Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2014, Vol. 6, No. 2

Evaluation of a Query-Obfuscation Mechanism for the Privacy Protection of User Profiles

José Estrada-Jiménez
Dept. of Telecommunications and Information Networks, Escuela Politécnica Nacional
C/Ladrón de Guevara E11 - 253. Quito, Pichincha (Ecuador)
Tel: +593-22507144 ext. 2304 E-mail: jose.estrada@epn.edu.ec

Ana Rodríguez
Dept. of Telecommunications and Information Networks, Escuela Politécnica Nacional
C/Ladrón de Guevara E11 - 253. Quito, Pichincha (Ecuador)
Tel: +593-22507144 ext. 2304 E-mail: ana.rodriguez@epn.edu.ec

Javier Parra-Arnau
Dept. of Telematics Engineering, Universitat Politècnica de Catalunya (UPC)
C. Jordi Girona, E-08034. Barcelona (Spain)
Tel: +34 93 4011871. E-mail: javier.parra@entel.upc.edu

Jordi Forné
Dept. of Telematics Engineering, Universitat Politècnica de Catalunya (UPC)
C. Jordi Girona, E-08034. Barcelona (Spain)
Tel: +34 93 4011871. E-mail: jforne@entel.upc.edu

Received: April 29, 2014 Accepted: June 8, 2014 Published: June 30, 2014

DOI: 10.5296/npa.v6i2.5395 URL: http://dx.doi.org/10.5296/npa.v6i2.5395

Abstract

Traces related to our identity are left every day while we browse the Internet. Being the user's information a very valued asset for most of the companies, user activities on Internet are permanently monitored, and the information obtained from this process is used by big advertising companies. Accurate user profiles are built based on web searches, tags, tweets and even clicks issued by users. Collecting and processing this huge amount of personal data represents a serious risk for the user's privacy but most of people are not aware of such risk.

We describe in this paper a way to measure the effectiveness of a query obfuscation method. Since privacy level of user profiles is only estimated theoretically in some previous work, we firstly create a privacy risk measuring tool in a Firefox add-on which we named PrivMeter. Besides warning the user about his privacy levels, this tool is especially useful to evaluate the obfuscation mechanism offered by another very well-known add-on called TrackMeNot. We find that, for identification attacks, TrackMeNot importantly improved the user's privacy. However, against more sophisticated attacks, such as classification attacks, the obfuscation mechanism was not successful enough.

Keywords: privacy evaluation, privacy protection, query obfuscation, profiling, Shannon's entropy, Kullback-Leibler divergence, user profile.

1. Introduction

User profiling and classification are nowadays very common practices thanks to the significant advances on data analysis techniques. Content personalization systems use all the information brought by the user to provide services that improve the user experience on the Internet. Users sometimes suspect these processes are being performed, but are usually unaware of the involved privacy risk that personalization brings.

Privacy is the price that people have to pay for using personalized recommendation systems that provide content and advertising. The success of such personalization is based on the accuracy of user profiles which are built from user browsing patterns. Traces left by users on the Internet, even when obfuscated, could reveal very sensitive information related to personal preferences [1] [2].

The information susceptible to be analyzed by adversaries goes from the content of visited web sites, browsing time, number of clicks, search queries, data delivered through web forms, cookies and even specific characteristics of the web browser configuration [3][4].

In this context, search engines, recommendation platforms, social networks and tagging systems are potential adversaries that collect user activity information. These data are interesting not only for advertising companies but particularly for powerful governments that are able to compel huge companies to “collaborate” with them.

As if this were not enough, users are completely unaware about the risks to which they are exposed on Internet. People immediately accepting privacy policies offered by different services on Internet is a proof of this.

Special attention has to be put on search engines and social networks which are becoming the Internet gateways at application level to help users to reach basic services such as blogs, news sites, etc. Most users connect to these services by querying Google or simply by following links published on personal Facebook walls. Nowadays, every web service collects search queries, labels, clicks and more metadata that can easily be mapped [2] to personal identities. Even when this data is exposed to anonymization processes, some investigations ([1] and [5]) show that privacy can still be compromised, so there is no warranty in the application of these countermeasures.

Privacy is part of a highly complex context that depends on the individual interests of the user. From our perspective, the shortest way to protect privacy is to increase the level of awareness of the user, by highlighting the strengths and weaknesses of their behavior on the Internet. The problem is that there are no tools to deliver this information (privacy level). Certainly, there are some measures that implement obfuscation or blockage to the flow of personal information, but it is unclear what their actual effectiveness is. Privacy and therefore privacy protection tools are related to a multi-dimensional concept, and, therefore, very difficult to be measured or evaluated. It is unclear, then, whether existing tools really reduce these risks, and this is the reason why measuring the effectiveness of these mechanisms is essential to compare them and to decide which to use in certain user environments.

1.1 Contribution

Privacy is a concept whose extent depends on multiple dimensions which are commonly related to the adversary been faced and even the interests and thoughts of the user whose privacy needs to be protected. Several mechanisms have been proposed, both theoretical and practical, aiming to protect the user privacy by employing heuristics and not considering the multiple dimensions of privacy. TrackMeNot (TMN) [15] is one of the most popular of such mechanisms and works by forging search queries in an attempt of obfuscating a user profile.

The main contribution of this work is the evaluation of the actual success that TrackMeNot has when obfuscating user profiles. As justified in [15], entropy and KL divergence are used as suitable privacy metrics for user profiles. Specifically, we compare the privacy level of the actual user profile with the apparent user profile built after generating bogus queries with TMN. We test different forgery strategies available in TMN to show if it is effective in increasing the user privacy. To the best of our knowledge this is the first attempt to quantitatively assess the level of privacy protection achieved after using a heuristic obfuscation tool such as TMN.

Since the privacy level of a user profile is relative to the context around the user (adversaries, for example), we suppose that the success of any protection tool is linked to such context. Hence, in front of different adversary models, TMN would offer different levels of privacy protection to a user. Our contribution, then, is to determine the capability of TMN to effectively obfuscate a user profile when he faces different adversary models. Moreover, query forgery is a mechanism whose intensity is tunable (amount of forged queries), but we suspect that a focused and smart process of forging would have better results than just generating millions of forged queries. In this sense, we also contribute on measuring the impact of the amount and the source of forged queries when obfuscating user profiles with TMN.

1.2 Organization

This paper is organized as follows. Sec. 2 explores some existing tools and mechanisms which offer privacy protection services. Sec. 3 summarizes the adversary models and the metrics used to determine the privacy risk levels. Sec. 4 describes the architecture and the modules part of our privacy measuring add-on and some implementation details. Sec. 5 describes the evaluation process of a privacy protection mechanism in the browser and, finally, Sec. 6 exposes some conclusions about our work.

2. State of the Art

Currently, there are some tools that try to protect user privacy on the Internet, essentially by means of blocking traffic that transports personal information. These mechanisms, generally based on heuristics, do not measure the user privacy risk nor evaluate the level of protection they offer.

2.1 Privacy Enhancing Tools (PETs)

PETs are technical means to protect user privacy [6]. Privacy is a wide concept that involves various approaches, from the intrinsic characteristics of communication traffic to the content of transmitted messages. Then, PETs can be classified in basic anti-tracking technologies, cryptographic methods, third-trusted-party, user collaboration and data perturbation based approaches.

2.1.1 Basic Anti-tracking Technologies

Tracking is a mechanism by which an entity identifies another one in a communication process. This is vital for personalized services since tracking allows them to map identities to their corresponding preferences. For this, there are various parameters that identify an entity, such as an IP address or a cookie.

Blocking or "hiding" these identity parameters is part of the basic anti-tracking mechanisms. The problem of doing this is related to the fact that some Internet services cannot be offered if these blocking strategies are implemented.

2.1.2 Private Information Retrieval (PIR)

PIR allows a user to retrieve information from a database with the provider ignoring the content of the retrieved information [7]. A simple implementation of PIR, but not very practical, could be that the user downloads the entire database and then, locally, accesses to the content of interest.

Another option, proposed for recommendation systems, is to reveal aggregated profiles obtained from a group of users, instead of revealing individual profiles.

2.1.3 Trusted Third Party (TTP) based Mechanisms

A TTP is an intermediary entity that receives user requests and forwards them (to the destination) on behalf of the user, in an attempt to anonymize communications. Messages in the destination seem to have been originated by the TTP, so that they ideally may not be linked to the user. Traffic bottlenecks, however, are a potential issue with TTP solutions.

Mixes [8] are TTP implementations that receive a message and forward it to its destination in such a way that the arriving event cannot be associated with the leaving one. This helps to prevent tracking which is executed by "listening" forwarding events in order to follow a message from its source to its destination.

Onion Routing is also a TTP-based implementation that forwards a received message which was previously ciphered a number of times in the source router. As the message goes to its destination its ciphering layers are decoded one by one.

2.1.4 User collaboration

There exist a myriad of alternatives based on user collaboration. One of the most popular is the one that implements cooperative participation among users in a system in order to gain privacy. *Crowds* [9] and the protocol for private LBS [10] are two examples of mechanisms that take advantage of the participation of various entities to route the information in an unpredictable and, then, anonymous manner.

Closely inspired by Crowds, [11] proposes a protocol that enables users to report traffic violations anonymously in vehicular ad hoc networks. This protocol differs from the original Crowds in that, first, it does take into account transmission losses, and secondly, it is specifically conceived for multi-hop vehicular networks, rather than for wired networks.

Also in the case of lossy networks, [12] provides a mathematical model of a Crowds-like protocol for anonymous communications. The authors establish quantifiable metrics of anonymity and quality of service, and characterize the trade-off between them.

In [13] a mechanism is proposed to provide privacy in web searches. It consists on users exchanging search queries before sending them to their destination, in such a way that real user profiles get obfuscated.

Another protocol for enhancing privacy in communications, also relying on user collaboration and message forwarding, is [14]. The objective of the cited work is to hide the relationship between user identities and query contents even from the intended recipient, an information provider. The main difference with respect to the Crowds protocol is that instead of resorting to probabilistic routing with uncertain path length, it proposes adding a few forged queries.

2.1.5 Data Perturbation

Data perturbation aims at hindering an attacker in its efforts to build a precise user profile, for example, by sending fake data combined along with genuine data.

Query forgery is an application of this technology where forged queries are generated from the client, so that the search engine cannot derive an accurate user profile due to the fact that the received queries are obfuscated.

TrackMeNot [15] is a well-known implementation of query forgery. It is a browser extension that generates fake search queries and sends them to different search engines. These queries are derived from RSS content hosted in different information sources.

Another proposal at application level in the browser is *GooPIR* [16]. This tool obfuscates each query that a user sends to Google and, to get this objective, GooPIR uses words locally hosted. However, the obfuscation of sensitive search queries related, for example, to health or politics affinity, is very difficult.

Perturbation proposals have also appeared with the objective of increasing privacy at recommendation systems, such as those based on ratings sharing. In [17], for example, an algorithm is proposed to send perturbed rating data to the recommendation system.

The combined usage of both strategies, that is, forgery and suppression, is studied in the scenario of personalized recommendation systems [18]. With the adoption of those strategies, users may wish to submit false ratings to items that do not reflect their preferences, and/or refrain from rating certain items they have an opinion on.

