNets)*, pp. 12:1–12:7, 2013.

[3] J. M. Carrascosa, J. Mikians, R. Cuevas, V. Erramilli, and N. Laoutaris, "I Always Fell Like Somebody's Watching Me. Measuring Online Behavioral Advertising," in *CONEXT*, 2015.

[4] R. Gonzalez, C. Soriente, and N. Laoutaris, "User profiling in the time of https," in *Proceedings of the 2016 Internet Measurement Conference*, pp. 373–379, ACM, 2016.

[5] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431, Association for Computational Linguistics, April 2017.

[6] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner, "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016," in *USENIX Security Symposium*, 2016.

[7] E. Steven and A. Narayanan, "Online Tracking: A 1-million-site Measurement and Analysis," in *ACM CCS*, 2016.

[8] J. R. Mayer and J. C. Mitchell, "Third-Party Web Tracking: Policy and Technology," in *IEEE Symposium on Security and Privacy*, 2012.

[9] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft, "Breaking for Commercials: Characterizing Mobile Advertising," 2012.

[10] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill, "Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem," in *NDSS*, 2018.

[11] C. Leung, J. Ren, D. Choffnes, and C. Wilson, "Should You Use the App for That?: Comparing the Privacy Implications of App- and Web-based Online Services," 2016.

[12] B. Reuben, L. Ulrik, M. V. Kleek, J. Zhao, T. Libert, and N. Shadbolt, "Third Party Tracking in the Mobile Ecosystem," *CoRR*, 2018.

[13] P. Vallina, A. Feal, J. Gamba, N. Vallina-Rodriguez, and A. F. Anta, "Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem," in *Proceedings of the Internet Measurement Conference*, IMC '19, (New York, NY, USA), p. 245–258, Association for Computing Machinery, 2019.

[14] M. Pachilakis, P. Papadopoulos, E. P. Markatos, and N. Kourtellis, "No more chasing waterfalls: A measurement study of the header bidding ad-ecosystem," in *Proceedings of the Internet Measurement Conference*, IMC '19, (New York, NY, USA), p. 280–293, Association for Computing Machinery, 2019.

[15] R. Li, C. Wang, and K. C.-C. Chang, "User profiling in an ego network: Co-profiling attributes and relationships," in *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, 2014.

[16] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 251–260, ACM, 2010.

[17] V. Kumar, D. Khattar, S. Gupta, M. Gupta, and V. Varma, "User profiling based deep neural network for temporal news recommendation," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2017.

[18] G. Alotibi, N. Clarke, F. Li, and S. Furnell, "User profiling from network traffic via novel application-level interactions," in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2016.

[19] L. Partners, "Display LUMAscape." https://lumapartners.com/content/lumascapes/display-ad-tech-lumascape/. "[Online; accessed 13-May-2019]".

[20] A. Pastor, R. Cuevas, Á. Cuevas, and A. Azcorra, "Establishing trust in online advertising with signed transactions," *IEEE Access*, vol. 9, pp. 2401–2414, 2020.

[21] A. Datta, M. C. Tschantz, and A. Datta, "Automated experiments on ad privacy settings," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 1, pp. 92–112, 2015.

[22] Google AdWords. https://adwords.google.com/, 2018.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[24] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, pp. 2177–2185, 2014.

[25] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pp. 49–62, ACM, 2009.

[26] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

[27] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, ELRA, 2010.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[29] K. Volovich, "What's a Good Clickthrough Rate? New Benchmark Data for Google AdWords." https://blog.hubspot.com/agency/google-adwords-benchmark-data/. "[Online; accessed 13-May-2019]".

[30] WordStream, "Average CTR (Click-Through Rate): Learn How Your CTR Compares." https://www.wordstream.com/average-ctr. "[Online; accessed 13-May-2019]".

[31] R. Hof, "Study: Mobile Ads Actually Do Work - Especially In Apps." https://www.forbes.com/sites/roberthof/2014/08/27/study-mobile-ads-actually-do-work-especially-in-apps/. "[Online; accessed 13-May-2019]".

[32] M. Marciel, R. Cuevas, A. Banchs, R. González, S. Traverso, M. Ahmed, and A. Azcorra, "Understanding the detection of view fraud in video content portals," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 357–368, International World Wide Web Conferences Steering Committee, 2016.

[33] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted dns–> privacy," *A Traffic Analysis Perspective (Proc. of the NDSS)*, 2020.

[34] M. Ingram, "Here's Why Verizon Wants to Buy Yahoo So Badly." https://fortune.com/2016/04/19/verizon-yahoo/. "[Online; accessed 24-Sep-2019]".

[35] AT&T, "AT&T Launches New Advertising Company, Xandr." https://about.att.com/story/2018/att$_{l}$aunches$_{x}$andr.html. "[Online; accessed 24-Sep-2019]".

[36] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, "Users get routed: Traffic correlation on tor by realistic adversaries," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 337–348, ACM, 2013.

[37] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal, "Raptor: Routing attacks on privacy in tor," in *24th USENIX Security Symposium, pages=271–286, year=2015*.