The trade-off posed by these perturbative strategies in terms of privacy protection and data utility is investigated analytically in [19]. The authors find a closed-form solution to the problem of optimal simultaneous forgery and suppression of ratings, and evaluate their approach in the real-world recommender Movielens [20].

2.2 Privacy Protection Oriented Tools

In this section, we examine several state-of-the-art privacy tools that aim at protecting online user privacy, mainly by blocking browser functions that facilitate the release of personal information. Usually based on heuristics, these tools do not inform users of their level of privacy nor evaluate the extent to which their private data is protected.

Adnostic [21] is a browser add-on developed for Mozilla Firefox. It implements an architecture that deploys personalized advertising without compromising user privacy. The browser add-on decides which ads to show, based on a user profile locally calculated. This profile is obtained by processing the queries that users send and the content of the web sites they visit. This information is then classified by means of natural language processing in the browser. The ads, part of previously downloaded sets, are deployed depending on the user interests.

REPRIV [22] is another system developed to work in the browser. It offers an enhanced personalization of content and a control mechanism over the information delivered to third parties by the user. REPRIV employs user browsing information in order to discover which his interests are, and to communicate these details to third parties so that the content they deliver can be personalized depending on the user preferences. It promises a significant improvement in the provision of tailored content, thanks to highly detailed browser information. However, the privacy control may be affected by the lack of usability that protection policies may add to the tool for an average user. Proposed by Microsoft, REPRIV is an interesting approach, although the protection of privacy is offered as an added-value that implicitly supposes trust in a third.

A couple of proposals ([23] and [24]) have been done about measuring privacy of users in social networks (Facebook in both cases). These proposals describe some mechanisms to determine the risk of user privacy depending on the amount of information that can be inferred about the user from the relationship he maintains with others

TrackMeNot (TMN) [15] is another tool to protect privacy in the premises of the web browser. It proposes the obfuscation of query flows sent by the user to search engines. This is done by generating false queries. TMN has received much criticism regarding its effectiveness, although not many mechanisms have been proposed to do this evaluation. In [25], it is shown that these false queries could be easily identified using artificial intelligence-based classifiers. Undoubtedly, the lack of a tool for measuring user privacy prevents the user to realize his risk condition before and after applying a protection strategy like TMN.

Google Sharing [26] is another tool that implements a privacy protection mechanism by preventing the user tracking done by Google by means of analyzing user search queries. The mechanism consists on an external proxy that manages a group of identities that are associated to cookies. These cookies replace the ones on the original HTTP requests, masking, consequently the user identity. These forged cookies are finally sent to Google along with the original request. Even when there is the option to send ciphered requests from the user, the user privacy can be compromised if there is collusion among the proxy server and Google.

Ghostery [27] is another Firefox add-on created to protect user privacy by detecting and blocking trackers and other objects dedicated to track user activity. It is a very complete and popular tool, equipped with various modules that implement protection mechanisms in different web browsers.

Private navigation mode is also a privacy protection option for most known browsers. This option disables local storage of user information (history, images, videos, cookies, etc.) during web browsing. This significantly complicates access to many websites, so those who use this mode activate it during very short time intervals. The protection is limited to the local level.

Blocking or deactivating some characteristics in the web browser is a common mechanism implemented by various applications in the form of browser plugins (NoScript [28], AdBlock Plus [29], DoNotTrackMe [30]), and they help to avoid information release that may be used to identify the user.

However, none of these tools evaluates the user level of privacy. TMN is the only one that implements a proactive mechanism to protect privacy, but does not propose a way to measure its success, which discourages its usage. In general, only advertisers and social networking services are considered as adversaries, but Internet Service Providers (ISPs) are the entities that more information have about users, especially if client-server connections are not encrypted using HTTPS. In fact, ISPs have access to all the information that users send to Internet and its very detailed content represents an enormous stimulus for its commercialization.

In [31], [32], [33] and [34], the authors describe some mechanisms that could be employed to protect privacy in environments where the user queries or labels content; also considering the cost of these strategies which is reflected in the loss of usefulness of the data, loss of functionality of a service or the additional resource consumption. Included among these mechanisms are query forgery or label suppression in order to show a distorted version of the user profile that the attacker cannot exploit. Optimization of these mechanisms and their impact are also studied.

3. Adversary Models and Privacy Metrics

In this section we describe the two adversary models used throughout this work. Based on these models, we define metrics to evaluate user privacy. User models and privacy metrics are widely justified in [35].

3.1 Adversary Models

The proposed privacy metrics rely on the assumption that user profiles are modeled as probability mass functions (PMFs), conceptually histograms of relative frequencies of user data across a predetermined set of categories of interest. This model assumes a very common representation in the personalized information services.

The adversary model allows defining the "attacker" properties, considering any entity capable of accessing to user information as an attacker, if his objective is to obtain the user profile and violate his privacy.

The characterization of the adversary is crucial to evaluate the effectiveness of a PET, given that the level of privacy provided is measured with respect to it. Depending on the properties of the adversary, the user could implement mechanisms to protect his privacy, for example, by applying one of the data-perturbative mechanisms explored in the state-of-the-art section.

We contemplate two objectives for an attacker, in particular identification and classification. These two objectives are described next:

- *Identification*, when the attacker tries to distinguish the user among the rest of the population, by detecting deviations of his interests with respect to the average profile of the population.
- *Classification*, when the attacker attempts to classify the user into a group of people by comparing the user's profile with the profile of a representative group.

Further details on the adversary model assumed here can be found in [20].

3.2 Privacy Metrics

In [35], Shannon's entropy and Kullback-Leibler (KL) divergence are justified as privacy metrics. Interpretations of these metrics will depend on the hypothesis done with respect to the adversary model.

Another more general metric, not limited to the privacy of user profiles, is the one proposed in [36]. In that paper, the authors propose to measure privacy as the estimation error of the adversary, and interpret, by using information and Bayesian decision theory based arguments, other metrics from the state of the art as particular cases of hers.

In order to facilitate comprehension, the main proposed definitions are exposed below, in order to justify the metrics used for measuring privacy. An interpretation of these metrics is also made to justify their usage as privacy level parameters.

Considering *H* as the Shannon's entropy and *D* as the KL divergence, the entropy *H(p)* of a discrete random variable X with probability distribution *p*, is a measure of its uncertainty, defined as in Eq. (1).

$$H(X) = -E \log\, p\,(X) = -\sum_x p(x) \log p(x) \qquad (1)$$

The KL divergence, also called, relative entropy *D* (p || q) between two probability distributions *p(x)* and *q(x)* over the same alphabet is defined as in Eq. (2).

$$D(p\,||\,q) = E_p \log \frac{p(X)}{q(X)} = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{2}$$

The KL divergence is a measure of discrepancy between probability distributions, ensuring that *D*(p || q)⩾0 with equality if, and only if p=q. Consequently, it is deduced that entropy *H(p)* reaches its maximum value at *H(u)* = log *n*, being *n* the cardinality of the finite alphabet over which *D (p || u)* is calculated, for a uniform distribution u, as it is stated in Eq. (3).

$$D(p\,||\,u) = \log n - H(p) \tag{3}$$

Specifically, according to the analysis made in [35], we have that the entropy maximization is a special case of divergence minimization, ideally reached when the distribution to be optimized is identical to the reference one.

Being q a user's interest profile, t a perturbed or modified version of this profile, and $\bar{t}$ the population's profile, the interpretations of Shannon's entropy and KL divergence as privacy metrics are shown in Fig. 1. These ideas are detailed, with respect to the attacker's objective, in the following sections.

3.2.1 Metrics against Identification

If the goal of the attacker is to identify the user, in the sense mentioned above, Jaynes' rationale behind entropy maximization methods enables us to justify and interpret divergence and entropy as measures of privacy.

The entropy of the user's apparent profile, that is, the profile observed by the attacker, is justified in [35] as a measure of the probability of this perturbed profile, in the sense of the frequency of occurrence of such profile in the population. Considering this probability of the user's profile as a reasonable measure of his anonymity (or privacy), the authors in [35] also justify the entropy as a measure of privacy. In brief, the higher the entropy of a profile, the higher the probability of this profile, and therefore the larger the population of users in which the user's interests are blended.

Furthermore, as it is observable in the first branch of Fig. 1, if the distribution of population's profile $\bar{t}$ is known, the divergence between the user's profile t and the population's profile is a metric of privacy, so that, the lower is this divergence, more private can be considered the profile.

To conclude, choosing the best apparent profiles in order to minimize the KL divergence improves the user anonymity. In simple words, a lower divergence corresponds to a higher frequency of occurrence of such profile, allowing the user to be unnoticed. When having a reference profile of the population, this is the same as maximizing the Shannon's entropy.

3.2.2 Metrics against classification

If the attacker's objective is to classify the user as a member of a particular group, the divergence is used as a metric of privacy, according to the analysis made in [35], from hypothesis testing and the method of types. As shown in Fig. 1, in the second branch of the tree, if the profile of group *g* is unknown by the user, an alternative is to maximize the divergence between the real profile *q* and the observed (apparent) profile *t*, in order to avoid being classified according to the original profile.

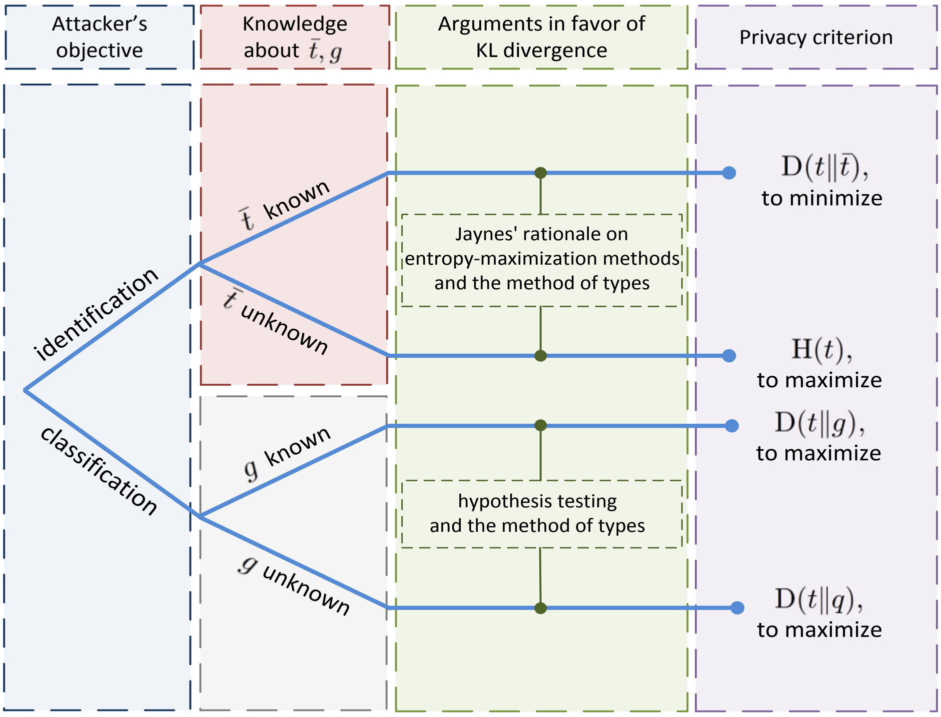

Attacker's objective
Knowledge about $\bar{t}, g$
Arguments in favor of KL divergence
Privacy criterion
identification
classification
$\bar{t}$ known
$\bar{t}$ unknown
g known
g unknown
Jaynes' rationale on entropy-maximization methods and the method of types
hypothesis testing and the method of types
$\mathrm{D}(t\|\bar{t})$, to minimize
$\mathrm{H}(t)$, to maximize
$\mathrm{D}(t\|g)$, to maximize
$\mathrm{D}(t\|q)$, to maximize

Figure 1. Summary of interpretations about Shannon's entropy and KL divergence (relative entropy) as privacy metrics, according to justification presented in [29].

It is necessary to note that in the classification problem, contrary to the identification problem, we are looking for the KL divergence maximization, instead of its minimization. The intuition underlying to the cited analysis is that we wish to increase the distance between the user's apparent profile and the real profile, or the group representative profile on which we wish to avoid the categorization.

3.3 User Profile Model

When doing a security analysis, we must consider the profile of the victim seen by the attacker, in order to envisage the parameters susceptible to be abused.

The scenario of this project involves a user sending queries to a search engine. Queries are sent to request information the user is interested in, so the words that are part of these queries strongly represent their interests, needs, problems, and many other personal details. The technology doing this linking from keywords to interests has being deeply studied during the last years and has been successfully implemented in a variety of recommendation systems.

For an easier interpretation, the user's profile is commonly modeled as a histogram of absolute frequencies, and where, for the sake of standardization, these interests are expressed as categories and subcategories; namely, a list of general topics linked to a weight value. This weight value is calculated as a score that measures the user's degree of interest in each topic, depending on the number of queries, tags or clicks sent or created by the user with relation to such topics.

The privacy criterion used as base for this work, and explained in the next headings, assumes a user model represented by a histogram of absolute frequencies.

4. Privacy Measuring in the Browser

One of the contributions of this work is the implementation of a tool that measures user privacy in the browser, in order to show him his privacy risk level. Honoring its functionalities, we called this tool as PrivMeter. The details of the implementation are described in [37] but we will summarize here some of its components.

The measurement of privacy, according to the metrics described in the last section, is essential to illustrate a comprehensive privacy indicator in terms of risk or gain of privacy. This is, by no means, the case of most of the existing tools created to protect the users' privacy on the Internet, as stated in Section 2. None of these tools is capable of showing the privacy level of a user nor are aware of the effectiveness of their (usually heuristic) mechanisms.

The lack of information about the state of user's privacy is a serious issue since, if the user himself is not aware of the danger derived from the digital trail he leaves, little he could do to protect himself. Clearly, the risk perception the user faces would end up in suspicion and then in a more proactive behavior (i.e. defensive attitude) regarding information management [38].

Here, we propose PrivMeter as a tool capable of displaying intelligible data about the privacy levels of the user, in order for him to understand the risks and probably, from his perspective and interests, to make a decision to protect himself.

4.1 Design Considerations

This section defines the main premises for the development of the browser extension. Fig. 2 illustrates the main components and interactions of the environment where the privacy is measured.

The first premise considers that the user does not trust any external agent or third party. This means that every process to obtain the user's privacy information should be done locally and, preferably, in the browser so that this tool can be easily ported. The user is not, therefore, interested in yielding more information like it happens with REPRIV, for the sake of his security.

Moreover, two potential attackers are considered, as it is described in Fig. 2: the search engine service and the Internet Service Provider, even when other adversaries can be integrated in this measuring process. The profiling activities of these entities represent a very serious risk for the user’s privacy due to the great volume of user information to which these entities have access. Search engines are capable of collecting all the queries submitted by the user. They are also able to trace the URLs clicked from the results page that is delivered to the user during his searching activities. ISPs, instead, have access to practically all the user’s information generated when he is interacting with Internet: queries, tags, URLs, HTML pages, plain text mail messages, visited time, and, in general, all the information delivered to third web services.

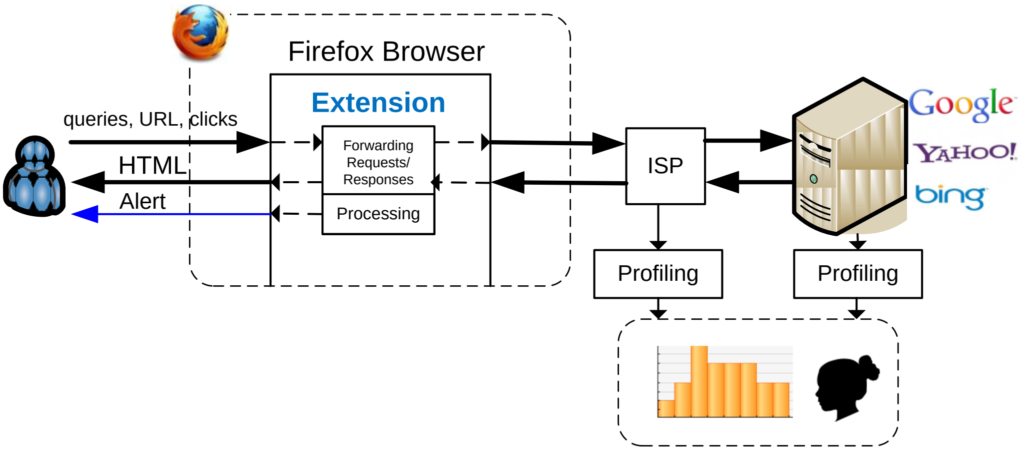


Fig. 2. Scheme of a browser and its extension as intermediaries between the user and Internet services (search engines and ISPs) showing the inherent risk of profiling.

The information obtained from the user’s browsing habits allows the attackers to build a detailed user profile. But, obtaining this profile is also crucial for the user to estimate the risk he is facing when browsing on the Internet. This profile is also important for generating alerts that would help the user to make a decision about his privacy.

It is not a surprise that the web browser is the first candidate to be the framework to build PrivMeter. It behaves as an intermediary (Fig. 2) between the user and Internet, since it is the agent that manages all the requests that the user sends, and all the received responses (commonly displayed as web pages). The information about the user activity obtained from the browser could be extremely detailed and, therefore, pretty useful to build a profile, in the same way as the mentioned attackers would do.

As it will be specified below, the hierarchical scheme of categories over which a user is profiled is also a key parameter to characterize the adversary. In this work, for the sake of convenience, the categories used in the profiling process are the ones that were used by Google to profile its users. Hence, the assumption is that the privacy is measured with respect to an adversary that profiles users according to Google Preferences’ tree of categories.

When classifying the user in the browser, in order to measure the risk of being categorized in a group, as it was mentioned in section 3, reference profiles must be available so that the discrepancy between these profiles and the user profile can be calculated.

Finally, according to the first premise of a user not trusting anything outside his local machine, the information of the profile will be kept locally in the browser storage structures.

4.2 Architecture

The main components structuring the browser extension are detailed along this section. These are grouped and interconnected to represent the architecture for the measurement of user's privacy risk. Fig. 3 illustrates this structure where the executed processes and the results are shown for each functional module.

4.2.1 Web Browser

As illustrated in Fig. 2, the web browser is the application agent through which all the information between the user and web services is exchanged. The browser manages all the HTTP requests and replies generated by the user's interaction with Internet. This interaction basically involves plain text generation from the user and the service provider side: queries and personal data from the user, and text, video and audio from the service providers. All this information is clearly visible for the web browser, as it is for the ISP and partially for the search engines.

Due to the last mentioned properties and its flexible components for development, the web browser is the ideal framework to implement a privacy measuring tool. Since it is located in the user side, the information can be maintained under the user control. Having interfaces to access all the user's information (words, particularly) yielded to the Web, it is possible to reproduce similar attacker models to measure the privacy risk of the user.

The idea is, essentially, to implement these analyses inside the browser, to reveal the user profile for his own use and interpretation.

Mozilla Firefox has been chosen as the browser application for measuring the privacy, because it has several interfaces for extension development that are necessary to access the user information. Additionally, Firefox is widely used and its components are very well documented.

Browser extensions (or add-ons) are software components used to add functions to the browser by retrieving and processing the web information managed or simply changing the browser graphical user interface. In Fig. 2, on the user side, we can observe how a Firefox extension is able to "capture" the information generated by the browsing activities to then process it and get some more specific data (e.g. a user profile).

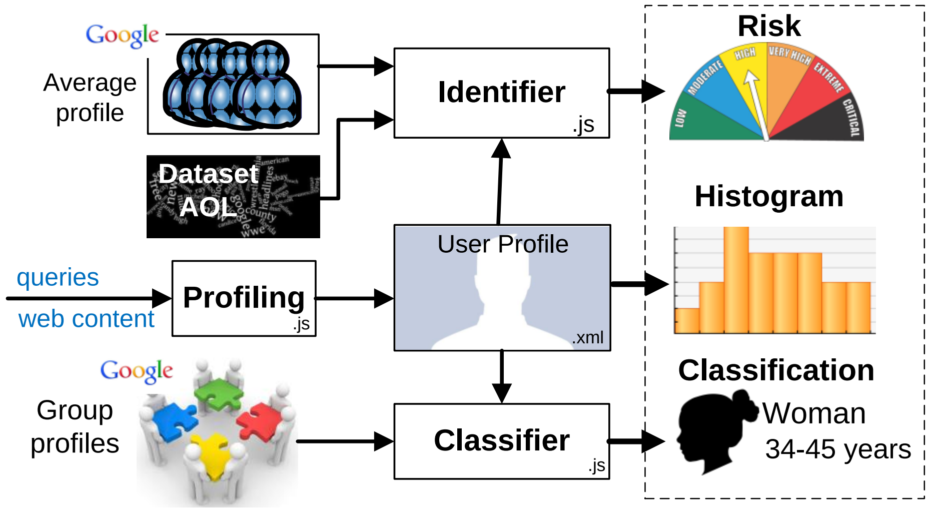


Figure 3. Architecture for privacy level calculation.

4.2.2 Profiler

Establishing the user’s profile, consists in doing a work similar to the potential attackers’ by tabulating the collected information from the user (words, basically) to model his behavior as a user profile.

The profiling process involves obtaining a table of frequencies from a set of pre-established categories. Part of this table is a weight or “score” for each category added to the profile. This punctuation for each category will be increased by one each time a related preference is revealed from the user’s activity. Fig. 4 illustrates the architecture of this profiling process.

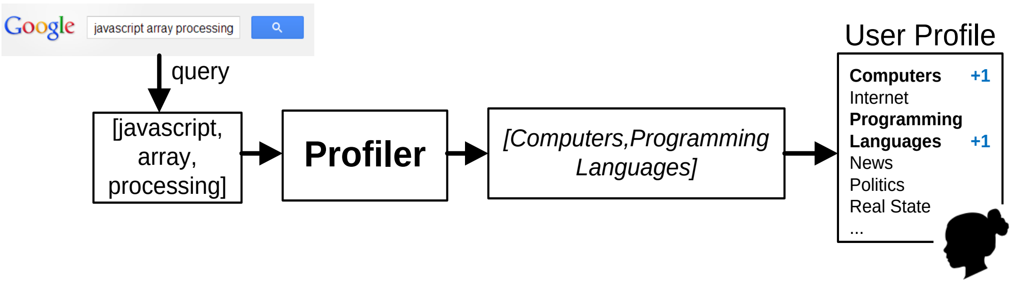


Fig. 4. Profiler architecture.

4.2.3 Histogram

The histogram is essentially the user model. It is a graphical representation of categories, drawn as bars, whose size is proportional to the “popularity” of each category in the profile. The hierarchical scheme of categorization, which in this work is inherited from Google Ad Preferences, used to have 3 levels with a total of 602 categories. The first level of the hierarchy was composed by 27 categories.

The histogram will show the 8 more representative categories from the first level of hierarchy and this graph will provide an initial basic impression about the user's profile, more or less as it is illustrated in Fig. 5.

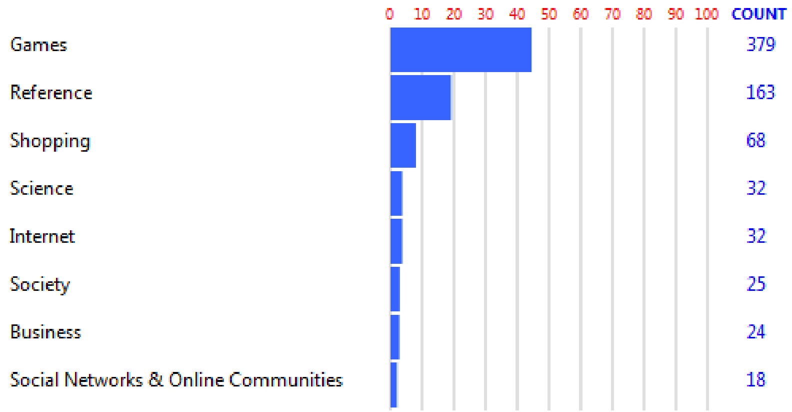


Fig. 5. Histogram representing a basic impression of the user profile.

4.2.4 Identification Module

This module determines the privacy level of a user profile under an identification attack, as explained in Section 3.1. This privacy level is shown here by means of three different ways:

- The entropy of the user's profile, seen as an anonymity or a privacy gain metric.
- The entropy of the user's profile, interpreted with relation to the values of entropy of the user profiles from a real population. These profiles are obtained from a subset of an AOL [39] dataset of queries that was made public some years ago.
- Having an approximated distribution of the average population's profile (obtained from Google Ad Planner tool [40]), the third way to show the privacy level is by calculating the KL divergence of the user's profile with respect to the average population's profile. A value 0 of this divergence would indicate that the user's profile distribution is equal to the population's one, with this state understood as the lowest level of privacy risk. This value, however, cannot be normalized with respect to a maximum because this maximum is not upper bounded. Then, this value could be used to measure the privacy gain, after having used some privacy protection mechanism, to verify the effectiveness of such protection mechanism. In Fig. 6, the architecture of this module is illustrated.

4.2.5 Classification Module

This module uses the KL divergence between the user's profile distribution and the profile distribution of some predefined groups (see Fig. 7). It recreates the attack that would be done by an adversary trying to classify the user within a certain group. In order to do that, the module calculates the KL divergence between the user's profile distribution and the average distribution of each group of population in which Google classifies its users according to their preferences (data obtained from the Google Ad Planner tool).

The lower the value of divergence between the user's profile and the group profile, the lower is the discrepancy between them. Therefore this would be the group to which the user has the highest probability of belonging.

From the user's point of view, this information is pretty illustrative since it gives him a very clear idea about how predictable his profile is in Internet and, especially, how much can be inferred from his digital trail.

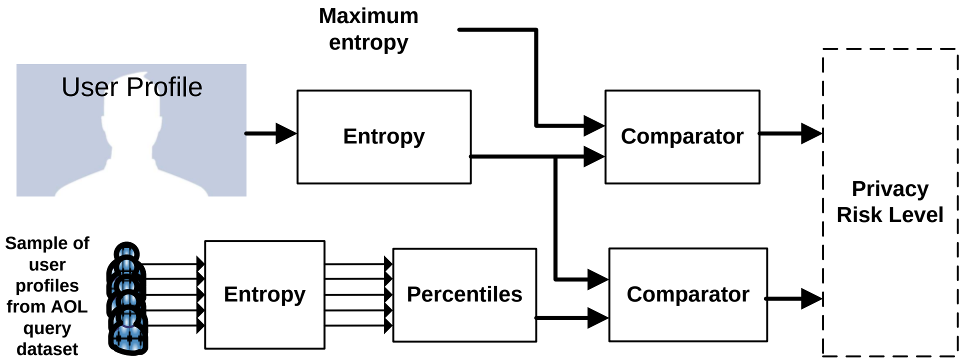


Fig.6. Architecture of identification process.

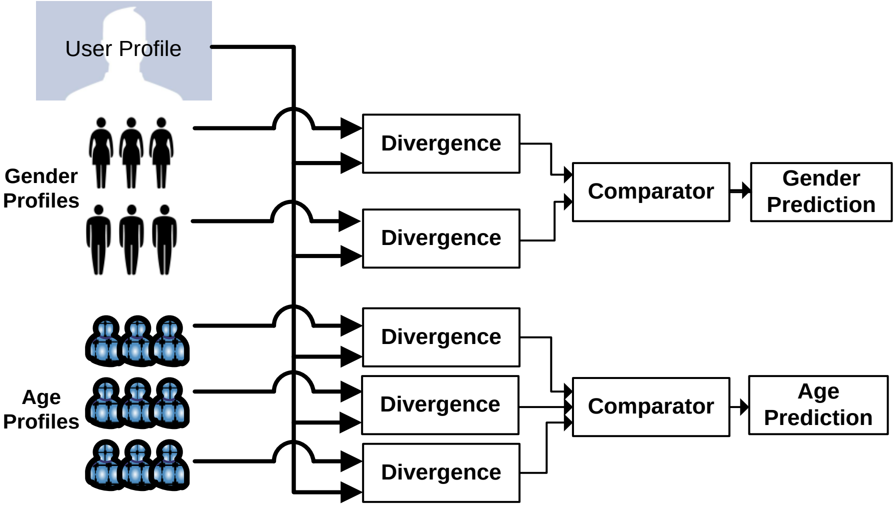


Fig. 7. Architecture of classification module.

This classification method is consistent with the metric and the attacker described in the section 3.2 and also illustrated in the Fig. 1, where the group's representative profile is chosen as the average among the pertaining profiles.

As a marginal note, any method for supervised classification (e.g. Support Vector Machines) could be used by the attacker or the architecture to classify a profile in one of the predefined subsets of the population. The chosen method in this architecture is conceptually simple, and consistent with the metric proposed in [35].

4.3 Implementation Details

The Firefox extension here proposed has been developed by using both Javascript and XUL languages. As explained along the preceding sections, this extension aims first to measure the level of privacy risk a user is exposed to when searching the Web, and secondly, to show this information in a comprehensive, intuitive manner. The level of privacy risk is computed according to the considerations analyzed in Section 3 where two models of attack were described.

The interfaces available for the Firefox extensions development are harnessed to access to the data yielded by the user in terms of words that are then processed in the profiling module. The user's profile is the input of the identification and classification modules that obtain the privacy level and allow the generation of some indicators or even alarms to the user.

Noteworthy that the profiling module used in this implementation was reused from the one implemented in Adnostic [21] extension.

4.3.1 Profiling Module

The profiling module is in charge of building the user's profile after categorizing the information yielded by the user to the Internet.

This component is reused from the profiler module that is part of the Adnostic [21] extension. It captures user information from the browser by detecting some particular events and data generated by the user browsing activity. These parameters are listed below:

- Search queries to the main search engines (Google, Yahoo and Bing).
- Title and meta-tags in the header of HTML code from visited web pages.
- Time during which the user visited a web page.
- Number of clicks done over a web page.

Adnostic's profiler module does a behavioral profiling of the user by means of machine learning techniques that process each query and HTML code to further obtain a category (up to five categories) related to these parameters. Then, the user's profile is updated accordingly.

Adnostic uses these categories and the user's profile to show personalized ads to the user, but in our work the user's profile is the input of identification and classification modules that will calculate the user's privacy to evaluate the potential risk he is facing.

The categories to which the user-generated keywords (queries, essentially) are mapped are the ones that Google Ad Preferences used to employ (now the number of categories is higher) to classify their users and deliver them ads related to the assigned categories. There are 602 categories distributed in 3 hierarchical levels but, essentially, each category could be considered independently from each other.

The user profile is locally stored as an XML file consisting of nodes representing each category and basically two attributes: id, corresponding to the name of the category, and count, whose value registers the weight a category accumulates during the profiling process.

4.3.2 Population Data

As mentioned, to measure the user's privacy when facing classification attack, the extension must classify the user in a predefined population group. According to the user's profile, our extension is capable of classifying the user by age and by gender

Section 3.2.2 describes how KL divergence could be used as a privacy metric by "comparing" the user's profile with the average profile of a predefined group. The risk perception will depend on how worried is the user about the fact of being classified in one group or another.

In this extension, the classification is made by getting the lowest KL divergence between the user's profile and each of the profiles of population groups. The group with which the user profile has the lowest KL divergence (i.e. "discrepancy"), would be the group in which the user gets classified.

In order to calculate such divergence, as illustrated in Fig. 1, we need to have these group profiles based, of course, in the same categorization scheme.

These profiles were obtained from the base of information available in an advertisement tool, which is property of Google, called Google Ad Planner [40]. This is a service that gives advertisers the opportunity of deciding what audience they want to reach, depending on the demographics and interests of users. So this tool contains the information related to this audience, organized in the same Google categorization scheme that is used here to represent the user's profile.

Google Ad Planer shows the projected audience for groups by age, gender and location, by using percentages, but it also has an interest projection of the average population which is expressed in millions of people.

Profiles are stored as XML files and this information is also included in the extension so the classification can be done in real time whenever the user's profile changes.

4.3.3 Graphical User Interface

The graphical interface is coded by using XUL (XML-based User Interface Language) which is an XML based language and includes very simple and portable GUI interface definitions. It is used by default by Mozilla applications for developing user interfaces and easily integrated with controlling functions through Javascript code.

The GUI is critical for the user to interpret his privacy risk levels. Graphics and some context information in this extension tremendously help to evaluate the privacy level.

Our extension's interface is made up of dialogs, bars and some other tools that allow the user to be aware of details about his privacy level during browsing activities.

The GUI is made up of 4 basic elements:

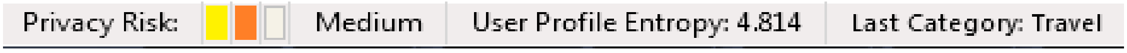
- A quick information bar that is located in the extension bar of the browser
- The main window of privacy metrics
- An extended privacy metrics dialog
- A module for web history import

These components are described with more detail in the next sections.

4.3.4.1 Quick Privacy Information Bar

This bar is located at the bottom of the browser and is immediately visible for the user when starting Mozilla Firefox. As illustrated in Fig. 8, a level indicator shows the user how high is his profile's risk level. From white to red, this icon alerts about the privacy risk level, according to the user's profile entropy, which is also displayed.

As a complement, the extension also shows the last category updated in the user profile after the last sent query.



Privacy Risk: 🟨🟧⬜ Medium | User Profile Entropy: 4.814 | Last Category: Travel

Fig. 8. Quick privacy information bar.

4.3.4.2 Privacy Metrics Window

This is the main dialog where privacy information is shown for the user. It is accessible from the Tools menu and the PrivMeter option (Tools → PrivMeter → Privacy Information). When it is called, the dialog contains the following information:

- The user's profile drawn as a histogram of categories
- Privacy information against an identification attack, which details info about:
  - Entropy of the user's profile
  - Maximum entropy value
  - Divergence with respect to the average population's profile
  - Privacy risk level measured with relation to the entropy of user's profiles in the population sample taken from the AOL query logs.
- Privacy information against a classification attack

As illustrated in the Fig. 9, this window initially includes a histogram representing the user profile, showing the 8 most popular categories with their corresponding count value.

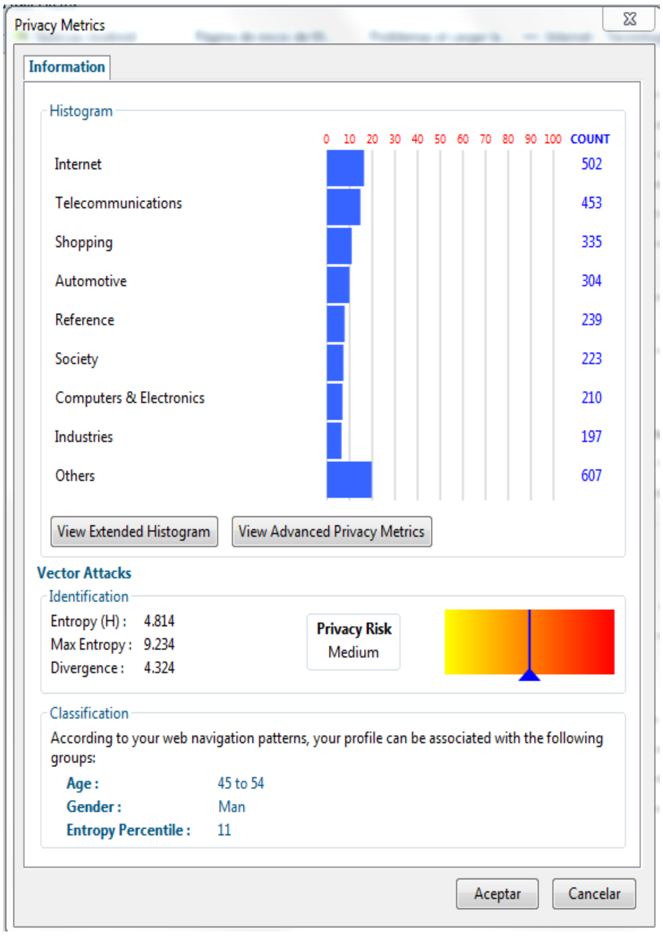
This graphical version of the user profile may help to have a first feeling about his privacy by seeing the most important categories where his queries have been classified.

Moreover, there are two buttons: one to display an extended histogram composed by the 27 categories of the first hierarchical level, and another to open a dialog showing some more advanced privacy metrics.

Additionally, privacy information related to the two attacker models previously described is also shown in this window: the user's profile entropy, the maximum entropy value and the divergence between the user's profile and the average population's profile. Although this last measure is not immediately useful since it is not upper bounded, an increasing value of this divergence shows that the user's profile is more easily identifiable, given that its discrepancy with respect to the average population's profile has increased. A level rule also displays the privacy risk level of the user according to his entropy value, whose magnitude is interpreted in relation with the values of entropy belonging to the profiles of a population sample taken from the AOL dataset.

By locating the user's entropy value in its corresponding percentile (along the user entropies of this dataset) a more realistic measure of privacy can be obtained.

4.3.4.3 Web History Import



Privacy Metrics

Information

Histogram

	0 10 20 30 40 50 60 70 80 90 100	COUNT
Internet		502
Telecommunications		453
Shopping		335
Automotive		304
Reference		239
Society		223
Computers & Electronics		210
Industries		197
Others		607

View Extended Histogram | View Advanced Privacy Metrics

Vector Attacks

Identification

Entropy (H) : 4.814
Max Entropy : 9.234
Divergence : 4.324

Privacy Risk
Medium

Classification

According to your web navigation patterns, your profile can be associated with the following groups:

Age : 45 to 54
Gender : Man
Entropy Percentile : 11

Aceptar | Cancelar

Fig. 9. Main privacy metrics window.

In order to give the user a start point when installing the extension, there is a web history import tool accessible as a dialog in the Tools menu. When clicking on “Privacy Options”, a dialog shows the button “Load Profile from Firefox History”. Pressing this button, our extension will import the user’s navigation history and will use it to generate an initial profile based on the entries of that history. Specifically, the information used as input of the profiler is the titles of visited web pages and also the user’s queries that were sent by means of a web form. This information is locally stored by the browser in a SQLite format and is accessible from the Firefox browser interfaces to be parsed and profiled in order to build an initial user’s profile, in the same way that queries to search engines are being categorized.

5. Evaluation of a User Profile Obfuscation Mechanism

In this section we evaluate the efficiency of the popular privacy protection tool TrackMeNot, in terms of the privacy metrics defined in Sec. 3.

It is a fact that, in the privacy research literature, several proposals have been made to protect privacy. At the user level, however, not many available tools implement those privacy protection approaches. Instead, we can find some mechanisms, based on heuristics, trying to isolate the user from the privacy risks by means of blocking certain interactions with the Web. As described in section 2.1, this blockage of functionality (cookies) in the browser complicates the access to several applications on the Web, hence reducing the usability of the browser.

A first step towards the developing of more efficient privacy protection applications is, thus, evaluating the existing ones. Determining their real benefits would allow us to take advantage of well-designed mechanisms and to detect wrong strategies. As explained in Section 2.1 there are several mechanisms and tools available that claim to protect privacy but almost no analysis or study about the effectiveness of such privacy-enhancing-technologies. Moreover, due to the fact that the user privacy depends on multiple factors related to both the adversary and the user (as illustrated in Fig. 1), measuring privacy is a multidimensional process which will vary if those parameters change. It would be really useful, therefore, to study the impact that every privacy protection mechanism have on the different conditions that a user could face.

Evaluating privacy protection applications means comparing privacy levels offered before and after the mechanism was implemented. A sort of privacy gain can be calculated, interpreted as the potential benefits of applying a given tool. The problem here is that measuring privacy is not a simple task; multiple variables could be taken into account and, in the same manner, multiple approaches of protection are proposed both theoretically and in the practice. Evidently, though, very few of the existing tools could be evaluated based on the same parameters, especially because most of those tools try to solve privacy issues using heuristics in different manners.

Along this section we explain the methodology used to evaluate one of these privacy protection tools, by using the privacy metrics proposed in [35] and some modules implemented in PrivMeter [37].

5.1 Queries as User Identifying Information

Information search has become a very common activity for users when browsing on Internet. By themselves, search queries can profoundly reflect our interests, our worries or problems. In combination with social network interactions and tagging activities, search querying could precisely reveal our identity. It was publicly demonstrated when released "anonymized" search logs in 2006 were used by the New York Times to expose the identity of a woman [2]. Not in vain Google, the biggest search engine, obtains most of its profits from advertising. Then it is evident that they are doing a pretty good job in profiling users to accurately personalize ads.

As with the rest of providers of personalized information services, Google argues that maintaining such amount of user data helps them to improve services and prevent fraud [41]. But, as we argued in the Introduction, the risk lays especially in the cooperation that companies like this one are offering to governments and other external entities. As far as we know from the newspapers and other sources on Internet, this cooperation apparently implies the sharing of personal data to powerful government security agencies.

As already stated in Section 2.1.3, data perturbation is a privacy preserving technique that can be applied in the user side, independently from third parties. Perturbation of queries or query obfuscation is a mechanism that has been implemented in the TrackMeNot Firefox extension in order to protect individuals against profiling activities when they browse on the Web.

5.2 TrackMeNot

TrackMeNot (TMN) is a Firefox browser extension whose aim is to obfuscate the user's profile, through introducing bogus material in the search query stream. This mechanism is based on the artificial generation of query-like phrases that are then sent to search engines via HTTP requests. In order to prevent the search engines to detect an automatically generated query from the real ones, some mechanisms are also implemented in TMN to simulate a human search behavior.

The architecture of this application is illustrated in Fig. 10 and some of its involved modules are briefly described below.

- **Dynamic Query-lists**. TMN needs to dynamically generate search queries to make it more difficult for search engines to accurately identify users by profiling these keywords. TMN essentially uses publicly available sources of information (RSS feeds) to build a seed list of query terms from which the search queries will be obtained and then sent to the search engine. This list evolves in the time according to the RSS feeds configured in the TMN options, especially because a new list is retrieved each time the Firefox browser is restarted. Also, given that the default RSS feeds belong to on-line newspapers' sites, the available words will change in a daily basis.
- **Real-time search awareness**. Some search engines like Google are capable of detecting automated requests, fundamentally by identifying too many requests in a short period of time. In order to prevent this, TMN provides a module that monitors the user search behavior so that it sends a number of bogus queries whenever a real user query is detected.

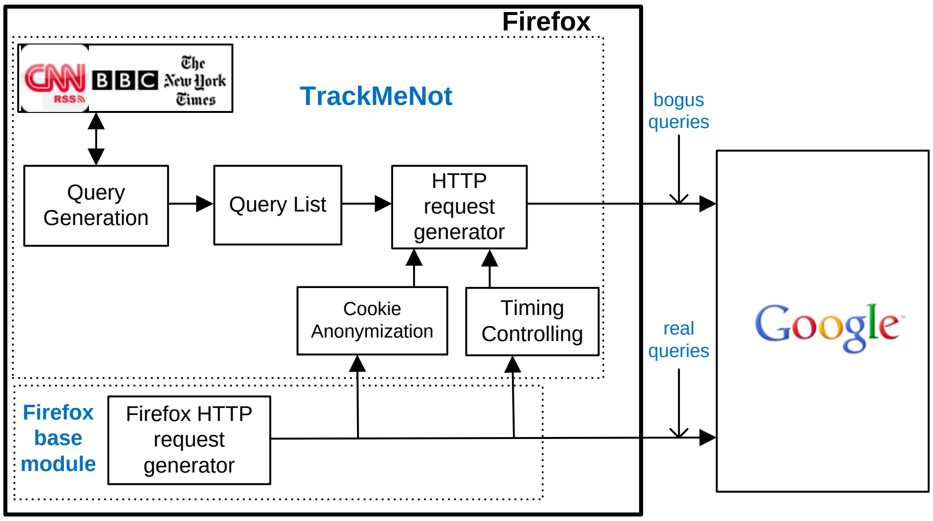


Fig. 10: TrackMeNot Architecture.

- **Live Header Maps**. To more closely mimic the user's search behavior, TMN adapts its HTTP requests headers according to the headers used by the user to send the most recently query. Depending on the web browser used, information in headers can also be used to identify a user in a region [3].

- **Burst-Mode Queries.** This mode of working is implemented in TMN to send a batch of fake queries in the moment that a user sends his. This mechanisms contributes to mimic the user behavior that usually submits several queries in a short period of time.

- **Cookie Anonymization.** This TMN module blocks cookies for user's search queries but, instead attaches them to the (artificially) generated search queries, so that the search engine registers only the fake queries in the user profile.

TrackMeNot is a widely known tool that uses an innovative mechanism to simulate the user search behavior in order to obfuscate the user profile seen by search engines. Not many other tools are available for the user that involve obfuscation of queries or tags, even when there exists theoretical contributions that proposes such strategies.

5.3 Evaluation Methodology

We evaluate TMN by comparing the level of privacy before and after its application. If there is a gain, we shall interpret if this gain is enough to protect the user against the external adversaries (i.e. if it is effective).

The main steps, performed to estimate the effectiveness of TMN with the objective of increasing privacy protection, are the followings:

- Obtain a significant sample of real user query logs, whose privacy will be measured.
- Generate sets of fake queries by means of the mechanism offered by TMN.

- Obfuscate the users' real queries by mixing real queries with fake ones.
- Obtain users' real and obfuscated profiles, from the corresponding query logs.
- Measure privacy of real and apparent profiles, using the metrics already described.
- Determine the privacy gain of user profiles after obfuscation.

As indicated in Fig. 11, in order to start the evaluation process we first obtained a sample of query logs belonging to a significant number of users (the users' queries). We used, with this goal, the AOL dataset [2] released in 2006, which contains around 20 million queries of 650 thousand users. The sample we took belongs to 6674 users, each of them having from 501 to 1000 queries. From these 6674 users we chose the 50 users with the greatest amount of queries.

Then, fake queries were obtained by means of the TMN query generation process. The code from TMN extension were hooked to create a function capable of generating 700 search queries from the RSS feeds configured by default (feeds of CNN, BBC, The Register and The New York Times).

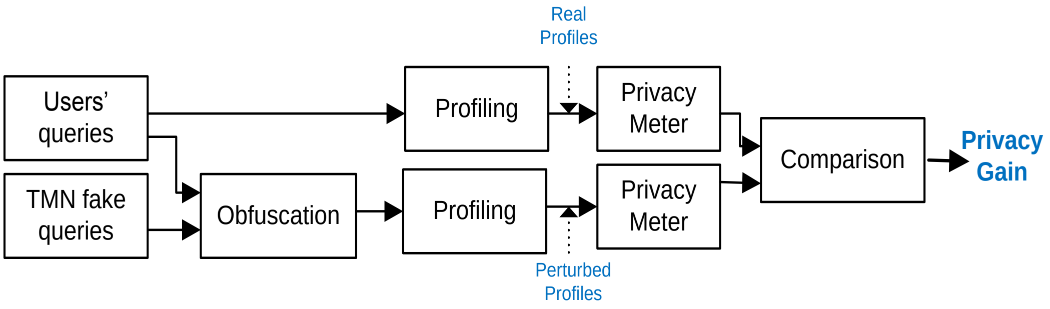


Fig. 11: TMN Evaluation Process.

As showed in Fig. 11, the obfuscation of real user queries was done by mixing these queries with the already generated fake queries (we got the obfuscated queries for each user).

Both real and obfuscated queries were separately profiled to get the corresponding real and obfuscated profiles. At this point we had the real profiles whose owners would be interested in hiding and the obfuscated (apparent) profiles which would be the resulting profiles of users after implementing TMN.

Evidently, the last step was to measure the privacy (according to metrics explained in Section 3) of each type of profiles to finally compare the values obtained for each user. This comparison determined a value of privacy gain that allowed us to know if the obfuscation was beneficial for the users' privacy.

Some of these steps will be explained with more detail along the next headings.

5.4 Profiling User Information

We used two existing mechanisms in order to get a user profile (as described in Section

3.3) susceptible to be measured by the metrics described in Section 3.

- **Adnostic's profiling module.** This module was used in [37], as part of PrivMeter. It builds a user profile from the continuous categorization of the user's search queries. We have modified its functions so that it can receive all the users' queries from our sample in order to obtain the user profiles.

  The categories in which the text can be classified do not include sensitive categories related to health, racism or porn. The scheme of categorization is based on Google hierarchy that contains 602 categories.

- **TextWise** [42]. It is a semantic technology implemented on the Web to be freely accessed through a web API (although with some restrictions in the usage). Among other services, TextWise offers a categorization service that accepts requests containing keywords and returns up to 5 categories in which the sent text can be classified.

  Unlike Adnostic's profiling module, TextWise uses a hierarchical schema based on ODP (Open Directory Project) that is built of 770 categories.

  We built a program to use this API in order to categorize each query from the AOL sample. This information allowed us to create user profiles according to this strategy of categorization.

  The code of the program, written in Python to consume the TextWise API and build the profile, receives a set of files containing search queries (each one represents a user and contains his queries), and returns the corresponding profiles.

As stated in Section 3, the measurement of privacy may depend on the capabilities of the adversary. If we measure the privacy of users whose profiles are obtained by using two strategies of profiling, we are modeling two different adversary capabilities. Hence, it is interesting to see if the impact of the privacy enhancing mechanism is the same no matter what profiling strategy is considered.

5.5 Privacy Measuring

During the TMN evaluation, as showed in Fig. 11, once both real and apparent profiles are available, the next step is measuring their privacy. Privacy metrics used in this work have been already introduced in Section 3. Essentially, the entropy of a user profile is the first way to measure the user's privacy. Also, the divergence of the user's profile with respect to the profile of a predefined population group can be used to measure his privacy.

From the users' profiles gotten after the profiling process, both of the last metrics were obtained (illustrated in Fig. 12):

- **Entropy** of the user profiles.
- **Divergence** of the user profiles relative to the average population profile obtained from Google Ad Planer.

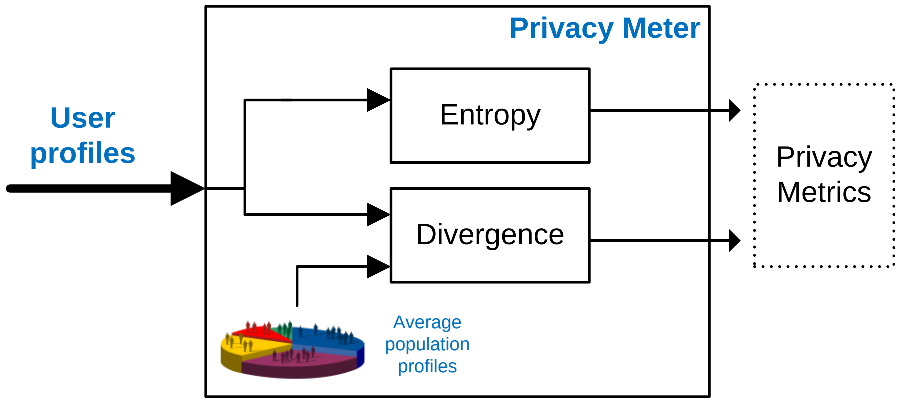


Fig. 12: Structure of privacy meter module, which receives the user's profiles and returns privacy measurements based on entropy and KL divergence.

5.6 Privacy Gain

According to the analysis in Section 3.2.1, an increasing value of entropy means a higher level of user's privacy. When getting the divergence of the user's profile with respect to the average population profile, a privacy gain is obtained when this divergence is decremented.

The privacy gain level is, then, calculated by comparing the privacy before and after the TMN perturbation mechanism was implemented, using the following expressions.

Being *q* the user profile, *t* the obfuscated user's profile, *p* the average population profile and *M* the privacy gain level, we have that for the privacy level, measured as the entropy of the user's profile *(H(q))*, the privacy gain relative to the initial privacy level is M_H, as defined in Eq. (4).

$$M_H = \frac{H(t) - H(q)}{H(q)} \tag{4}$$

For the privacy level measured as the divergence of the user's profile with respect to the average population's profile *(D(q|p))*, the privacy gain, M_D, is defined in Eq. (5).

$$M_D = \frac{D(q \mid p) - D(t \mid p)}{D(q \mid p)} \tag{5}$$

These values would give us an initial view of how privacy is being enhanced when using obfuscation of queries. Calculating the percentiles in the population profile to which these values belong will give us a more realist measure of the level of privacy.

5.7 Evaluation Environment

The objective of this evaluation was to measure the privacy gained after using the TMN obfuscation mechanism, but also, trying to determine which of its configuration parameters have an important influence on the effect of enhancing privacy.

The main inputs of this environment were the queries obtained as a sample from the AOL dataset. The fake queries used to obfuscate the users' profiles were artificially generated from the TMN mechanism. Basically, we programmed TMN to generate different amounts of fake queries, during 5 days, in order to then verify how the proportion of fake queries with respect to the real ones influenced the user´s privacy gain. Hence, being ρ the relation of the number of fake queries with respect to the number of real ones, we determined how privacy gets increased with ρ.

$$\rho = \frac{Number\ of\ \ fake\ queries}{Number\ of\ \ real\ queries}$$

Some evaluations were also done by modifying the RSS feeds that TMN uses as source to generate the fake queries.

The evaluations were done considering both the entropy and divergence as metrics of the users' profiles privacy, as explained in Section 5.5.

5.8 Results of Privacy Measuring

5.8.1 Analysis of Fake Queries

The 700 fake queries obtained (during 5 days) from TMN are analyzed first in this section. As if those generated queries were user queries, we profiled them to analyze how these profiles could impact on the user profiles. This was done by means of the Adnostic's profiling module.

Each profile, obtained from the fake queries, was built of about 140 categories (from the 602 available in the categorizer).

Fig. 12 shows the histograms of the fake queries obtained in the first day out of 5 (only the 30 most popular categories are showed). The results for the next 4 days were very similar to those obtained in the first one.

We observed that, during the experiment, the categories to which the TMN fake queries belong are basically the same, and their impact (popularity) is similar during the 5 days. In addition, the category "References" repeatedly appears, having an important influence in these histograms. From our experience with the Adnostic's profiling module we have seen that "References" is a sort of generic category used (but not strictly) to classify all the queries related to sensitive information (e.g. health condition). This minimizes the influence of such categories in the profile; so it does not provide much information.

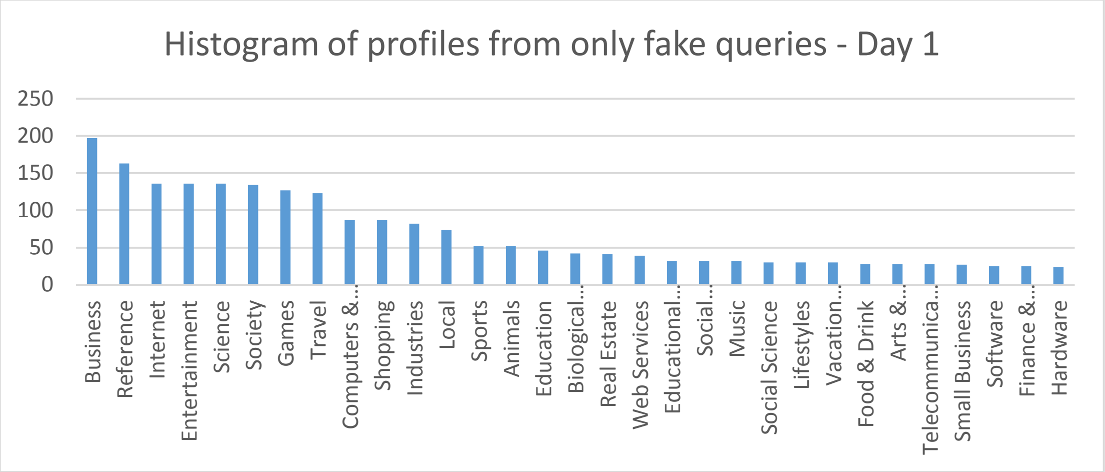


Fig. 13: Histogram of categories obtained after the profiling of TMN fake queries, generated using the default 5 RSS feeds on day 1 (only the 30 most popular categories are showed).

The profiles were very similar, which means that, at least during these five days, the topics of fake queries do not changed much. This would make easier the work for an adversary when trying to separate the influence of such queries in the obfuscated profile, in order to obtain the real user’s profile.

5.8.2 Privacy gain using default TMN´s RSS feeds and Adnostic’s profiling module

There is actually a gain in users’ privacy when obfuscation of their queries was implemented using TMN, both in terms of users’ profiles entropy and in terms of users’ profiles divergence with respect to the average population’s profile.

In terms of entropy, the mean gain was 14.58% and 20.17% in terms of divergence, taken from the 5 consecutive days. This results from using 100% of fake queries (the same number of real queries as the number of fake ones).

The entropy values of users’ profiles after obfuscation were also compared with the original values by using non-central position measures (percentiles in this case). It was found that after obfuscation, the users’ privacy value (users’ profiles entropies) got increased in about 50 percentiles, according to the distribution of entropies obtained from the whole population where the sample queries were taken. This is a considerable gain since a user whose privacy was in the 4th percentile, after obfuscation it went to the 50th.

5.8.3 Privacy gain using default TMN’s RSS feeds and Textwise-based profiling module

This test was implemented by using TextWise as the categorization engine. As it involves a wider scheme of categories (about 770), it was supposed to obtain more accurate users’ profiles from their queries.

In terms of entropy the average gain obtained from the obfuscation process was of 24.19%, also during a 5-day period. The benefits, are about 5% higher than the ones obtained using a somewhat more limited profiling mechanism

5.8.4 Privacy gain with respect to ρ

Privacy gain was also measured as a function of ρ (described in Section 5.6). We observed how the privacy gain got increased with the percentage of fake queries, both in terms of entropy and divergence. These measures of relative privacy gain are illustrated in Fig. 14 and Fig. 15.

5.8.5 Privacy gain against classification attacks

As stated in Section 3, the KL divergence could be used as a metric against classification attacks. If we have the profile of a population group, we can classify a user profile as part of such group if the divergence between the two profiles is small enough.

We measured the KL divergence of the users' profiles with respect to the profiles of some population groups (age and gender groups) to classify these users in one of these groups. Almost no user was changed from its original category after the obfuscation process, which means that this mechanism was not successful against classification attacks.

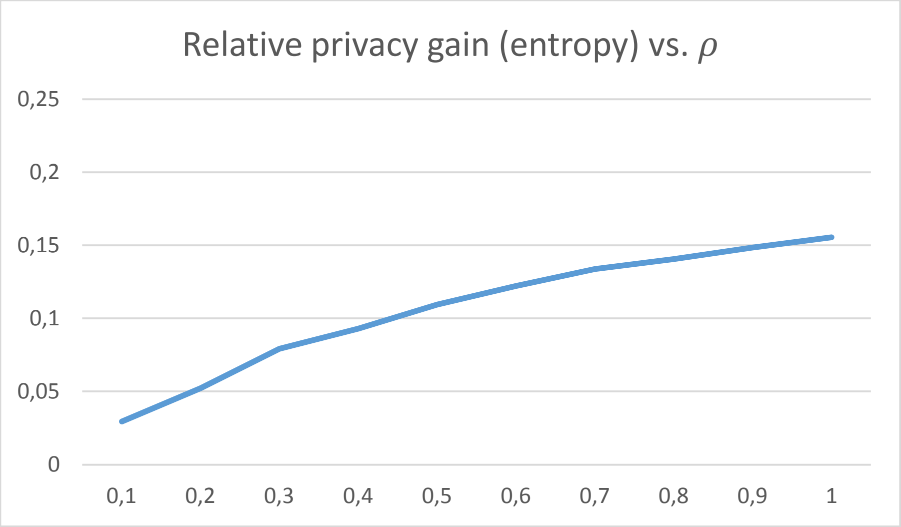


Fig. 14: Increasing of relative privacy gain (in terms of entropy) according to ρ. The greater the amount of fake queries, the higher the privacy gain.

The impact of TMN's fake queries do change the users' profiles, increasing their discrepancy with the category where the users were originally classified. But, this impact is not enough to classify the users in a different category. This suggests that false queries should not be randomly generated because if so, apparently, the influence that they have may be so scattered along multiple categories that the final effect is almost null. Instead, a more directed strategy, say, with the specific objective of provoke the classification of the user in a predefined category, would give better results.

So, the challenge is to implement an intelligent generation of fake queries capable of efficiently obfuscating the users' profiles against classification attacks. This means that the obfuscation process must be adaptable to the particular privacy needs of the users. This process of obtaining keywords (which then would be combined to get fake queries) related to certain topics, without having to depend on third parties, is a very interesting research field.

5.8.6 Privacy gain using a more specific RSS feed

The privacy gain was also measured when using more specific RSS feeds to generate the fake queries. Default RSS feeds belong to well-known newspapers and point to sections where a summary of the more important news is published. Instead, we configured RSS feeds that point to sections of these newspapers related to Sports. The category Sports tends to be a category of greater interest for male public. The intention was to obfuscate the users' profiles with bogus 'male' queries so that the profiles that were initially classified as female, could be categorized as male profiles after obfuscation.

This strategy effectively modified the female profiles, reducing its divergence with respect to the reference male profile. Once again, changing the category where the users were originally classified was not enough.

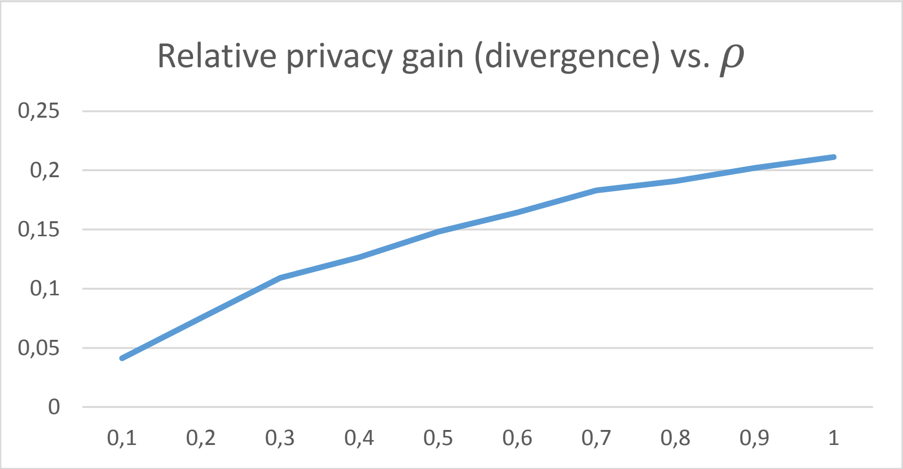


Fig. 15: Increasing of relative privacy gain (in terms of divergence) according to ρ. The greater the amount of fake queries, the higher the privacy gain.

5.9 Discussion about parameters involved in the evaluation

As far as we have shown here, measuring privacy is not only important to illustrate the privacy risk for a user when he browses on Internet. The privacy level can also be calculated to compare profile states of a user. We talk about the original and the obfuscated profile of a user, when the latter is obtained from a tool such TMN in its effort to protect the user privacy. This evaluation, however, is limited to the specific conditions we have taken into account, which are mostly subject to the adversary being faced.

As seen in the last section, several elements are involved in the evaluation of this particular privacy-enhancing technology. From the side of entropy and divergence of the users' profiles with respect to the average population's profile, the privacy is efficiently improved but at the cost of additional traffic. Apparently, the more privacy is needed, the more bogus queries the system must generate. This may also have an important impact on personalized services offered by search engines. Moreover, sending a large number of fake queries may not be so effective, since search engines are able to detect and block automatically generated requests, so additional mechanisms have to be implemented in order to mimic a real user behavior. But, users have been browsing since a long time ago, and adversaries have already collected a huge amount of real information about them. Thus, effectively obfuscating so dense profiles, using not intensive mechanisms of query generation, seems a titanic task. A more focused strategy to generate bogus queries should be implemented, so that the impact of using them can multiply the effect of sending randomly generated queries.

The way the users are modeled is also an important factor when measuring privacy (see Section 3), but it depends on the adversary capabilities of user profiling. So, evaluating a privacy-enhancing mechanism by using different profiling methods will give us a better idea of the performance of the mechanism against diverse adversaries. The task of profiling, however, is not trivial, and neither it is to precisely know what technics search engines are using. We reused here two methods of categorization whose hierarchy is similar to the one used by Google. This scheme, however, is now different, since Google would be using more than 1000 categories to build the user profiles. The capabilities of the adversaries to profile users are significantly higher than ours, because their activities generate a lot of money. Therefore, figuring out how users are being modeled by the myriad of adversaries out there on Internet, is also very complicated.

Query processing is part of the profiling capabilities we chosen to simulate, both to measure the user's privacy, and so to evaluate a protection mechanism. The idea is, again, to mimic the adversary intentions of discovering the user's interests, even when the queries are written with typos, to more accurately profile the user. Considering that about 20% of the queries of the obfuscated set were not successfully classified, improving their processing would contribute to enhance the quality of the profiles obtained from them. Once again, since our language processing capabilities are limited, the user profiles, which are the basis of privacy measuring, may not be as accurate as the obtained by our potential adversaries.

The availability of real user information, in the form of queries or tags, may greatly help to interpret the results of the evaluation of this privacy-enhancing technology. If the privacy is measured as a value relative to the values in a population, a more realistic evaluation can be done of the risk levels. Unfortunately, this information is only accessible by the adversaries, those who control the user data. Collecting this information from data sets existing on Internet, is also a pending and non-trivial task.

Finally, this evaluation shows that query obfuscation is not efficient enough against more sophisticated attacks (e.g. classification attacks) if the fake queries are randomly generated. The solution would be the implementation of a strategy of queries generation based on the user's profile and his particular needs. A bunch of bogus queries is not able to effectively obfuscate a user profile when the adversary implements a more complex attack.

5.10 TrackMeNot Integration with PrivMeter

As showed by the evaluation of TMN, its efficiency on improving privacy level of the user, basically depends on the number of fake queries that are generated with respect to the real ones. This means that a great part of TMN benefits will depend on the periodicity of fake query generation and on the number of queries sent by the user.

The mentioned parameters will change for each user (fake query generation can be configured by the user). Thus, in order for the user to be aware of the relative gain of privacy obtained thanks to TMN, it is helpful to integrate TMN and PrivMeter so that privacy enhancements thanks to obfuscation can be measured and showed to the user.

We rewrote some of the functions of PrivMeter used to create and update the user's profile. These functions were used to build the obfuscated user's profile, also called apparent profile.

The apparent user profile has the same structure as the real profile (hierarchical scheme of categories and corresponding scores). The difference is that the queries being categorized to build this profile are the user's queries and the queries generated by TMN. This means that our code detects both real and fake search queries, categorizes them and updates the user apparent profile.

Once both real and apparent profiles are available, the relative privacy gain can be obtained similarly to what we did during the TMN evaluation. This gain value can be calculated in terms of the entropy increasing or in terms of divergence (with respect to average population profile) reduction.

In order for the user to visualize the impact that TMN is having, we incorporated two information items to the privacy bar in PrivMeter: the entropy of the apparent user profile and the privacy gain measured in terms of entropy.

6. Conclusion

Considering the big risk to privacy that users are facing when browsing the Internet, the creation of an application to measure the user's privacy level in real time definitely worth it.

The usage of justified metrics to obtain this privacy level could also help to solidly evaluate how effective are some other privacy protection mechanisms that exist already in the landscape. Data about privacy is essential for the user, in order for him to be aware of how his browsing behavior is revealing very sensitive information about his profile.

Our browser extension aims at quantifying the level of privacy offered by TMN, a popular privacy-enhancing mechanism that capitalizes on the principle of query forgery. The developed extension is equipped with a couple of information-theoretic quantities that are interpreted and justified through Jaynes' rationale on entropy-maximization methods and hypothesis testing. Such metrics are used to assess then the efficiency of TMN under the assumption of two adversary's goals, namely individuation and classification.

Our extensive experimental evaluation shows how TMN may provide levels of privacy gain around 20% in average, when the adversary’s goal is to individuate users. . Against classification attacks, however, the results of the evaluation were not so successful, since the impact of fake queries were not enough to change the original classification of the users.

As expected, the level of privacy was increased as the number of bogus queries was greater. This clearly defined the inherent tradeoff between privacy and total traffic produced by the additional queries.

Some improvements could be done on the attack model we reproduce on the browser, when measuring privacy. More detailed user information can be extracted so that the accuracy of the user profiling process can be increased. This information could be obtained from online sources such as social networks where users spend much of their browsing time. Using other profiling engines would also help emulate different attackers of user privacy.

From the perspective of the evaluation of TrackMeNot, a smarter mechanism of search (obfuscated) query generation must be developed; a mechanism capable of changing its behavior according to the particularities of the user profile. Probably, it should accept direct configuration from the user, who could tune it up to adapt the mechanism to his particular needs.

Another pending work on the evaluation of an obfuscation mechanism such TrackMeNot is the analysis of the tradeoff between the privacy obtained and the additional data generated by obfuscated traffic.

Acknowledgement

This work was partially supported by the Ecuadorian government through its National Secretary of Science and Technology (SENESCYT) and the scholarship it granted to J. Estrada-Jiménez and A. Rodríguez for their Master studies.

This work was also supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES” and TEC2010-20572-C02-02 “CONSEQUENCE”.

References

[1] Narayanan, A., Shmatikov, V., "Robust De-anonymization of Large Sparse Datasets". Security and Privacy. 2008. Pp 111-125. IEEE Symposium. 2008. http://dx.doi.org/10.1109%2FSP.2008.33

[2] Barbaro, M., Zeller, T., "A Face Is Exposed for AOL Searcher No. 4417749". The New York Times. Technology. Available at: http://www.nytimes.com/2006/08/09/technology/09aol.html?pagewanted=all. August, 2006.

[3] Eckersley, P., “How Unique Is Your Web Browser?”. Available at: https://panopticlick.eff.org/.

[4] Electronic Frontier Foundation, “Panopticlick”. Available at: https://panopticlick.eff.org/.

[5] TechCrunch, “AOL Proudly Releases Massive Amounts of Private Data”. Available at: http://techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data.

[6] Wang, Y., Kobsa, A., “Privacy-enhancing technologies. Social and Organizational Liabilities”. Information Security. Pp 203-227. 2006.

[7] Ostrovsky, R., Skeith III, W., “A survey of single-database private information retrieval: Techniques and applications”. Public Key Cryptography–PKC 2007. Pp 393-411. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-71677-8_26

[8] Chaum, D. L., “Untraceable electronic mail, return addresses, and digital pseudonyms”. Communications of the ACM, 24(2). Pp 84-90. 1981. http://dx.doi.org/10.1145/358549.358563

[9] Reiter, M., Rubin, A., “Crowds: Anonymity for web transactions”. ACM Transactions on Information and System Security (TISSEC), 1(1). Pp 66-92. 1998. http://dx.doi.org/10.1145/290163.290168

[10] Chow, C., Mokbel, M., Liu, X., “A peer-to-peer spatial cloaking algorithm for anonymous location-based service”. Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems. Pp 171-178. 2006. http://dx.doi.org/10.1145/1183471.1183500

[11] Barba, C. T., Aguiar, L. U., Igartua, M. A., Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., Pallarès, E. “A collaborative protocol for anonymous reporting in vehicular ad hoc networks. Computer Standards Interfaces. Pp 188-197. http://dx.doi.org/10.1016/j.csi.2013.06.001

[12] Rebollo‐Monedero, D., Forné, J., Pallarès, E., Parra‐Arnau, J., Tripp, C., Urquiza, L., Aguilar, M. (2013). On collaborative anonymous communications in lossy networks. Security and Communication Networks. 2013. http://dx.doi.org/10.1002/sec.793

[13] Rebollo-Monedero, D., Forné, J., Domingo-Ferrer, J., “Query Profile Obfuscation by Means of Optimal Query Exchange between Users”. IEEE Trans. Depend., Secure Comput. 2012. http://dx.doi.org/10.1109/TDSC.2012.16

[14] Rebollo-Monedero, D., Forné, J., Solanas, A., Martínez-Ballesté, A. “Private location-based information retrieval through user collaboration”. Computer Communications, 33(6). Pp 762-774. 2010. http://dx.doi.org/10.1016j.comcom.2009.11.024

[15] Howe, D., Nissenbaum, H., “TrackMeNot: Resisting surveillance in web search”. Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society. Pp 417-436. 2009. http://dx.doi.org/10.1007/978-3-642-14527-8_

[16] Domingo-Ferrer, J., Solanas, A., Castellà-Roca, J., “k-private information retrieval from privacy-uncooperative queryable databases”. Online Information Review, 33(4). Pp 720-744, 2009. http://dx.doi.org/10.1108/14684520910985693

[17] Polat, H., Du, W., “Privacy-preserving collaborative filtering using randomized perturbation techniques”. Data Mining, 2003. ICDM 2003. Third IEEE International Conference. Pp. 625-628. IEEE. 2003. http://dx.doi.org/10.1109/ICDM.2003.1250993

[18] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., “A privacy-protecting architecture for collaborative filtering via forgery and suppression of ratings”. Data Privacy Management and Autonomous Spontaneus Security. Pp 42-57. Springer Berlin Heidelberg. 2012.

http://dx.doi.org/10.1007/978-3-642-28879-1_4

[19] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., "Optimal Forgery and Suppression of Ratings for Privacy Enhancement in Recommendation Systems. 2013. http://dx.doi.org/10.3390/e16031586

[20] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., "Privacy Protection of User Profiles in Personalized Information Systems". Proc. 2012 Forum PhD Research Information and Communication Technologies (ICT). 2012.

[21] Toubiana, V., Boneh, D., Nissenbaum, H., Barocas, S., "Adnostic: Privacy Preserving Targeted Advertising". Proc. of the 17th Annual Network and Distributed System Security Symposium (NDSS). 2009. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.154.1112

[22] Vi-a, S., News, G., Vi-b, S., "REPRIV: Re-Envisioning In-Browser Privacy". Available at: http://research.microsoft.com/apps/pubs/default.aspx?id=137038.

[23] Becker, J., Chen, H., "Measuring Privacy Risk in Online Social Networks". Proceedings of W2SP 2009: Web 2.0 Security and Privacy. 2009.

[24] Fire, M., Kagan, D., Elishar, A., Elovici, Y., "Social Privacy Protector - Protecting User' Privacy in Social Networks". SOTICS 2012, the second international conference on social eco-informatics. Pp 46-50. 2012.

[25] Peddinti, S. Teja, Saxena, N., "On the Privacy of Web Search Based on Query Obfuscation: A Case Study of TrackMeNot". 10th International Symposium, PETS. Pp 19-37. 2010. http://dx.doi.org/10.1007/978-3-642-14527-8_2

[26] Google Sharing. Available at: https://addons.mozilla.org/en-us/firefox/addon/googlesharing/

[27] Ghostery. Available at: http://www.ghostery.com/

[28] Maone, G., NoScript. Available at: http://noscript.net, 2009.

[29] Palant, W., Adblock Plus: Save your time and traffic. Available at: http://adblockplus.org/.

[30] DoNotTrackMe. Available at: https://addons.mozilla.org/en-US/firefox/addon/donottrackplus/?

[31] Polat, H., Du, W., "Privacy-preserving collaborative filtering using randomized perturbation techniques", Data Mining, 2003. ICDM 2003. Third IEEE International Conference. Pp. 625-628. IEEE. 2003.

[32] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., "A Privacy-Preserving Architecture for the Semantic Web based on Tag Suppression". Proc. Int. Conf. Trust, Priv., Secur., Digit. Bus., Bilbao, España. Pp 58-68. 2010. http://dx.doi.org/10.1007/978-3-642-15152-1_6

[33] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., Muñoz, J. L., Esparza, O., "Optimal tag suppression for privacy protection in the semantic Web", en Data, Knowl. Eng.. Vol. 81-82. Pp 46-66. 2012. http://dx.doi.org/10.1016/j.datak.2012.07.004

[34] Parra-Arnau, J., Perego, A., Ferrari, E., Forné, J., Rebollo-Monedero, D., "Privacy-Preserving Enhanced Collaborative Tagging". IEEE Trans. Knowl. Data Eng. 2012. http://dx.doi.org/10.1109/TKDE.2012.248

[35] Parra-Arnau, J., Rebollo-Monedero, D., Forné, J., "Measuring the Privacy of User Profiles in Personalized Information Systems", Future Generation Computer Systems. 2013.

http://dx.doi.org/10.1016/j.future.2013.01.001

[36] Rebollo-Monedero, D., Parra-Arnau, J., Diaz, C., Forné, J., "On the Measurement of Privacy as an Attacker's Estimation Error". Springer, International Journal of Information Security. Vol. 12, n. 2. Pp 129-149. 2013.

[37] Estrada, J., Rodríguez, A., Parra-Arnau, J., Forné, J., Rebollo-Monedero, D., "Medición de la Privacidad de Perfiles de Usuario mediante un Add-on de Navegador," XI Jornadas de Ingeniería Telemática (JITEL 2013). Granada, Spain. Pp 93-100. 2013.

[38] Drennan, J., Sullivan G., Previte, J., "Privacy, Risk Perception, and Expert Online Behavior: An Exploratory Study of Household End Users". Journal of Organizational and End User Computing (JOEUC). Volume 18(1). Pp 1-22. 2006. http://dx.doi.org/10.4018/joeuc.2006010101

[39] Pass, G., Chowdhury, A., Torgeson, C., "A Picture of Search". The First International Conference on Scalable Information Systems, Hong Kong. June, 2006.

[40] Google Ad Planner. Available at: https://www.google.com/adplanner/ #audienceBuilder.

[41] Schmidt, E., "Global Privacy Standards". Available at http://www.peterfleischer. blogspot.com/. 2007.

[42] TextWise. Available at: http://www.textwise.com/

Copyright Disclaimer